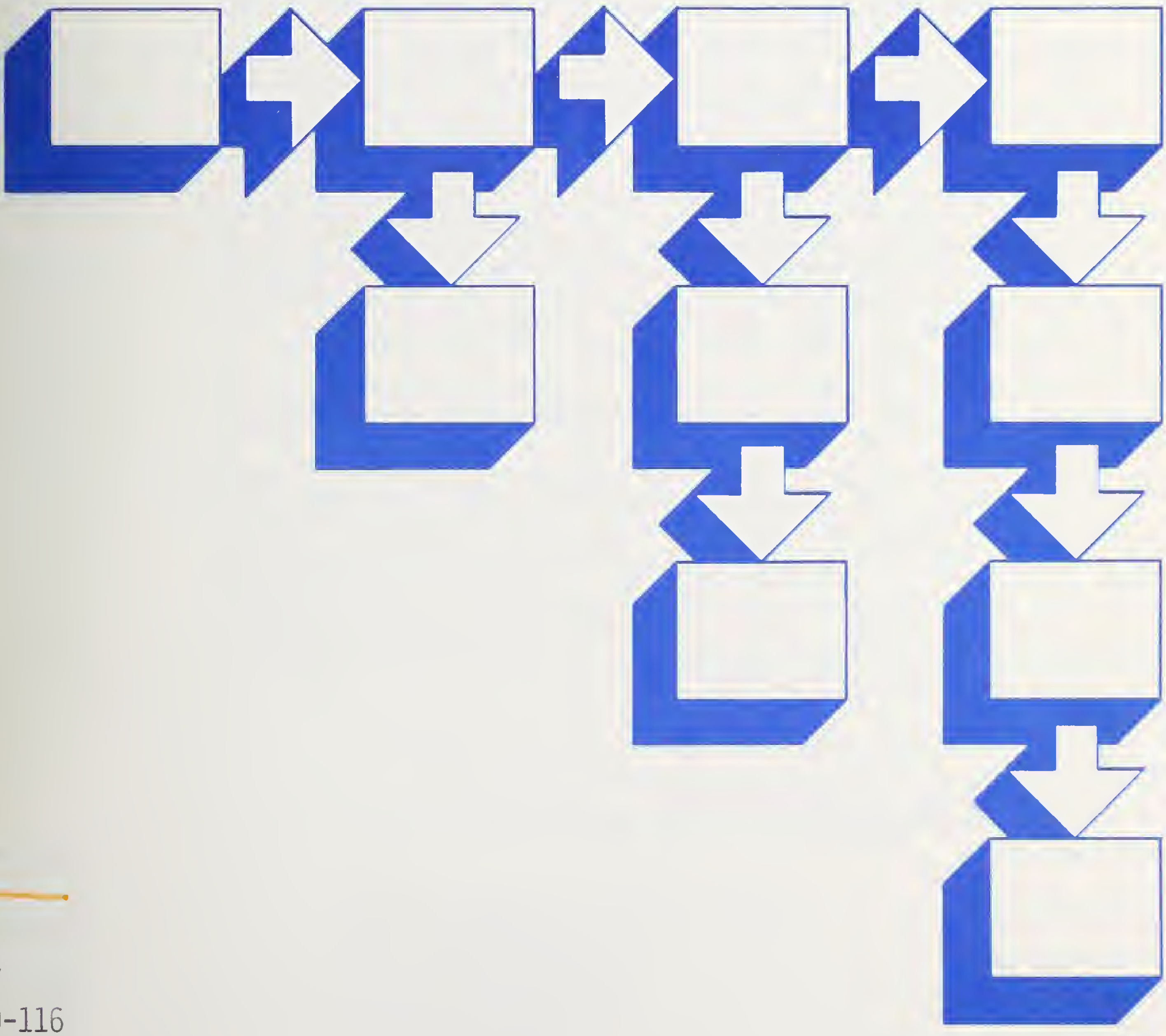
National Bureau
of Standards

Computer Science and Technology

NBS Special Publication 500-116

Toward an Improved FIPS Cost-Benefit Methodology, Phase II: Descriptive Models— General Purpose Application Software Development and Maintenance



NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards[1] was established by an act of Congress on March 3, 1901. The Bureau's overall goal is to strengthen and advance the Nation's science and technology and facilitate their effective application for public benefit. To this end, the Bureau conducts research and provides: (1) a basis for the Nation's physical measurement system, (2) scientific and technological services for industry and government, (3) a technical basis for equity in trade, and (4) technical services to promote public safety. The Bureau's technical work is performed by the National Measurement Laboratory, the National Engineering Laboratory, and the Institute for Computer Sciences and Technology.

THE NATIONAL MEASUREMENT LABORATORY provides the national system of physical and chemical and materials measurement; coordinates the system with measurement systems of other nations and furnishes essential services leading to accurate and uniform physical and chemical measurement throughout the Nation's scientific community, industry, and commerce; conducts materials research leading to improved methods of measurement, standards, and data on the properties of materials needed by industry, commerce, educational institutions, and Government; provides advisory and research services to other Government agencies; develops, produces, and distributes Standard Reference Materials; and provides calibration services. The Laboratory consists of the following centers:

Absolute Physical Quantities[2] — Radiation Research — Chemical Physics — Analytical Chemistry — Materials Science

THE NATIONAL ENGINEERING LABORATORY provides technology and technical services to the public and private sectors to address national needs and to solve national problems; conducts research in engineering and applied science in support of these efforts; builds and maintains competence in the necessary disciplines required to carry out this research and technical service; develops engineering data and measurement capabilities; provides engineering measurement traceability services; develops test methods and proposes engineering standards and code changes; develops and proposes new engineering practices; and develops and improves mechanisms to transfer results of its research to the ultimate user. The Laboratory consists of the following centers:

Applied Mathematics — Electronics and Electrical Engineering[2] — Manufacturing Engineering — Building Technology — Fire Research — Chemical Engineering[2]

THE INSTITUTE FOR COMPUTER SCIENCES AND TECHNOLOGY conducts research and provides scientific and technical services to aid Federal agencies in the selection, acquisition, application, and use of computer technology to improve effectiveness and economy in Government operations in accordance with Public Law 89-306 (40 U.S.C. 759), relevant Executive Orders, and other directives; carries out this mission by managing the Federal Information Processing Standards Program, developing Federal ADP standards guidelines, and managing Federal participation in ADP voluntary standardization activities; provides scientific and technological advisory services and assistance to Federal agencies; and provides the technical foundation for computer-related policies of the Federal Government. The Institute consists of the following centers:

Programming Science and Technology — Computer Systems Engineering.

[1]Headquarters and Laboratories at Gaithersburg, MD, unless otherwise noted;
mailing address Washington, DC 20234.
[2]Some divisions within the center are located at Boulder, CO 80303.

Computer Science and Technology

NBS Special Publication 500-116

Toward an Improved FIPS Cost-Benefit Methodology, Phase II: Descriptive Models— General Purpose Application Software Development and Maintenance

Mary Lou Chipman
Marco Fiorello

TITAN Systems, Inc.
6833 Beverly Drive
McLean, VA 22101

Peg Kay
Patricia Powell

Institute for Computer
Sciences and Technology
National Bureau of Standards
Gaithersburg, MD 20899

Monty Snead

Aurora Associates, Inc.
1140 Connecticut Ave., N.W.
Washington, DC 20036

U.S. DEPARTMENT OF COMMERCE
Malcolm Baldrige, Secretary

National Bureau of Standards
Ernest Ambler, Director

Issued June 1984

Reports on Computer Science and Technology

The National Bureau of Standards has a special responsibility within the Federal Government for computer science and technology activities. The programs of the NBS Institute for Computer Sciences and Technology are designed to provide ADP standards, guidelines, and technical advisory services to improve the effectiveness of computer utilization in the Federal sector, and to perform appropriate research and development efforts as foundation for such activities and programs. This publication series will report these NBS efforts to the Federal computer community as well as to interested specialists in the academic and private sectors. Those wishing to receive notices of publications in this series should complete and return the form at the end of this publication.

National Bureau of Standards Special Publication 500-116
Natl. Bur. Stand. (U.S.), Spec. Publ. 500-116, 55 pages (June 1984)
CODEN: XNBSAV

Library of Congress Catalog Card Number: 84-601071

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1984

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402

TABLE OF CONTENTS

EXHIBITS

ABSTRACT

This report presents a functional-flow descriptive model that can be used to categorize the application software (ASOF) development and maintenance activities of Federal data processing facilities. ASOF-related activities may be conceptually represented in descriptive model form by combining one or more of the basic model tasks. The comprehensive framework for ASOF development and maintenance provided by the descriptive model can be used in the identification of impacts from standards and guidelines and in the preparation of cost-benefit impact assessments. The framework provides both macro and micro levels of detail in order to link the descriptive models to additional data processing issues.

Key words: application software development; computer standards; cost-benefit analysis; data processing management; descriptive models; evaluability assessment; information systems.

FOREWORD

The Information Processes Group (IPG) of the Institute for Computer Sciences and Technology (ICST) was created in 1981 and is responsible for, among other things, the development of methodologies for assessing the costs and benefits of Federal Information Processing Standards (FIPS). Almost immediately it became apparent that existing cost-benefit methodologies were inadequate for the purpose, partly because they were based on "rhetorical" (or idealized) models of the Federal DP environment* and did not take into account the multiplicity of operational variations among the agencies, the complex interactions among the components of data processing system management and operations, or the sometimes extreme difference between the way things are and the way things ought to be. In other words, existing methodologies are sufficient for "ball park" estimates; but, particularly in cases where anticipated costs and benefits are less than enormous, the expected margin of error is unacceptably large.

In order to reduce the margin of error, the IPG embarked on a program to describe more accurately the Federal DP environment. Using functional flow diagrams as the basic tool for description, we hope eventually to have a realistic model of the Federal DP environment. This report on Application Software Development and Maintenance marks the completion of Phase II of the program.

The bulk of the work on this project was done by Mary Lou Chipman of TITAN Systems, Inc., working with Dr. Monty Snead of Aurora Associates, Inc. Significant contributions were made by Dr. Marco Fiorello and Peter Eirich of TITAN Systems, Inc., and Peg Kay and Pat Powell of the IPG/ICST. The descriptive modeling methodological framework used in this study was developed by Peg Kay of ICST. Like most of the IPG projects, results were obtained and validated through interviews and reviews with personnel from other Federal agencies. We received a number of useful comments which have been incorporated into this report. We are particularly grateful to personnel from: the Department of Energy (both regional offices and contractors); the Department of Justice; National Aeronautics Space Administration; the Brookhaven, Los Alamos, and Sandia National Laboratories; the Federal Home Loan Bank Board; the Federal Communications Commission; the Federal Emergency Management Agency; the Department of Commerce, the Tennessee Valley Authority; and the Department of Housing and Urban Development.

---

* Another serious barrier to accurate cost-benefit projections is the absence of reliable Base Case data. This series of reports does not address that issue.

While the IPG program is primarily directed toward the improvement of ICST's products, it is clear that the descriptive models being developed are useful tools for ADP managers throughout the Federal Government--and probably for ADP managers in sub-Federal jurisdictions and in industry. Their usefulness is not confined to cost-benefit related studies, but is applicable to a host of management concerns (e.g., reorganization, workload forecasting, functional specifications for procurement, and so on). We are therefore making these reports widely available in the NBS Special Publications series.

Questions or suggestions related to the program are welcome and should be addressed to Pat Powell, Institute for Computer Sciences and Technology, Building 225, Room B246, National Bureau of Standards, Washington, DC 20234.

I. INTRODUCTION

A. General

The objective of this Phase II report is to present and describe a "descriptive model" of how the Federal Government develops and maintains its general purpose application software (ASOF). By a "descriptive model" we mean a qualitative portrayal of the activities and their interrelationships involved in the development and maintenance of ASOF.

The Phase I effort, our initial step in the model development process, focused on developing the overall descriptive modeling framework and presenting a model of the "Data Processing Operations" portion of Exhibit I-1. The Phase II model is the second step toward the development of a comprehensive set of Federal data processing descriptive models which, ultimately, will depict the breadth of DP system management and operation portrayed in Exhibit I-1. A brief summary of the role of descriptive modeling in standards evaluations is presented in the following section. Further discussion of the role of descriptive modeling in standards evaluation can be found in [ICST-82].*

B. Evaluation of Standards

As noted in our Phase I report [ICST-82, p.1], ICST conducts an assessment of expected costs and benefits before it develops a standard or guideline. Based on a review of six impact assessments which applied the 1978 set of preliminary guidelines for performing these assessments [FIOR-78], several relevant conclusions were offered [FIOR-81]:

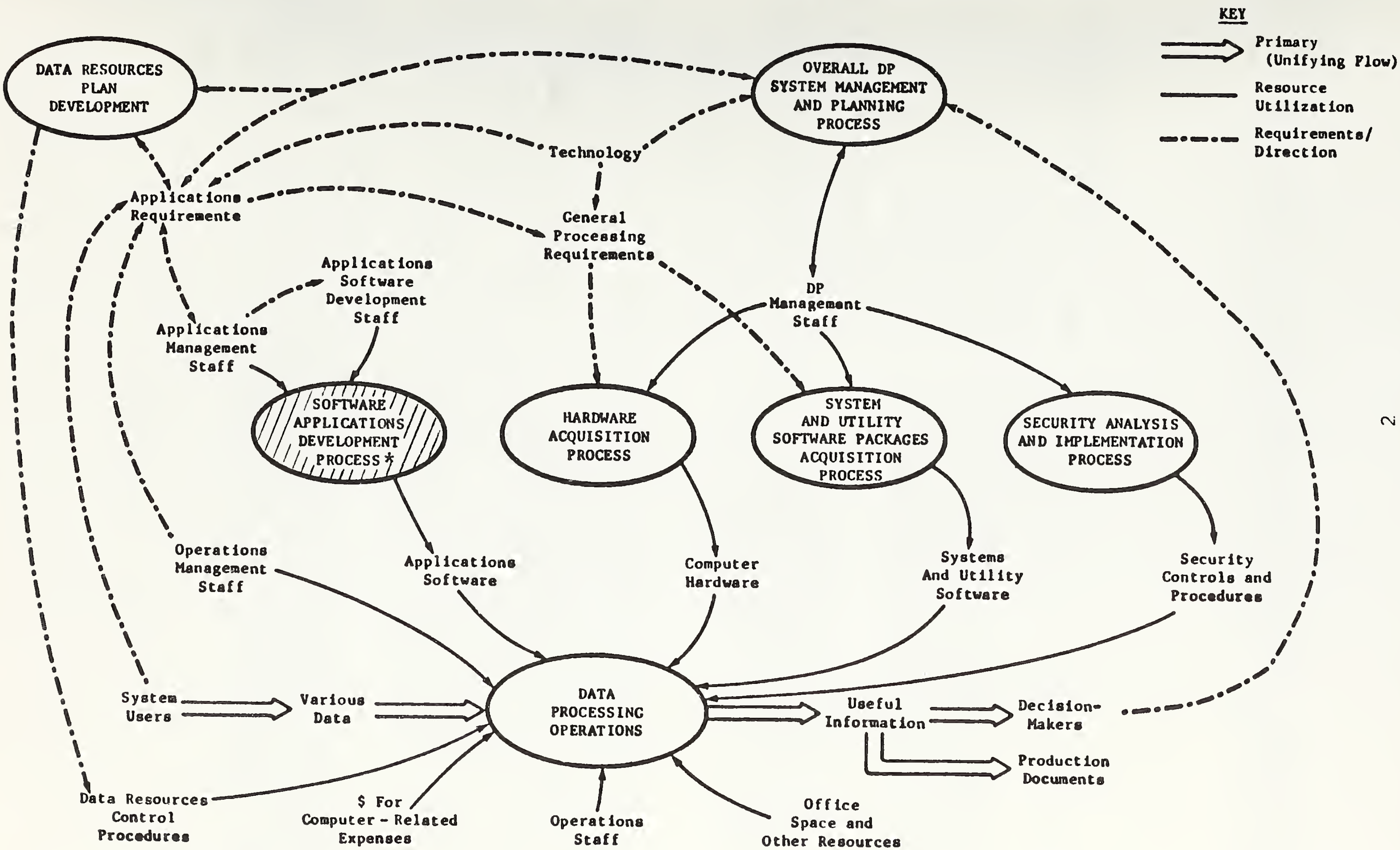
- o The analyses tended to focus exclusively on the components or parts of the ADP system or process, and did not mention the relevant procedures and management actions that must occur to achieve the impact modeled.
- o The interpretation of the nature of impacts was inconsistent across the studies.

It was recommended that, when revised, the impact assessment guidelines should:

---

* References are listed at the end of the text. They are cited in the text by two to four alphanumeric characters followed by the last two digits of the year of publication, all within brackets.

Exhibit I-1. MAJOR COMPONENTS OF DP SYSTEM MANAGEMENT AND OPERATION



* The focus of this analysis.

- Improve the discussion of the formulation of the base case and impact process. The explicit use of scenario constructs and a descriptive model (functional diagrams with unifying flows) should be introduced.

It was noted that the following benefits result when such an approach is applied [ICST-82]:

- The cost-benefit analyst's implicit mental model is replaced with an explicit, documented and commonly understood representation of Federal DP activities.

- Consistency among the different impact assessments of standards performed by ICST is facilitated.

- Comparisons are permitted among the expectations of standards developers, the benefits projected in the impact assessment, and the actual benefits that may be computed after a standard has been implemented.

C. Project Goals

The end-product of the overall effort will be a set of functional-flow descriptive models providing a comprehensive representation of Federal data processing activities. Using this framework, it will be possible to specify the impacts of different standards, collect data organized so as to aid impact assessments, and define points of measurement for evaluating a standard's actual benefits.

The Phase I and Phase II descriptive models can be applied to the development of a cost-effective set of computer-related standards and guidelines. For example, they can improve planning for ICST products by identifying the types of personnel to be affected by a proposed standard or guideline. Improved standards will, in turn, assist computer system and application managers to do a better job of developing and monitoring data processing systems.

D. Descriptive Modeling and the Unifying Flow Principle

Much of the modeling terminology presented throughout this report and our Phase I report [ICST-82] is based on the evaluability assessment literature [NAKA-82]. Here we will briefly define some of the major terms that will be applied throughout this report.

As indicated previously, by "descriptive model" we mean a type of functional model depicted by a function and flow diagram that presents what is <u>actually observed to be occurring</u>. Here we are equating descriptive models with "equivalency" models discussed in the evaluability assessment literature.

A second type of model, the "rhetorical" or "testable model", is a portrayal of what is believed to be occurring during the process under scrutiny. Frequently, people with authority to affect the process or who teach about the process believe the process to work quite differently than it actually does. When there is a wide discrepancy between what is believed to be occurring and the reality of day-to-day work, those in charge of the agency may make decisions that either cannot be implemented or -- if implementable -- may have unintended effects. It is therefore useful to examine both the rhetorical and descriptive models of a given process in order to compare and contrast them. In common with the descriptive models, the rhetorical models are depicted by functional flow diagrams.

The key to preparing both the descriptive and rhetorical models in a way that captures the essence of a system or organization, and portrays it in a straightforward manner, is to identify a "unifying flow" that ties the system together; that is, some item or characteristic that can be traced by the model analyst through the system or organization to be modeled.

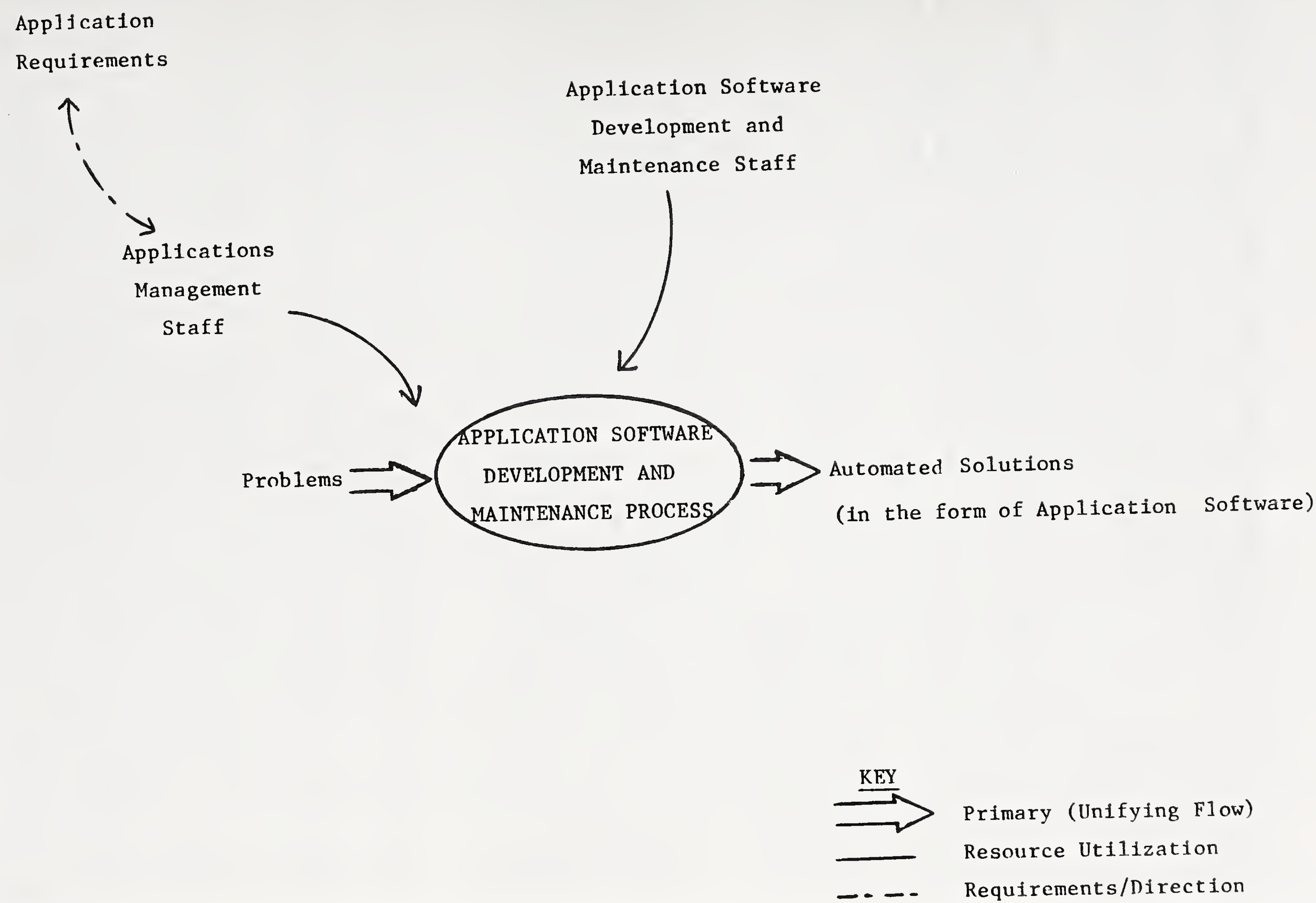
In our Phase I effort, the transformation of data into information was selected as the unifying flow for the descriptive model of Data Processing Operations in Exhibit I-1, taken from [ICST-82]. For the Phase II study of the applications software (ASOF) development and maintenance process, our literature review led us to select problem to automated solution as the unifying flow. Utilizing this unifying flow, in the descriptive model of ASOF development and maintenance, facilitates the identification and partitioning of relevant activities into the essential steps necessary to conduct the process and define the interactions among the steps.

E. Scope of the Phase II Effort

The scope of the Phase II modeling effort includes the following components of the data processing environment: "Applications Software Development and Maintenance Process", "Applications Management Staff", "Applications Software Development and Maintenance Staff", "Applications Requirements", and "Applications Software". Exhibit I-2 portrays the basic relationships among these ASOF-related components and the unifying flow.

ASOF is defined as a computer program (or set of programs) that automates a task. ASOF development is a process that starts with the identification, analysis, and interpretation of an

Exhibit I-2. APPLICATION SOFTWARE DEVELOPMENT AND MAINTENANCE PROCESS



application problem or requirement, proceeds through the development of an automated solution, and results in a program that implements the solution. Maintenance is defined as any post-development activity to: correct programming errors (corrective), modify the program to accommodate data processing changes (adaptive), or improve or augment program capabilities (perfective).

The Phase II effort was restricted to ASOF developed to accomplish the Federal Government's standalone, general purpose, data or information processing functions, e.g., payroll, financial reporting, inventory tracking and monitoring. Contractor-developed software falling within the above scope was included. However, the development of system (e.g. operating system, compiler) and utility software or software developed for unique applications within the Federal Government (e.g., embedded software/hardware systems, or the Federal Reserve's check sorting and processing system) were excluded.

F. Methodology

The descriptive model presented in this report was empirically derived on the basis of data collected during on-site interviews with a total of 17 DP staff, including managers, systems analysts, and programmers, at four Federal DP facilities in the Washington, D.C., area. A preliminary literature review guided the construction of structured interview protocols. The results of the interviews, along with review of written documentation at each site, were used to derive the descriptive models presented in this report.

G. Overview

Chapter II presents the rhetorical model of the ASOF development and maintenance process derived from a literature review.

Chapter III introduces several descriptive models based on interviews with Federal Government staff involved in ASOF development and maintenance.

Chapter IV presents a narrative comparison of the rhetorical and descriptive models, and discusses basic observations made during the interviews.

Chapter V summarizes the work done in this report. A glossary of selected terms is provided at the end of the text. The bibliography contains the references used in this study as well as a number of references for general background discussions on the application software development and maintenance process. The Appendix describes the data collection procedure used in the survey.

II. THE APPLICATION SOFTWARE DEVELOPMENT AND MAINTENANCE RHETORICAL MODEL

A. General

A rhetorical model* of the ASOF development and maintenance process represents what is thought to be occurring during the process. The rhetorical model presented here is an amalgam of traditional and recently developed "academic" approaches to software development and maintenance. It is important to recognize that this model is not a prescription. We do not believe that it is normally implemented in its entirety. It represents an idealized set of ASOF development and maintenance activities that provides a useful basis for comparison with actual observations.

Contemporary ASOF development and maintenance models can be classified into two basic types: a classical (management oriented) approach, and a neoclassical (process oriented) approach. This breakout of model types is somewhat artificial, but is useful to describe the range of approaches to ASOF development and maintenance.

Several classical models of ASOF development and maintenance are presented in Exhibit II-1. Variations in these classical models of the software development and maintenance process exist (see [PETR-78]), but they are, in general, definable within the basic Brandon and Gray [BRAN-70] framework. The most prominent feature of the classical model is the presumed sequential, and essentially non-iterative nature of the steps performed. The biggest problem with the classical approach is that it does not reflect the dynamic, interactive nature of the applications software development process.

The neoclassical approach to ASOF development and maintenance includes modern software development strategies that can reduce or eliminate many of the problems associated with the classical approach [DEMA-78], [DIST-80], [BOEH-76], [BERG-78], and [GAO-81]. In general, these strategies incorporate portions of the iterative, evolutionary, and feedback-oriented nature of the ASOF process into their paradigm. One of the major strategies in these neoclassical approaches is the use of structured techniques. The classical approach shaped the ASOF development process so that it would conform to traditional management approaches, while the neoclassical approach better captures the essence of the process and then fits an appropriate management scheme to it. In preparing the rhetorical model of the ASOF pro-

---

* See Section I.D for definition of "rhetorical model"

Exhibit II-1. COMPARISON OF SOME OF THE APPROACHES TO APPLICATION SOFTWARE DEVELOPMENT AND MAINTENANCE

<table>
<tr><th colspan="6">AUTHORS OF THE PROPOSED PHASES</th></tr>
<tr><th>[BRAN-70]</th><th>[DALY-77]</th><th>[BENJ-71]</th><th>[FREE-76]</th><th>[METZ-73]</th><th>[BOEH-76]</th></tr>
<tr><td>Application Identifiction and Project Selection</td><td rowspan="2">Planning</td><td rowspan="2">Feasibility Study</td><td rowspan="2">Needs Analysis</td><td rowspan="3">Definition</td><td></td></tr>
<tr><td>System Survey</td><td>System Requirements</td></tr>
<tr><td>Data Gathering</td><td rowspan="2">Specify</td><td rowspan="2">System Specification</td><td>Specification</td><td>Software Requirements</td></tr>
<tr><td>System Analysis</td><td rowspan="2">Architectural Design
Detail Design</td><td rowspan="2">Design</td><td>Preliminary Design</td></tr>
<tr><td>System Design</td><td>Design</td><td>System Engineering</td><td>Detailed Design</td></tr>
<tr><td>Programming</td><td>Code</td><td rowspan="2">Programming and Procedure Development</td><td rowspan="3">Implementation</td><td>Programming</td><td>Code & Debug</td></tr>
<tr><td>Program Testing</td><td rowspan="2">Test</td><td>System Test</td><td rowspan="3">Test and Preoperations</td></tr>
<tr><td>System Testing</td><td rowspan="2">Implementation</td><td>Acceptance</td></tr>
<tr><td>Conversion and Installation</td><td></td><td></td><td>Installation & Operations</td></tr>
<tr><td>System Maintenance</td><td></td><td rowspan="2">Maintenance</td><td rowspan="2">Maintenance</td><td rowspan="2"></td><td rowspan="2">Operations and Maintenance</td></tr>
<tr><td>System Evaluation</td><td>Evaluation</td></tr>
</table>

Source: Peters, L. J. and Tripp, L. L. "A Model of Software Engineering," Boeing Computer Services Co., Seattle, Washington, 1978.

cess, we used components of the classical and neoclassical frameworks.

B. Detailed Description

The rhetorical model of the ASOF development and maintenance process consists of 11 basic steps:

1. Application Identification and Project Selection
2. System Requirements
3. System Analysis and Cost-Benefit Analysis
4. System Design
5. Program Development
6. Testing
7. Training
8. Acceptance Test
9. System Turnover
10. System Maintenance
11. System Evaluation

These 11 steps are normally presumed to be performed in a sequential manner as shown in Exhibit II-2. However, even in the rhetorical model, changes and reviews throughout the process can cause iterations within and among the steps. Generic types of personnel believed to be involved in each of these steps in primary or secondary roles are indicated in Exhibit II-3. Each of the rhetorical model steps is described next.

1. Step 1 - Application Identification and Project Selection

In our rhetorical model, the identification of the candidate application or project for automation originates in a user (of-the-automation product or service) group that has organizational responsibility for managing and performing the application. There is rarely just one type of user in a typical automation project, and for convenience we utilize two generic types: user and user-management: "user" refers to individuals that perform the "hands-on" activity in the existing or new application being considered for automation; "user-management" refers to individuals responsible for the function or application, and for requesting the automation project. A user group consists of users and user-management.

Once a user group has identified an area for potential improvements, and its user-management approves the idea, a written request for support from the DP group is forwarded to DP management. This request for support includes an identification/description of the problem/process to be automated.

Exhibit II-2. ASOF RHETORICAL MODEL - FLOW DIAGRAM

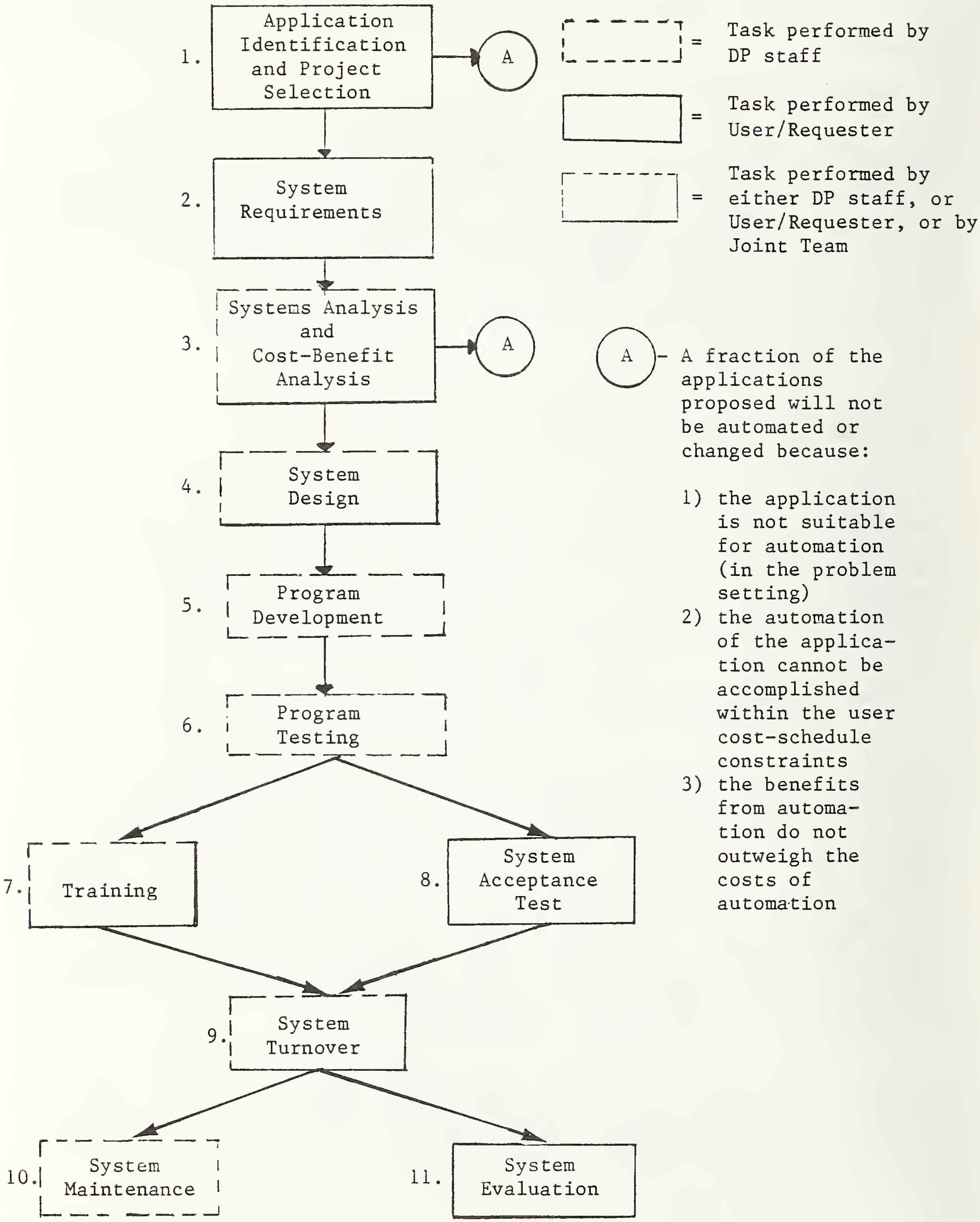


Exhibit II-3. GENERIC TYPES OF PERSONNEL AND THEIR ROLES BELIEVED TO BE INVOLVED IN THE RHETORICAL MODEL'S STEPS

RHETORICAL MODEL BASIC STEPS	GENERIC TYPES OF PERSONNEL						
	USER *	USER-MGMT. *	DP MGMT.	ANALYST	PROGRAMMER	OTHER ANALYSTS	OTHER PROGRAMMERS
1. Application Identification and Project Selection	P	P	A/P				
2. System Requirements	P	A	A				
3. System Analysis and Cost Benefit Analysis	A	A/P	A	P			
4. System Design	A			P		A	
5. Program Development				A	P		A
6. Program/System Testing	A			P	P		
7. Training	P		A	P	P		
8. Acceptance Test	P	P	A				
9. System Turnover	P		P	A			
10. Maintenance	P			P	P		
11. System Evaluation	P	P	A				

P = Primary

A = Also Involved

* It should be recognized that there is rarely just one "user" in a typical automation project. In this report, the term "user" refers to the hands-on user of the automated function, and "user-management" refers to the individual(s) responsible for the function being automated and for requesting the automation project.

The user request for DP support is reviewed by DP management to verify that the request is suitable for automation. If DP management accepts the project, a preliminary estimate of cost and scheduling is developed.

These preliminary cost and scheduling estimates are then reviewed by user-management to determine whether the new application will fit into the organization's budget and time constraints. Approval by user-management of these preliminary estimates is required before the process can continue.

2. Step 2 - System Requirements

In the rhetorical model, once the Application Identification and Project Selection step is completed, the user group begins the development of a systems requirements document. The assumption is that this document is developed by the user group to provide the DP department with as much information as possible about the proposed system. The document is supposed to contain such information as functional requirements, data available/required, input/output formatting requirements, interaction with other manual and automated processes, operational environment, etc. The document is also supposed to define clearly what an acceptable system must do, and to provide a basis for establishing system test criteria.

Once the Systems Requirements document is developed it is presumably reviewed by user-management before being sent to DP management. Upon receiving the requirements document, DP management reviews it, indicates areas where additional information is needed, and, if necessary, updates the preliminary cost and scheduling estimates and requests user approval of the updated estimates.

3. Step 3 - System Analysis and Cost-Benefit Analysis

In this "rhetorical" step, the DP systems analyst reviews the Systems Requirements document, performs user group interviews, and investigates related automated systems in order to learn as much as possible about the proposed system. The systems analyst then develops system alternatives which will meet the user's requirements. A basic cost-benefit analysis of the new application and its solution alternatives is then supposed to be performed by the systems analyst with inputs from the user group and DP management as necessary.

Some rhetorical models we reviewed indicate the use, in this step, of structured analysis techniques, such as Data Flow Diagrams, Data Structure Diagrams and Data Dictionaries to help produce systems analysis documents which are easily reviewed and understood by the user group.

Upon completion, the cost-benefit analysis and the solution alternatives are presented to the users and user-management by the systems analyst.

This information is then evaluated by the users, user-management and DP management in order to identify the "preferred" solution alternative. The cost-benefit analysis may indicate that none of the solutions are feasible, in which case the entire project would be cancelled, redefined, or reanalyzed to define new solutions not yet identified.

4. <u>Step 4 - System Design</u>

After the solution approach has been selected, the DP systems analyst begins in this step to design the proposed system. In this rhetorical model, the analyst develops the design using top-down design techniques suitable for system definition and develops test data, program, and file specifications.

Once the systems analyst has completed the design, a formal peer review is to be conducted, which may involve a "walkthrough" of the design. Any problems identified during the review would then be corrected by the DP systems analyst.

The user-group is then supposed to be given a presentation of the final design to ensure the developer fully understands how the new system is intended to function.

After clearing the final design with the user-group, the analyst turns over the program specification to the programmer, and initiates the development of a User's Manual for the system.

5. <u>Step 5 - Program Development</u>

In this rhetorical model, the programmer develops the modules' logic based on the program specifications prepared in the System Design step and illustrates the program logic using block diagrams, decision tables or other similar techniques. Any questions arising during the logic definition effort are to be addressed to the systems analyst responsible for the design.

As part of this rhetorical model, the completed program logic is submitted for peer review. Techniques which may be used are inspections, walkthroughs, code reading, and round-robin reviews. The programmers begin coding on the program after all logic errors identified during the peer review are corrected. The code is generated according to the organization's putative standards which may employ a modular top-down approach. Programmers may also use automated tools and techniques including program generators and interactive debugging.

The documented program logic developed in this step, along with the program specifications, is to be filed with other program documentation, required by the organization, to establish a dynamic history of the program for maintenance personnel.

6. Step 6 - Testing

Unit testing is presumed to begin upon completion of module coding. Unit testing involves executing the module with the test data defined during the System Design Step. Output from these test runs is to be reviewed by the programmer. When the programmer believes the module is functioning correctly, the systems analyst reviews the output. If the analyst believes the module is functioning correctly, it is ready for integration testing. However if there are problems, unit testing continues.

Once two or more related modules have been successfully unit tested, integration testing is supposed to begin. This phase involves testing the individual modules as an integrated group to ensure they work together properly. Output from these tests are to be reviewed by the programmers involved and the systems analyst.

After all the modules have been integrated, system testing would then verify that the overall system is functioning as specified. Output of the testing would be reviewed by the systems analyst.

7. Step 7 - Training

At this point in the rhetorical model the systems analyst initiates the user training process. This training includes a description of what is covered in the User's Manual.

8. Acceptance Test

When the DP staff are assured that the system is working properly, user acceptance testing begins. During this test, the system performance is to be compared to the test criteria developed by the user from the specifications in the requirements document. The output from the system acceptance test would be reviewed by the user-group to verify that the system does meet the requirements.

9. System Turnover

While user training is being completed, the system is installed and turned over to the user for production running.

The user then makes the first few runs of the system, with the assistance of the systems analyst, if necessary, in parallel with the existing procedure for handling the application. These first production runs are to be used to verify that the system runs properly in an operational environment. Any problems identified during this parallel testing would be corrected by the DP systems analyst and the programmer. Any perfective maintenance requirements identified during this time are postponed until after parallel testing is completed. Once the user is satisfied that the system is working as specified in the acceptance test, user management would notify DP management that the delivered system is acceptable. Normal production running of the system would then begin.

10. Step 10 - System Maintenance

Once the system is operating in a normal production mode, any changes necessary to the system are classified as maintenance. These changes may be corrective, adaptive or perfective. Normally corrective, adaptive and small perfective changes presumably are taken care of as soon as possible by the DP systems analyst and programmer in the maintenance group. Large perfective changes are handled in much the same way as new software development efforts.

11. Step 11 - System Evaluation

This "rhetorical" step consists of a follow-up evaluation after four or five full cycles of the new system. It is the responsibility of the data processing shop to conduct interviews, observe system operation, and prepare the performance evaluation report. Anticipated system benefits are to be compared with actual benefits to determine if the system has fulfilled requirements. The user also identifies and documents needed improvements in a user acceptance test.

The rhetorical model presented in this chapter is compared with descriptive models of ASOF development and maintenance observed in several Federal agencies in Chapter IV.

III. THE DESCRIPTIVE MODEL

A. General

The rhetorical model, in the previous chapter, represents what some contemporary theorists believe (or say they believe) occurs in application software development and maintenance. In comparison, the descriptive model, presented in this chapter, is based on actual experience in several Federal agencies. After reviewing the ASOF development and maintenance processes in these agencies, we found a number of differences in how software is being developed. In order to represent the process in all of the shops surveyed, we developed a "composite" descriptive model which consists of 21 basic tasks. These tasks can be assembled in different combinations to represent consistently the variety of ASOF development and maintenance processes in the different agencies. In general, it is unlikely that all 21 tasks would be represented explicitly in any one particular data processing facility. Several illustrations of descriptive models for ASOF development and maintenance are presented in Section C.

The basic tasks which make up the descriptive model are:

1. User Makes Request
2. Data Systems Problem Determination
3. Feasibility Study
4. Feasibility Study Review
5. Evaluation of Alternative Approaches
6. Preparation of Systems Development Plan
7. Software Package Investigation
8. Preliminary Program Design
9. Program Design Approval
10. Module Specification
11. Module Logic Development
12. Module Logic Peer Review
13. Coding
14. Test Data Development
15. Unit Testing
16. System Testing
17. Parallel Testing
18. Documentation Finalization
19. Training
20. Turnover to Production Mode
21. Maintenance

A detailed description of each of these tasks is presented in the following section. These descriptions also identify the key players and major products of each of the tasks. Please note that while we often refer to a user, analyst, or programmer (in the singular) the words are intended to cover those cases where groups or teams are involved. Exhibit III-1 shows the key players in each of the 21 tasks. In general, it is rare to find any organization which uses all of the 21 tasks as part of its software development and maintenance process. The entries in Exhibit III-1 indicate the "typical" settings observed in the ASOF descriptive modeling activity. In the review of the draft report, a number of variations were noted by the reviewing agencies: additional personnel categories were cited, including user data processing liaison, standards/quality control, contractors, project manager, and computer specialist (programmer and analyst); and, the degree of overall involvement for all personnel types increased for large and/or complex tasks relative to the usual project profiles indicated in the exhibit.

B. Detailed Description

1. User Makes Request

This task can be initiated by an informal request, by a formal written request (created by the users, signed-off by user-management, and forwarded to the DP shop management for sign-off and action), or by a procedure somewhere between these two levels of formality. The larger the DP shop or the larger the system being requested, the more formal the request is likely to be. Regardless of the initiation mode, the product from this task is a user functional requirements statement describing the problem or requirement and the request for an automated solution.

2. Data Systems Problem Determination

The user request is reviewed by the DP shop to determine whether it is a problem which can best be solved by automation. This task usually consists of a meeting of data processing staff and requesting users to determine what the user's problem really is and what the possible solutions are. The complexity of the system to be developed often indicates whether this meeting will be just an informal phone conversation, or will be a series of more formal meetings involving the analyst (and perhaps the manager from the DP shop) and a group of prospective users and their management. The product from this task is a "go" or "no-go" formal decision by the DP shop to continue with the ASOF project to at least the next task activity.

3. Feasibility Study

Given an affirmative decision in task 2 to proceed, a feasibility study is then initiated. The feasibility study is concerned with whether or not the user's functional requirements can be satisfied, and how. This study is often only a technical evaluation of the problem, sometimes with only one possible approach being identified to meet the user's needs. Other feasibility studies may identify several possible alternative

Exhibit III-1. GENERIC TYPES OF PERSONNEL INVOLVED IN THE DESCRIPTIVE MODEL'S TASKS

(for a "typical" setting)

DESCRIPTIVE MODEL TASKS	ROLES							
	USER	USER MGMT.	DP MGMT.	ANALYST	PROGRAMMER	OTHER ANALYSTS	CONTRACTOR	OTHER PROGRAMMER
1. User Makes Request	P	D	D	P				
2. Data Systems Problem Determination	P	A	D	P				
3. Feasibility Study	A		A	P		C	C	
4. Feasibility Study Review	P	D	C					
5. Evaluate Alternative Approaches	P	C	C	P		C	C	
6. Prepare System Development Plan				P				
7. Software Package Investigation				P				
8. Preliminary Program Design	A			P			C	
9. Program Design Approval	C	C	P	A		C		
10. Module Specifications	C			P			C	
11. Module Logic Development	C			A	P	C	C	
12. Module Logic Peer Review				D	P	C	C	D
13. Coding				C	P		C	
14. Test Data Development	D			D	D		D	
15. Unit Testing	C			A	P		C	
16. System Testing	C			P	P		C	
17. Parallel Testing	P			P	P		C	
18. Documentation Finalization				D	P		C	
19. Training	P	C		P	A		C	
20. Turnover to Production Mode				D	D			
21. Maintenance	A			D	P		C	

P = Primary D = Primary in Some Organizations

A = Also Involved C = Sometimes Is Involved solutions to the problem. When several approaches are identified, it is more likely that the feasibility study will contain some type of cost-benefit analysis of the alternatives. Some DP shops do require formal cost-benefit analysis on all projects; however, most of the shops we visited estimated only development time and cost and provided these numbers as part of their feasibility study.

In shops which require a formal feasibility study, or in instances where it is determined that a feasibility study must be performed for a particular project, there is usually a written feasibility study document produced as part of the effort.

The feasibility study is normally conducted by an analyst in the DP department, with assistance from the manager and the requesting user as necessary. In DP shops with staffing constraints this step is often contracted out, with little involvement of the in-house DP staff during the study itself.

4. <u>Feasibility Study Review</u>

At this point in the process, the user reviews the feasibility study and decides whether or not to go ahead with the system development. By this time, preliminary estimates of at least development time and cost have been completed. These estimates may have resulted from a cost-benefit analysis conducted during the feasibility study, or they may simply be the analyst's subjective assessments based on discussions with the user. Note that formal cost-benefit analyses were more the exception than the rule. Thus, the user makes a "go" or "no-go" decision based on the time and cost estimates, the results of the feasibility study (if one was completed), any scheduling information DP management can provide, and any other information available at that time.

5. <u>Evaluation of Alternative Approaches</u>

There are often several ways in which a particular problem may be formulated and solved in ASOF development and maintenance. Some of these approaches may have been identified during the feasibility study; others or major variations may be identified in this task. Each alternative solution is reviewed with the user to first determine if it can satisfy the user's requirements and, second, to identify the technically preferred approach from those solutions that are acceptable to the user.

This evaluation effort is usually performed by the DP analyst and the user. Both data processing and user management may become involved in this review process, but more often, they are presented with the results of the review by the analyst and user. In DP shops with staffing constraints, this task may be performed by a contractor and the user with little DP group involvement.

6. Prepare Systems Development Plan

This task, when performed, consists of developing a written plan for the rest of the ASOF development process. The systems development plan normally contains a list of all the events in the process, and a preliminary schedule for their completion. The plan may also include milestone charts and other tools (e.g. PERT schedules) to support the management of the project. This plan is usually prepared by the analyst responsible for the project.

7. Software Package Investigation

In this task, the analyst responsible for the new system development reviews those software packages currently available on the target computer system to determine if there is a package which can be utilized to meet all or part of the user's requirements. If a software package exists that can satisfy the user requirements, and was not considered earlier as a possible solution alternative, it may be necessary to go back to task 5 to reevaluate the previously identified alternatives relative to the alternative's cost, effectiveness, availability, etc.

8. Preliminary Program Design

In this task, a preliminary design of the proposed solution is prepared. Depending upon its size and complexity, only a few staff hours or, occasionally, up to several staff years may be required. The design is normally done by the analyst with assistance from the user. This task usually results in a formal program documentation which may include:

- Program Narrative
- Operational Requirements
- Program Flowchart
- Input Definition/Description
- Output Definition/Description
- Implementation Plan
- Cross References of Procedures, Modules, Data
- Audit Trail Description
- Test Plan for the Program (in some instances)
- Security/Privacy Requirements
- Host Computer System Description

The document may also contain other technical information which the analyst believes may be of use in later stages of the development effort. For very simple, one-time efforts, much of the design information may be documented informally for later reference.

If this task is performed by contractors, they are normally required to produce a formal design document containing much of the information identified above no matter what the size or complexity of the system being designed.

9. Program Design Approval

In this task, the completed program design document is reviewed by DP management, and occasionally by the user, to verify its completeness and correctness. The product from this task is the preliminary program design document which has the formal approval of the DP management.

10. Module Specifications

Specifications are developed by the analyst to identify what is to be accomplished by each module in the system. The software module specifications can include:

- general description of the program
- definition/description of inputs
- processing requirements
- definition/description of outputs

The amount of detail included in the specifications depends on the agency, the staff's level of experience, and the complexity of the program.

If it has been determined that a software package will be used to meet the user's needs, then any routines needed, (i.e., interface, report generator, etc.), are designed as part of this task.

11. Module Logic Development

In this task, the programmer assigned to the job generates the module logic using

- the module specification, if developed;
- the detailed program design, if developed;
- discussions with the analyst and/or user, where necessary.

Depending on the complexity of the module and organizational requirements, the descriptions of the logic flow for the module may be:

- informally recorded (e.g. programmer's memory, partial notes, ...)
- a traditional flowchart
- a series of stepwise refinement charts
- a Nassi-Schneiderman Chart
- Warnier-Orr Diagrams
- or charts based on other design methodologies

Any clarification questions necessary on the function, inputs, or outputs of the module are normally addressed to the analyst responsible for the project.

12. Module Logic Peer Review

The module logic developed by the programmer may be reviewed in different ways. In some shops, this review is performed by the analyst on the project. Other sites use a structured walkthrough technique where the programmer presents to a group of peers (i.e., other programmers) the working specification and the proposed logic.

Any errors detected in the module logic are corrected before further work continues.

13. Coding

The programmer responsible for the development of a module produces the code necessary. Some Federal data processing shops require that the programmer follow certain standards during development (i.e., variable naming conventions, structured techniques), while other Federal shops allow a great deal of flexibility in the way code is written. Some shops require that the code generated by a programmer be reviewed by an analyst before it is tested.

14. Test Data Development

Test data are designed and developed by the analyst, the programmer, or some outside source, including contractors and users. The test data are usually designed to test each of the following:

- logic of the different pieces in a program;
- the logic/structure of one module;
- the interaction between two or more modules;
- the entire system.

The test data are created either by hand, by using a test data generator, by extracting data from existing data, or by some combination of the above.

15. Unit Testing

Each of the modules or program units is tested by its programmer using test data. If someone other than the programmer designed/developed the test data, that person is often responsible for reviewing the output of the test runs. Unit testing continues until all concerned believe that the module or program unit is functioning properly.

16. System Testing

Once some portion of the individual modules have completed unit testing, integration testing followed by system testing begins. Two or more related modules are tested to make sure that all of the module interfaces are correct. When the modules of the program(s) comprising the system have been tested, system testing can begin. Output from the system tests is reviewed by the analysts and programmers involved in that particular test. In some cases, the user may also become involved in reviewing the system test results.

17. Parallel Testing

If the new system is replacing an existing manual or previously automated system, the two are usually run in parallel for a period of time long enough to verify that the new one is working properly. At this time, when performed, the acceptance test criteria prepared by the user group are applied to evaluate the performance of the system delivered. During this process users, analyst, and programmer work together to identify and correct as many errors (bugs, incomplete requirements, etc.) as possible. Enhancements identified during this testing are normally postponed until after the testing has been completed.

18. Documentation Finalization

In this task, the documentation of the system is completed. The documentation effort could have started at any point in the development effort up to and including this point. The amount and type of documentation produced varies depending on the system size, mode of operation, agency standards, etc. Common types of documentation produced are User's Guides, Maintenance Manuals, and Operator Guides.

19. Training

The users are provided with instruction on how to use the new system. Some data processing shops train all potential users of a system while others train only representative users who must go back and train additional users. Some shops make programmer/analysts available to provide refresher training to users once a system has been implemented, and others provide help-centers to facilitate the learning process.

20. Turnover to Production Mode

At this point the programmer and/or analyst completes the documents necessary to transfer the system from a test mode to a production library and turn responsibility for running the system over to operations/production control or to the user to carry on the day-to-day running of the system.

Some systems can require constant modifications to the code in order to sustain an operational status, and in a sense are never turned over to the user group.

21. Maintenance

Maintenance is the process of: correcting programs to remove logic errors; perfecting programs to produce new output, use new input, or perform additional functions; or adapting programs to changes in the hardware/software environment. Requests for maintenance are usually initiated by the person responsible for running the system or by the user of the system. Requests for corrective maintenance are usually initiated by a phone call, with hard-copy follow-up to help identify the problem. Requests for enhancements to the system are usually more formal and are often handled in the same way as requests for new software. Requests for adaptive maintenance are usually generated from within the DP organization whenever new hardware or software will change the operating environment of the subject system.

C. Sample Descriptive Models

This section includes 5 descriptive models of the ASOF development and maintenance process found in 5 different Federal DP shops.

Exhibit III-2 shows which of the 21 tasks in the composite descriptive model are performed in the Federal DP shops surveyed, and were, for the most part, confirmed by the agencies reviewing the descriptive model. For the organizations that were included in the survey and that reviewed the "draft" descriptive model, Exhibit III-2 indicates the distribution of the 21 tasks by the different types of data processing shops. The 21 tasks in the composite model were adequate to describe the highly structured to the resource constrained environments encountered in shops surveyed. Depending upon the type of DP shop, the frequency of occurrence of the tasks varies as indicated in the exhibit. Some of the data processing organizations surveyed experience all of the operating modes noted by "Type of DP Shop" in Exhibit III-2.

EXHIBIT III-2. DESCRIPTIVE MODEL TASKS BY TYPE OF DP SHOP

DESCRIPTIVE MODEL TASKS	TYPE OF DP SHOP				
	STAFF SHORTAGES	"CRUNCH" MODE	STRUCTURED TECHNIQUE	PACKAGED SOFTWARE	TRADITIONAL
1. USER REQUEST	●	●	●	●	●
2. DATA SYSTEM PROBLEM DETERMINATION	●	◐	●	●	●
3. FEASIBILITY STUDY	●		●	●	●
4. FEASIBILITY STUDY REVIEW	●		●	○	●
5. EVALUATE ALTERNATIVE APPROACHES	●	○	●	●	◐
6. PREPARE SYSTEMS DEVELOPMENT PLAN	○	○	◐	○	●
7. SOFTWARE PACKAGE INVESTIGATION	○	○	◐	●	◐
8 PRELIMINARY PROGRAM DESIGN	●	●	●	●	●
9. PROGRAM DESIGN APPROVAL	●		●	○	●
10. MODULE SPECIFICATIONS	◐	●	●	●	◐
11. MODULE LOGIC DEVELOPMENT	●	○	●		◐
12. MODULE LOGIC PEER REVIEW	●	○	●		●
13. CODING	●	●	●	●	●
14. TEST DATA DEVELOPMENT	●	●	●	○	●
15. UNIT TESTING	●	●	●	●	●
16. SYSTEM TESTING	●	●	●	●	●
17. PARALLEL TESTING	●	○	●		●
18. DOCUMENTATION FINALIZATION	●	◐	●	○	●
19. TRAINING	●	◐	●	●	●
20. TURNOVER TO PRODUCTION MODE	●	○	●	●	●
21. MAINTENANCE	●	●	●	●	●

Blank - Not done
○ - Done rarely
◐ - Done half the time
● - Done almost always

In at least half of the cases, however, one of the "Type of DP Shop" categories represents their typical ASOF development environment. In most cases, where some formal procedures are adhered to for ASOF development and maintenance, the majority of the tasks in the composite descriptive model were carried out. Variations occur where there are staff, budget or time constraints imposed on the DP shop. For example, Exhibit III-3 illustrates the descriptive flow model for a shop with staffing constraints that relies heavily on various contractors for different tasks throughout the ASOF development and maintenance process. In contrast, Exhibit III-4 depicts the descriptive flow model for a DP shop which is under heavy time or budget constraints, a "crunch" mode, in its ASOF process. In this case, 13 of the 21 tasks are skipped or at best implicitly accommodated in the tasks performed.

In Exhibits III-5, III-6 and III-7, there were no severe resource constraints and the differences in the descriptive flow models is principally due to procedural differences in the ASOF process. Exhibit III-5 illustrates a traditional DP shop that has implemented structured techniques to develop application software instead of attempting to utilize "off-the-shelf" packages. In that setting, Task 5 - Evaluation of Alternative Approaches, tends to "absorb" much of Task 6 - Prepare Systems Development Plans, and Task 7 - Software Package Investigation is largely omitted.

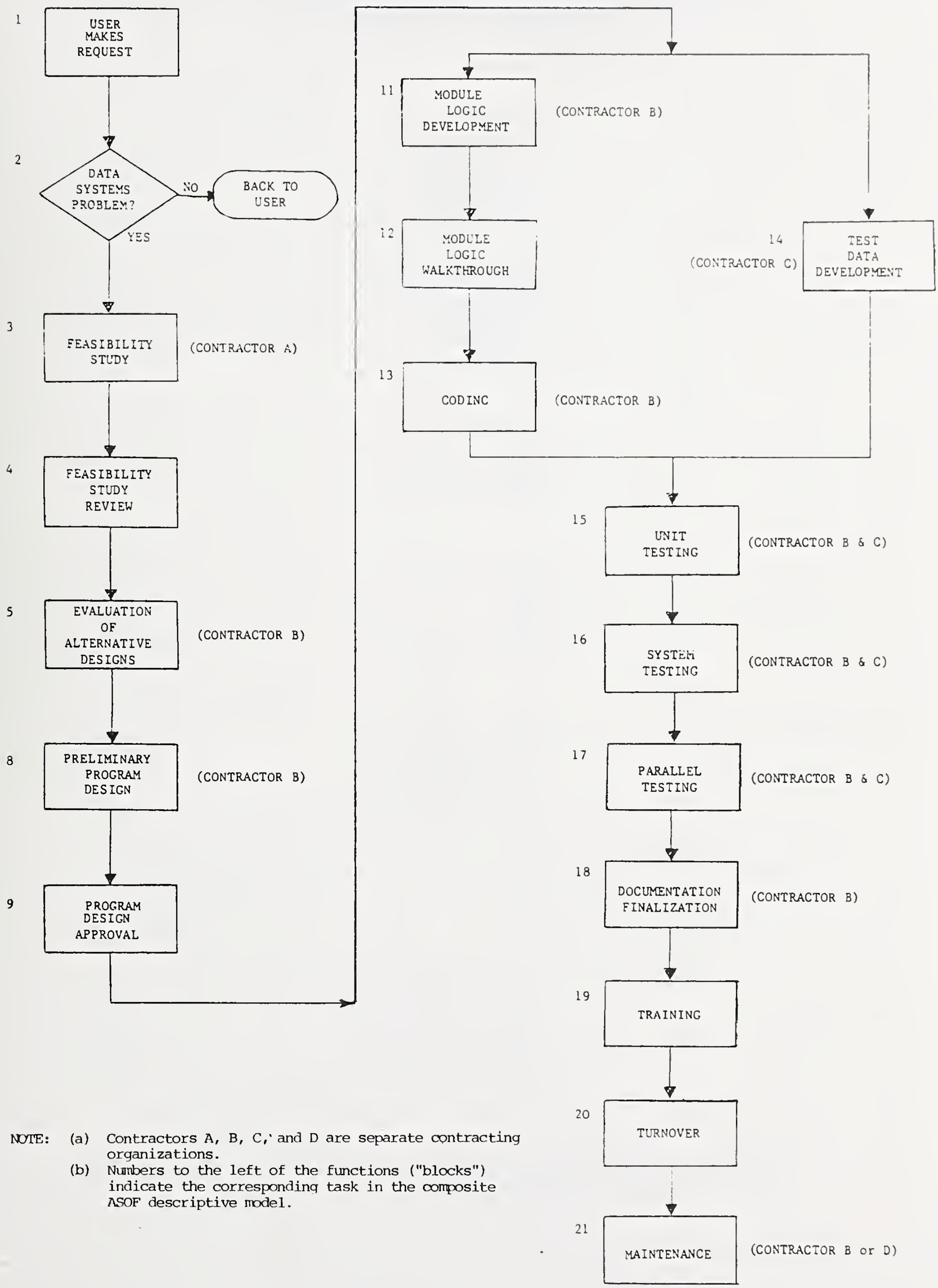
On the other hand, Exhibit III-6 depicts a small DP shop that relies heavily on available, off-the-shelf, software packages. In this case, several tasks:

4. Feasibility Study Review
9. Program Design Approval
11. Module Logic Development
12. Module Logic Peer Review
17. Parallel Testing
18. Documentation Finalization

are essentially omitted or at best dealt with implicitly in one or more of the other tasks. In a sense, software packages are used by this shop to short-cut the traditional ASOF development and maintenance process.

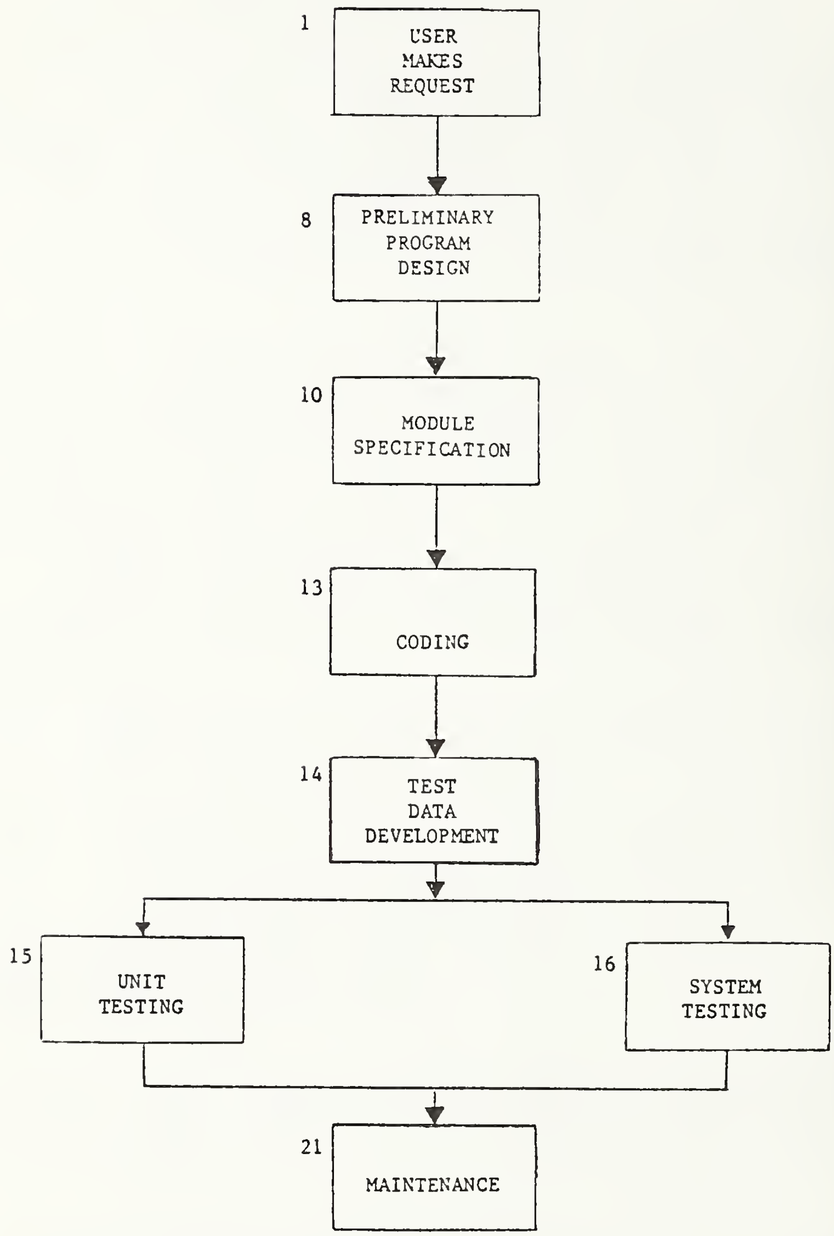
The last descriptive model, Exhibit III-7, is of a "traditional" DP shop that does ASOF development in a rather classical way. In this case, there is no apparent emphasis on feasibility and cost-benefit analyses, module specification, acceptance testing and formal documentation.

EXHIBIT III-3. SHOP WITH STAFF SHORTAGES USING CONTRACTORS FOR MOST OF ASOF DEVELOPMENT & MAINTENANCE



NOTE: (a) Contractors A, B, C, and D are separate contracting organizations.
(b) Numbers to the left of the functions ("blocks") indicate the corresponding task in the composite ASOF descriptive model.

Exhibit III-4. DATA PROCESSING SHOP WHICH IS ALWAYS WORKING IN "CRUNCH" MODE



NOTE: "Crunch" mode indicates that the shop is constantly facing tight deadlines.

EXHIBIT III-5. TRADITIONAL ASOF DEVELOPMENT SHOP USING SOME STRUCTURED TECHNIQUES

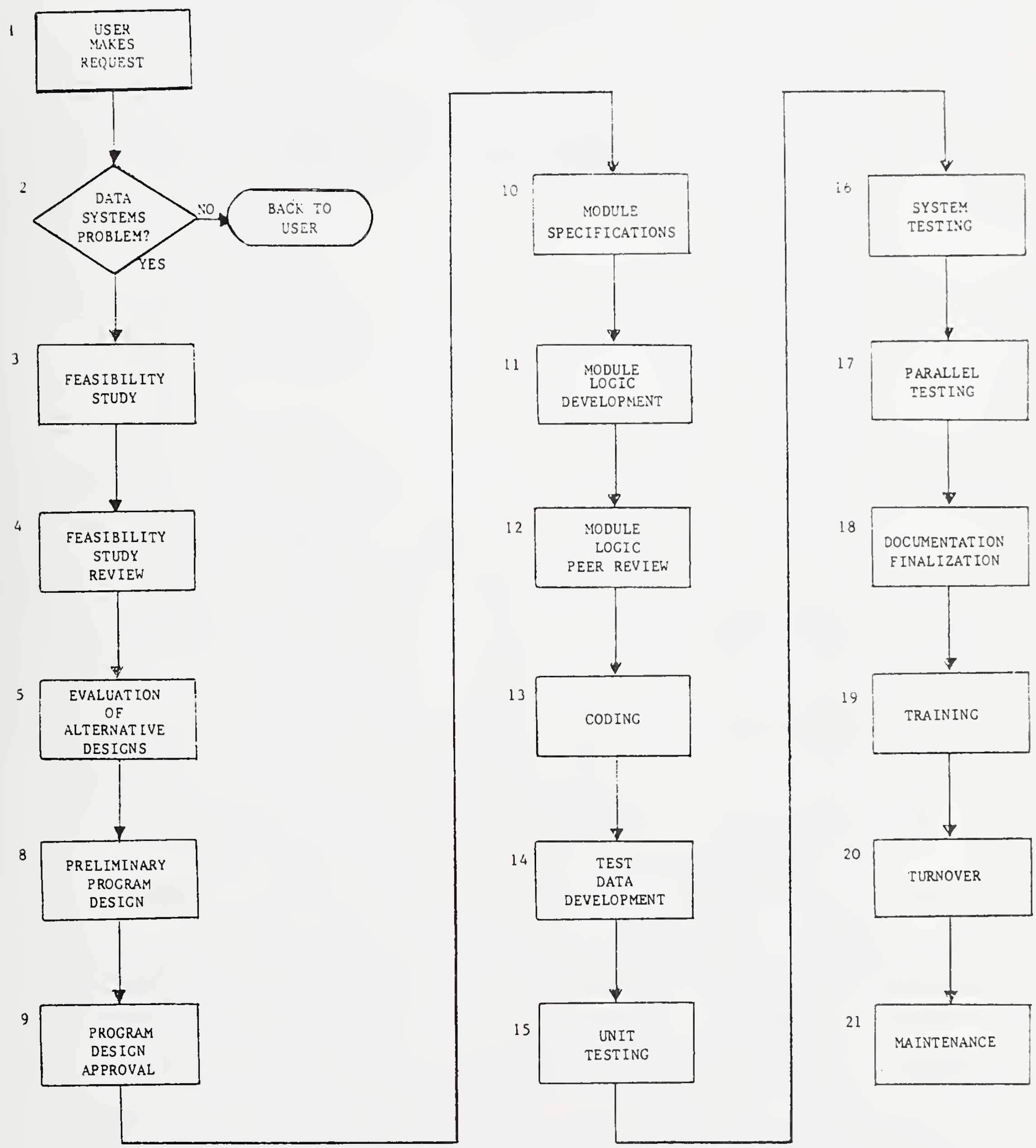


EXHIBIT III-6. SMALL SOFTWARE SHOP SOLVING ASOF PROBLEMS WITH PACKAGED SOFTWARE

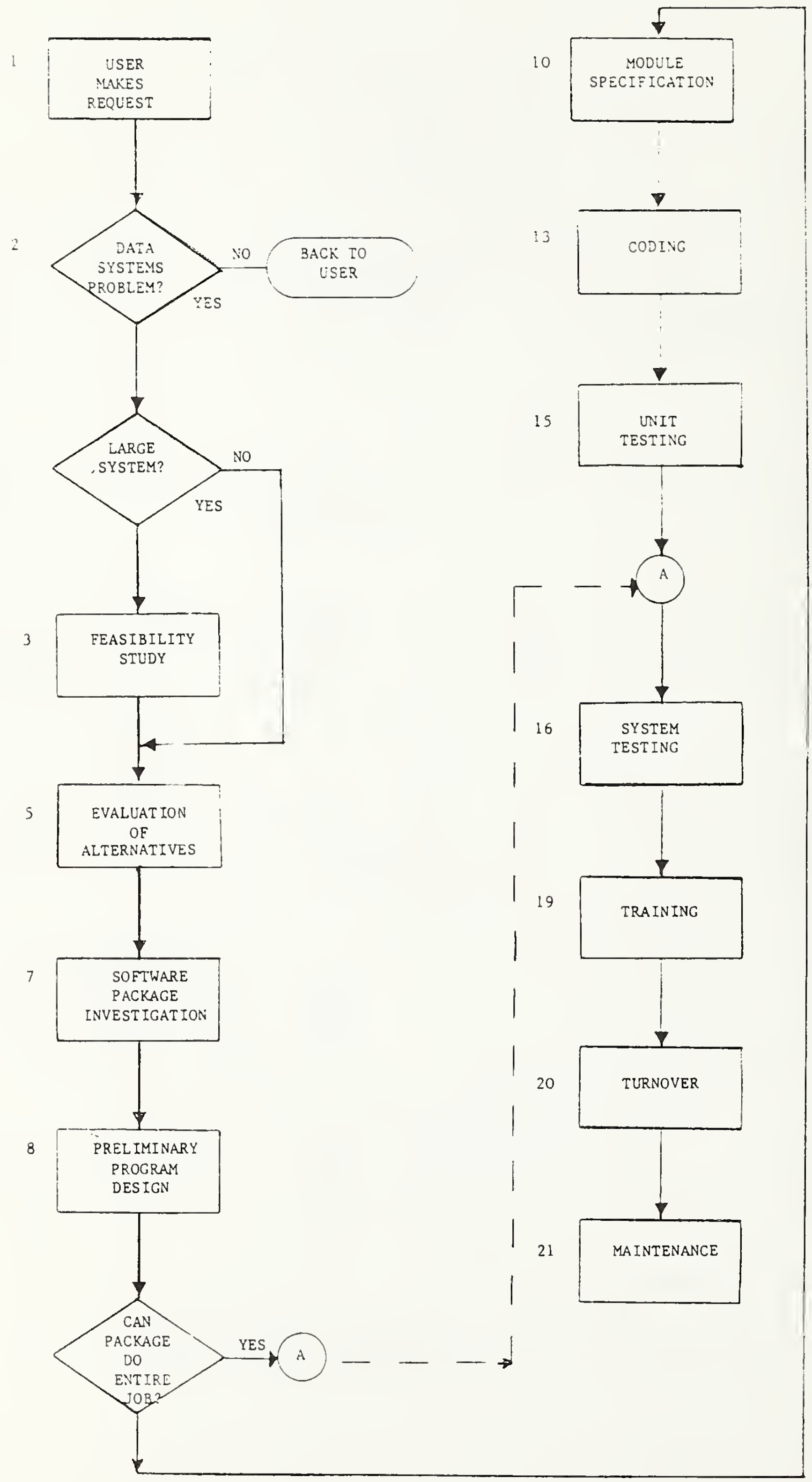
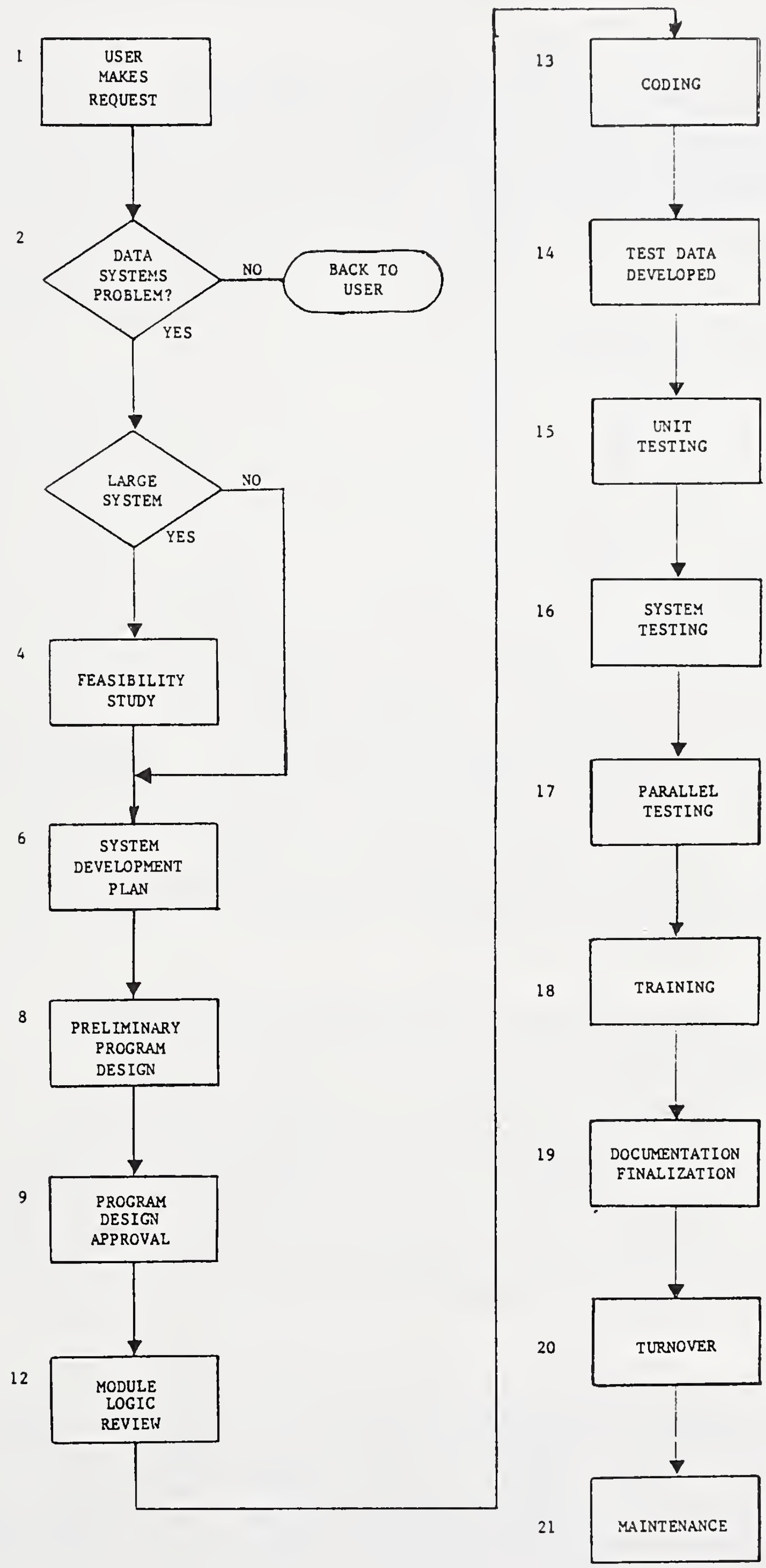


EXHIBIT III-7. TRADITIONAL DP SHOP



IV. COMPARISON OF THE MODELS

A. General

We have now described in detail both a rhetorical model (i.e., how the process is thought to be occurring) and a descriptive model (i.e., how the process was observed to be occurring) of the software development and maintenance process. Since both models represent the same process, there are a great number of similarities in the descriptions, as can be seen in Exhibit IV-1. There are, however, several areas where there are significant differences, and we will identify and discuss these differences in more detail in Section B.

Those in the Federal DP community who compare their own installation's process with that of the rhetorical model will undoubtedly find differences. We do not mean to imply that those differences necessarily indicate that something "wrong" is being done or that the rhetorical model describes the only correct way for software to be developed. It is, however, useful -- for several reasons -- to compare processes to determine where differences occur. First, these differences may suggest the need for further investigation to determine whether or not they are problem areas. Second, the rhetorical model represents the way many people believe that applications software is developed. Some of these people are in positions where their actions or decisions may affect the ASOF process. If those actions or decisions are based on incorrect assumptions about the process, unintended impacts could easily result. By way of contributing to an improved software development process, a number of observations based on the on-site interview results are presented in Section C.

B. Significant Differences Between the Models

In the initial part of the development process, Step 1 - Application Identification and Project Selection, of the rhetorical model roughly corresponds to Tasks 1 - User Request and Task 2 - Data System Problem, of the descriptive model. The rhetorical model indicates that a preliminary plan of personnel time and budget should be done here. In fact, we found that if a plan is prepared at all, it usually occurs later in the process when the analyst and DP management have a better idea of what is to be done.

Step 3 - Systems Analysis and Cost-Benefit Analysis, of the rhetorical model includes elements of what the descriptive model calls Feasibility Study (Task 3), Feasibility Study Review (Task 4) and Evaluation of Alternative Approaches (Task 5). One major difference between the models in this area is in the use of structured analysis techniques. While the rhetorical model indi-

Exhibit IV-1. THE RHETORICAL MODEL VS THE DESCRIPTIVE MODEL

Rhetorical Model Steps	Descriptive Model Tasks
Step 1 Application Identification and Project Selection	Task 1 - User Makes Request Task 2 - Data Systems Problem? Task 6 - Prepare Systems Development Plan
Step 2 System Requirements	
Step 3 Systems Analysis and Cost-Benefit Analysis	Task 3 - Feasibility Study Task 4 - Feasibility Study Review Task 5 - Evaluation of Alternative Approaches
Step 4 System Design	Task 8 - Preliminary Program Design Task 9 - Program Design Approval Task 10 - Module Specifications Task 14 - Test Data Development
Step 5 Program Development	Task 11 - Module Logic Development Task 12 - Module Logic Peer Review Task 13 - Coding
Step 6 Testing	Task 15 - Unit Testing Task 16 - System Testing Task 17 - Parallel Testing
Step 7 Training	Task 18 - Documentation Finalization Task 19 - Training
Step 8 Acceptance Test	
Step 9 Turnover	Task 20 - Turnover to Production Mode
Step 10 System Maintenance	Task 21 - Maintenance
Step 11 System Evaluation	

cates that structured techniques such as, Data Flow Diagrams, Data Dictionaries, Data Structure Diagrams and Structured Diagrams would prove useful at this time, the descriptive model, drawn from on-site interviews, shows little evidence that these techniques are regularly being used in the Federal Government. (We did interview one organization which tries to do a "data dictionary" at this stage if time is available.)

Step 4 - System Design, of the rhetorical model corresponds with parts of Task 8 - Preliminary Program Design, and Task 14 - Test Data Development, and all of Task 9 - Program Design Approval, and Task 10 - Module Specification, in the descriptive model. The most significant difference here seems to be the amount of user involvement. The rhetorical model indicates that the user is involved in the System Design step to review the design and to gain familiarity with what the system would or would not be doing. In reality, as the descriptive model indicates, the user is a source of information for the analyst and, in some shops, may review the program design document developed during this stage. However, we did not find any shop which actually did design reviews, and we saw little evidence that a user had review responsibilities during the Preliminary Program Design.* Another difference here is in the definition of test data. The rhetorical model indicates that test data is defined by the Systems Analyst during the System Design step. In most of the shops we visited to develop the descriptive model, the test data was defined/developed by the programmer after program coding was complete.

Step 5 - Program Development of the rhetorical model corresponds to Task 11 - Module Logic Development, Task 12 - Module Logic Peer Review and Task 13 - Coding, of the descriptive model. Here, one significant discrepancy between the rhetorical and descriptive models becomes apparent. The rhetorical model indicates that each step or substep will always occur during the development process. In fact, there are many shops which do <u>not</u> always do all of the steps in the rhetorical model, or which sometimes perform some of the steps in a different order. An example of this is that many organizations do not do Module Logic Reviews, although the rhetorical model indicates the belief that they do.

Step 6 of the rhetorical model, Testing, corresponds very closely to Tasks 15 through 17 of the descriptive model (i.e., Unit Testing, System Testing and Parallel Testing).

---

* One of the draft document reviewers did note that they conduct a Program Requirements Specification after Task 6, that includes an investigation of Software Packages (Task 7), Preliminary Program Designs (Task 8) and Program Design Approval (Task 9).

Step 7 - Training, of the rhetorical model corresponds closely with Task 18 - Documentation Finalization, and Task 19 - Training, of the descriptive model. However, the rhetorical model indicates when both the User's Manual and the Program Maintenance Manual should be started (i.e., Step 4 and Step 5) while the descriptive model allows for documentation to begin at any point in the process. In fact, several of the shops we visited did not start program documentation until testing was complete.

Step 9 - System Turnover, of the rhetorical model corresponds very closely with Task 20 - Turnover to Production Mode of the descriptive model.

Step 10 - System Maintenance, of the rhetorical model represents essentially the same functions as Task 21 - Maintenance of the descriptive model.

This, however, is where the close similarity between the two models ends.

The rhetorical model includes three steps which are not formally incorporated into the descriptive model, and the descriptive model has one task which is not explicitly indicated in the rhetorical model. The three steps missing from the descriptive model are System Requirements, Acceptance Test and System Evaluation. The task missing from the rhetorical model is Software Package Investigation.

The System Requirements step as described in the rhetorical model occurs rarely in the Federal DP shops interviewed. The rhetorical model states that during this step the users develop a document completely defining the desired system as they see it. In fact, we did not find any Federal DP shop which required or expected this type of document from its user community. It appears that much of the System Requirements step is actually performed not by the user, but by a DP analyst during the Feasibility Study and the Preliminary System Design tasks.

The System Evaluation step of the rhetorical model also appears to occur rarely in the Federal DP shops interviewed. One shop we visited indicated that they had performed a few of these evaluations, but that they were not part of their formal process. Most of the DP shops indicated that their workload was so great that there was not sufficient time available to go back and look at a completed system, unless it had problems. They also indicated that they were unaware of user shops regularly performing any type of formal user acceptance tests. In most of the Federal ASOF development processes with which we are familiar, the user group participates in the Parallel Testing effort (Task 17) but does not perform a separate acceptance task based on their own criteria.

Another area of difference between the two models is that the rhetorical model does not have a step or substep which corresponds to Task 7 - Software Package Investigation. We found that one of the most common complaints of DP shops was the shortage of staff available to them. Many shops are solving this problem by contracting out large portions of the ASOF development and maintenance process; however, one group we visited, whose staff cutbacks have made it impossible to manage a large number of contractors, has gone a step beyond. This group investigates and finds software packages which can meet all (or large portions of) their users' needs and then implements systems around these packages. We believe that this step in the ASOF process is becoming more frequent within the Federal government due to the staff reductions and the increased availability of generic software packages across the DP hardware spectrum.

C. Other Observations

As part of the on-site interviews, questions were included on organizational structure, standards, training, quality assurance, and many other related topics. These survey questions were designed to give a better understanding of the Federal DP shop environments in which the ASOF development and maintenance process occurs. After completing the survey, we made several observations which may or may not be pertinent to any given DP shop but which could require further investigation.

1. Standards Availability and Usage

Most of the shops we visited did have some type of software development standards based on the FIPS Pubs or Guidelines. However, most of these standards dealt only with programming methods, file naming conventions, standards on filling out forms, etc. None of the standards we reviewed included, for example, guidelines for analysts on how to perform a system analysis or design.

Even though standards were available, none of the shops visited had a planned review of the standards with new employees. The shops expected the new employees to learn the standards "when they need them" or from other staff members.

In several of the shops, no check was made to see if standards were being followed unless some document or program was being forwarded to another group or department, in which case it was reviewed for compliance with the standards at that time.

2. Quality Assurance

Only one of the shops which we visited, or with which we are otherwise familiar, has a functioning quality assurance group. The quality assurance group with which we are familiar reports to the same person as the head of the ASOF development group. The group is, thus, functionally separate from the ASOF development shop and has sufficient authority to enforce quality assurance. The group reviews all formal documents and programs produced during the ASOF development and maintenance process for completeness, correctness of form, standards adherence, and validity. No software produced in this shop can go into production until completely approved by the quality assurance group.

The other shops we are familiar with do not have groups which perform these functions. The closest to having quality assurance groups were organizations where the computer operations function has instituted some type of review in order to verify that systems turned over to them include adequate documentation and would run in the normal production environment.

3. Software Development Staff Training

In all of the shops which we visited, we were told that there was training available to all members of the software development group. However, it seems that, with the exception of trainees and senior staff attending seminars, little use is made of the training available. The main reason seems to be that the staff doesn't have time to learn anything new unless it is immediately required for a particular project. Therefore, the overhead associated with training staff in order to use a new language/package/technique for one particular application is regarded as unreasonable.

4. Software Development Tools

A few of the shops which we visited had, at one time or another, considered the possibility of using automated software generation and documentation tools. In those shops which had tried these tools, they had proved unsatisfactory and had been dropped. This left the shops with few tools available to assist the programmer/analyst in the ASOF development and maintenance process. Most shops used interactive input, editing and job submission, but few of the shops seemed to use interactive compilations, interactive scanning of job output, or interactive debugging tools in order to reduce the turnaround time and the number of batch printouts to be scanned by the programmer.

Several of the shops visited used automated test data generators; however, one shop indicated it had tried one of these generators and found it easier to generate test data by hand.

5. Program Reviews

Only one of the shops visited was actually using program reviews. The technique used was structured program logic walkthroughs. The staff indicated that this was one of the best things that had happened to their ASOF development and maintenance process. Each program was required to be presented by the programmer to the other programmers working on the project. This helped to reduce debugging time by identifying logic errors early. It also served as a training vehicle as there was always at least one senior programmer in attendance who could provide guidance in new or different approaches/solutions to a problem. The walkthrough also made other programmers on the team familiar with the program in case of personnel reassignments.

6. Cost Estimating for Program Coding/Testing

Each of the shops visited had techniques available for use in estimating time and costs for the development of a particular program. These techniques were usually implicit or hidden in the back of some file drawer. One group which we visited had a simple estimating formula which they had been using for this purpose, but hoped to get better estimates by having an outside contractor develop a new formula for them. After investing a good bit of time and effort, the contractor turned over a new formula which was very complex and considered many factors. After comparing the two formulas on a number of projects, it was determined that for that shop the old formula worked best. (The Beta distribution approximation of the normal distribution as applied in PERT, [MILL-63].)

Estimating Formula

$$\text{Mean value} = \frac{A+4B+C}{6}$$

where A = optimistic case estimate of time
B = modal estimate
C = worst case estimate of time

The time estimates using this simple formula were made after considering "all the factors I know from my own experience that go into the development process", i.e., subjective estimates based on the estimator's past experience with managing ASOF development projects.

Notwithstanding the accuracy of this subjective estimating approach or the implicit, diverse, and ever-changing methodologies employed by other facilities, it might be possible for DP managers to improve their estimates by incorporating some of the formalized approaches to time and cost estimating already available in the literature. Some of these approaches, [BOEH-81] and [PUTN-78], base their models on data derived from actual ASOF development projects. However, it has been noted that the accuracy for state-of-the-art software cost estimates is about ±20% of the actual time/cost 60-70% of the time [BOEH-81].

7. Software Package Usage

In spite of the growing number of application software packages and the slowly growing sophistication of the user, we found only one shop among those visited that conducted regular reviews of available software packages to use as a solution or partial solution for a particular ASOF problem.

Several of the staff interviewed were unaware of what packages were currently available on their own computer system. Also, only a few of those interviewed were familiar with the GSA Software Exchange Program. The generally held opinion was that packaged software could not meet the unique needs of their shop's users, but that it might meet the needs of some other agency.

8. Lack of Cost-Benefit Analysis

The use of a thorough cost-benefit analysis of potential systems seems to be extremely rare. Indications were that cost-benefit studies were only done for very large systems and then only considered the automation alternatives available, not whether or not the system should be implemented in the first place. In one case, a "benefit analysis" was performed during the final stages of the development process in an *a posteriori* attempt to cost-justify the implementation of the system.

9. Reduction in Maintenance with Packages

There seems to be an overall reduction in the amount of maintenance requested or necessary for those systems implemented using software packages. Both the group which makes extensive use of software packages, and other individuals familiar with systems using software packages indicated that use of a well-developed, debugged, and thoroughly tested package could decrease significantly the amount of maintenance necessary for the software application.

10. Lack of User Involvement

We found that in some shops there was a, perhaps surprising, lack of involvement of the user in the ASOF development and maintenance process. In these shops the users initiate the process with a request for a system, answer the analyst's questions regarding what they think is really needed, perhaps review one of the initial analysis documents, and then hear and see no more until the system is ready for implementation.

V. SUMMARY

This report presents a rhetorical model of how the ASOF development and maintenance process is thought to be performed, as derived from the current literature. This rhetorical model combined portions of the classical and neoclassical approaches to ASOF development and maintenance. After the development of the rhetorical model, interviews were held with Federal staff actually involved in the ASOF development and maintenance process in order to determine how software was actually being developed in the Federal Government. These interviews and the subsequent review process demonstrated that no two shops actually follow the same steps in the same sequence in order to develop their software. The descriptive model, presented in Chapter III, was developed to synthesize these findings. This model consists of 21 tasks, subsets of which can be combined in some sequence to represent the ASOF development and maintenance process of a particular shop. With the exception of certain scientifically oriented software development efforts, the composite descriptive model was found to be robust enough to describe the ASOF process in all the different types of Federal DP shops surveyed and in the set of Federal agencies that reviewed the draft document.

In Chapter IV, these two models were compared. As a result of this comparison, several differences between the models were noted. In addition, a number of observations derived from the interviews were presented. It is clear that not every ASOF development and maintenance group within the Federal Government goes about this process in exactly the same way. The important thing is to be able to identify those areas where differences exist. If there are differences, it may be helpful to understand how and why they are appropriate for that particular shop and organizational environment. The comparison of an individual shop with the model may suggest areas where improvements can be made.

GLOSSARY OF SELECTED TERMS

In the literature discussing application software a number of basic terms are used with varying interpretations. To avoid confusion in this document, the following terms are used as defined below:

Integration Testing	-	evaluation of the logic of a combination of program units or modules and their interfaces.
Module	-	a program unit that is discrete and identifiable with respect to compiling, combining with other (program) units or loading.
Peer Review	-	a process in which project personnel perform a detailed study and evaluation of code, documentation, or system specifications [POWE-82].
Program	-	a sequence of logic statements suitable for processing by a computer. A program is comprised of one or more modules.
System	-	a program or a logical combination of programs that performs one or more specific functions.
System Testing	-	evaluation of the logic of a combination of programs and their interfaces to determine if they satisfy a functional requirement.
Unit Testing	-	evaluation of the logic of a module or program unit.

BIBLIOGRAPHY

[AFDR-80] AFDSDC, ADS Project Approval/Development Process, AFDSDCR 300-8, USAF Headquarters, AFSDC, Gunter AFS, Alabama, March 31, 1980.

[AFR3-77] Department of the Air Force, "Procedures for Managing Automatic Data Processing Systems Documentation" Development, Acquisition, and Implementation," AF Regulation 300-12, USAF Headquarters, Washington, D.C., September 12, 1977.

[AUSH-81] Aurora Associates, Inc. and Shoemaker, D., "Product Utilization Survey of the Federal Information Processing Standards Program," (Test Phase Report), Report to the Institute for Computer Sciences and Technology, National Bureau of Standards, U.S. Department of Commerce, September 1981.

[BENJ-71] Benjamin, R.I., Control of the Information System Development Cycle, Wiley Interscience, New York, 1971.

[BERG-78] Bergland, G.D., "Structured Design Methodologies," 15th Annual Design Automation Conference Proceedings, 1978.

[BOEH-76] Boehm, B.W., "Software Engineering," IEEE Transactions on Computers, Vol. I, C. 25, No. 12, December 1976.

[BOEH-81] Boehm, B.W., Software Engineering Economics, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1981.

[BRAN-70] Brandon, D.H. and Gray, M., Project Control Standards, Brandon Systems Press, Inc., 1970.

[BROO-75] Brooks, F.B., The Mythical Man-Month: Essays on Software Engineering, Addison, Wesley, Reading, Massachusetts, 1975.

[BROO-80] Brooks, R.E., "Study Programmer Behavior Experimentally: The Problems of Proper Methodology," Communications of the ACM, Vol. 23, No. 4, April, 1980.

[CURT-80] Curtis, B., "Measurement and Experimentation in Software Engineering," Processings of the IEEE, Vol., 68, No. 9, 1980.

[CURT-79] Curtis, B., Sheppard, S.B., Milliman, P., "Third Time Charm: Stronger Prediction of Programmer Performance by Software Complexity Metrics," Proceedings of the 4th IEEE International Conference on Software Engineering," 1979.

[DALY-77] Daly, E.B., "Management of Software Development," IEEE Transactions on Software Engineering, Vol. SE-3, No. 3, 1977.

[DEMA-78] DeMarco, T., Structured Analysis and System Specification, Yourdon, Inc., New York, 1978.

[DERO-78] DeRoze, B.C. and Nyman, T.H., "The Software Life Cycle - A Management and Technological Challenge in the Department of Defense," IEEE Transactions on Software Engineering, Vol. SE-4, No. 4, July, 1978.

[DIMM-80] Distaso, J.R., Munson, J.B., Manley, J.H., and Stucki, L.G. "Software Technology - Key Issues of the '80's" Digest of Papers COMPCON '80, February, 1980.

[DIST-80] Distaso, J.R., "Software Management - A Survey of the Practice in 1980," Proceedings of the IEEE, Vol. 68, No. 9, September 1980.

[FIOR-78] Fiorello, M. and Jaffin, S., Costs and Benefits of Federal Automated Data Processing Standards: Guidelines for Analyses and Preliminary Estimating Techniques, Logistics Management Institute, Washington, D.C., 1978.

[FIOR-81] Fiorello, M., "Preliminary FIPS Cost-Benefit Guidelines: Assessment and Improvements," (Draft Working Note for ICST), Fiorello, Shaw and Associates, McLean, Virginia, May 1981.

[FREE-76] Freeman, P., and Wasseman, A., Ed., Tutorial on Software Design Techniques, third edition, IEEE Computer Society, 1980.

[GAO-81] Federal Agencies Maintenance of Computer Programs: Expensive and Undermanaged, General Acccounting Office, AFMD-81-25, February 26, 1981.

[GRAY-81] Gray, M.G., An Assessment and Forecast of ADP in the Federal Government NBS Special Publication 500-79, Institute for Computer Sciences and Technology, National Bureau of Standards, Washington, DC, 1981.

[ICST-82] Toward an Improved FIPS Cost-Benefit Methodology, Phase I: Descriptive Models - Data Processing Operations, NBS Special Publication 500-100, Institute for Computer Sciences and Technology, January 1983.

[LIEN-79] Lientz, B.P., and Swanson, E.B., "Software Maintenance: A User/Management Tug of War," Data Management, April, 1979.

[METZ-73] Metzger, P.W., Managing a Programming Project, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1973.

[MILL-63] Miller, R.W. Schedule, Cost and Profit Control With PERT, McGraw-Hill Book Company, Inc., New York, New York, 1963.

[MOHA-81] Mohanty, S.N., "Software Cost Estimation: Present and Future," Software-Practice and Experience, Vol. 11, 1981.

[MUNS-78] Munson, J.B., "Software Maintainability: A Practical Concern for Life-Cycle Costs," Proceedings of COMPSAC, November, 1978.

[NAKA-82] Nay, J.N. and Kay, P., Government Oversight and Evaluability Assessment, D.C. Heath and Company, Lexington, Massachusetts, 1982.

[OTTE-79] Ottenstein, L.M., "Quantitive Estimates of Debugging Requirements," IEEE Transactions on Software Engineering, Vol. SE-5, 1979.

[PARI-79] Parikh, G., "The World of Software Maintenance," Computer World, 1979.

[PETR-78] Peters, L.J. and Tripp, L.L., "A Model of Software Engineering," 3rd Annual Software Engineering Conference, Atlanta, Georgia, 1978.

[POWE-82] Powell, P.B., Ed., Software Validation, Verification, and Testing Techniques and Tool Reference Guide, NBS Special Publication 500-93, U.S. Department of Commerce, Institute for Computer Sciences and Technology, September 1982.

[POWE-83] Powell, P.B., Ed., Profiles of Computer Programmers in the Executive Branch of The Federal Government, NBSIR 82-2565, U.S. Department of Commerce, Institute for Computer Sciences and Technology, March 1983.

[PUTN-78] Putnam, L.H., "A General Empirical Solution to the Macro Software Sizing and Estimating Problem," IEEE Transactions on Software Engineering, July 1978.

[ROSS-77] Ross, D.T., and Scherman, K.E., "Structured Analysis for Requirements Definition," IEEE Transactions on Software Engineering, Vol. SE-3, January, 1977.

[SHEI-81] Sheil, B.A., "The Psychological Study of Programming," Computing Surveys, Vol. 13, March, 1981.

[SHEP-79] Sheppard, S., Curtis, B., Milliman, P. and Love, T., "Modern Coding Practices and Programmer Performance," Computer, Vol. 12, 1979.

[SCHN-80] Schneiderman, B., Software Psychology, Winthrop, Cambridge, Massachusetts, 1980.

[WINE-71] Wineberg, G.M., The Psychology of Computer Programming, Van Nostrand Rheinhold Company, New York, 1971.

[YOUR-75] Yourdon, E., Techniques of Program Structure and Design, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1975.

[YOUR-76] Yourdon, E., Managing the Structured Technique, Prentice-Hall, Englewood Cliffs, New Jersey, 1976.

[YOUR-78] Yourdon, E., Structured Walk-throughs, Yourdon, Inc., New York, 1978.

[YOUR-79] Yourdon, E., The Second Structured Revolution, Yourdon, Inc., New York, 1978.

APPENDIX

DATA COLLECTION PROCEDURE

This appendix describes the data collection procedure used to select candidate sites and prepare the data to construct the descriptive models presented in this Phase II report.

A. General

The data collection phase of this study was conducted over a seven month period beginning in September 1982, and ending March 1983. The four Federal sites ultimately included in the study* were selected on the basis of the following criteria:

- o Washington, D.C., area location;
- o desire of facility manager(s) to participate in the study;
- o time availability of candidate interviewees;
- o civilian facility
- o variation among selected facilities along key dimensions (e.g., staff size, varying percentage of in-house vs contractor-developed ASOF, custom developed ASOF vs predominant use of packaged ASOF).

Review of existing information on Federal DP facilities [POWE-83] and [AUSH-81] guided this selection process.

ICST reviewed and approved site selection criteria, and provided the project team with initial contacts at the sites.

B. Respondents

The same two members of the project team conducted a total of 17 on-site interviews, each lasting one to two hours. The professional categories of the 17 interviewees are broken out by site in Table 1.

C. Survey Instrument

Structured survey instruments to guide data collection efforts were constructed on the basis of an extensive literature review, the Federal DP experience of project team members, and briefings with ICST staff. Variations of the interview protocol were geared toward one of the following three types of DP personnel: managers, analysts, and programmers. Although the instruments served to standardize and guide the overall data collection effort, flexibility was maintained by pursuing, when appropriate, issues and areas not included on the instruments when the need arose.

---

* Four additional sites were approached to solicit their participation in the study, but, for varying reasons, they were not included in this study.

Table 1. STUDY RESPONDENTS
(By Site and Job Skills)

	Manager	Manager/ Analyst	Analyst	Programmer/ Analyst	Programmer	Total
Site 1	2	1	2	1	-	6
Site 2	2	1	1	-	1	5
Site 3	1	1	1	-	-	3
Site 4	1	-	1	1	-	3
TOTAL	6	3	5	2	1	17

D. Data Collection

The process involved in collecting the data for this study included the following three steps:

1. Gaining Entry to Each Site

Initial contacts with key individuals at selected sites were provided by ICST. The project team was responsible for initiating contacts with each facility based on this information as well as conducting all subsequent site visit and data collection activities.

Initial meetings were scheduled with the DP facility's manager(s) to discuss the purpose of the study and to request permission to interview staff members. An overview of all procedures involved in collecting the data was also provided, and the introductory letter and survey instruments were provided to the manager(s) at that time.

The DP managers selected the interviewees at each facility based on the project team's request to meet with a manager, analyst, and a programmer.

2. Conducting the Interviews

Each of the 17 interviews were conducted on-site by the same two members of the project team. Following each interview, one member of the project team consolidated the interview notes which were then reviewed by the other team member before forwarding them to the interviewee for comment and review. Interviewee comments were either received back in writing, or in cases where few changes were necessary, noted over the phone.

These edited notes from the 17 interviewees, along with the written documentation provided at the four sites visited, formed the basis for the findings presented in this report.

☆U.S. GOVERNMENT PRINTING OFFICE: 1984 420 997 202

NBS-114A (REV. 2-80)

U.S. DEPT. OF COMM. **BIBLIOGRAPHIC DATA SHEET** (*See instructions*)	1. PUBLICATION OR REPORT NO. NBS SP 500-116	2. Performing Organ. Report No.	3. Publication Date June 1984

4. TITLE AND SUBTITLE

Computer Science and Technology:
Toward an Improved FIPS Cost-Benefit Methodology, Phase II:
Descriptive Models--General Purpose Application Software Development and Maintenance

5. AUTHOR(S)

Mary Lou Chipman, Marco Fiorello, Monty Snead, Peg Kay and Patricia Powell

6. PERFORMING ORGANIZATION (*If joint or other than NBS, see instructions*)

National Bureau of Standards
Department of Commerce
Gaithersburg, MD 20899

TITAN Systems, Inc.
6833 Beverly Drive
McLean, VA 22101

Aurora Assoc., Inc.
1140 Connecticut Ave. NW
Washington, DC 20036

7. Contract/Grant No.

NB80SBCA0405

8. Type of Report & Period Covered

Final

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (*Street, City, State, ZIP*)

Institute for Computer Sciences and Technology
National Bureau of Standards
Gaithersburg, Maryland 20899

10. SUPPLEMENTARY NOTES

Library of Congress Catalog Card Number: 84-601071

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (*A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here*)

This report presents a functional-flow descriptive model that can be used to categorize the application software (ASOF) development and maintenance activities of Federal data processing facilities. ASOF-related activities may be conceptually represented in descriptive model form by combining one or more of the basic model tasks. The comprehensive framework for ASOF development and maintenance provided by the descriptive model can be used in the identification of impacts from standards and guidelines and in the preparation of cost-benefit impact assessments. The framework provides both macro and micro levels of detail in order to link the descriptive models to additional data processing issues.

12. KEY WORDS (*Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons*)

application software development; computer standards; cost-benefit analysis; data processing management; descriptive models; evaluability assessment; information systems.

13. AVAILABILITY

☒ Unlimited
☐ For Official Distribution. Do Not Release to NTIS
☒ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.
☐ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

55

15. Price

ANNOUNCEMENT OF NEW PUBLICATIONS ON COMPUTER SCIENCE & TECHNOLOGY

Superintendent of Documents,
Government Printing Office,
Washington, DC 20402

Dear Sir:

Please add my name to the announcement list of new publications to be issued in the series: National Bureau of Standards Special Publication 500-.

Name ______________________________

Company ______________________________

Address ______________________________

City ______________ State ______________ Zip Code ______________

(Notification key N-503)

NBS TECHNICAL PUBLICATIONS

PERIODICALS

JOURNAL OF RESEARCH—The Journal of Research of the National Bureau of Standards reports NBS research and development in those disciplines of the physical and engineering sciences in which the Bureau is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Bureau's technical and scientific programs. As a special service to subscribers each issue contains complete citations to all recent Bureau publications in both NBS and non-NBS media. Issued six times a year. Annual subscription: domestic $18; foreign $22.50. Single copy, $5.50 domestic; $6.90 foreign.

NONPERIODICALS

Monographs—Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NBS, NBS annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NBS under the authority of the National Standard Data Act (Public Law 90-396).

NOTE: The principal publication outlet for the foregoing data is the Journal of Physical and Chemical Reference Data (JPCRD) published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements available from ACS, 1155 Sixteenth St., NW, Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Bureau on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NBS under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NBS administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order the **above** *NBS publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*

Order the **following** *NBS publications—FIPS and NBSIR's—from the National Technical Information Service, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NBS pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NBS Interagency Reports (NBSIR)—A special series of interim or final reports on work performed by NBS for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.

U.S. Department of Commerce
National Bureau of Standards
Gaithersburg, MD 20899