

UNCLASSIFIED

AD NUMBER
AD465228
NEW LIMITATION CHANGE
TO Approved for public release, distribution unlimited
FROM Distribution authorized to U.S. Gov't. agencies only; Administrative/Operational Use; May 1965. Other requests shall be referred to Office of Naval Research, 800 North Quincy Street, Arlington, VA 22217-5660.
AUTHORITY
ONR ltr, 1 Apr 1981

THIS PAGE IS UNCLASSIFIED

AD 465 228

FILE COPY

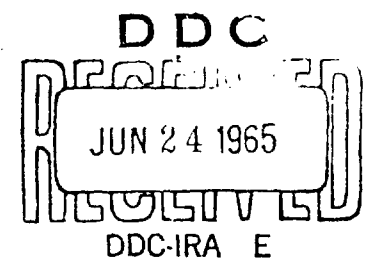


TM-2314/000/00

Planning Guide for Computer Program Developer

10 May 1965

Best Available Copy



7351

④ System Development Corp.
Santa Monica, Calif.

TM-2314/000/00

⑨ Technical memo.

⑭ Rept. no. TM-2314/000/00

⑮ Contract Nonr 4543 00

TECHNICAL MEMORANDUM

(TM Series)

This document was produced by SDC in performance of contract Nonr 4543(00)

all upper case
⑥ Planning Guide for Computer Program Development.

Prepared for
Office of Naval Research
Methodology Division

⑩ by Leonard Farr,
Victor LaBolle and
Norman E. Willmorth.

⑪ 10 May 1965,

⑫ 1119 P.

SYSTEM
DEVELOPMENT
CORPORATION
2500 COLORADO AVE.
SANTA MONICA
CALIFORNIA

The views, conclusions or recommendations expressed in this document do not necessarily reflect the official views or policies of agencies of the United States Government.

Reproduction in whole or in part is permitted
for any purpose of the United States Government.

Although this document contains no classified information it has not been cleared for open publication by the Department of Defense. Open publication, wholly or in part, is prohibited without the prior approval of the System Development Corporation.



10 May 1965

1
(page 2 blank)

TM-2314/000/00

FOREWORD

This Planning Guide for computer program development was prepared by the System Development Corporation to fulfill a contract with the Office of Naval Research, Nonr-4543(00). This Guide is intended to be a planning aid for Project Leaders at the Naval Command Systems Support Activity (NAVCOSSACT), particularly during their preparation of Planning Estimates and Project Development Plans. Two preliminary versions (TM-WD-1954/103 and /104) preceded this document, and these are hereby superseded.

The work leading to this Planning Guide is also part of a continuing study of programming management conducted by personnel from SDC's Programming Management Project. The material used in the Guide combines data from the files of the study members and data gathered at NAVCOSSACT. The material and concepts in the Guide reflect contributions from both organizations. Particularly, we wish to acknowledge the assistance and time given by NAVCOSSACT Project Leaders who participated in the early interviews that formed part of the data base for the Guide. Also, we wish to recognize the guidance and contributions of the following NAVCOSSACT personnel: R. Dolan, B. Mandel, J. Salvail, W. Kent, and E. Wolf. Guidance was also provided by J. R. Simpson, Office of Naval Research.

The following SDC personnel participated in the work: B. Nanus, J. N. Wallace, R. S. Steinert, V. LaBolle, N. E. Willmorth, and L. Farr.

10 May 1965

(page ³₄ blank)

TM-2314/000/00

ABSTRACT

This document offers a systematic approach for planning projects to develop computer-based information systems. The primary emphasis is placed on the computer program portion of such systems. A descriptive model of the development process forms the basis for a set of prescribed planning and management tasks. The model includes eight phases: (1) System Analysis, (2) System Design, (3) Program Development, (4) Program Coding, (5) Program Checkout, (6) User Documentation, (7) User Training and Assistance, and (8) Turnover. Each phase is further divided into tasks and subtasks for the purpose of more clearly understanding the elements of the development process. A detailed sequence of planning activities provides guidance for planning, scheduling and costing the tasks that comprise the development process, and forms are supplied to record the planning results and to serve as checklists for the required work. The forms and procedures also provide a basis for project control and for collection of data that may be used to improve estimates based upon experience. Although this Guide was prepared for use at the Naval Command Systems Support Activity, the material can easily be adapted to apply to programming in other organizations.

TABLE OF CONTENTS

FOREWORD	1
ABSTRACT	3
I. INTRODUCTION	9
A. The Source of Information	9
B. The Audience	9
C. Organization of the Guide	10
D. Maintenance and Use of the Planning Guide	11
II. BACKGROUND	
A. Assumptions for the Planning Guide	13
B. Some Characteristics of Information Processing Systems and Their Development	14
C. Characteristics of Computer Programming Projects at NAVCOSSACT	16
D. Characteristics and Trends of Programming at NAVCOSSACT	18
E. Communication and Coordination in Technical Management	20
F. Terminology	22
III. USING THE PLANNING GUIDE	23
A. Planning a Project	23
B. The Outputs of Planning	23
C. Inputs and Aids to Planning	25
D. The Planning Process	31

IV. GUIDELINES FOR ESTIMATION	53
A. Cost Factors in Program Development	53
B. Guidelines for Estimation	54
C. Research in Estimation of Programming Costs	59
V. SYSTEM ANALYSIS AND DESIGN	69
A. Objectives	69
B. Tasks	70
C. Communication, Coordination, and Control	71
D. Supervision	72
E. Cost Factors	73
F. The Check Sheets	74
VI. PROGRAM IMPLEMENTATION	109
A. Objectives	109
B. Tasks	110
C. Communication, Coordination, and Control	112
D. Supervision	113
E. Cost Factors	115
F. Schedules	116
VII. SUPPORT AND TURNOVER	145
A. Objectives	145
B. Tasks	145
C. Communication, Coordination, and Control	148
D. Supervision	149
E. Cost Factors	149
F. Schedules	150

PLANNING FORMS

10 May 1965

7
(page 8 blank)

TM-2314/000/00

LIST OF ILLUSTRATIONS

Chart I	The Program System Development Process	17
Chart II	Sample Project Summary Sheet (Page 1)	27
Chart III	Sample Project Summary Sheet (Page 2)	28
Chart IV	Sample Computer Program Planning Sheet	29
Chart V	Sample Program System Planning Sheet	30
Chart VI	An Example of a Gantt Chart for a Programming Project	47
Chart VII	An Example of Network Analysis for System Analysis	48
Chart VIII	Cost Factor-Task Matrix	55
Chart IX	Contract-Letting Processes and Delays	61
Table I	Relative Costs of Programming Project Phases	57
Table II	Time Estimates for Review and Other Activities at NAVCOSSACT	60
Figure I	Number of Man Months versus Number of Instructions	65
Figure II	Number of Computer Hours versus Number of Machine Language Instructions	66
Figure III	Computer Hours versus Man Months	67

I. INTRODUCTION

This Planning Guide is intended as an aid for managers of automatic data processing (ADP) development efforts. To help a manager plan for development of computer programs, the Guide supplies techniques and advice on how to sequence the work and estimate the costs and schedules. The nature of program development is depicted to identify its influence upon the job of the technical manager. Although the Guide was prepared for use at the Naval Command Systems Support Activity (NAVCOSSACT), the material applies to planning for a much broader spectrum of programming in other organizations.

A. THE SOURCE OF INFORMATION

As a relatively new discipline, computer programming lacks proven, systematic techniques for planning and comparing the planned efforts with completed efforts. On the other hand, considerable experience has been amassed, and this Guide extracts and organizes elements of this accumulated experience. More specifically, the Guide synthesizes analyses of programming experience at NAVCOSSACT and other organizations and divides the programming process into 36 tasks. A generalized example of how to plan for these tasks is given. For example, a Gantt Chart shows the time sequence of the tasks.

B. THE AUDIENCE

The Guide provides advice on how to plan the development of a program system. At NAVCOSSACT, such an effort is known as a Project and is managed by a Project Leader. Therefore, the contents of this Guide are addressed primarily to Project Leaders at NAVCOSSACT and, to the extent that similarities exist in the work of other programming organizations, the remarks are then also addressed to the Project Leader's counterpart. The Guide may well be helpful to other levels of programming management as well as to users and buyers of computer programs.

Although the primary aim is to provide guidelines for planning a programming project, i.e., an effort that results in delivery of a computer program to a customer, the Project Leader will find some of the initial steps useful for both planning and conducting a feasibility study of the proposed effort. As he progresses through the preliminary analysis and design tasks called for in planning, he will recognize the specific requirements and assess the feasibility of meeting them.

C. ORGANIZATION OF THE GUIDE

Some of the organization and the vocabulary used in the Guide reflect current guidance for programming managers within NAVCOSSACT. In addition to interviews and data gathered at NAVCOSSACT, the primary reference is NAVCOSSACT Instruction 5230.1A, Project Management Manual, 17 July 1964. To facilitate the possible use of this Guide in other programming organizations, the various terms, documents, and procedures that are unique to NAVCOSSACT have been explained in the text, particularly in Section II, Background, which provides several kinds of material:

- . translation from the NAVCOSSACT framework to a more general environment for computer program development;
- . a generalized concept of computer programming to establish a common ground with the reader. Included are some characteristics of computer program development and computer programs as products and their implications for the manager of program development; and
- . some additional terminology for NAVCOSSACT readers to help them interpret the advice in the Guide.

Following this background material, Section III, Using the Planning Guide, provides a step-by-step procedure for developing a Project plan. This section introduces the various aids that are contained in the remainder of the document, such as forms for recording planning information.

Section IV provides some specific guidance on how to estimate the various resources and the elapsed time needed for computer program development. It also lists factors that influence computer programming costs and shows their influence on various tasks that constitute program development.

Each of the remaining sections, V through VII, corresponds to one of three broad activities in computer program development at NAVCOSSACT. These activities, labeled Analysis and Design, Program Implementation, and Support and Turnover, are in turn divided into Phases to further describe that particular activity.

ANALYSIS AND DESIGN ACTIVITY

System Analysis Phase
System Design Phase

PROGRAM IMPLEMENTATION ACTIVITY

Program Development Phase
Program Coding Phase
Program Checkout Phase

SUPPORT AND TURNOVER ACTIVITY

User Documentation Phase

User Training and Assistance Phase

Turnover Phase

The sections V through VII are organized alike. First, an introduction outlines and describes the purpose of the broad activity, followed by a review of the work in the Phases, and then typical technical management problems and cost factors are discussed. In addition, the task descriptions that constitute the bulk of each of these sections are previewed. These task descriptions, called Check Sheets, are provided in a format modeled after a typical computer program transfer function. That is, the Check Sheets provide a statement of Inputs, Subtasks, and Outputs for each task. In addition, they include statements about the Environment in which the task will be conducted as well as specific factors that influence the cost of performing this task. For easy reference, each section of the Guide, as well as the Check Sheets for each Phase, have index tabs.

D. MAINTENANCE AND USE OF THE PLANNING GUIDE

Although the Guide is intended principally for planning, the text and forms provide a means for continuous control of a Project. Actual progress on a Project and the corresponding expenditures of time and resources may be compared to plans to determine if changes are needed.

Such changes are "normal" in programming efforts because of the relative absence of systematic, reliable techniques for prediction. Based upon this lack, the themes or the generalized advice for the planner in this Guide are:

- . plan in detail,
- . review and revise plans frequently, and
- . coordinate plans, both new and revised, with support organizations within NAVCOSSACT and with the system user.

To help the planner make numerical estimates, quantitative rules of thumb are given throughout the Guide. Based upon experience, these rules lack rigorous validation under controlled conditions, and the reader is cautioned to temper them with his own judgment and experience when he uses them. Information on the relative success of these rules is supplied when it is available. Use of the Guide and the forms provided is one way to accumulate detailed data to characterize programming experience. After sufficient data have been recorded for many Projects, they may be analyzed to derive quantitative planning factors that could be inserted in modifications of the Guide. An example of results from analysis of this type is shown in Section IV, Guidelines for Estimation.

Since this effort to set down planning principles for computer program development is essentially part of a learning process, the contents of the Guide are subject to change. (The looseleaf format permits pages to be inserted or removed easily.) Such changes may result from:

- a. the development of new insight into the programming process,
- b. changes in requirements for programs, and
- c. advances in technology such as new computers or programming techniques.

II. BACKGROUND

To make the best use of the Planning Guide, the reader should have some understanding of the concepts on which it is based. Therefore, this section presents background material on:

- . Some characteristics of computer programs and programming projects, both in general and in particular, at NAVCOSSACT, and
- . Some factors that contribute to the difficulty of programming management.

A. ASSUMPTIONS FOR THE PLANNING GUIDE

In preparing this Planning Guide, certain assumptions were made about the development of program systems. These assumptions, some of which are elements of the programming process, management structure, and policy now used at NAVCOSSACT, are as follows:

- . The production of a program system should be organized as a Project with a series of Phases (System Analysis, System Design, Program Development, Program Coding, Program Checkout, User Documentation, User Training and Assistance, and Turnover) composed of tasks and subtasks. This model conforms to the way in which program development is organized at NAVCOSSACT and does not differ in any major way from practice in other programming organizations.
- . A Project begins with the receipt of a Project Request,* that may be preceded by gross requirement analyses, feasibility studies, and overall system design, and ceases with the acceptance of the system by the customer after a shakedown period.

*Project Request--a document submitted by a user indicating the objective, the concept of operation, the tasks, the security classification of the desired data-processing capability. The document also calls out the earliest date the capability could be accepted, the latest date for turnover and an optimum date. These Requests trigger the planning process at NAVCOSSACT.

- . A programming Project exists in a dynamic environment, requiring that such documents as Project Estimates,* Project Development Plans,** functional descriptions, and other statements of system requirements and program specifications be kept up to date.
- . Advice for the Project Leader should be devoted to actual program development, e.g., evaluation and feedback on completed work, shifting forces as tasks prove difficult or easy, etc., rather than being devoted to basic and concurrent supervisory tasks such as training and personnel evaluation.
- . Documentation of programs and programming should be encouraged in intermediate as well as final stages, to:
 - a. increase the tangibility of work results;
 - b. promote continuity of work;
 - c. create a file of program designs, development techniques, test plans, etc., for future use; and
 - d. promote continuity of information and its communication in systems following an evolutionary design and implementation plan.
- . Review, validation, and inspection of products should be stressed to insure proper performance and compatibility with other systems and products affected by their design and operation.

B. SOME CHARACTERISTICS OF INFORMATION PROCESSING SYSTEMS AND THEIR DEVELOPMENT

Consideration of the nature of information processing systems and their development will help the Project Leader to plan. Projects to develop computer program systems are characterized by the following:

*Project Estimate--a document prepared by a Project Leader containing estimates of dollar costs, man month costs (contractor and in-house), computer time, a gross schedule (total elapsed time) and NAVCOSSACT rating factors for Project complexity (see NAVCOSSACT Instruction 5230.1A). This early document is used primarily to obtain management agreement to proceed with the Project.

**Project Development Plan--a more detailed forecast dividing the Project work into the eight Phases with schedules and manpower estimates for each. Number of instructions, computer time, and dates for key milestones are estimated. This document is used as a basis for status reports once the Project is underway.

- . The products are, by and large, intangible. That is, they are procedures and algorithms, ideas and concepts, organizations and flows, efficiencies and optimizations rather than hardware.
- . The products are intimately related to the operations, and mode of operation, of the system user.
- . The user, in many cases, may not have a clear idea of precisely what he needs when a Project begins, may be unable to communicate the need to program developers, or may be reluctant to release the information for security or proprietary reasons.
- . Program system products and programming processes are subject to frequent change.
- . A program system typically takes a long time to design and produce, and consequently suffers from loss of information through obsolescence and turnover of personnel.

These characteristics are the sources of many problems inherent in the development of program systems:

- . The intangibility of programs and procedures makes it difficult to evaluate the product that has been produced.
- . Ambiguity in specifications makes comparison of systems and system features indecisive and leads to disagreements over whether the product really satisfies stated requirements and specifications.
- . Ambiguity in statements of work leads to the failure to recognize the importance of individual tasks and the impact that poor performance in a task may have upon others.
- . Intangibility and ambiguity result in overemphasis upon tasks that lead to visible "hard" products and underestimation of the difficulty and cost of less tangible tasks such as "supervise," "coordinate," and "evaluate."

One way to resolve uncertainty and to provide tangible products for programming activities is through documentation. Some tasks have documentation built into them ("develop tests," "flowchart," "specify"), and others may be designated as documentation tasks by requiring reports ("produce user documentation"). The number of different types of documents usually depends upon user needs and the policy and work procedures in the programming organization. How much documentation a Project will require depends upon its complexity, size, duration, and changeability, and is partially determined by the individual Project Leader. Although a theme in this Planning Guide is thorough, accurate,

and up-to-date documentation during program development, the danger of developing large amounts of expensive, useless documentation exists. Therefore, managers must determine the usefulness of various document types and their content; but, in general, too little documentation is the problem in programming rather than too much.

C. CHARACTERISTICS OF COMPUTER PROGRAMMING PROJECTS AT NAVCOSSACT

The sequence of work to develop a computer program may be divided and labeled in different ways. But, when these various descriptions are examined in detail, they are quite similar. Chart I is a model in block diagram form of the computer program development process as it is viewed at NAVCOSSACT and described in this Guide. The diagram illustrates three major Activities: Analysis and Design, Program Implementation, and Support and Turnover. Within these broad areas are the specific subordinate activities that constitute the process. The subordinate activities are known as Phases, at NAVCOSSACT, and are so termed in this Guide. The Phases are grouped within the major Activities as follows:

1. Analysis and Design Activity

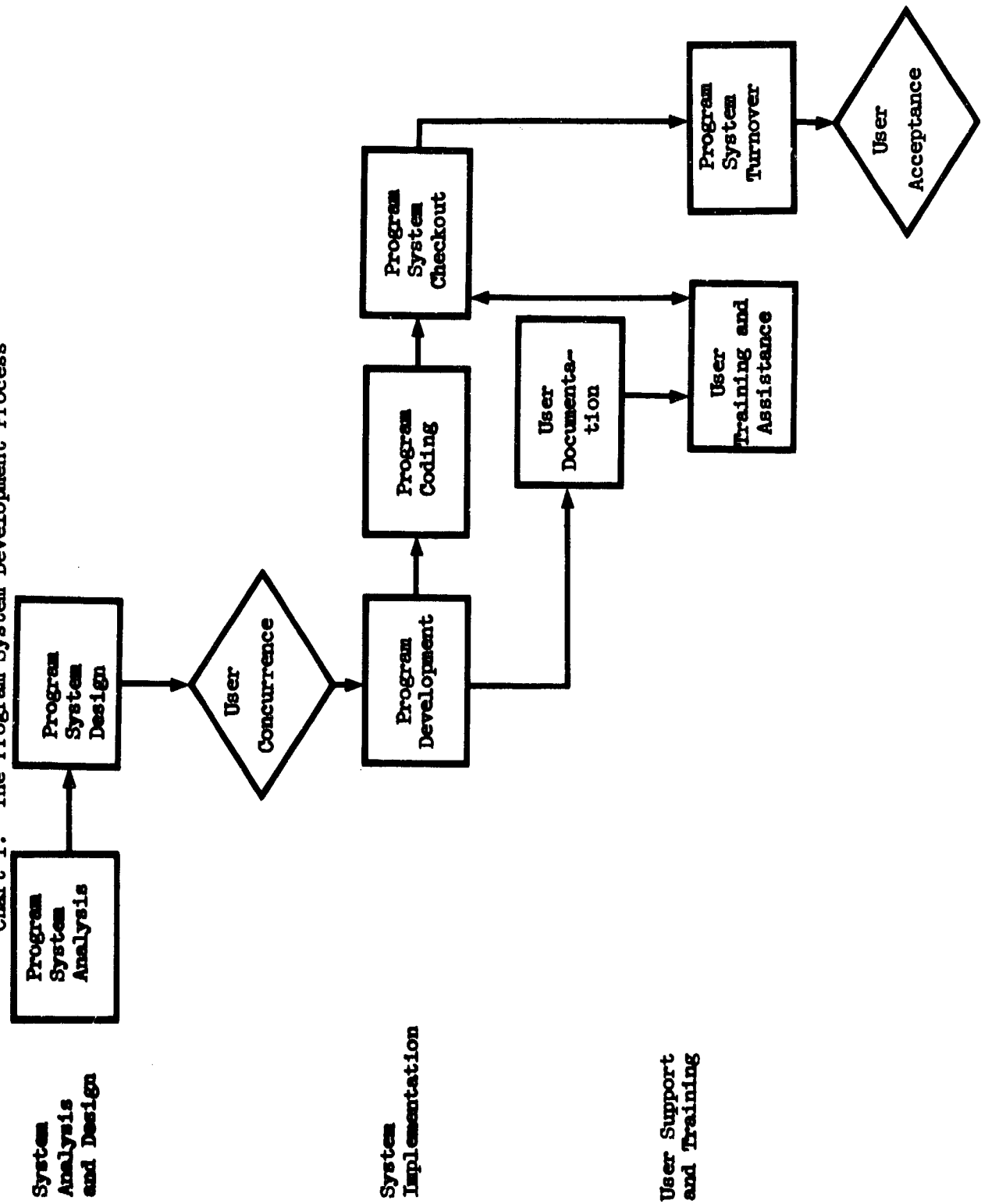
- . System Analysis Phase--the investigation of an information processing problem, particularly the need for a new ADP system or a change to an existing one and the conditions that may surround the development.
- . System Design Phase--the creation of a scheme, or ADP design, to satisfy the requirements of the user.

At NAVCOSSACT, this Activity also includes the Requirements and Feasibility Studies.

2. Program Implementation Activity

- . Program Development Phase--the actual design of the programs, program system test, and program and system files.
- . Coding Phase--the production of computer instructions to implement the program designs.
- . Checkout Phase--the inspection and test of the coded programs to insure that they satisfy both design specifications and operational requirements.

Chart I. The Program System Development Process



3. Support and Turnover Activity

- . User Documentation Phase--the ongoing documentation of the system, in the form of manuals or reports, and the preparation of formal documentation.
- . User Training and Assistance Phase--planning for and assistance in the indoctrination of the user into the use of the system. This Phase also includes the collection and conversion of system data, which at NAVCOSSACT is a user responsibility.
- . Turnover Phase--the turnover of the program system to the customer and the shakedown of the programs in their operational environment. This work is shared with the customer, but NAVCOSSACT assumes the major responsibility. The word "Phase" implies a time sequence of these types of work. In general, this is true for the process of program system development, but some Phases, such as User Documentation, require continual work during the entire process. In some sense the time-sequence connotation of the word "Phase" still applies, since the time roughly indicated for this Phase would be a period of increased intensity for documentation.

Within, or supporting these broader Phases, many other major efforts may be needed, such as:

- . The production of utility and support programs.
- . The development of using and operating procedures.
- . The conduct of simulation studies.

D. CHARACTERISTICS AND TRENDS OF PROGRAMMING AT NAVCOSSACT

During preparation of this Guide, certain characteristics and trends were noted at NAVCOSSACT. To some extent, these are not unique to that organization, but reflect patterns of growth in other programming organizations (particularly in the government), as well as general trends in automatic data processing.

1. Characteristics

- . Program development only part of the total activity. Although program development is the central core of the work, other significant resource-consuming activities are also carried out. Thus, at NAVCOSSACT, the term "Project" refers to many efforts besides program development, e.g., operation of the Navy Information Center (NAVIC) computer and research and development work on programming standards.

- . A diversity of users. Requests for programs come from many widely dispersed sources.
- . A large number of small Projects. Most efforts appear to be small, in terms of machine-language instructions (e.g., less than fifteen thousand) and number of persons involved (e.g., less than twenty).
- . A growing organization with a division of the labor force. Continuing growth results in a shortage of trained personnel. In government organizations such as NAVCOSSACT, the labor force may consist of a mixture of contractor personnel, Civil Service personnel, and military personnel. Also, at NAVCOSSACT, reliance on outside help has caused Project Leaders to act as monitors of contracted Projects.
- . A fixed array of computers. The basic computers now and for the near future are relatively fixed, e.g., at NAVCOSSACT, the basic computers are the IBM 1401, IBM 7090, AN/FSQ-20, CDC 1604 and CDC 160 (or models of them).
- . Many reprogramming efforts. Many programming efforts require one data-processing capability at one operational center to be adapted to serve another center and/or the conversion of programs from one machine to another.
- . Many revisions of existing programs. Feedback from users and changes in procedures demand modifications to existing programs.
- . A "service bureau" policy. To provide program and development services to a large number of users, the organization must try to preserve its resources. To bound the ADP development process, an organization such as NAVCOSSACT defines maintenance as a user responsibility. Also, as indicated earlier, the extent of NAVCOSSACT participation during development is bounded; e.g., the user is primarily responsible for a potentially significant task--Data Collection and Conversion.

2. Trends

- . A reduction in software diversity. Despite the rapid expansion of ADP into many new fields, programming languages and tools and application programs have tended toward a common information-processing technology that cuts across areas of application. As a result, the stockpile of standard programs and programming techniques grows, and more work can be done to satisfy the need to transfer programs from center to center and convert programs from one machine to another. For example, NAVCOSSACT has used similar programs at more than one center and has standardized with a single programming language.

- . A growing adaptability to change. Programming techniques that permit more generality and flexibility are being developed to ease the accommodation of change. For example, at NAVCOSSACT, the data base has been "divorced" from the programs to permit quick changes to it.
- . A trend toward integrated systems. Although many applications still call for independent programs and will undoubtedly do so for some time, there is a trend toward integrating computers and program systems into operations. For example, even independent programs must operate under executive program control and communicate with centralized data base structures. With increasing use of computers on-line via data links, communication networks, and multiprocessors, an increasing number of multipurpose, multi-service, multiuser centers may be expected, with increasing demands on programming organizations such as NAVCOSSACT for tightly integrated computer programs to operate within these centers.
- . A continued growth of ADP. The actual and projected use of automatic data processing continues to grow in all areas of operations. For NAVCOSSACT this has meant a growing number of Project Requests that, combined with the shortage of available resources, leads to the use of a priority system and the "moth-balling" of low-priority projects for short periods of time.

E. COMMUNICATION AND COORDINATION IN TECHNICAL MANAGEMENT

The preceding material provides some background on the process and the products of programming. In this brief discussion, the Project Leader's job is examined, particularly in terms of how it intersects with the more elusive aspects of programming that do not yield hard products.

The uncertain operating environment of computer program development needs extensive and intensive communication, coordination, and control. The Project Leader must actively seek information, encourage and provide coordination and communication, and solicit feedback on every decision. Documentation appears again as a prime tool to promote feedback and verify information as well as to record and transfer experience and information.

In a job characterized by continual change, few standards, and fewer techniques for estimating costs, control is particularly difficult. The Project Leader cannot specify exactly what is to be produced, or precisely what must be done; as a result, he cannot easily evaluate completed work. Nevertheless, he must establish some mechanism for insuring that everything is done that should be, and that things that are not essential are not. Further, he needs a way to control, or at

least to know about, all changes. Precise documentation, extensive personal attention, and insistence upon coordination and concurrence procedures for all designs, procedures, and changes thereto are his management techniques.

The Project Leader must be aware of, integrate and coordinate major interactions or interdependencies among the following:

- . Tasks--the processes necessary to produce the system. Failure to recognize the importance of the individual tasks and their impacts upon one another is common and invariably leads to slipped schedules and degraded products.
- . Organizations--the people and agencies who are banded together to produce and use the system--the customers and the hardware and software developers. Although cost of the program system may be only a small part of the total system cost, the success of the entire system depends upon an effective program system. Program design and operation reflect (1) the design of equipment such as the computer, communication devices, and weapons, and (2) the information-processing policies and procedures in the using organization and others with which the program system interfaces. These design dependencies demand coordination contacts by the Project with equipment developers, using organizations, and interfacing organizations. Face-to-face interchange must supplement documents to overcome the language barrier that often exists between user or equipment development personnel and system analysts in the Project.
- . Products--the programs themselves and their corresponding documents. In using ADP, part of the trend toward integration of information processing using computers, computer programs, and communication equipment, particularly integration of many centers with on-line use of computers, is increased program system size and complexity. Interaction among programs in a program system is a factor frequently overlooked or underestimated in Project planning and costing. Such interactions call for increased coordination and communication between programmers working upon the individual programs that interface. Increased work for analysis and design is also needed, as well as the additional code to handle the communications among programs. In large, complex programs, this part of the program system for functions such as control and housekeeping may be a high percentage of the total code. Finally, larger, longer, and hence more expensive tests are needed to check out programs, subsystems, and the total program system.

F. TERMINOLOGY

This Planning Guide uses many of the technical and managerial terms in the NAVCOSSACT Project Leader's Guide (NAVCOSSACT Instruction 5230.1A). In addition, the following common terms not defined in the reference are defined here for all readers:

- . Routine--a section of code (computer instructions) in a program that performs some identifiable function and that is organized and identified as a logical entity.
- . Program--a set of one or more routines or sections of code that perform some identifiable function or set of functions and are organized to operate as a self-contained unit.
- . Program Subsystem--a set or subset of one or more functionally interdependent programs that operate together to perform a more or less self-contained data-processing task or phase within a larger system of program or other system components.
- . Program System--an interrelated set of one or more programs or program subsystems that perform the automatic data-processing functions of a system and are identified as belonging to the set. Routines, subsystems, and systems often relate to one another, through the operation of a control or executive routine, program, or subsystem.
- . Program Test--the application of a set of data and procedures to a program to assure the developers that the program will operate as specified (also known as a parameter test and "debugging").
- . Program Subsystem Test--a test of a subsystem to insure correct communication between the various interdependent programs that comprise the subsystems.
- . Program System Test--a program test applied to a total system of programs, often, but not necessarily, in a live environment, using "live" data (data generated during the actual operation of the system) to assure that various programs interact as specified and required, to determine if operational requirements for information processing are satisfied, and to evaluate ease of use and maintenance.
- . Program System Data Base--central data files, excluding tables and constants that are used only by individual programs and so are not stored centrally.

Following these background remarks on the nature of program development, its characteristic problems, and the more difficult parts of a Project Leader's job, Section III provides a recommended procedure that outlines, step by step, how to use this Guide to plan a programming Project.

III. USING THE PLANNING GUIDE

A. PLANNING A PROJECT

The objectives of Project planning are (1) to state, in some detail, the intermediate and final products of the Project, the work needed to develop these products and the expected conditions under which the work will be done, and (2) to estimate the kinds of resources needed and their costs in terms of elapsed time, manpower, and machine hours.

To plan a Project, in addition to specific knowledge of the job to be done, the Project Leader must know, in general, what to plan for, what sources of information he has, what aids he has to help him plan, and what procedures he should follow in planning. To help meet these needs, the Planning Guide (1) describes the contents of plans in terms of schedules, cost estimates, and product lists, (2) suggests sources of planning information, and (3) presents in this section some planning aids, e.g., forms, and a procedure for planning.

B. THE OUTPUTS OF PLANNING

The planning procedure outlined below carries planning for a programming project from its inception to the point where a detailed plan exists for the production of a system. As mentioned earlier, well-defined procedures do not exist for formulating an effective plan by using only a statement of requirements. The absence of such techniques means that to develop a plan, work must be done to define or analyze the particular information processing problem and even some work must be done to design programs that solve the problem. In addition to its contribution to planning, work of this type constitutes the first activity in program development, i.e., System Analysis and Design. Generally this work proceeds in iterative stages--first analysis then design--in which the detail increases and various alternatives are rejected at various decision points. At NAVCOSSACT, three major decision points and allied products are identified: first, a gross estimate of system feasibility and overall costs; second, an estimate of more detailed system requirements and a plan for system development; and, third, a detailed design for the system and a plan to produce it. Planning does not stop at this point; a Project Leader must take many more plans to detail the actions of his Project, and the existing plans, always subject to change, must be maintained. The issuance of these plans, however, may result in a management action to cease planning and development at any of these major decision points. At NAVCOSSACT these plans mark the following decisions:

1. Planning Estimate

As a result of a preliminary analysis of the Project Request and the work expected, the decisions that might be made are:

- . The system implied is desirable and technically and economically feasible and, tentatively, can be developed using internal resources. A rough estimate of Project duration and cost is made and more detailed analyses of requirements are undertaken. At NAVCOSSACT, this statement, called the Planning Estimate, must be prepared within 60 days following receipt of the Project Request.
- . The system is feasible, but cannot be produced with the available internal resources. Both the Planning Estimate and a more detailed analysis of requirements are used to prepare a statement of work and a request for proposal by an outside contractor.
- . The system is not feasible. A report of nonfeasibility is sent to the system requestor and the planning ceases.

2. Project Development Plan

As a result of more detailed analyses, a gross inventory of system requirements, functions, and environment is generated, and either the Project Leader or the selected contractor prepares a Project Development Plan that presents:

- . A gross estimate of system development and implementation requirements.
- . A preliminary flow diagram for the system.
- . A tentative schedule for the Project.
- . A tentative estimate of the manpower and computing time required to produce the system.

On the basis of this more detailed analysis, the decisions to be made are:

- . The system is technically and economically feasible and desirable and planning should continue. If appropriate, a contract is let.
- . The system is not feasible or desirable and development and planning should cease.

3. Project Implementation Plan

As a result of the completion of the System Analysis and Design activities, a set of detailed plans for the production of the system is made including:

- . Specifications of system requirements.
- . Specifications of the system design.
- . Evaluation of system implementation requirements.
- . Schedules for system production.
- . Cost estimates for system components and system production tasks.

On the basis of these, it is decided that:

- . The proposed system design and implementation plan are satisfactory and the implementation of the system is initiated.
- . The proposed plans are not satisfactory, but may be modified until they are acceptable.
- . The proposed plans cannot be made satisfactory and development and planning should cease.

C. INPUTS AND AIDS TO PLANNING

The planning procedure assumes that the Project Leader has the following aids:

1. A Project Request

A customer's statement of need and requirements that contains a general description of the system to be planned and the problems it must solve for the customer; the statement includes the user's indication of objective, operational concept, tasks, and security classification of the desired data-processing capability. The document also states the dates of earliest acceptance and latest turnover.

2. A Model of the Program Development Process

Sections V, VI, and VII of this Planning Guide contain Check Sheets that describe, a model of the program system development process, in terms of inputs, tasks and subtasks, expected outputs, and environmental factors. The process has been divided into a hierarchy--three activities consisting of eight phases, each divided into tasks and these in turn divided into subtasks.

3. A Project Summary Sheet*

Charts II and III present a suggested form for recording Project planning information as it is derived. One section of the form is used to record general descriptive information about the Project--its identification, its description, its environment, its operating conditions, its size, and various assumptions concerning the production of the system. The second section of the form is used to record the time and cost estimates that are made for each of the major tasks and phases of the program system development for the particular Project.

4. A Computer Program Planning Sheet*

Charts IV and V present suggested forms to assist in the detailed costing of designing, coding, and checking out individual computer programs, as well as an integrated set of these that constitute a system. The form shown in Chart IV is divided first into three parts--Program Development, Program Coding, and Program Checkout--that represent the major products and activities in the production of a program. Each subpart is divided into a description of the product or activity, and a series of major tasks to be performed in producing each product. Information derived using this form for each program comprising the Program System is totaled and summarized on the Project Summary Sheet. For large program systems, several programs may be tested together prior to test of the total system. Chart V, Program System Planning Sheet, is a form for recording planning information for program subsystem and system checkout.

5. Access to Previous Experience

Much of the analysis and costing of a program system will benefit from experiences with similar systems and programs. Because costing and scheduling of computer programs are presently subject to large errors, access to and use of such experience in planning will usually result in more accurate estimates. Sources of such information are the personal experiences of the Project Leader and his colleagues (such as other Project Leaders and Project members), files of previous and current Projects, summary descriptions and evaluations of completed Projects, technical journals and books, and documentation of Projects from other organizations (e.g., technical reports, plans, schedules, and actual costs and experiences in producing the system).

* Large copies of these forms, suitable for actual use, are included at the end of the Planning Guide.

Chart III. Sample Project Summary Sheet (Page 2)

PROJECT SUMMARY SHEET
(PAGE 2)

PROJECT PHASES & TASKS	NUMBER OF MAN MONTHS			START DATE			COMPLETION DATE			NUMBER OF COMPUTER HRS.			PROPOSED MANNING			REMARKS (DATES AND REASONS FOR REVISIONS, ETC.)
	REV EST	ACT		REV EST	ACT		REV EST	ACT		EST	ACT		SR ANAL	SR ANAL	SR PROD	
SYSTEM ANALYSIS																
1. PLAN THE PROJECT																
2. ANALYZE SYSTEM REQUIREMENTS																
3. ANALYZE USER ENVIRONMENT																
4. ANALYZE PRODUCTION REQUIREMENTS																
5. ANALYZE SIMILAR SYSTEMS																
6. EVALUATE CONTRACT PROPOSALS																
7. ANALYZE CHANGE REQUESTS																
SYSTEM DESIGN																
1. DESIGN TOTAL SYSTEM																
2. DESIGN PROGRAM SYSTEM																
3. OUTLINE PRELIMINARY FUNCTIONAL DESC.																
4. PRODUCE PRELIMINARY FUNCTIONAL DESC.																
5. FAMILIARIZE USER																
6. OBTAIN PFD CONCURRENCE																
7. INDICATE PRODUCTION PERSONNEL																
PROGRAM DEVELOPMENT																
1. DESIGN PROGRAM SYSTEM TEST																
2. DESIGN PROGRAMS																
3. DESIGN PROGRAM FILES																
4. ESTABLISH SYSTEM FILES																
PROGRAM CODING																
1. CODE PROGRAMS																
2. DESK CHECK PROGRAMS																
PROGRAM CHECKOUT																
1. LEARN TEST ENVIRONMENT																
2. COMPILE AND CHECK CODE																
3. TEST PROGRAMS																
4. TEST SUBSYSTEMS																
5. TEST SYSTEM																
USER DOCUMENTATION																
1. VERIFY SPECIFICATION DOCUMENTATION																
2. OUTLINE USER DOCUMENTATION																
3. PRODUCE USER DOCUMENTATION																
4. OBTAIN CONCURRENCE																
OVER-USER DOCUMENTATION																
USER TRAINING AND ASSISTANCE																
1. DATA COLLECTION AND CONVERSION																
2. DEVELOP USER TRAINING PLAN																
3. PROVIDE USER TRAINING AND ASSISTANCE																
TURNOVER																
1. DEVELOP TURNOVER PLAN																
2. CONDUCT DEMONSTRATION																
3. OPERATIONAL SHUTDOWN																
TOTALS																

★ PHASES ABOVE THIS LINE WILL BE COMPLETED AFTER PLANNING ESTIMATE IS APPROVED.
PHASES BELOW THIS LINE WILL BE COMPLETED AFTER PRELIMINARY FUNCTIONAL DESCRIPTION IS APPROVED.

10 May 1965

30

TM-2314/000/00

Chart V. Sample Program System Planning Sheet

PROGRAM SYSTEM PLANNING SHEET

SYSTEM CHECKOUT DESCRIPTION	NUMBER OF SUBSYSTEM TESTS						NUMBER OF SYSTEM TESTS						ASSIGNMENTS	
	TESTS			TRIALS			TESTS			TRIALS				
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL		
PROGRAM SUBSYSTEM TESTS														
DESIGN SUBSYSTEM TEST														
TEST DATA PRODUCTION														
TEST SUBSYSTEM														
TOTAL AND ENTER ON SUMMARY SHEET, PROGRAM CHECKOUT, TASK 4														
PROGRAM SYSTEM TESTS														
DESIGN SYSTEM TEST														
TEST DATA PRODUCTION														
TEST PROGRAM SYSTEM														
TOTAL AND ENTER ON SUMMARY SHEET, PROGRAM CHECKOUT, TASK 5														
REMARKS														
PREPARED BY REVIEWED BY APPROVED BY														

6. Intermediate Products of System Analysis and Design

As the Project progresses, it produces much information that is relevant to Project planning. The processes of system analysis and design produce analysis of user's requirements and environment, program requirements and data environment, production requirements and environment, and design specifications for the system, for programs, and for file and data structures, plus many less formal products such as trip reports, conference reports, correspondence, and other documents.

7. Planning Instructions

The Project Leader's management must specify how the leader is to plan. It should specify how the leader and his organization must interface with others, what reports are to be made, what forms must be completed and sent out, and what his other responsibilities and commitments are. At NAVCOSSACT, the Project Leader's Guide (NAVCOSSACT Instruction 5230.1A) and this Planning Guide provide general instructions on the way Projects are to be planned.

D. THE PLANNING PROCESS

The planning process recommended is detailed in this section of the Planning Guide. The sequence of steps outlined tells how to use the Check Sheets contained in Sections V, VI, and VII; explains how to complete the Project Summary Sheets, the Computer Program Planning Sheet and the Program System Planning Sheet; and describes the other activities and products produced in the planning process. This process is assumed to occur over a period of time during which work on the Project proceeds. Therefore, the planner will find that the sequence of planning steps includes or is intermingled with much of the work described in Section V, The System Analysis and Design, and in the Check Sheets for these Phases.

The sequence of the principal planning tasks detailed below is:

1. Preliminary Analysis
2. Information Collection
3. Gross System Analysis
4. Preliminary Program System Design
5. Determination of Work Tasks
6. Schedule Estimation
7. Preliminary Review
8. Obtain Concurrence

PLANNING TASK 1. PRELIMINARY ANALYSIS

Evaluate available information to determine how much more information must be collected, estimate the cost required to collect and analyze it, and make a gross estimate of system size and implementation costs.

The steps to be taken are:

Step 1. Begin to complete the Project Summary Sheet (using the form provided at the end of the Guide) by recording the Project Identification from the Project Request:

- . Request code and date
- . Request title
- . Requesting organization
- . Requesting letter reference number
- . Division assigned responsibility for the Project
- . Project Leader's name and date assigned
- . Contractor and contract date

Step 2. Obtain, in addition to the actual Project Request, other information, such as:

- . Requirements for the proposed computer program system.
- . Proposed information processing, e.g., logistics planning or weapons control that will use the program system.
- . User's environment (present as well as future) including mission statements, organization charts, physical location of facilities, operational functions, and mode of operation.
- . Proposed system hardware including the computer, input and output equipment, and communications networks.
- . Requirements for tests, inspections and/or demonstrations of the system.
- . Location and availability of the computer to be used during development.
- . The names and descriptions of hardware components that may interact with the program system.

- . The number and locations of stations and organizations that will use or participate in developing the system.

Record on the Project Summary Sheet the Project Description:

- . Statements of the missions and objectives of the system.
- . Description of the using organization and the placement of the program system within its operation.
- . The primary functions the system is to perform.

Record on the Project Summary Sheet the System Environment:

- . The names of "super" (or larger) systems that include this system as part of their functioning.
- . The names of systems that interface with this system, feeding information to it or receiving information from it.
- . The names of similar systems that have been developed by NAVCOSSACT or others.

Step 3. Determine and list, briefly, the information that is at hand and that must be obtained. Estimate the effort to collect the required information, and, consequently, the costs of performing the first five tasks of System Analysis as described in the Check Sheets. These estimates are to be entered on page 2 of the Project Summary Sheet.

Step 4. At this point, complete the Planning Estimate by making gross estimates of:

- . Total cost
- . Total program system size
- . Data base size and structure and storage
- . Relative complexity of development requirements
- . Program system complexity
- . Percentage of time to allot to the various phases
- . Project staffing and duration

10 May 1965

34

TM-2314/000/00

- Step 5. Make a recommendation on whether or not (1) the system is feasible, (2) it should be contracted, and (3) planning should be continued.
- Step 6. Coordinate the review of the Planning Estimate and the decisions on the recommendations in Step 5 above and revise the Planning Estimate and recommendations to incorporate the results of the review. Cease or continue planning, according to the decisions made.

PLANNING TASK 2. INFORMATION COLLECTION

Collect information (from the user and other sources) about the proposed and existing system and its environment, equipment configurations, and modes of operations, and about system production requirements.

The information to be gathered in this planning task is similar to, but less detailed than, the information required for the actual system design. For an internal Project, this information represents the beginning of system analysis. For a Project to be contracted, it represents the information necessary to formulate a statement of requirements and a statement of work for a contract proposal request.

The steps to be taken are:

Step 1. Contact the user or Project requester, hardware manufacturers, and other development agencies, as necessary, and arrange for:

- . Conferences and briefings
- . Security clearances and access to information
- . The collection and transmittal of pertinent documentation
- . Visits to existing installations
- . Interviews with user personnel

Enter the names of contacts on the Project Summary Sheet.

Step 2. Conduct conferences and briefings, interview users, and observe current operations. During Steps 1 and 2, look for the following problems and clues to both design requirements and improved planning estimates (see Check Sheets for System Analysis, Tasks 2 and 3):

- . Existing and potential system inefficiencies
- . Possible approaches and or methods of attack for design of the total system or parts of it
- . Possible problem areas, such as functions that will be difficult or costly to program
- . Possible interaction problems such as points of friction, e.g., users or other organizations participating in the Project that seem reluctant or unable to provide information

- . Especially easy areas of development, e.g., possible use of (1) routines or programs from an existing system, (2) other systems or libraries, and (3) other design or programming techniques that may be directly applied. In these cases, try to get the specific products to actually assess the ease of transfer.

Step 3. Determine the computer to be used in producing the program system (see Check Sheet for System Analysis Task 4) and make gross estimates of:

- . Turnaround times
- . Priority and work loads
- . Procedures and operating system

Step 4. Identify similar and interfacing systems (see Check Sheet for System Analysis Task 5) to determine possible processing requirements and constraints and to identify routines, functions, and data files that might be transferable and/or furnish a basis for costs.

PLANNING TASK 3. GROSS SYSTEM ANALYSIS

From an analysis of the information gathered, design and evaluate a preliminary model of the operational system. Determine system size and complexity more accurately and evaluate the expected equipment configuration and usage.

This analysis must be detailed enough to provide material for a contract statement of work and to permit evaluation of proposals. The work is an iteration of System Analysis Tasks 2, 3, 4 and 5. Therefore, inputs will be preliminary results of the earlier work done in these tasks. In this planning task, the Project Leader (with substantial contractor help in case a contract is let) produces a Project Development Plan.

The steps to accomplish this analysis are:

Step 1. Prepare preliminary descriptions of:

- . The existing system
- . The existing user organization for related information flow
- . The proposed system
- . The proposed user organization for related information flow
- . System inputs and outputs
- . Operational functions and the proposed mode of operation
- . The proposed equipment configuration
- . System interfaces
- . Available programming tools and facilities

Step 2. Issue these descriptions for review and also as part of a Request for Proposal when contract help is sought.

Step 3. Summarize the planning information obtained on the Project Summary Sheet:

- . On page 2, enter estimate the costs of performing System Analysis Tasks 6 and 7.
- . In Manning Assumptions, enter estimates of experience and skills levels required to perform the various tasks, and names of key personnel who are needed for successful Project conduct.

- . In Lead Time Assumptions, enter notes on key events or conditions such as elapsed time to reach assumed levels of manning or training, expected delivery dates for computers and any other key equipment, or delivery dates for compilers and operating systems.
- . In Contacts, identify individuals who will represent the various agencies involved in this Project.

Step 4. Determine the computer and support program requirements and availability and enter the following under Computer Usage on the Project Summary Sheet:

- . Computers to be used for program production and test and their location.
- . Programming language and operating system to be used.
- . Expected date of first use of the computer, the probable maximum usage of the machine, and desired turnaround time.

In the Comments column, note the following and other relevant information:

- . Evaluation of the reliability of the computer and programming tools.
- . Probable competition with other uses of the computer, and the priority of these Projects or activities.

Step 5. Coordinate the review and approval of the above analysis work and/or contract proposal and issue the Project Development Plan.

The Project Development Plan is a more detailed and accurate forecast than the Planning Estimate and contains an estimate of the calendar time, the effort in man months, and other quantities related to Project progress. It lists starting date, completion date, man months allocated, and progress indicators (quantity required and unit of measure) for each Phase in the Project Development Plan. Depending upon the complexity of the Project, the Project Development Plan must be prepared either 60 or 90 days after Project initiation.

PLANNING TASK 4. PRELIMINARY PROGRAM SYSTEM DESIGN

To help complete the estimate of Project costs and schedules, lay out a preliminary program system design in terms of overall functional blocks, and compare the proposed system to similar existing systems. The steps in planning during preliminary program system design are:

Step 1. Identify systems with similar features to exploit this experience, if possible, by using cost data, designs, and development methods for planning and developing the proposed system. Look for similarities in programs and routines, applications, computers and other equipment, and Projects for the same user.

Sources to consult are:

- . Past and current Project files
- . Libraries of professional books, journals, and reports
- . Lists of SHARE, CO-OP, and other subroutine libraries
- . Experienced personnel such as other Project Leaders and consultants
- . Colleagues in the field

Step 2. Divide the program system into subsystems, by identifying relatively independent sets of functionally related requirements. Here, the words program systems are used in a relative sense. For example, in a large, new information system, one division may identify (a) operational programs (i.e., those programs that contribute directly to the system mission), (b) operational support programs (those programs that do not directly perform operational work, but are necessary to support the operation of the system), and (c) program production and checkout (utility) programs. (At NAVCOSSACT this type of system development would embrace a large number of Projects.) In other cases, the program system to be divided would be only one or part of one of these programs. If necessary, divide the subsystems into smaller blocks such as programs and major subroutines. It is suggested that individual Project Summary Sheets be initiated for each major subsystem and that subsequent tasks be repeated for each subsystem.

Note that the Summary Sheets, Program Planning Sheets, and Program System Planning Sheets are sufficiently flexible to be used for a range of possible program system heirarchies.

- Step 3. For each subunit identified, record the identifying information on a Computer Program Planning Sheet (see work sheets at the end of the Guide), including program name, function and type of job.
- Step 4. On the second page of the Summary Sheet, record estimates of the cost of performing the seven designated tasks of the System Design Phase.

PLANNING TASK 5. DETERMINATION OF WORK

To assess in detail the work to be done, examine each program and estimate the man months, computer hours, and elapsed time necessary to produce and test the programs. Identify and establish the program flows, functions, inputs and outputs, and testing requirements for each program. To be sure that tasks or important aspects of the work are not overlooked, refer to the appropriate Check Sheets in costing the various programming tasks. The costs to develop individual programs and to test subsystems are to be recorded on the Computer Program Planning Sheet and the Program System Planning Sheet, respectively. Later these are to be summed and transferred to the Project Summary Sheet. Initial entries should be tentative, subject to revision after planning of this phase is complete and prior to recording them in the Summary Sheet.

Step 1. For each program, develop a tentative, broad flow diagram of system operations (see Program Development Tasks 2 and 3) that shows:

- . Inputs--messages, number of types, rates
- . Outputs--messages, number of types, rates
- . Data flows through the system
- . Intermediate data structures
- . Processing functions
- . Feedback, monitoring, interrogation, and response loops
- . Cyclical or temporal relations of functions and data flows

Step 2. For each program, estimate the work required to perform the following Program Design tasks (see System Design Task 2 and Program Development Task 2) and enter the values on the Computer Program Planning Sheet.

- . Logic Analysis and Flow Chart
- . Timing/Analysis
- . Design Review
- . Program Specifications

- Step 3. Analyze the broad flow diagram to identify data structures and manipulation requirements and to determine functions and data structures that:
- . Are similar or identical, e.g., with respect to design, I/O sources, and operating speeds
 - . Operate in the same time intervals
 - . Are highly interdependent or interactive
- Step 4. Record the Data Description entries for each program on the Program Planning Sheets:
- . Number of items of data to be processed
 - . Number of different input and output formats
 - . Number of data tables to be designed
 - . Number of constants and parameters
 - . Number of files
 - . Number of pages of documents to describe all the data
- Step 5. Estimate the work necessary to perform the following Data Design tasks for each program and enter these estimates on the respective Program Planning Sheets:
- . Data Analysis
 - . Input and Output Formats
 - . Data Design
 - . Storage Allocation
 - . Data Review
 - . Documentation
- Step 6. Describe each program in terms of size, complexity, and familiarity. Some techniques to help make these estimates are:
- . Comparison with similar programs or routines in previous Projects, in the literature, or in subroutine libraries.

In addition to comparison of size and complexity, special attention should be paid to differences in generality, modularity, language, operating system, and function.

- . Make use of short design and coding trials by roughly flowing and actually coding sections of the programs. These trials should sample most functions but stress possible difficulties such as control and feedback loops and interfaces. Although time-consuming, this sampling of the work yields fairly accurate estimates of size, complexity, and difficulty.
- . Solicit and use expert opinion and diagnosis. Although subject to possible bias, experienced persons, when dealing with systems that are similar to them, can often make relatively good estimates of size and complexity, and can also detect conditions likely to cause difficulty. However, experts in a single function may overlook program system communication, housekeeping, and I/O requirements. Be sure an estimate includes these, since, depending upon the system design, they may comprise up to 40 percent of small programs.

Step 7. Record the following information in the Program Description section of the Computer Program Planning Sheet:

- . Number of instructions, divided into two estimates--one of new code and one of old or revised code--to get an idea of the proportion of innovational or unfamiliar coding involved. Reference any old programs with usable code or design in the Remarks area on the reverse side of the Program Planning Sheet.
- . In the space for Number of Blocks, enter the number of functions and subfunctions involved.
- . The entry for Complexity must be a local standard, such as a scale of one to five, that reflects not only the number of interactions among subfunctions and the number of interfaces with other programs, but also the number and variety of data types input, manipulated, and output; or, the standard could actually be a rough count of these items.

At NAVCOSSACT a Complexity Factor* has been defined for the total system to be developed and is presented in the Project Leader's Guide.

- . Enter estimates of the number of pages of documentation for the program. If no better estimate is available, a minimum documentation estimate of one page per hundred instructions may be adopted. If the program is new or complex, and study is thus required for design and use of the program, more documentation may be required.

Step 8. After estimating the program size, estimate the cost of performing the tasks of coding, desk checking (see Program Coding Tasks 1 and 2) and keypunching, and enter the cost of the tasks required for each program on the Program Planning Sheet.

Step 9. Estimate the cost of compiling and checking the code for each program, and of designing and running individual program tests. (See Program Checkout Tasks 2 and 3.) Assume a minimum of three trials per program for test. Enter the results under Checkout Description on the Program Planning Sheet.

Step 10. Although the detailed planning for system testing will occur later, make rough estimates of the number of tests that will be run for subsystem and system testing, and the number of trials that the tests will require, and enter these estimates on the Checkout Description area of the Program System Planning Sheet.

- . The number of tests for subsystem testing may be estimated by determining the number of pairs of interfacing programs, the number of triplets, and the number of other subsets of interfacing combinations, and by scheduling at least one test for each interface combination. (Other assembly procedures are possible; for example, plan tests for each required operational function and string together (often via an executive routine) those programs that contribute

* At NAVCOSSACT, the Degree of Factor Complexity as defined in NAVCOSSACT Instruction 5230.1A provides a technique for rating complexity of the development using a matrix of factors versus degree. The factors include originality required, degree of generality, span of operation (geographic dispersion), change in scope and objective (anticipated), equipment complexity (ranging from one machine to multicomputer with automatic I/O), number of personnel (including contract size), and cost.

to the function, whether they interface or not. One program may be involved in several "string" tests since it normally contributes processing to several functions. Tests may also be planned for evaluating interactions with particular items of system equipment.)

At least two system tests should be planned, an easy one (with simple inputs) to determine that the system will operate, and one quite difficult (heavy load) to evaluate performance. As with subsystems, individual system functions should also be tested to evaluate them separately, i.e., free of the obscuring effects of interacting functions. If several versions of the system are to be produced for installation at different locations, tests should be planned for the different versions. This system test planning may also include plans for the Demonstration test.

- Despite careful programming and thorough debugging, programs seldom pass the first test. Hence, at least three trials per program system test should be planned. Early tests will probably take even more trials; later tests may take less. If the system has several versions e.g., one for each of several locations, similar trials should be planned for each unique version.

Step 11. Estimate the cost of performing the Program System Checkout tasks (see Program Checkout Tasks 4 and 5) and enter the results on the Program System Planning Sheet.

Step 12. From the Program Planning Sheets and the Program System Checkout Planning Sheets, sum the costs of Program Development, Program Coding, and Program Checkout tasks for all programs in the particular system and enter the results on the Project Summary Sheet. If separate Summary Sheets are maintained for subsystems, sum onto the subsystem Summary Sheets and then onto a master Project Summary. Entries should be regarded as tentative, subject to later revision.

Step 13. Estimate, on the basis of the results of the previous planning, the costs of the components, the planned work for user documentation, user training and assistance, and turnover phases and enter these estimates on the Project Summary Sheet. (See User Documentation, User Assistance, and Turnover Task Check Sheets.)

PLANNING TASK 6. SCHEDULE ESTIMATION

Prepare detailed schedules for the overall Project and for the tasks and components of the system. (Section IV of the Planning Guide provides additional discussions of schedule considerations.)

Step 1. Lay out rough time-line schedules to indicate the general time periods during which various tasks are to be performed. See Chart VI for an example of a Gantt Chart. For large program systems in which some subsystems may have to be developed first, e.g., utility, separate graphs should be drawn for each independent program subsystem.

Step 2. Lay out time-line schedules for:

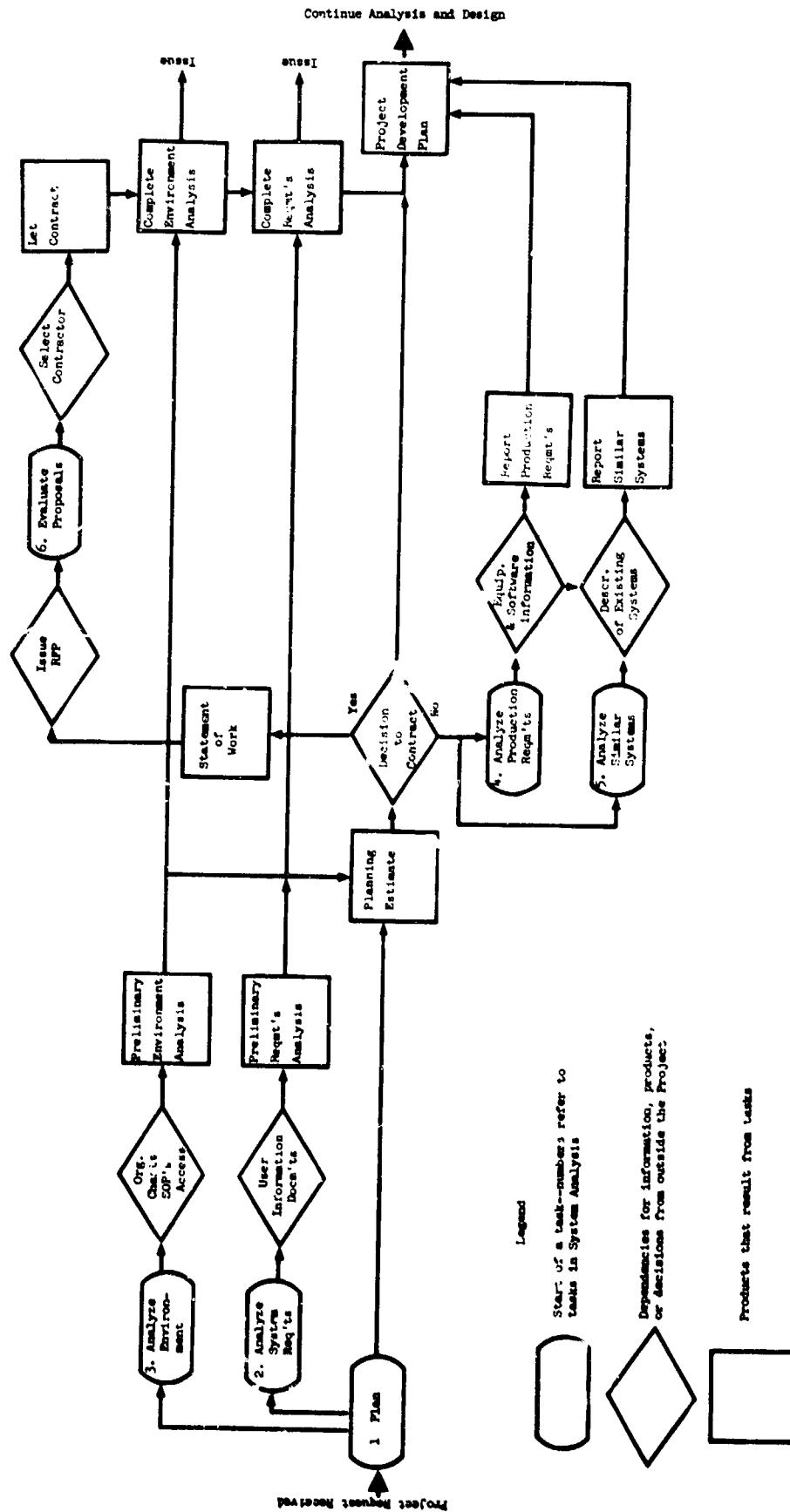
- . Each of the individual programs as recorded on the Computer Program Planning Sheets.
- . Each of the subtasks of the tasks involved in the System Analysis, System Design, Documentation, User Training and Assistance, and Turnover Phases.

Step 3. Prepare a general sequence of work or network analysis schedule to depict, in graphic form, the sequences in which work must be done. Network analyses show the dependencies of one task upon the successful completion of others and indicate the shortest time (the longest or "critical path") in which the job can be done with the cost, productivity, and delay time assumptions that have been made. A way in which these dependencies are diagrammed is shown in Chart VII, an example of a network analysis for the System Analysis Phase. In preparing a network analysis, give special attention to:

- . Task dependencies.
- . Critical deadlines, such as the delivery and availability dates for computers and the utility system.
- . Review, concurrence, and decision points.
- . Large periods of "slack time" that permit some work to be done in parallel to reduce lead times.

Step 4. Reconcile differences between the detailed schedules (Step 2) and the overall schedules (Steps 1 and 3). All of these schedule representations, Planning Sheets, Gantt Charts, and networks should be aligned so that they agree. To make this alignment, tentative plans should be adjusted by:

Chart VII. An Example of Network Analysis for System Analysis



- . Increasing or decreasing estimates of manpower allocations.
- . Reducing or removing proposed system features.
- . Reducing proposed quality and performance levels.

Step 5. Enter proposed start and completion dates on Program Planning Sheets and the Project Summary Sheets. On the Summary Sheets the Coding and Checkout dates should span the entire range of time for coding and checking out individual programs as indicated on the individual Program Planning Sheets. It may be necessary to adjust earlier manpower and computer time estimates, to account for changes that result from the schedule analysis in Step 4. Record any assumed conditions that critically influence lead time in the Lead Time Assumptions section of the Summary Sheet.

Step 6. Review the assumptions that have been made for computer usage by:

- . Summing up estimates of computer time by computer, language, operating system, and location.
- . Realistically estimating the number of competing users of the computer and the Project priority, and then estimating the expected average amounts of computer time for the Project per day and per week.
- . Estimating the average amount of computer time each programmer is expected to need.
- . Estimating the expected turnaround time for Project work, considering competition and priority, plus other factors such as the accessibility of the computer location and the known or estimated efficiency of the computing facility.

Step 7. Enter these revised values, along with the forecast date of first use, in the Computer Usage section of the Project Summary Sheet.

PLANNING TASK 7. PRELIMINARY REVIEW

Integrate costs, schedules, and all plans by reviewing them with the Project members and with other Project Leaders.

The basic costing is now done; the sequence of study, information collection, analysis, design, costing, and scheduling is complete. This first detailed analysis may contain many oversights, redundancies, and contradictions. The steps taken to detect and eliminate such discrepancies are:

- Step 1. Study the costs and dates recorded on the Project Summary Sheets, Computer Program Planning Sheets, and various schedules, to evaluate their reasonableness and to detect contradictions between detailed and overall schedules.
- Step 2. Review the reasonableness of Project plans, particularly costs and schedules, with other Project Leaders or objective expert personnel.
- Step 3. Look for and identify omissions and redundant efforts.
- Step 4. Correct oversights, remove redundancies, and adjust contradictions.

PLANNING TASK 8. OBTAIN CONCURRENCE

Document the plans for the implementation of the program system and obtain the concurrence of higher management and the customer. The coordination and concurrence for the Project Implementation Plan and Preliminary Functional Description (see System Design Task 6) should be concurrent.

The steps taken are:

- Step 1. Draft the Project implementation plans. Emphasize statements of mission and objectives, descriptions of activities and products, and discussions of assumptions and limitations that will make graphic and numeric information meaningful to the reader who is inexperienced with ADP. Recommendations may be made concerning the continuance, priority, and feasibility of the Project.
- Step 2. Submit plans and recommendations to management for coordination, review, and approval.
- Step 3. Cooperate with management in their evaluation of the plans by:
 - . Presenting briefings.
 - . Attending conferences.
 - . Providing additional information.
 - . Clarifying assumptions, specifications, and estimations.
 - . Trying to answer objections and responding to suggestions raised during the evaluation.
- Step 4. Revise plans until management concurrence and approval is received.
- Step 5. Prepare drafts of the plans for coordination with the user.
- Step 6. Cooperate with the user and his agents in their evaluation of the plans as in Step 3.
- Step 7. Revise plans until customer concurrence is achieved.
- Step 8. Publish the Project Implementation Plan.

IV. GUIDELINES FOR ESTIMATION

In the development of a specific program or program system, the cost of any task or subtask depends on numerous factors, such as the size and complexity of the program being developed, the resources including the personnel, methods and tools used for development, and the particular conditions under which the programs are produced. This section describes some cost factors that the Project Leader should consider; presents some guidelines that may be used for cost estimation and scheduling; and displays some equations and raw data that are the results of research in cost estimation.

As one might expect, larger Projects incur larger costs. Experience shows that increases in size, complexity, and integration of computer programs into a system lead to a need for increased division of labor and coordination as follows:

- . To meet reasonable development schedules, work must be divided into tasks that can be handled by a single person.
- . Tasks that may be subsumed in a small effort now become large enough and important enough to have one or more persons assigned to them full time.
- . The time and effort needed for (and usually spent on) system analysis, design, and testing increase much faster than the time spent on detailed programming and coding.
- . The interdependencies of programs increase rapidly so that the need for coordination and communication among the programs and in the corresponding development work grows exponentially.
- . Correspondingly, the need for management, supervision, and control increases greatly.

Although the guidance given below reflects some of these, the Project Leader should keep these characteristics in mind to aid his judgment.

A. COST FACTORS IN PROGRAM DEVELOPMENT

The many factors that influence Project difficulty or cost may be divided into three groups:

- . The nature of the job to be done--the nature, clarity, and extent of system requirements.
- . The wherewithal--the amount and availability of the various resources (personnel, machines, and information) required to do the job.

- . The environment--the conditions under which the Project is managed and the program must be developed.

Developed from a survey of experience in program development, Chart VIII is a matrix showing a summary of cost factors as they are qualitatively related to the tasks. The Check Sheets include a more detailed list of cost factors. The 36 tasks describing the system development process are listed on the left side of the matrix, the factors above. Factors are grouped into the three categories: Requirements, Resources, and Development Environment. A plus sign appearing at the intersection of one of the factors and one of the tasks means that the presence of that factor will increase the cost of the task; a minus sign indicates that its presence will decrease the cost of that task. The extent to which the factor is present determines the degree to which it increases or decreases the task cost; e.g., the greater the amount of programmer experience, the greater will be the extent to which the cost of the programming task is decreased. On the other hand, some factor/task relationships do not exist in degrees; e.g., if the computer used for development is not the same one as the computer used in actual operation, project planning and development costs will be higher. (If data were available, this chart could be used to make detailed comparisons of factors in a new Project with factors in completed Projects.)

B. GUIDELINES FOR ESTIMATION

Although the prediction of programming costs is still largely uncertain and inaccurate, better costing formulas are gradually evolving. However, as long as programming includes a large amount of development work and information generation, some inaccuracy of prediction must be expected. The accuracy of prediction depends upon the accuracy of assessment of the many factors that influence the work. Also, until the influence of these factors can be established conclusively, the initial estimation of costs must rely upon experience and rough rules of thumb. But in any specific Project, as work proceeds, the influence of specific factors will become clearer and, hopefully, quantitative. Therefore, planning, including the estimation of costs and schedules, is viewed as an ongoing function of the Project Leader, and estimates will change several times during the course of a Project. Generally, each revision of the plan is more accurate than the preceding one.

In planning, the Project Leader must make estimates of the costs of the programs in terms of:

- . Manpower
- . Computer Time
- . Elapsed Time

Chart VIII. Cost Factor-Task Matrix

TASKS	FACTORS	
	REQUIREMENTS	RESOURCES
SYSTEM ANALYSIS 1. PLAN PROJECT 2. ANALYZE SYSTEM REQUIREMENTS 3. ANALYZE USER'S ENVIRONMENT 4. ANALYZE PRODUCTION REQ'TS. 5. ANALYZE SIMILAR SYSTEMS 6. EVALUATE CONTRACT PROPOSALS 7. ANALYZE CHANGE REQUESTS	1. INNOVATION IN OPERATIONAL SYSTEM 2. USER PARTICIP. IN REQ'TS ANALYSIS 3. SYSTEM CHANGES 4. SYSTEM COMPLEXITY 5. CONCURRENT SYSTEM DEVELOPMENT 6. SYSTEM DISPERSION 7. SIZE OF PROGRAM SYSTEM 8. NUMBER OF INPUTS & OUTPUTS 9. INNOVATION IN PROGRAM SYSTEM 10. PROGRAM SYSTEM COMPLEXITY 11. DESIGN CONSTRAINTS 12. PROGRAM CHANGES 13. SIZE OF DATA BASE 14. ADEQUACY OF TEST REQUIREMENTS 15. NUMBER OF DISPLAYS	1. AMT. DOCUMENTATION AVAILABLE FOR UTILITY PROGRAMS & ADP EQUIPMENT 2. AVAILABILITY OF PROG'G. TOOLS 3. AVAILABILITY OF COMPUTER TIME 4. PROGRAMMER EXPERIENCE 5. PERSONNEL TURNOVER DEVELOPMENT ENVIRONMENT 1. DOCUMENTED MANAG'T PLANS FOR: a. SYSTEM DESIGN CHANGES b. PROGRAM DESIGN CHANGES c. ERROR CORRECTIONS d. COMPUTER USAGE e. STANDARDS, CODING, FLOW CHARTS 2. NUMBER OF CONCURRING AGENCIES 3. USER EXPERIENCE 4. CONCURRENT EQUIPMENT DEVELOPMENT 5. PROGRAM DEVELOPED MORE THAN ONE LOC'N. 6. COMPUTER FOR DEVEL. DIFF. THAN OPER'NS.
SYSTEM DESIGN 1. DESIGN TOTAL SYSTEM 2. DESIGN PROGRAM SYSTEM 3. OUTLINE PFD 4. PRODUCE PFD 5. FAMILIARIZE USER 6. OBTAIN PFD CONCURRENCE 7. INDOCT. PRODUCT'N PERS.		
PROGRAM DEVELOPMENT 1. DESIGN PROGRAM SYST. TEST 2. DESIGN PROGRAMS 3. DESIGN PROGRAM FILES 4. ESTABLISH SYSTEM FILES		
PROGRAM CODING 1. CODE PROGRAMS 2. DESK CHECK		
PROGRAM CHECKOUT 1. LEARN TEST ENVIRONMENT 2. COMPILE & CHECK CODE 3. TEST PROGRAMS 4. TEST SUBSYSTEMS 5. TEST SYSTEM		
USER DOCUMENTATION 1. VERIFY SPEC'N. DOCUMENT'N. 2. OUTLINE USER DOCUMENT'N. 3. PRODUCE USER DOCUMENT'N. 4. OBTAIN CONCURRENCE 5. PUBLISH USER DOCUMENTATION		
USER TRAINING & ASSISTANCE 1. DATA COLLECT. & CONVERSION 2. DEVELOP USER TRAINING PLAN 3. PROVIDE USER TRAINING & ASSIST.		
TURNOVER 1. DEVELOP TURNOVER PLAN 2. CONDUCT DEMONSTRATION 3. OPERATIONAL SHAKEDOWN		

1. Computer Time Estimates

On most Projects, gross estimates of computer usage are adequate. That is, estimates of the computer time needed to check out any one program are usually quite inaccurate, but rules of thumb appear to come surprisingly close to actual expenditures. Two basic methods are:

- a. Estimate computer hours as a function of the number of programmers and weeks of computer usage. Depending upon the conditions of use and the number of programmers competing for time, computer usage averages between eight and fifteen minutes per man day, or between two-and-a-half and four hours per man month. That is, 30 to 60 programmers can use one to two shifts of computer time per day in compilations, code checks, data generations, and program tests.
- b. Estimate computer hours as a function of number of instructions. Great variation exists in the amount of time taken to check out any one routine or program. However, a rough rule of thumb for moderately large systems is one checked-out instruction per minute of computer time. For small and simple programs, for program rewrites, and for program conversions from one computer to another, less time will be required. For large systems with many interfaces and tightly integrated functions, more will be required.

For initial estimates, these rules are usually adequate. Because of unforeseen difficulties in computer usage and availability it is good practice to reforecast computer time requirements frequently.

2. Programmer Productivity

Estimating programmer production is more difficult than estimating computer time. For a programming process divided roughly into three parts, Analysis, Coding, and Checkout, the percentage of total effort devoted to these parts is shown in Table I.

From these data it may be estimated that the average allocation of effort is as follows: Analysis and Design will consume 40 percent of the manpower; Coding, 15 percent; and Checkout and Test, 45 percent. Roughly, the Analysis and Design work represented by these figures includes equivalent tasks; the same is true of the Checkout work. At NAVCOSSACT, the Coding Phase is a larger part of the work (includes more tasks) than that implied by the other entries under Coding in the Table. For example, at NAVCOSSACT, design of individual programs and their associated documentation would be included in Coding, but in the other programming efforts this work was included in Analysis and Design. Despite these differences, the average values agree with NAVCOSSACT experience.

TABLE I. RELATIVE COSTS OF PROGRAMMING PROJECT PHASES*

	SETE** (%)	SAGE*** (%)	NTDS**** (%)	NAVCOSACT (%)	AVERAGE (%)
Analysis and Design	35	39	30	34	34.5
Program Coding and Checking	17	14	20	20	18
Checkout and Test	48	47	50	46	47.5

*The data on SAGE (Semi-Automatic Ground Environment) refer only to early models and not to model changes that logically might reflect costs in different proportions than are characteristic of new systems. The Project SETE (Secretariat for Electronic Test Equipment, New York University) values are averages derived from data on twelve systems produced in support of automatic test equipment for missile and other projects. The NTDS data do not include information on preliminary design and operational system analyses.

**A Survey of Programming Aspects of "Computer-Controlled" Automatic Test Equipment, Volumes 1 and 2, SETE 210/78, New York University, New York, June, 1964.

***J. P. Haverty, Programming Language Selection for Command and Control Application, Symposium Address, Symposium on Computer Programming for Military Systems, The Hague, The Netherlands, September, 1964.

****G. G. Chapin and P. A. Hensel, Programming for the Naval Tactical Data System, Symposium Address, Symposium on Computer Programming for Military Systems, The Hague, The Netherlands, September, 1964.

The analysis and design work for information processing systems involves many tasks and is frequently underestimated, undercosted, and inadequately performed. Because the later work in the process is more susceptible to measurement and can be estimated more easily, the analysis and design work may be estimated as a percentage of the coding and checkout costs using data such as those in the Table.

Coding and checkout productivity are usually estimated at 250 to 300 machine instructions per man month estimated from the start of coding to the completion of the Project (i.e., approximately 60 percent of costs). Utility and support programming fall at the upper end of this productivity range, and complex, tightly integrated systems at the lower end.

Conversion and reprogramming Projects should enjoy a much higher productivity rate, if there is adequate documentation and there are no major changes. If no major redesign of the program or system is necessary, savings are possible because most of the analytic work is eliminated. Although coding is reduced somewhat, checkout and turnover will run about the same as an original development. Savings will also be realized if the same test designs and test data can be used for the converted programs as were used originally.

Program modification, notably of large systems, is frequently undercosted. Coding a small change is usually trivial--a few minutes' work--but thoroughly investigating the impact of the change on many interrelated programs and operations may take days. Also, updating the coding is often less costly than updating the documentation of the program. As changes become larger, the proportionate costs of coding and checkout gradually rise toward the relative costs that hold for original programming. The cost of testing small changes to a large system, to be sure that the change has been made correctly and has not adversely affected any other item, is sometimes very large in comparison to the other costs of change.

3. Elapsed Time

In estimating the amount of time that a Project will take, people are often quite good at estimating the time for their own jobs, but are very poor at estimating how long others will take. This is particularly true when estimates are made of time to be used by non-Project members. For example, the times needed by higher level managers or the user to review documents are usually underestimated. Some typical examples of work that must be done by non-Project members are:

- . Product inspections and reviews (e.g., reviews of documents).
- . Editorial and technical reviews of documents prior to publication.

- . Decisions or choices between alternative modes of procedure.
- . Decisions about or approval of a product or procedure (e.g., concurrence on a document).
- . Coordination and concurrence procedures.
- . Evaluation of proposals (e.g., processing a change proposal).
- . Establishing contact and making arrangements to meet and discuss.

At NAVCOSSACT, estimates based on experience show a minimum of 14 weeks elapsed time for processing such as obtaining documentation concurrence, security clearances for people and information, and approval for a particular design. Most of these periods may be predicted, once experience data for average turnaround time are obtained. Although the accuracy of an estimate for individual reviews may be poor, the collection of such estimates will usually contain some over- and underestimates that will cancel one another. Since the number of lengthy reviews is usually greater than the number of unexpectedly quick responses, overestimating the individual periods will pay off in a more accurate total Project schedule.

Some typical times for various non-Project activities estimated by NAVCOSSACT personnel are shown in Table II. Chart IX shows the breakdown of elapsed times for letting contracts with some comparison of these for sole source and solicited bid contracts. Project members tend to view these periods as delays. This is not true, of course, but even so, any periods of time that involve transit and wait may be shortened.

C. RESEARCH IN ESTIMATION OF PROGRAMMING COSTS

Under an Air Force Electronic Systems Division contract, SDC recently completed the first part of an exploratory study of computer programming cost factors.* Aimed at developing cost estimating equations or relationships, the analysis included use of various statistical techniques such as correlation analysis, multivariate regression analysis, and factor analysis.

*TM-1447/001/00, Factors that Affect the Cost of Computer Programming: A Quantitative Analysis, L. Farr and H. J. Zagorski, 31 August 1964.

<u>Product</u>	<u>Review or Activity</u>	<u>Range</u>
Project Request	Rating of Project, e.g., preliminary user contact	1-2 weeks
Planning Estimate	Establish contact with user	1-2 weeks
Planning Estimate	Approve draft in-house and non-in-house organizations	3-12 weeks
Project Development Plans	Approve in-house document	2-4 weeks
Manning	Acquire and assign personnel	4-8 weeks
Security Clearance	Clear new people	6-12 weeks
Security Clearance	Establish right-to-know and need-to-know to obtain information from user and others in researching this and other related systems	1-6 weeks
Contract-Letting Process	See Chart IX	
Preliminary Functional Description	Obtain document draft including typing and duplication	1-4 weeks
Preliminary Functional Description	Approve in-house document	3-7 weeks
Preliminary Functional Description	Approve document by users	6-14 weeks

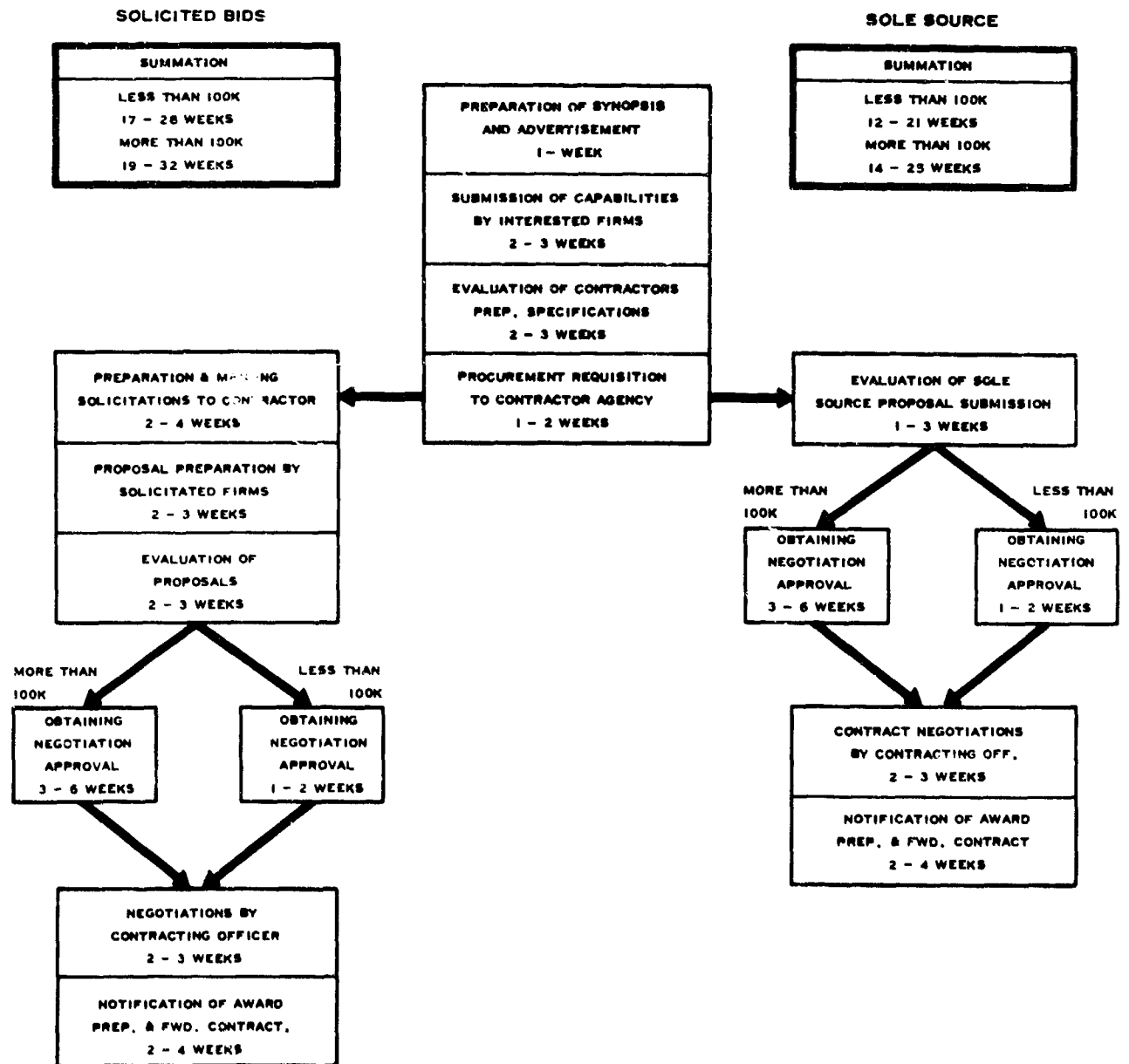
Above Activities apply to both Feasibility Studies and Programming Projects. Those below apply only to Programming Projects.

Computer Time	Maintain equipment and facilities	@ 3 weeks/year
Change Request	Process change request (average: 5 changes per project)	1 week
Project Manual	Obtain draft document including typing and duplication	1-4 weeks
Project Manual	Approve in-house document	2-8 weeks
Turnover Plan	Obtain draft document including typing and duplication	2-3 days
Turnover Plan	Approve document in-house and by non-NAVCOSACT organizations	2-8 weeks
Statement of Project Completion	Obtain final acceptance	2-12 weeks

NOTE: Many of these activities are or can be pursued concurrently with other tasks. Therefore, the times indicated do not add linearly in any schedule but help a Project Leader to schedule dependent events such as beginning a new task.

Table II. Time Estimates and Other Activities at NAVCOSSACT

Chart IX. Contract-Letting Processes and Delays



Although the statistical techniques yield meaningful estimating relationships, the equations that emerged from the first phase have large confidence limits. That is, the statistical confidence with which one can use these equations is quite low. Work is proceeding to improve the equations; therefore, they are presented here, not as tested, well-proven tools, but simply as experimental aids to be used in conjunction with judgment and experience.

Data on over a dozen cost variables and almost a hundred predictor variables were collected on twenty-seven programs and subjected to statistical analysis. Below is a summary of four of the resulting equations and some cost data for man months and computer hours presented graphically. Figures I and II plot man months and computer hours against program size in terms of the number of delivered instructions. Figure III plots computer hours against months and reveals a fairly high correlation between these two cost variables. The variable, man months, represents the cost of designing, coding, testing, and documenting the program. The scope of work is about the same as that included in the following NAVCOSSACT Phases:

- . System Design (Program System Design Task only)
 - . Program Development
 - . Program Coding
 - . Program Checkout
 - . User Documentation
1. Man Months for Design, Code and Test

$$Y_1 = 2.8X_2 + 1.3X_3 + 33X_4 - 17X_5 + 10X_6 + X_7 - 188$$

Standard error of estimate* = 70 man months

Range of costs in sample = 20-900 man months

*The standard error of estimate is a measure of expected deviation of estimated data from actual data. Two thirds of actual costs should fall within one standard error of their predicted values. Since this measure tends to be constant throughout the cost estimation range, the relative percent of error to total cost will decrease as one proceeds from small programs to large programs. Thus, the larger programs are able to tolerate the estimating error much more readily than smaller programs.

Variables

- 1 Number of man months for program design, code, and test
- 2 Number of machine language instructions in delivered program (in thousands)
- 3 Number of man miles for travel (in thousands)
- 4 Number of document types delivered to the customer
- 5 System programmer* experience index
- 6 Number of display consoles
- 7 Percent instructions new to this program (not reused from previous versions)

2. Months Elapsed Time

$$Y_1 = 2.5X_2 - .14X_3 + .11X_4 + .3X_5 + 7.0$$

Standard error of estimate = 4.8 months

Range of elapsed times in sample = 5-56 months

Variables

- 1 Number of months of elapsed time for program design, code and test
- 2 Number of words in tables and constants
- 3 Number of words in core storage
- 4 Percentage of decision-making instructions
- 5 Number of document types delivered to the customer

*System programmer. As the most senior of four classes, he contributes to the formulation, planning, design, and development of computer program systems; experience index for the system programmer is the sum of the average number of years of experience with the specific computer-type, application, and language.

10 May 1965

ca

TM-2314/000/00

3. Computer Hours

$$Y_1 = 21.5X_2 + 985X_3 + 197X_4 - 3468$$

Standard error of estimate = 905 hours

Range of hours in sample = 130-9000 hours

Variables

- 1 Number of computer hours
- 2 Number of machine language instructions in original estimate
- 3 Complexity rating (scale 1 to 5--subjective from simple to highly complex)
- 4 Number of words in data base

4. Delivered Machine Language Instructions

$$Y_1 = 2.6X_2 + 1.2X_3 + 5.6X_4 - 13.9$$

Standard error of estimate = 25.7 instructions (thousands)

Range of program sizes in sample = 8-300 instructions (thousands)

Variables

- 1 Number of machine language instructions in delivered program (in thousands)
- 2 Number of input messages
- 3 Number of subprograms
- 4 Number of words in tables and constants

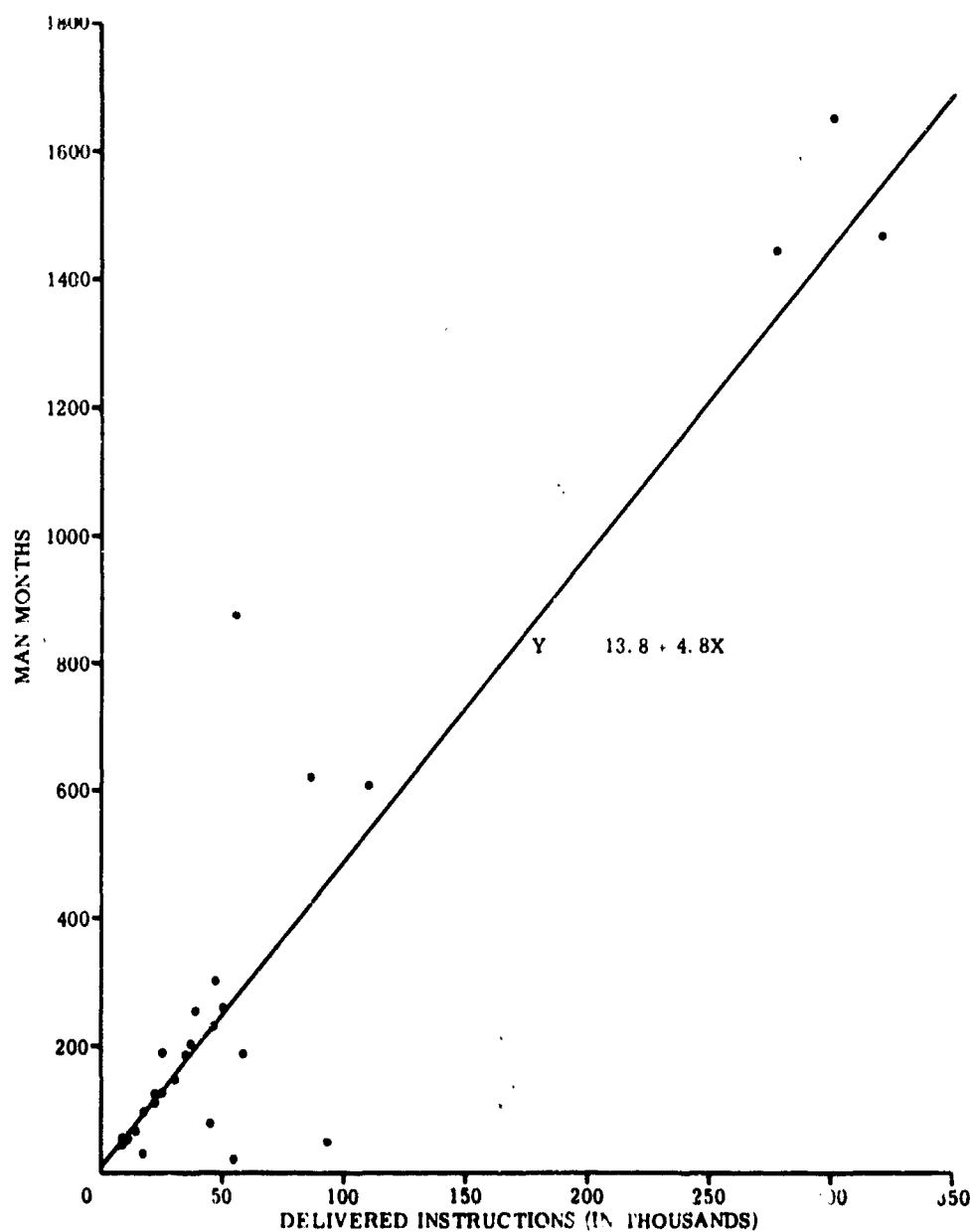


Figure 1. Number of Man Months versus Number of Instructions

(The equation represents a simple linear regression with the available data and is not a reliable predictor.)

10 May 1965

66

TM-2314/000/00

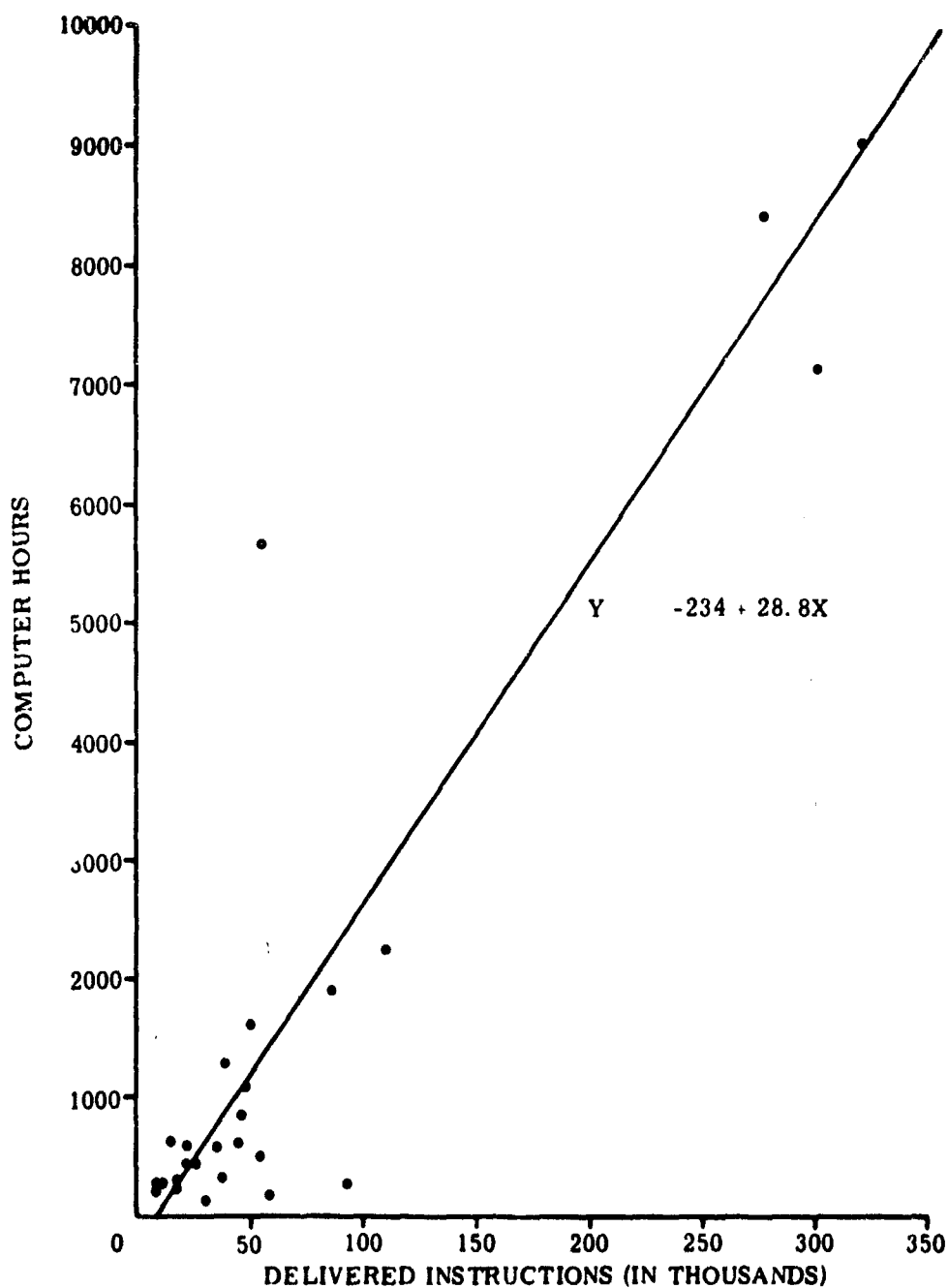


Figure 2. Number of Computer Hours versus Number of Machine Language Instructions

(The equation represents a simple linear regression with the available data and is not a reliable predictor.)

0

10 May 1965

67
(page 68 blank)

DA-2314/000/00

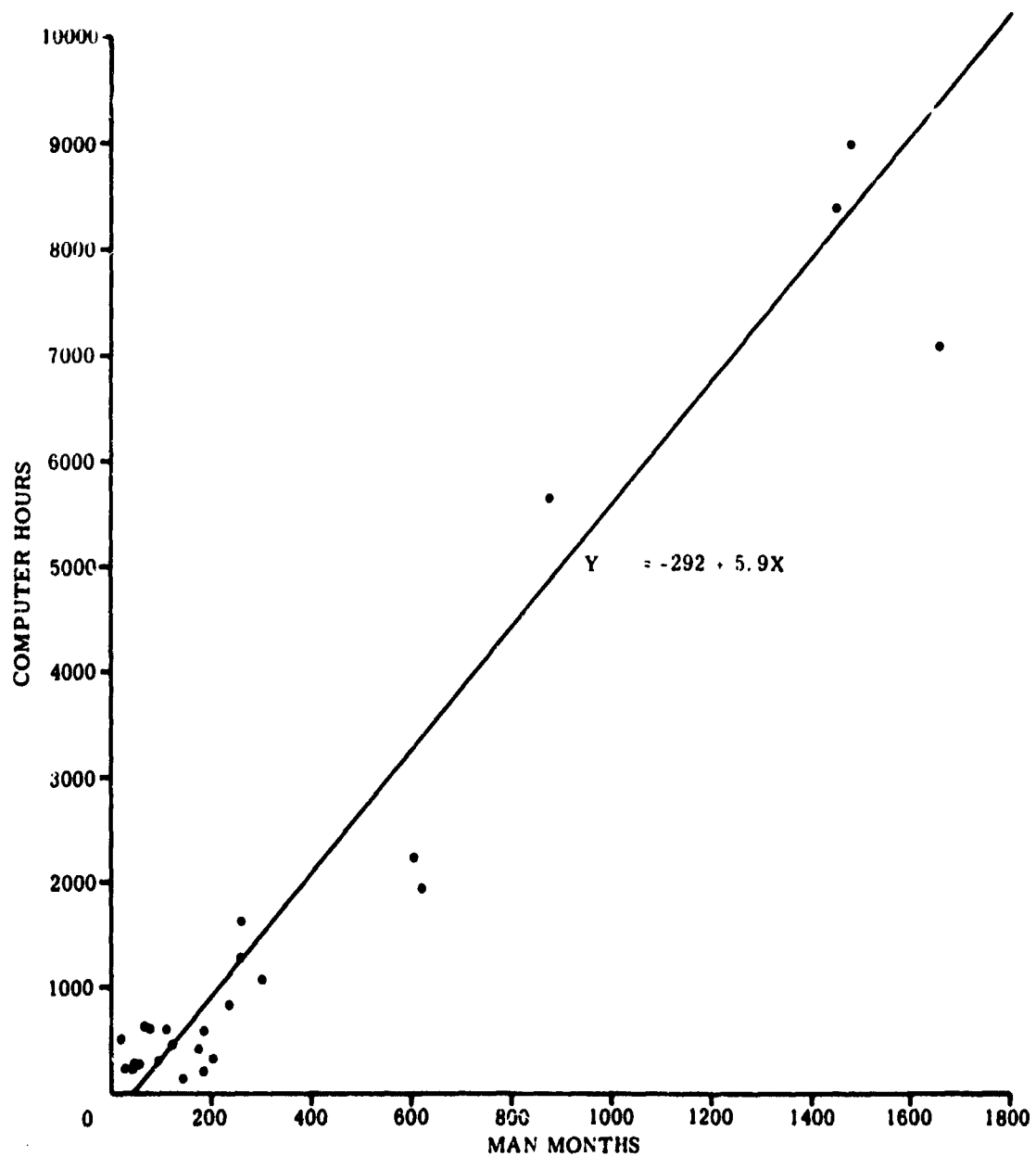


Figure 3. Computer Hours versus Man Months

(The equation represents a simple linear regression with the available data and is not a reliable predictor.)

V. SYSTEM ANALYSIS AND DESIGN

The process of determining the requirements for the program system and planning a set of programs capable of fulfilling them is divided into two Phases--System Analysis and System Design. The first Phase consists of investigating the particular information processing task that is to be adapted to automatic data processing methods; the second consists of attempts to devise a satisfactory solution to the data processing requirements involved.

System analysis and design is a complex process and, for a large system, can be broken down into many fine tasks and phases; in the case of a small or simple information processing task, it may be considered only a single step in the production of the program. The tasks of analysis and design that are largely intangible such as "study," "investigate," and "coordinate" may be made more tangible by requiring specific documents to record the thoughts and actions of the system analysts and designers.

A. OBJECTIVES

Generally, the mission of the analyzing and synthesizing process is to devise the most effective and efficient organization of program functions and elements possible within the constraints of available manpower, funds, and time, to perform the required data processing functions. In the case of feasibility studies, the goals of the work are more limited and involve accurate definition of the information problem and assessment of the possibility of solving it with ADP.

In the System Analysis and Design Phases, system analysts and senior programmers:

- . Define in detail the information processing problem indicated in the Project Request.
- . Devise one or more ways to perform the required functions.
- . Evaluate these alternate designs to select the most effective and efficient solution.
- . Detail the design of the program system specified by this solution.

B. TASKS

For System Analysis, the tasks are:

1. Plan the Project
2. Analyze system requirements
3. Analyze the user's environment
4. Analyze computer program production requirements
5. Analyze similar and interfacing systems
6. Evaluate contract proposals
7. Analyze requests for system change

For System Design, the tasks are:

1. Design the total system
2. Design the computer program system
3. Outline the Preliminary Functional Description
4. Produce the Preliminary Functional Description
5. Familiarize the user with the system design
6. Obtain concurrence on the Preliminary Functional Description
7. Indoctrinate production personnel

Although these tasks apply generally to all Projects, their intensity, i.e., the amount and/or quality of work needed, may vary among Projects. For example, to transfer an ADP capability that exists on one machine to another machine requires little analysis because the Project already has a "proven" program system design. Also, the Project can benefit from the earlier documentation (e.g., Project Estimate, Project Development Plan, and Preliminary Functional Description), the experience of personnel, the records of problems and their solutions, and the actual costs and schedules of the original Project. However, conversions and revisions of old programs invariably include some new analysis and programming. It is dangerous, then, for Project personnel to neglect these functions by assuming that adequate analysis and documentation has been done. Attempts to use poorly maintained documents, or subtle differences between an old and new system, sometimes lead to more costly design work than designing a new system. Therefore, the Project Leader, even in this case, must consider, in planning, all of the analysis and design tasks, and should expect to perform all of them in at least a rudimentary way.

Further, although in a small project analysis and design may be combined into a single task and done by one person, the Project Leader should recognize that the tasks are distinct in terms of time and effort. Some benefits accrue from combination but may be negated by failure to recognize individual task responsibility.

C. COMMUNICATION, COORDINATION, AND CONTROL

The analyst collects information from:

- . The Project Request.
- . Descriptions of the proposed and existing systems.
- . Descriptions of proposed hardware.
- . Descriptions of available program production tools.
- . Documents stating the mission and requirements of the system.
- . Conferences and briefings.
- . Interviews with user and other personnel.
- . Feasibility study reports.
- . Descriptions of interfacing systems.
- . Documents describing the user's mission, responsibilities, and organization.
- . Observations of the existing systems.
- . Files of previous projects.
- . Interviews with expert consultants and other Project Leaders.
- . Simulation studies.
- . Technical literature and professional meetings.
- . Analytic and feasibility studies of his own.
- . Progress reports, trip reports, minutes of meetings, and similar administrative documents.
- . Correspondence files.

The analyst communicates and provides coordination through:

- . Personal contact with users, programmers, Project Leaders of other projects, and other developers.
- . Conferences and briefings.
- . Circulation of documents for review.

- . Concurrence meetings
- . Dissemination of trip reports, minutes of meetings, reports of studies, and confirmatory (feedback) letters and reports following contacts and interviews.
- . Progress reports and other documentation.

Control is established by:

- . Schedules and budgets
- . Project monitoring and program reporting
- . Concurrence procedures
- . Design change procedures
- . Documentation procedures
- . Coordination procedures
- . Product lists and product status reports
- . Planning documents
- . Review procedures
- . Procedures for the verification of information

The analysis and design phases require communication because the personnel collect and generate information, and coordinate information among the programmer personnel, customer, and other developmental agencies. Most analysis work is recorded in documents whose contents must be coordinated and concurred with the customer and, sometimes, with other developers.

The analysis phase needs control mechanisms to (1) assure completeness and accuracy of information, (2) ensure complete coordination, (3) obtain decisions and concurrence, and (4) control change proposals, including their evaluation and implementation.

D. SUPERVISION

During the System Analysis and System Design Phases, this intense need for communication and coordination dictates the Project Leader's responsibilities. He and his subordinate supervisors must monitor the tasks, evaluate their products, coordinate activities, and resolve technical and administrative difficulties.

The Project Leader must make all important technical decisions. In a small Project, he himself must do the planning, set schedules and deadlines, keep abreast of progress, and evaluate all of the analyses and

designs produced. On a larger Project, although he may delegate much of the planning and product review to other senior team members, he remains responsible for the final review and the technical quality of the final products.

One of the most important and time-consuming tasks of the Project Leader is the external coordination of analysis and design plans. He represents the Project in contacts with user personnel, such as briefings, information gathering, and user review and concurrence on Project plans and designs. He also represents the Project to his management by presenting briefings, coordinating plans, and obtaining design approval. The Project Leader must deal with the computer and duplication facilities and other service organizations to arrange for computer time, duplicating and illustrating services, EAM work, and other support. Project success also depends upon other coordination activities, e.g., arranging for conferences and trips, getting reviews of plans and designs, and in obtaining decisions and concurrence.

Administrative matters may be a time-consuming chore for the Project Leader, but good secretarial support can ease this burden. The Project Leader (or, on a large Project, a delegate) must review, approve, and expedite requests for trips, conferences, clearances, and information to be sure these are necessary and accomplished quickly and effectively. The Project Leader is also responsible for work assignments, time reports, progress reports, performance evaluations, salary reviews, and other administrative details.

Responsible for the efficient operation of his team, the Project Leader not only makes plans, but sees that the work indicated is done on schedule. Two particularly difficult tasks are (1) to see that the analysis and design documents are completed on schedule and are accurate and complete and (2) to handle changes to design and the plan and to document the resulting changes in a timely way.

E. COST FACTORS

For small projects, the Project Leader will need only a few skilled analysts to do the work in the Analysis Phase. Since analysis is the first Phase, there is little time to train persons for the job. However, it is not always possible to get all experienced people, especially on larger projects, and less-qualified persons may have to be employed. In this case, their need to gain experience and learn may inflate costs or reduce quality.

For the analysis and design effort, availability of information is perhaps the key cost factor. Clear and complete statements of objectives and functions are usually not available and are not provided with all associated information. For example, an analyst might analyze and design an input or

output format in half a day, given the appropriate information. However, without the information, he may need days or even weeks to locate and actually obtain the information. For example, the analyst may have to arrange mutually satisfactory conference dates and places, possibly get security clearances, and plan his travel. These support activities require a great deal of the analyst's time--particularly in a large effort in which dozens or even hundreds of questions requiring such efforts may arise.

Further, much of system analysis and design work is creative--that is, the information required does not exist but must be generated, e.g., the model of information processing that is needed to design a system. It is extremely difficult to estimate this "cost of innovation." One way is to estimate the amount of new programming and new applications that are in a Project, and find past Projects of a similar nature and study their "innovation" costs.

Incidentally, two major reasons for reviewing old Projects are to avoid "reinventing" and its associated costs and to help the analyst determine the feasibility of an approach. Innovation, or "pioneering," makes meaningful comparison with completed Projects difficult and minimizes the opportunity to learn from experience. For this reason, the cost of doing something for the first time is usually much greater than for subsequent attempts.

Costs of changes made during analysis are less than for those made during any other phase of the program development process. Less work is scrapped, there are fewer elements affected, there are fewer documents to change, and there is much less detail to consider in evaluating a proposed change. Further, if a good job of forecasting is done in analysis, to determine the evolution of the system, fewer costly changes will have to be made. That is, one characteristic of good design is that it can anticipate and accommodate many changes easily.

F. THE CHECK SHEETS

The Check Sheets are organized into a standard format as shown in the following diagram.

10 May 1965

75
(pages 76 and 77 blank

TM-2314/000/00

Task Name		Task Description	
Inputs	Subtasks	Outputs	Costs
Environment			
Interactions and Dependencies		Resources and Working Conditions	

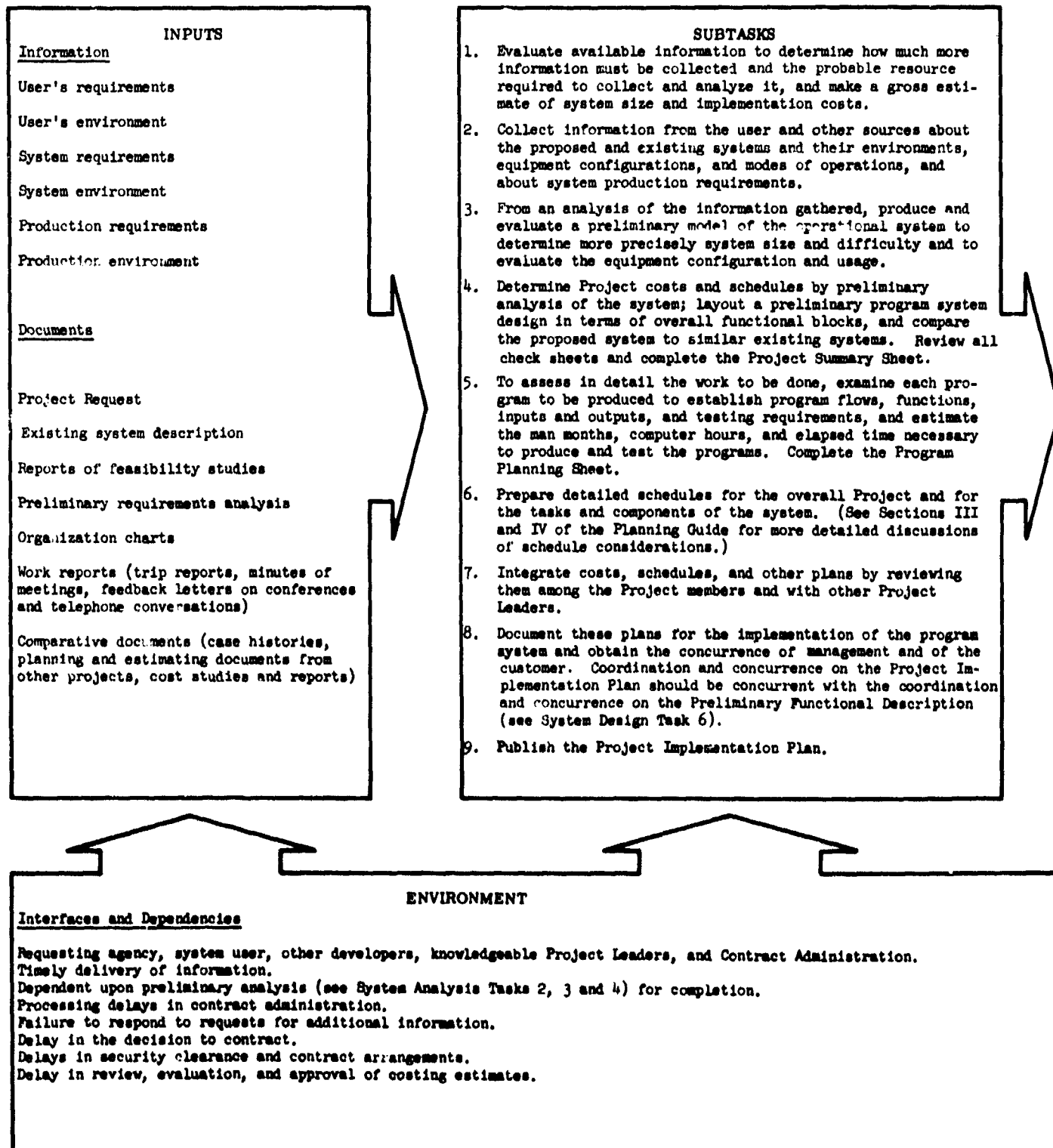
Included are the task name and a general description of the task to be performed. The Inputs are redundant in that they indicate both types of information and specific documents containing it that should be available to perform the task. The task is further detailed in terms of subtasks comprising the task. The Outputs or products of the task are, again, redundant, since they are also information or documents that are necessary inputs to succeeding tasks.

Under Costs are factors that should be considered in estimating the cost of performing the particular task. A costing formula or rule of thumb may also be included.

The lower portion of the sheet lists environmental factors, divided into two parts: interactions and dependencies on other personnel and tasks that may cause delays; and some statements about the nature of resources required and the difficulty of obtaining them are included to indicate the degree of difficulty in performing this task.

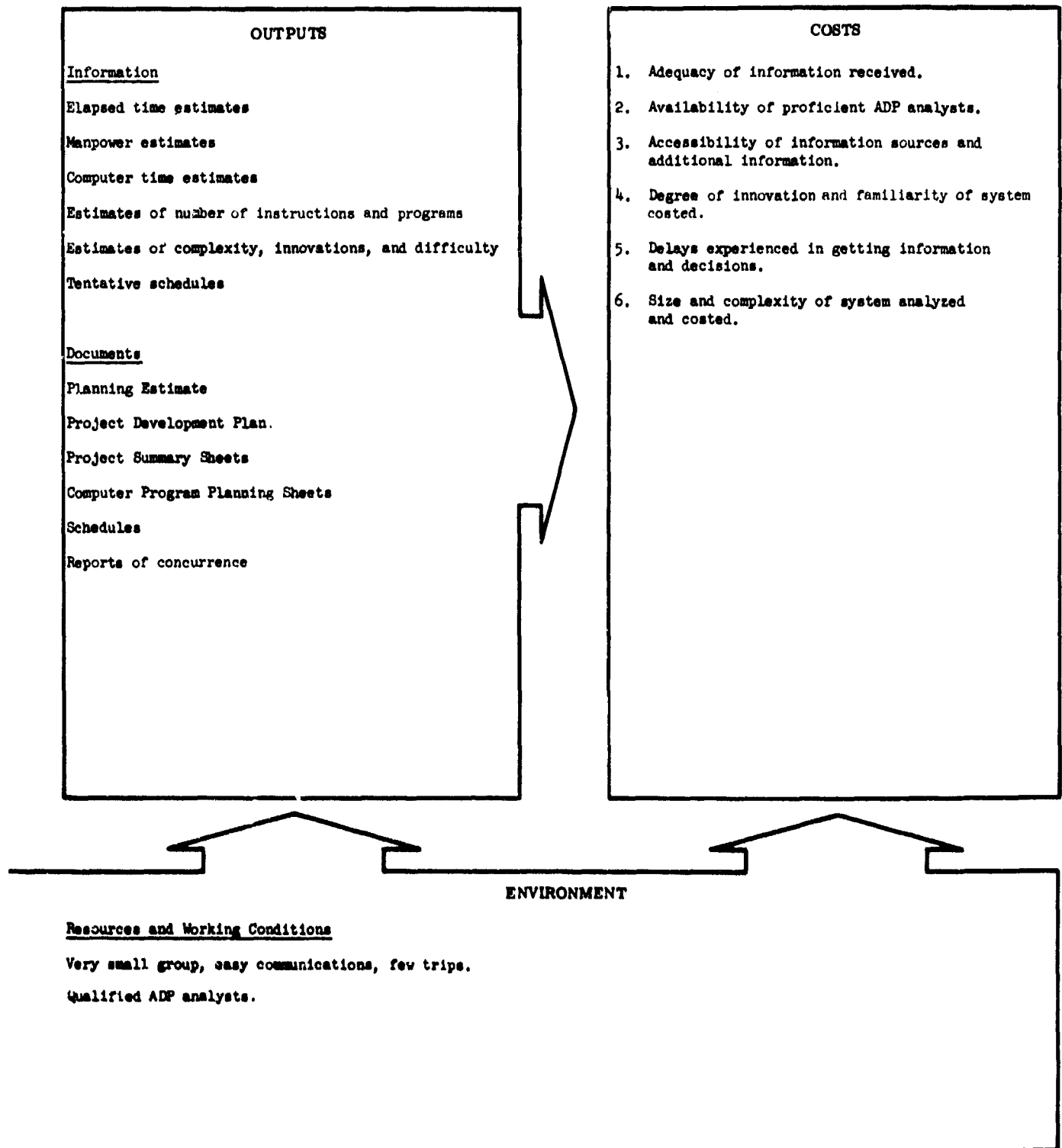
SYSTEM ANALYSIS TASK 1

PLAN THE PROJECT



DESCRIPTION

Study the system requirements and estimate the need for manpower, computer time, elapsed time, and other resources. Prepare the Planning Estimates, Project Development Plan and Project Implementation Plan.



SYSTEM ANALYSIS TASK 2

ANALYZE SYSTEM REQUIREMENTS

INPUTS

Information

(See System Analysis Task 1)

Documents

Project Request

Existing system description

Associated and referenced documents

NAVCOSACT documents on past systems

Allied projects of other agencies

SUBTASKS

1. Assist user in stating functional requirements.
2. Determine data base manipulation requirements.
3. Determine support system requirements.
4. Determine requirements for program operating time for support and day-to-day operation.
5. Study cost effectiveness and feasibility for critical equipment, including, if necessary, computer facility.
6. Study the user's present system by observations, interviews, and study of available documentation.
7. Discuss ambiguities and problem areas with user personnel.
8. Identify special documentation, phase-over, and/or training needs and problems.
9. Coordinate with other analysis tasks such as "analyze similar systems," "analyze the user's environment" (System Analysis Tasks 3, 5).
10. Document the results of the above subtasks.
11. Review requirements with project personnel and revise as needed.

ENVIRONMENT

Interfaces and Dependencies

User on system requirements.

Other analysts on associated system studies, environmental analysis, production requirements.

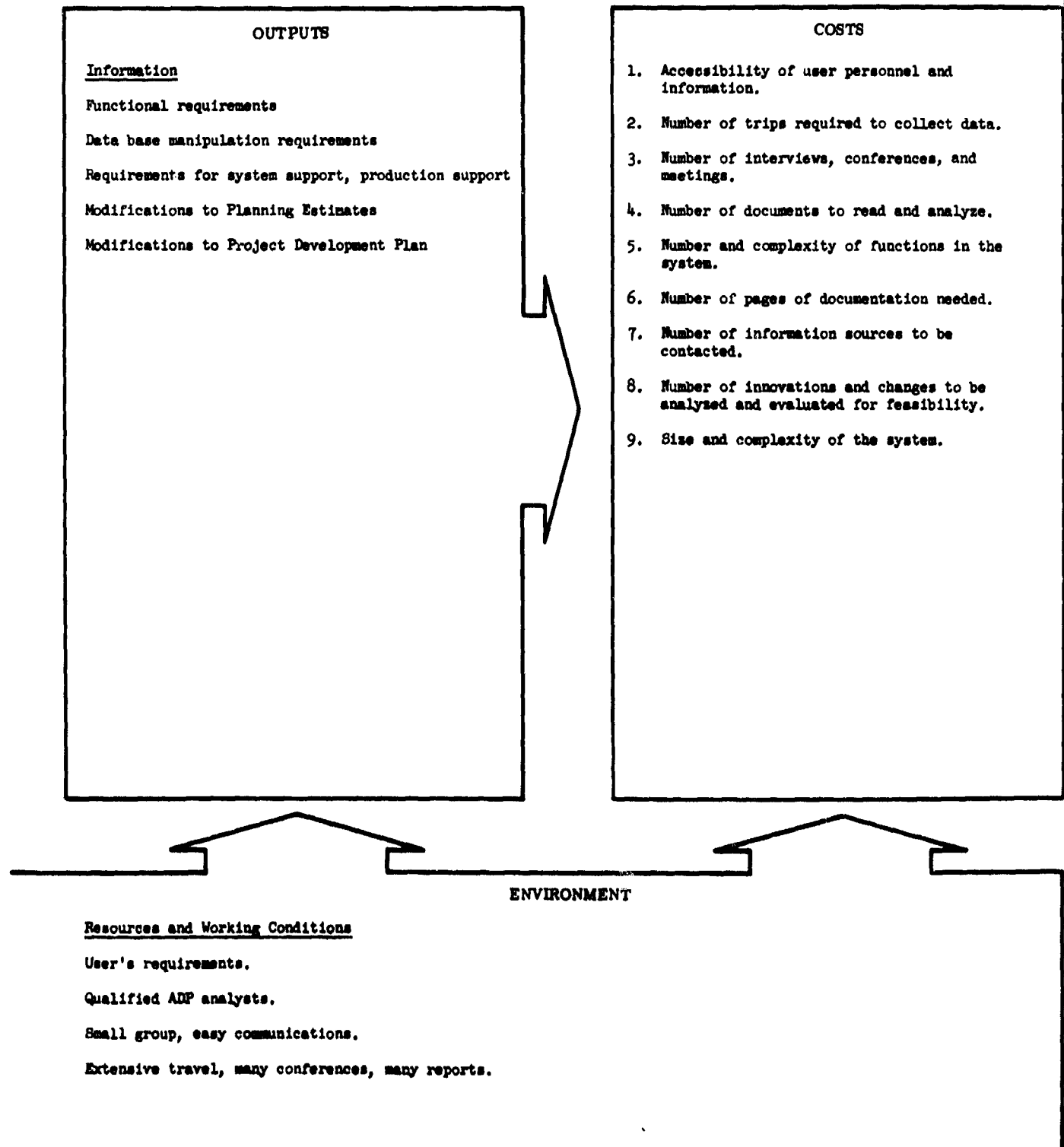
Leaders and other evaluation and review personnel.

Timely delivery of complete user requirements.

Need System Analysis Tasks 3 and 5 for information.

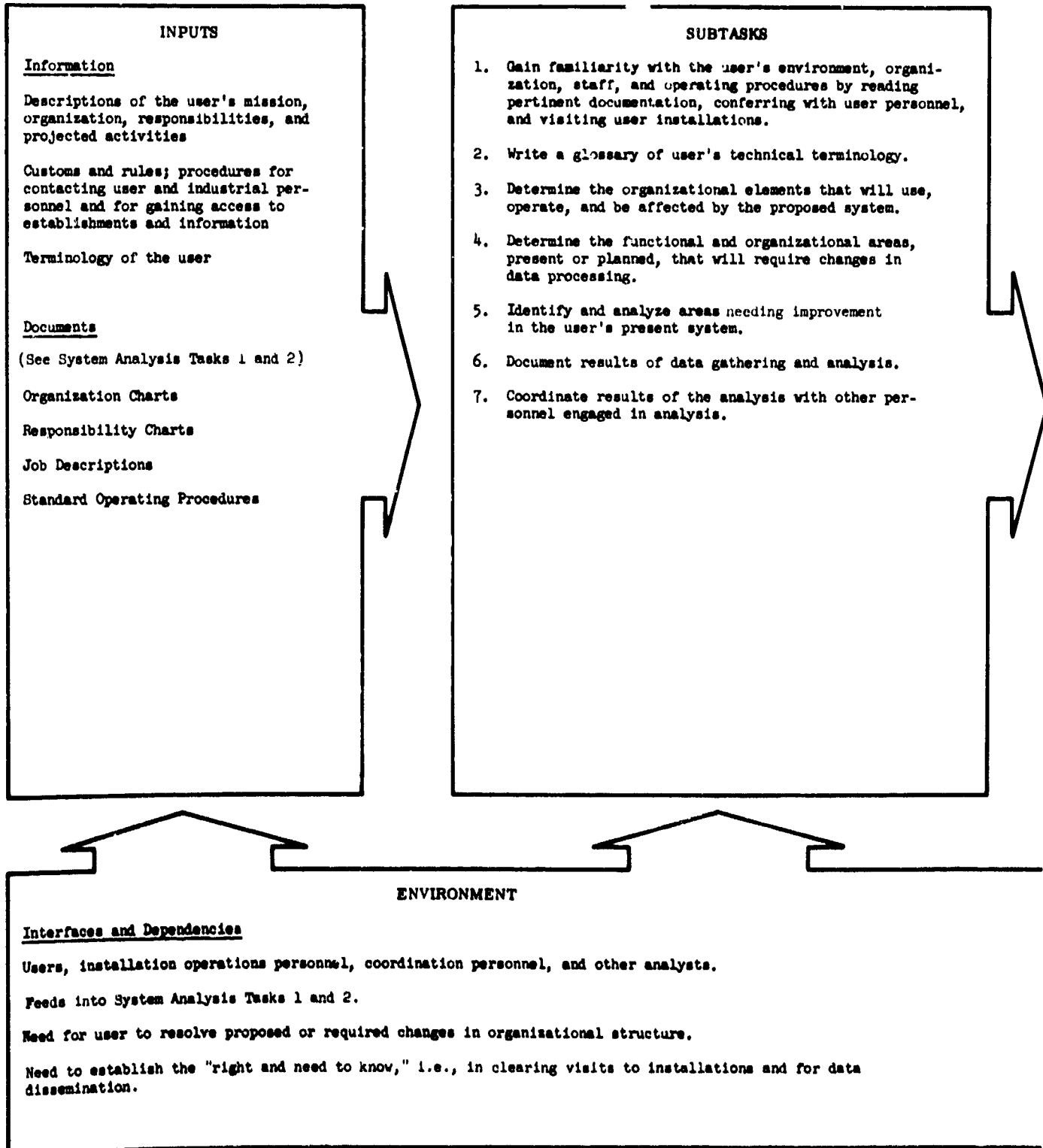
Potential delay in requirements review.

DESCRIPTION Determine the operational requirements of the system and evaluate their completeness, feasibility, and compatibility with other systems by studying the Project Request and its references and by contact and coordination with user personnel.

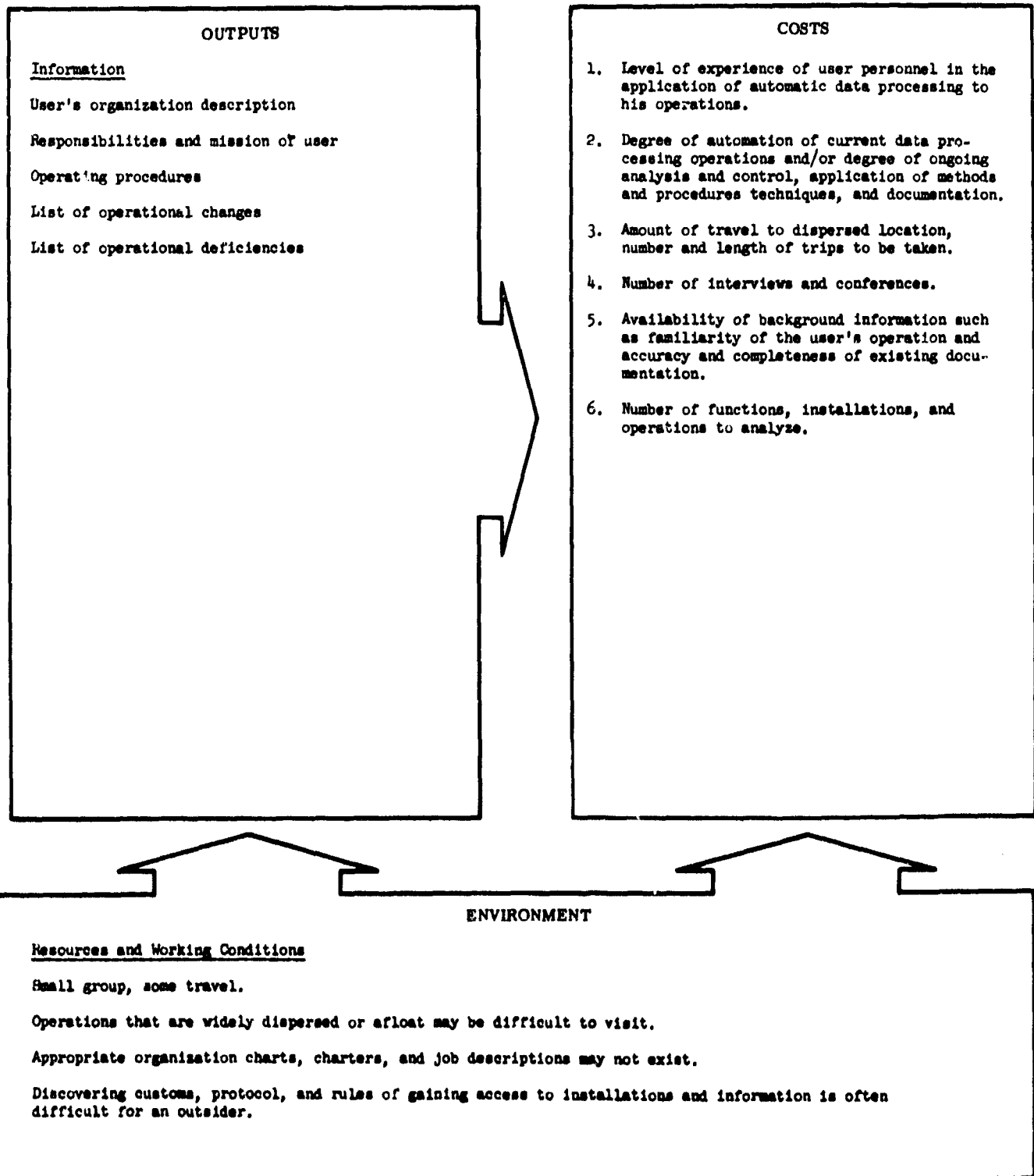


SYSTEM ANALYSIS TASK 3

ANALYZE THE USER'S ENVIRONMENT



DESCRIPTION Study the user's environment and operations, to determine how the system and equipment will be employed, where the operation will be based (ship or shore) and what the responsibilities of the user are, especially to other Naval operations; and to determine the effectiveness and deficiencies of existing data processing operations that might be improved by the proposed system.



SYSTEM ANALYSIS TASK 4

ANALYZE COMPUTER PROGRAM PRODUCTION REQUIREMENTS

INPUTS

Information

Descriptions of languages, compilers, assemblers, utility systems, monitors, test data generation, and recording and reduction systems

Descriptions of the computers, and other equipment, its machine language and command structure

Reports on EAM, computer and support program usage experience and delivery dates

Details of system to be produced (real-time, relocatable, task-oriented, cyclic vs. stacked job, etc.)

Computer operating procedures

Documents

Project Request

System descriptions

Project files

Manufacturer's equipment manual

NAVCOSACT Instructions (policies and procedures) on machine requests

SUBTASKS

1. Determine average turnaround and availability of effective time for the computer.
2. Determine project priority during the production period.
3. Examine procedures, schedules, and backlog of EAM shop.
4. Ascertain programming language or languages to be used.
5. Detect and evaluate differences between computer operations for production (e.g., at NAVCOSACT or contractor) and at the command center.
6. Determine distance and time separation between programming staff and the test computer.
7. Determine procedures for submitting programs for computer run.
8. Investigate utility and support program systems, including print, compile, assembly, etc., to determine reliability, availability, ease of use, and state of documentation.
9. Investigate the monitor or executive system controlling work on the computer, including state of checkout, state of modification, and functional capabilities.
10. Determine actual and potential hardware constraints such as amounts and kinds of storage, input/output devices, displays, etc., that influence the design of the program system.
11. Investigate potential back-up computers and conditions of use.
12. Coordinate with other analysis activities.
13. Advise those producing Implementation Plan and planning for the Indoctrination of Production Personnel (System Design Task 7) on characteristics and limitations of the computer and production tools.

ENVIRONMENT

Interfaces and Dependencies

User, on location and operation of proposed and existing equipment.

Computing operations, on operating procedures and use experience.

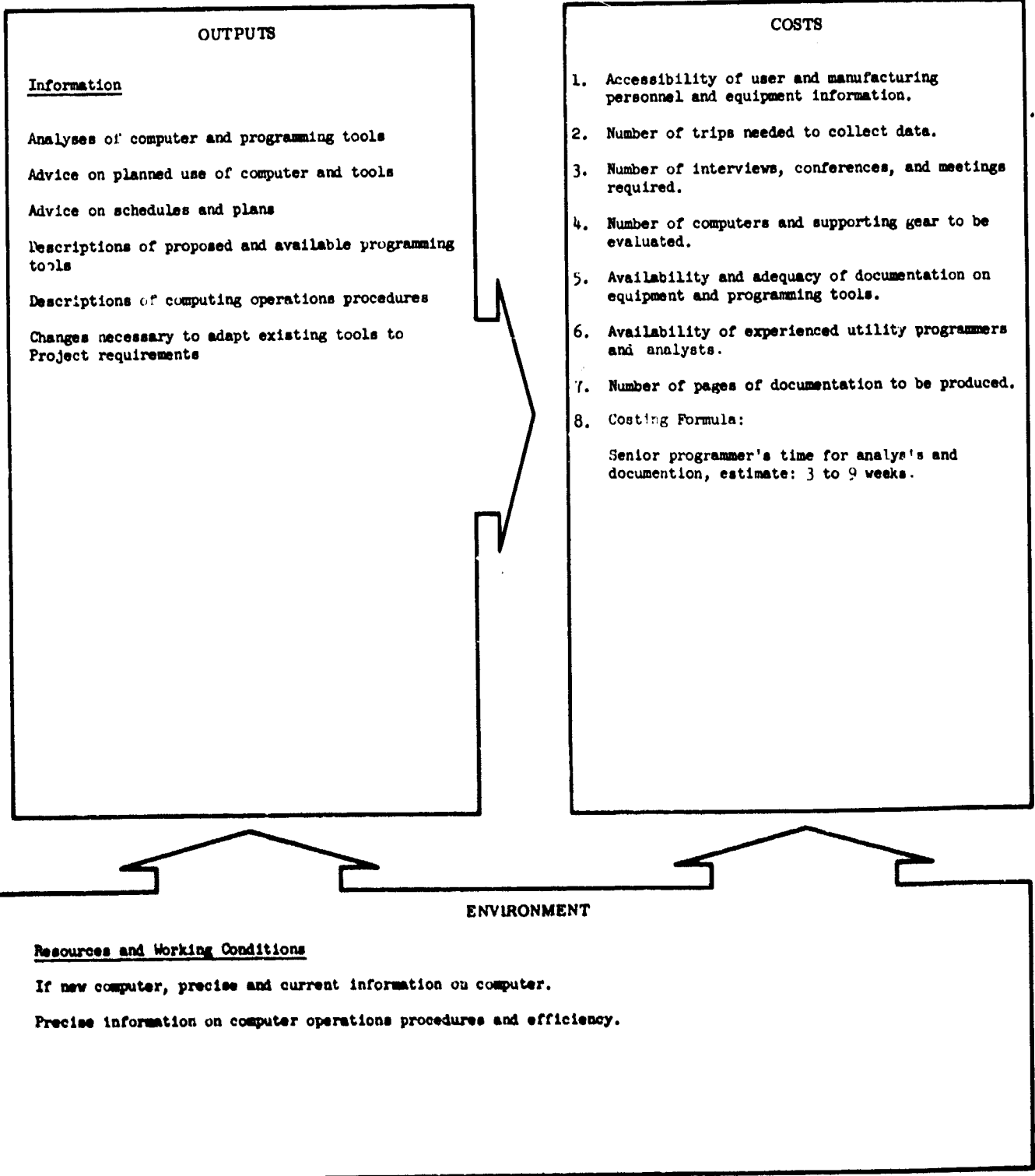
Equipment manufacturers, on equipment characteristics.

Software producers, on support program characteristics.

Dependent upon System Analysis Task 2 and 3 for system information.

Potential delay in contacting user and manufacturer for visits and information gathering.

DESCRIPTION Determine the requirements for program production and test, the adequacy of available tools, and the tools required to produce the proposed system by studying the total environment for program production, including computer operations, experience with the project computer and facility, the projected availability of the machine, the availability of back-up equipment, and amount and kind of programming languages, operating systems, and other programming support.



SYSTEM ANALYSIS TASK 5

ANALYZE SIMILAR AND INTERFACING SYSTEMS

INPUTS

Information

Similar systems in planning, production and use.

Other applicable programs, procedures, tools, and techniques

Documents

Project files

Library files

Planning documents and reports from other projects

Subroutine libraries

Preliminary system design documentation

Proceedings of professional meetings

SUBTASKS

1. Study Project files and reports, and the Catalog of ADP Capabilities (NAVCOSACT Report 0047) to identify similar systems and extract and evaluate useful facts.
2. Interact with NAVCOSACT departments, other agencies, organizations, and industry to identify systems that will interface with the projected system, and extract and evaluate the pertinent facts.
3. Identify applicable programs, procedures, techniques, and tools by searching technical books and journals, sources such as the IBM Catalog and SHARE Library listings.
4. Coordinate the results of the search with System Analysis Task 2 personnel.
5. Isolate elements of the projected system, such as routines and data files, that may be available from other projects and systems.
6. Confirm results with command and development personnel who have had experience on similar systems; add to or revise results, and publish.

ENVIRONMENT

Interactions and Dependencies

Personnel of other projects, users and developers of other systems, professional personnel and societies, and contractor and intergovernmental agencies.

Delays in getting data on systems outside NAVCOSACT (particularly on classified systems).

Delays in getting clearance for access to such data and access facilities of such systems.

Library research often overruns budgets and schedules unless closely monitored.

DESCRIPTION

Determine if there are systems, subsystems, procedures, tools, and techniques already in production or use, or planned, that may influence the current Project or provide useful information for Project plans.

OUTPUTSInformation

Pertinent facts on similar and interfacing systems, and applicable tools, techniques, and procedures

Documents

Reports documenting the above information

COSTS

1. Volume of files and literature to be searched.
2. Efficiency of information retrieval system.
3. Familiarity with the application involved.
4. Number of similar and interfacing systems identified that must be studied and evaluated.
5. Number of briefings, conferences, and interviews to be conducted, both to retrieve and to disseminate information.
6. Volume of documentation to be produced.
7. Costing Formula:

Estimate 2 to 10 man weeks depending upon the nature of the project.

ENVIRONMENTResources and Working Conditions

Considerable searching of files and literature.

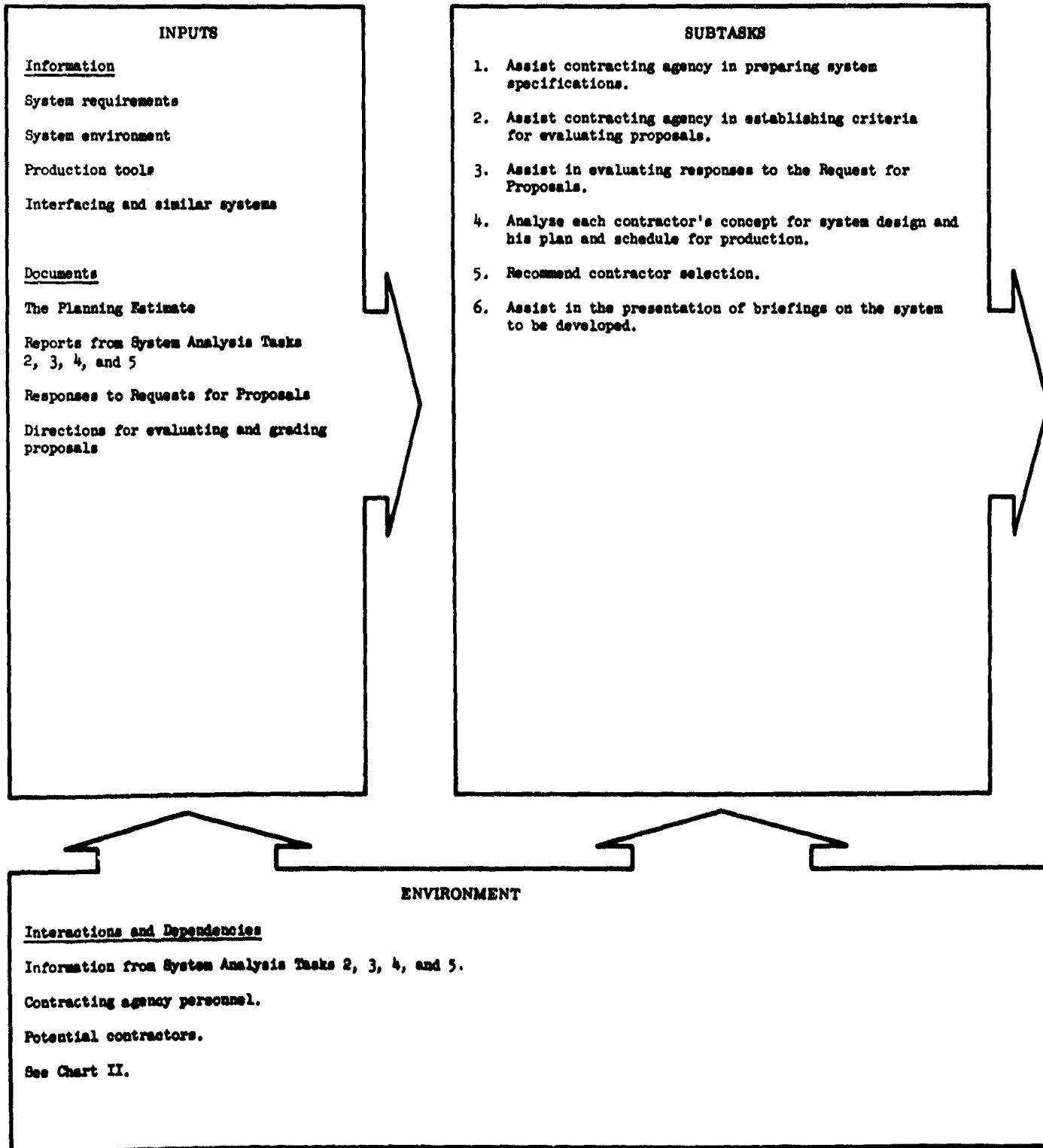
Some contact with outside personnel.

Trips and conferences.

Information about systems planned, in production, or in use--frequently vague, general and of limited usefulness.

Adequate and useful summaries of the professional literature seldom readily available.

Personal acquaintances, conferences, and briefings are usually the best sources of information.

SYSTEM ANALYSIS TASK 6**EVALUATE CONTRACT PROPOSALS**

DESCRIPTION

Assist contracting agency in specifying the work to be done, in specifying criteria for evaluation of proposals, and in evaluating the proposals that are submitted.

OUTPUTSInformation

System specifications

Evaluation criteria

Briefings on the system to be developed

Evaluation of proposals and relative scores

Recommendations regarding contractor's approach to system implementation

COSTS

1. Size of the planned system.
2. Type of Request for Proposal (sole-source or solicited).
3. Number of bidders, i.e., number of proposals to be evaluated.
4. Costing formula:

See Chart estimates range from 8 to 48 man weeks, depending upon the conditions.

ENVIRONMENTResources and Working Conditions

Experience in the field of information processing.

Detailed knowledge of user's requirements.

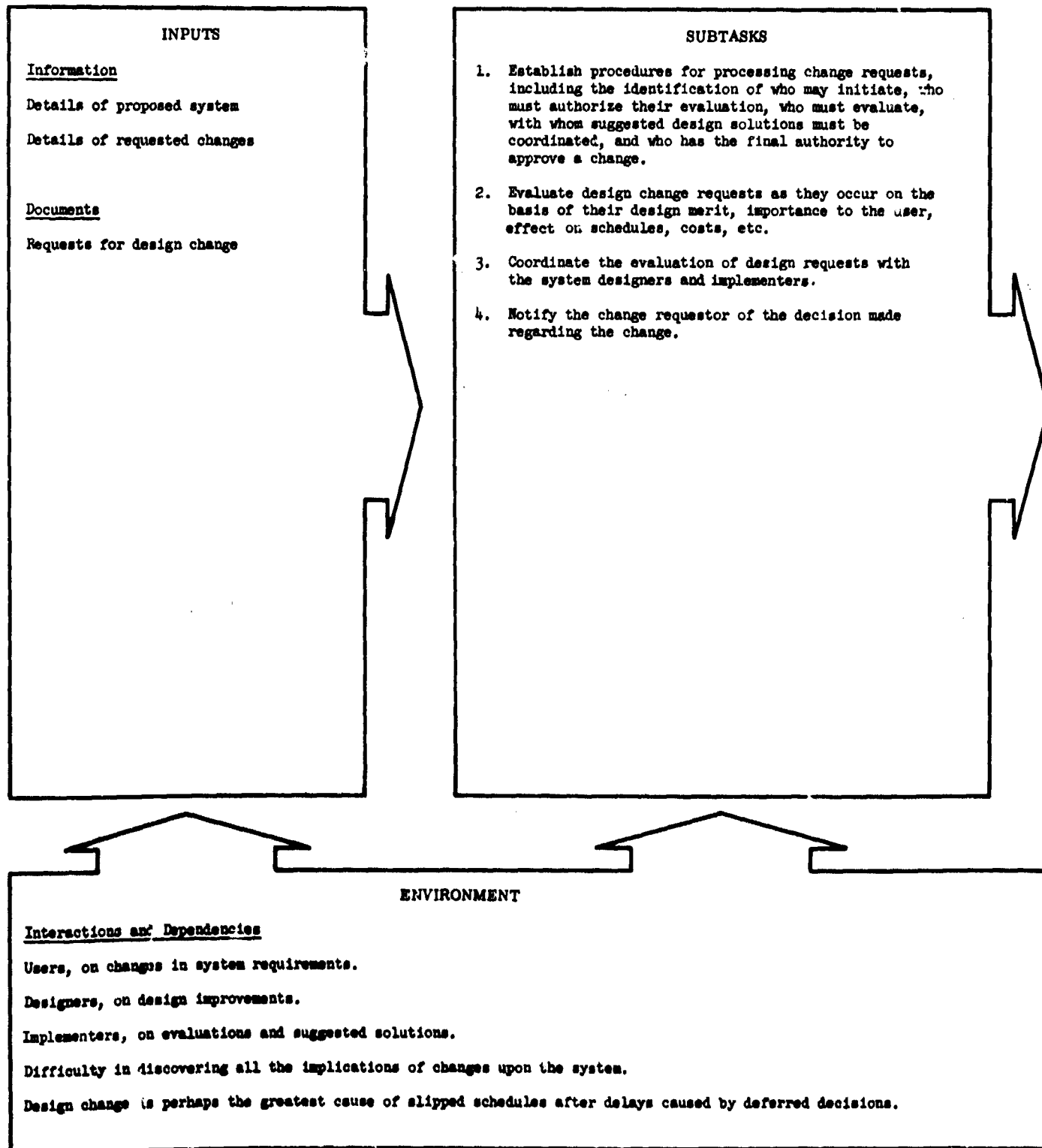
Experience in the evaluation of proposals.

Slow contract proposal and evaluation process sometimes requires long lead times.

Pressure to review quickly because of tight schedules.

SYSTEM ANALYSIS TASK 7

ANALYZE REQUESTS FOR SYSTEM CHANGE



DESCRIPTION

Establish procedures for processing requests for change, and receive, evaluate, and respond to requests for changes in system design.

OUTPUTSInformation

Evaluations of design change requests

Coordination of requested changes

Documents

Change evaluations

Coordination memos to requestor

Design change authorizations

Reports on formal disposition of requests

Changes to plans - Planning Estimate, Project Development Plan, Program Implementation Plan

COSTS

1. The size and extensiveness of the change requested.
2. Relative point in the system development process at which the change is requested. In general, the later in the process the more the cost because of more work scrapped, more work to be done to bring the change up-to-date, and more difficulty in changing, i.e., more decisions needed.
3. The size and duration of the Project, how much might be changed, how much time for system environment to change, how much time for the user to have second thoughts, and how much turnover among users.
4. Costing Formula:

Experience at NAVCOSSACT indicates the number of changes averages 4 or 5 for most projects, may range up to 15 for some.

Each change should be individually evaluated and costed, i.e., work necessary, work scrapped.

Gross estimates: 5-20% additional for costs, 10-15% for schedules.

ENVIRONMENTResources and Working Conditions

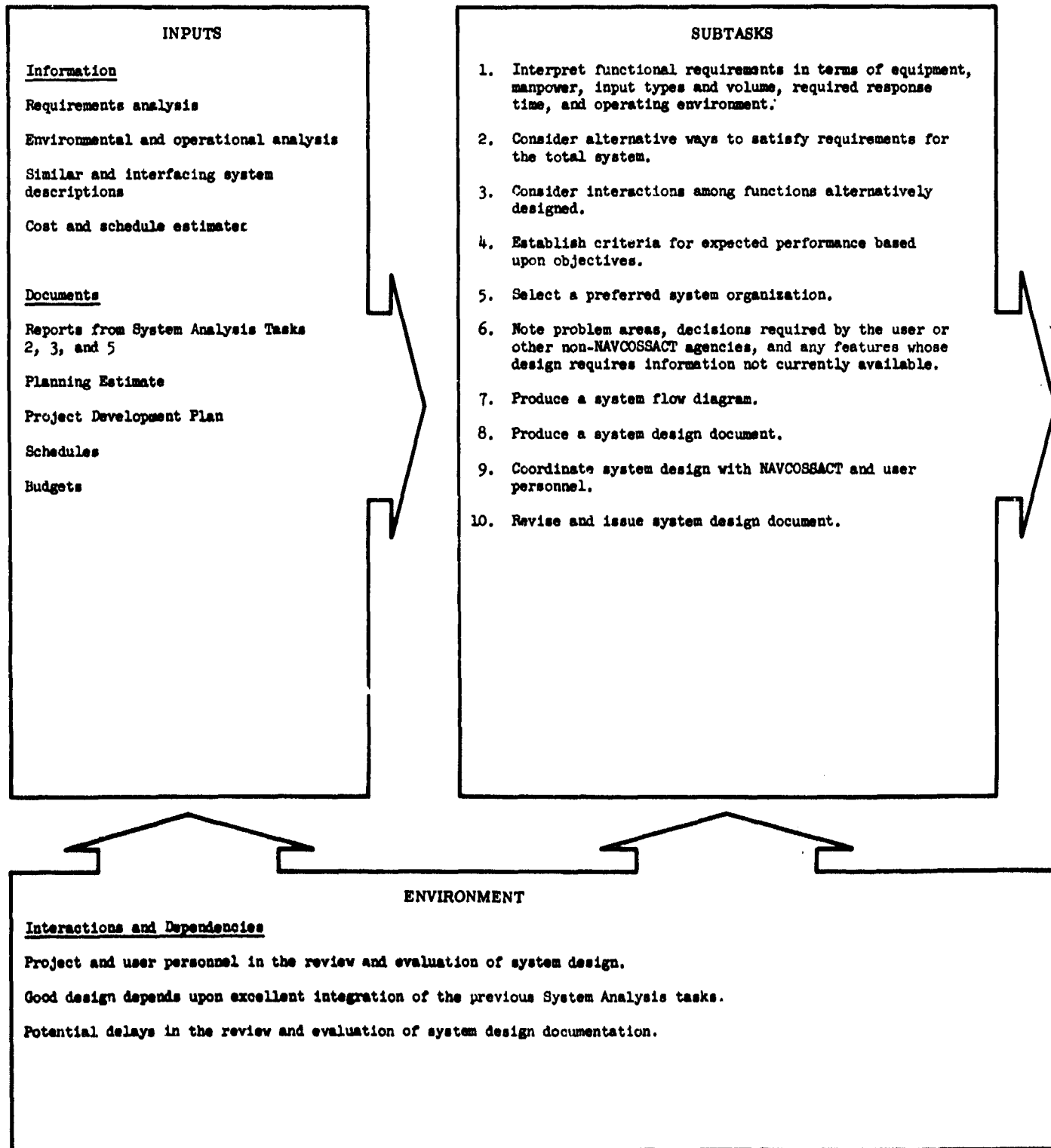
Experienced personnel to perform the required analyses, designs, and implementations.

Coordination task, little travel or outside contact necessary except on major changes.

Ease of evaluation depends upon the accuracy and detail of documentation and general knowledge about the system.

SYSTEM DESIGN TASK 1

DESIGN THE TOTAL SYSTEM



DESCRIPTION

Develop the total information processing system, and the system configuration that is expected to meet requirements to operate in the user's environment, and produce a system flow chart and system design document.

OUTPUTSInformation

System design

Evaluations of alternative designs

Documents

System design document

System flow diagrams

COSTS

1. Familiarity with system requirements and operations and degree of innovation required to handle them.
2. Amount of new design needed. Usefulness of existing designs.
3. Effectiveness with which earlier tasks are discharged.
4. Size and complexity of requirements.
5. Costing Formula:

Estimates range from 1 to 3 man months, depending upon the conditions indicated above and the delays experienced.

ENVIRONMENTResources and Working Conditions

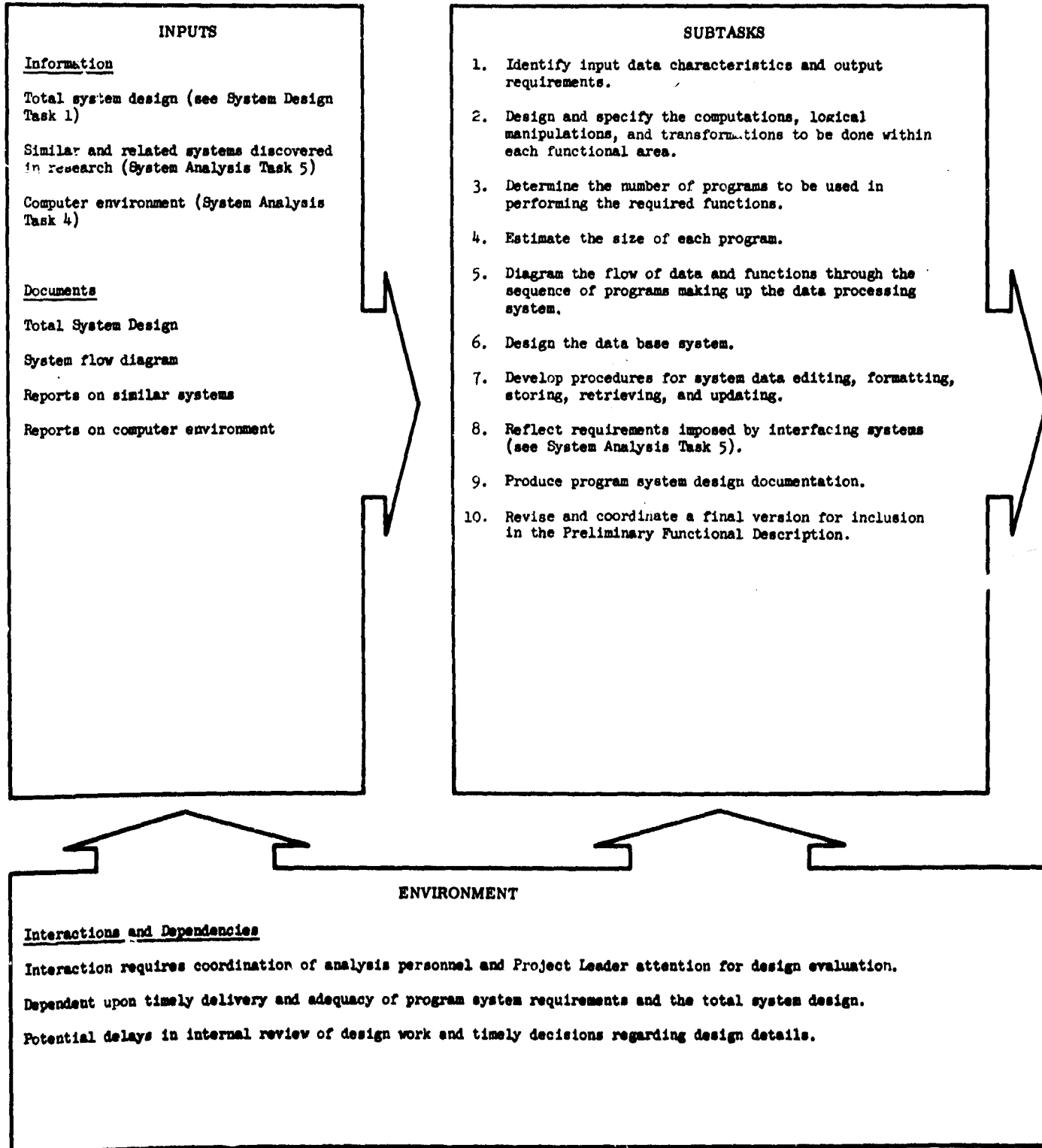
Complex, highly technical task.

Precision and accuracy of information crucial to good design task.

Availability of appropriate solutions to data-processing tasks requiring innovation, are often costly.

SYSTEM DESIGN TASK 2

DESIGN THE COMPUTER PROGRAM SYSTEM



DESCRIPTION

Develop the design for the program system part of the total information processing system.

OUTPUTS

Information

Program system design

Data base design

Documents

Program System Design

Program System Flow Diagram

Data Base Design

COSTS

1. System size and complexity.
2. Degree of innovation and creativeness required.
3. Adequacy of program system requirements and total system design.
4. Experience and skill of available program analysts.
5. Costing Formula:
Estimated at 10% of the total Project man months.

ENVIRONMENT

Resources and Working Conditions

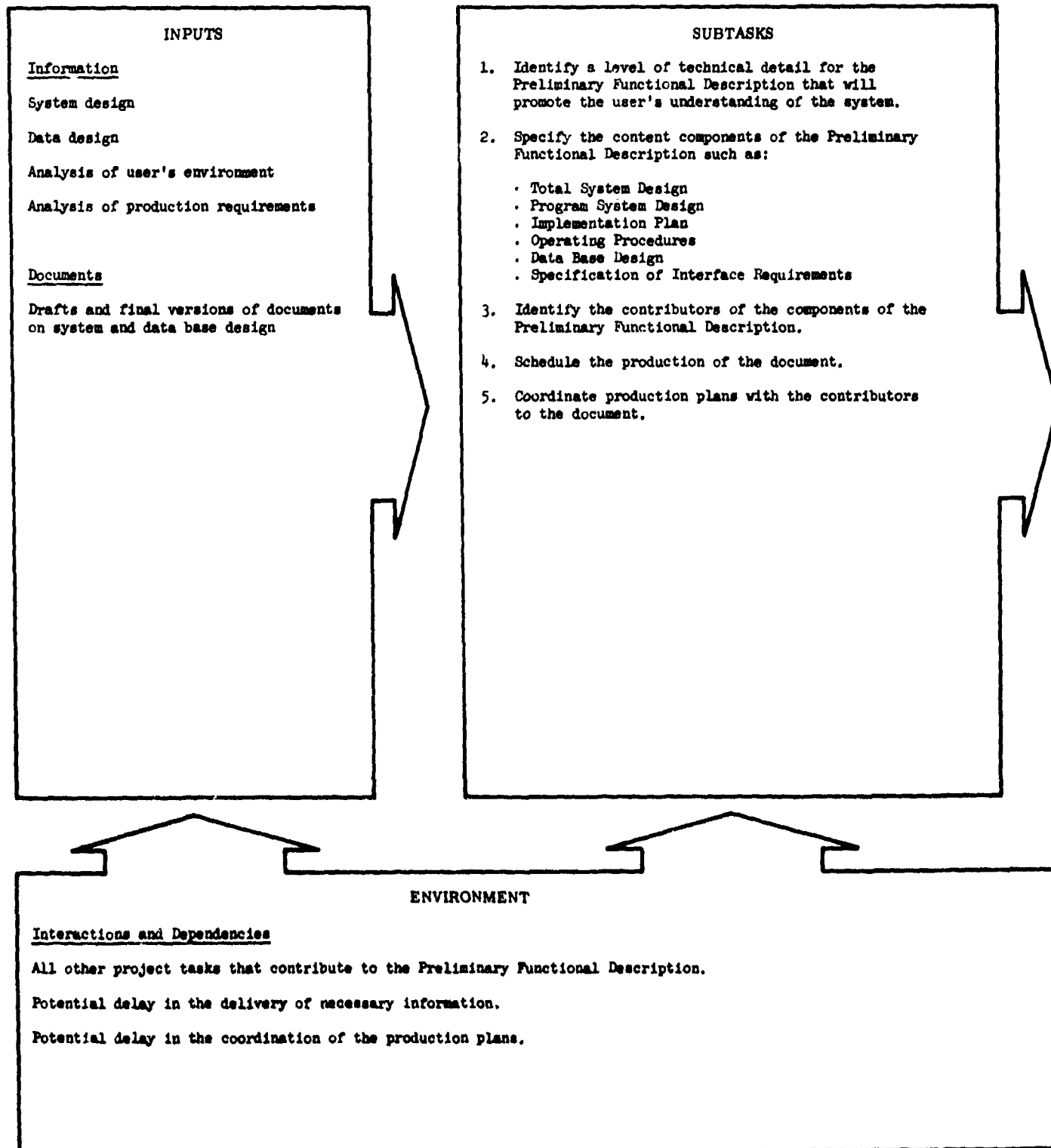
Creativity, knowledge, experience, and time of the senior programmers, program analysts, and Project Leader.

Complex, highly creative task.

Home office environment, little travel unless major changes must be included.

SYSTEM DESIGN TASK 3

OUTLINE THE PRELIMINARY FUNCTIONAL DESCRIPTION



DESCRIPTION

Determine the level of technical detail required in the Preliminary Functional Description, develop the outline of a document to satisfy these requirements, identify those who will contribute to the document, and prepare and coordinate plans for its production.

OUTPUTSInformation

Formats and contents of PFD
Production plans and schedules
Coordination of formats and plans

Documents

Specifications for the Preliminary Functional Description (outline)
Production Plans and Schedules for the Preliminary Functional Description

COSTS

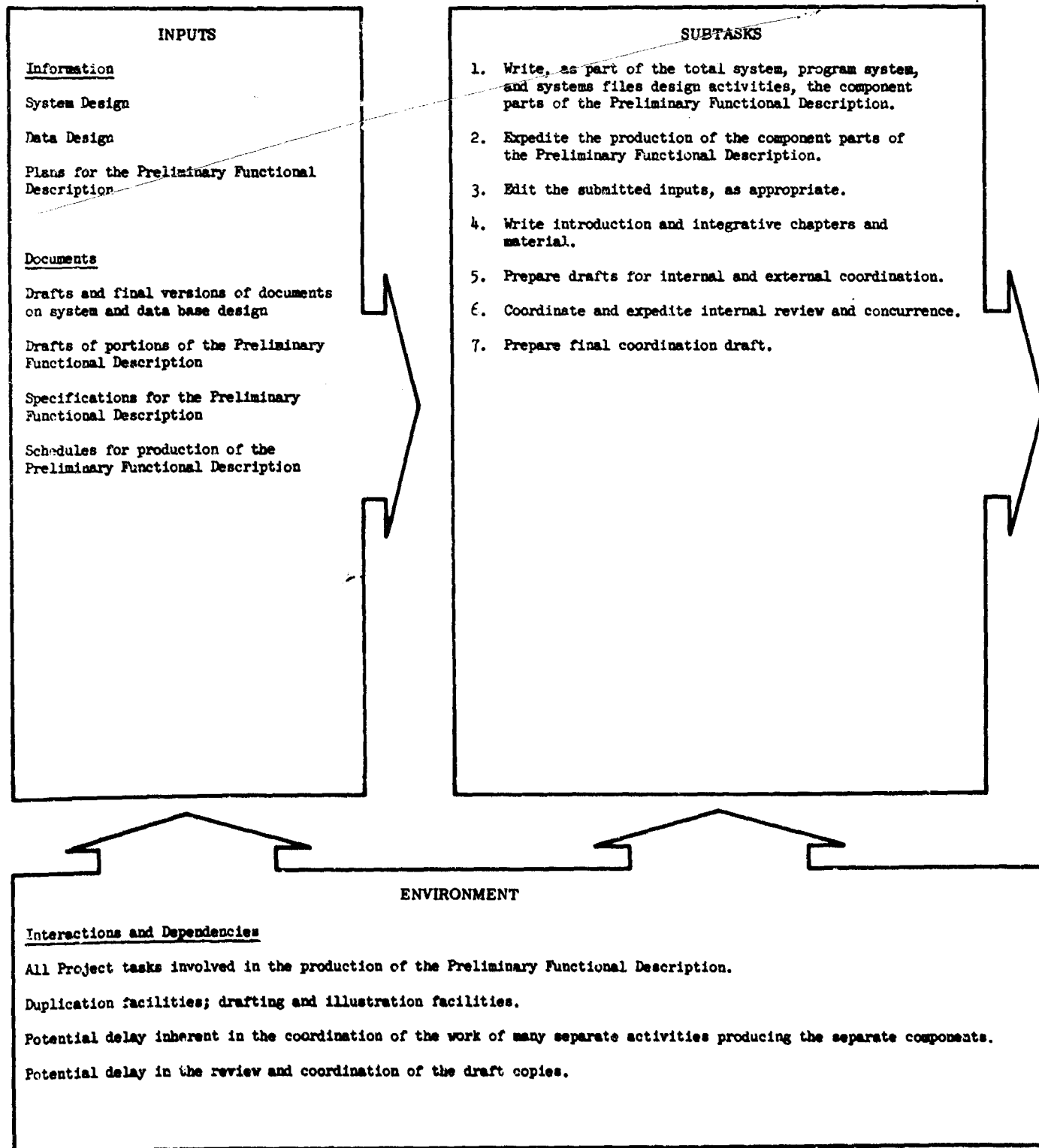
1. Degree of detail required in the analysis of the user's requirements and in the specifications of the document's contents.
2. Number of persons required to review the outline and production plans.
3. Costing Formula:
Two man days per page of outline and schedule.

ENVIRONMENTResources and Working Conditions

Usually a staff job, editorial staff, or technical writing and editing support activity.
Technical writing and graphic arts desirable; editorial and secretarial support required.

SYSTEM DESIGN TASK 4

PRODUCE THE PRELIMINARY FUNCTIONAL DESCRIPTION



DESCRIPTION

Produce and coordinate a document that describes in detail the system to be developed and the environment within which it is to operate.

OUTPUTSInformation

Descriptions of the system to be developed

Internal coordination and concurrence on the Preliminary functional Description

Documents

Drafts of the Preliminary Functional Description for internal review

Final coordination draft of the Preliminary Functional Description

COSTS

1. Number of pages of documentation to produce.
2. Number of illustrations to design and produce.
3. Number of separate parts to integrate and explain.
4. Number of drafts produced.
5. Productivity rates of those who write, type, review, coordinate, modify, edit, illustrate, reproduce, assemble, bind, and distribute the document.
6. Number of reviewers.
7. Costing Formula:

Ignoring reproduction and review costs, two man days per page of documentation.

NOTE: The bulk of the writing and revision costs are included in the costs of other system analysis and system design tasks.

ENVIRONMENTResources and Working Conditions

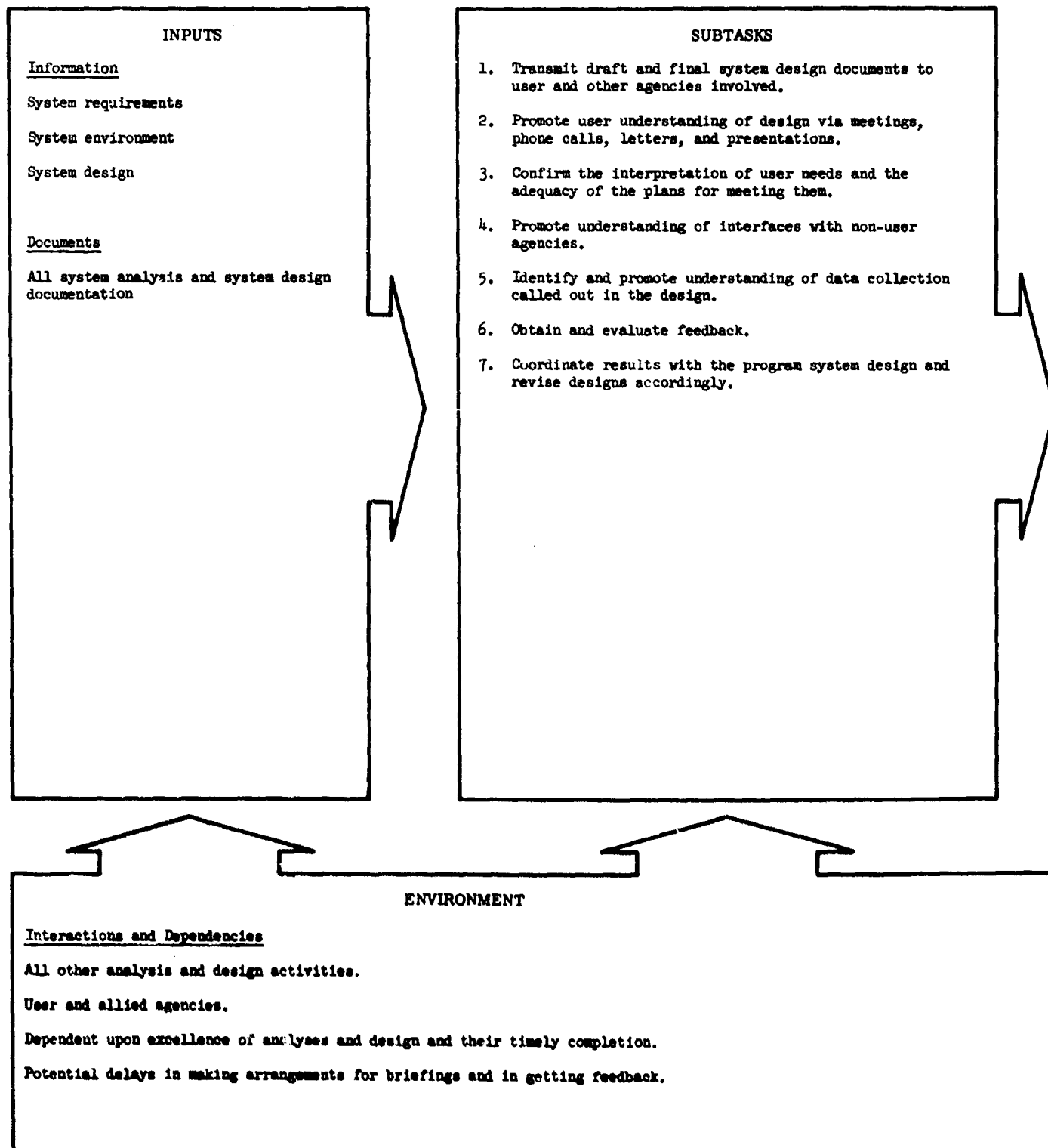
Technical writing and coordination task.

Requires editorial and duplicating support.

Writing skill required. Editorial and graphic arts support is desirable; secretarial and duplicating support required.

SYSTEM DESIGN TASK 5

FAMILIARIZE THE USER WITH THE SYSTEM DESIGN



DESCRIPTION

Inform the user and other interested agencies, at both working and command levels, of the system design and its expected operation.

OUTPUTSInformation

User understanding of the evolving system design
Interacting agencies' understanding of data collection and other interactions with the system
Feedback on the adequacy of interpretations and designs
Design changes
Coordination of Project plans

Documents

Briefings
Display material
Feedback reports
Change Requests

COSTS

1. Number of contacts, briefings, and conferences.
2. Number of agencies requiring coordination.
3. Sophistication (knowledge and experience level) of user and other agencies in ADP applications to their operations.
4. Size and complexity of system.
5. Adequacy of analysis and design work.
6. Number and length of trips taken.
7. Costing Formula:

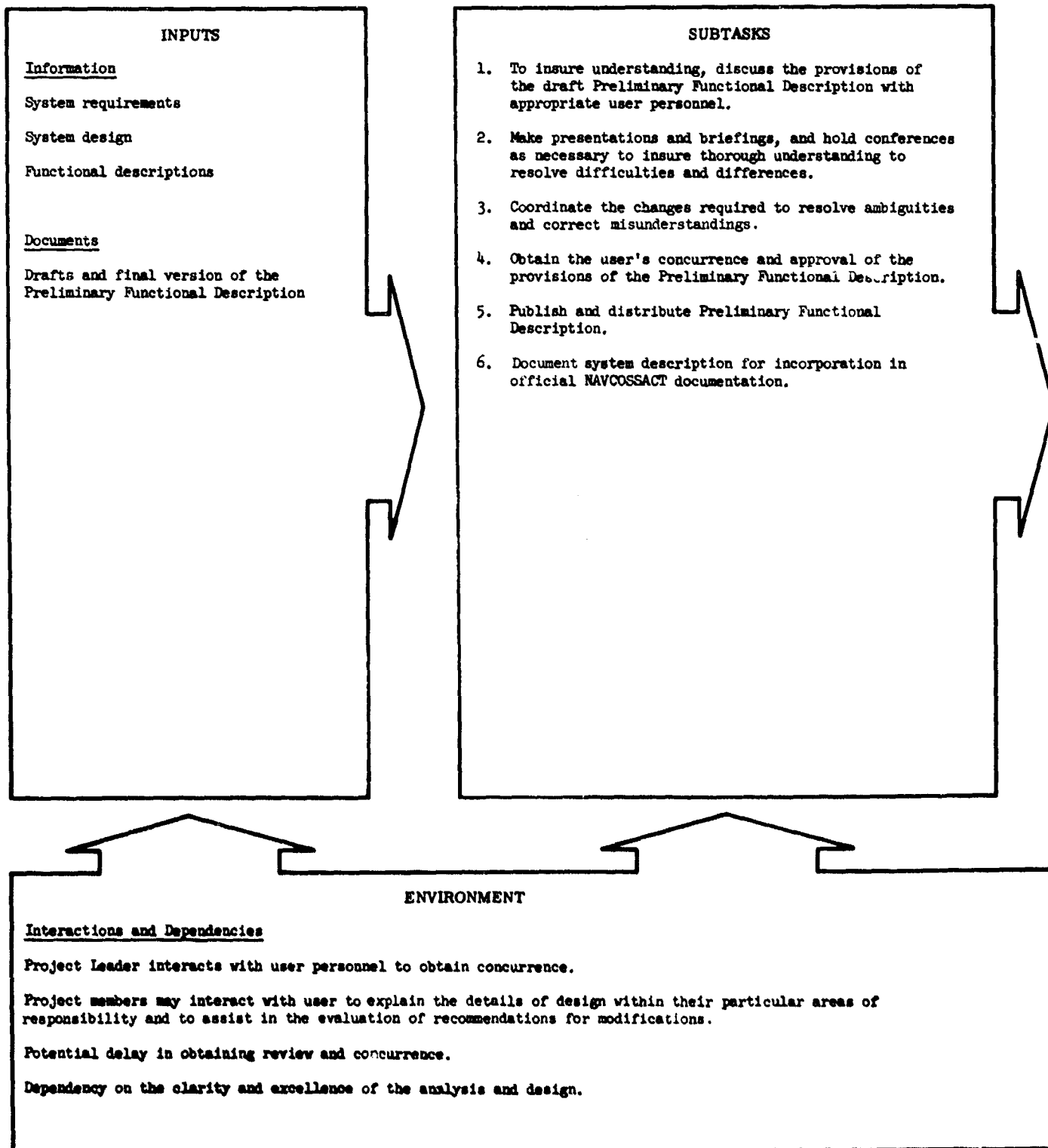
Three man days per design document per agency contacted, plus allowances in elapsed time for travel.

ENVIRONMENTResources and Working Conditions

Requires detailed knowledge of total system.
Good public appearance and ability to relate to others.
An ability to communicate clearly and concisely about complex technical subjects.
Many contacts with users--conferences and briefings.

SYSTEM DESIGN TASK 6

OBTAIN CONCURRENCE ON THE PRELIMINARY FUNCTIONAL DESCRIPTION



DESCRIPTION

Present draft Preliminary Functional Description to the user, discuss its contents with him to insure understanding, coordinate the changes necessary to resolve ambiguities, and obtain his concurrence on the details of the system.

OUTPUTS

Information

Briefings, conferences, and presentations

Changes and corrections to Preliminary Functional Description

Concurrence on PFD

Documents

Preliminary Functional Description

Notification of user's approval of the system

Memoranda on changes and corrections

Memoranda recording the results of briefings and conferences

System Description

COSTS

1. Adequacy of prior indoctrination and liaison (See System Design Task 5).
2. Degree of participation of user in the system analysis and design procedure.
3. Technical and editorial excellence of the Preliminary Functional Description.
4. Size and complexity of the system as reflected in the amount of documentation and information that must be considered in the review and concurrence process.
5. Data processing experience of the user as reflected in the amount of indoctrination that must be done and the number of misunderstandings that must be cleared.
6. Remoteness of user, difficulties in communication and contact.

ENVIRONMENT

Resources and Working Conditions

Principally, experience and skill of Project Leader in personal contact and in presenting the details of the design.

Management and coordination task.

Many conferences, personal contacts.

SYSTEM DESIGN TASK 7

INDOCTRINATE PRODUCTION PERSONNEL

INPUTS

Information

Knowledge of the computer programming tools, operating procedures, operating system, etc.

Program system design

Data base design

Formal NAVCOSSACT programming courses

Documents

Programmer manuals

System design documents

Program design documents

Data base documents

Computer manuals

Operating procedures

SUBTASKS

1. Arrange for programmer training in the use of the computer, the computing facility, the programming language, and the compiler, as needed.
2. Train the programmers, as above.
3. Indoctrinate the programming personnel in the use of the system data files.
4. Indoctrinate the programming (and other) personnel in the design of the system and the particular functions for which they are to be responsible.
5. Indoctrinate production personnel in design control and review procedures.
6. Indoctrinate contractor personnel in NAVCOSSACT environment.

ENVIRONMENT

Interactions and Dependencies

Computing facility, on setting up curricula and arranging for computer usage for training.

Equipment manufacturers, on course material.

Utility programmers, for material and lectures.

Other analysis activities for information on the system and programs.

Potential delays in obtaining details about computer, operating procedures, or programming language, or in producing teaching materials pertinent to these.

Dependency upon the timely delivery of system documentation.

DESCRIPTION

Train programmers in the use of the computer and production tools and indoctrinate them in the design and details of the programs to be produced.

OUTPUTS

Information

Understanding of the nature of the programming job and the tools to be used in producing the programs

Lectures and practicum

Documents

Training plans and schedules (curricula)

Manuals

COSTS

1. Experience level of production personnel.
2. Relative familiarity of the computer and programming language.
3. Availability of computer time for training.
4. Number of lectures and practicum scheduled.
5. Changes in system, equipment, or tools design.
6. Costing Formula:

Without handover (i.e., analysts also do the programming), training costs minimal, but hidden.

With handover (i.e., new Project members do the coding), estimate one month minimal normal training time per programmer.

On-the-job training costs not included.

ENVIRONMENT

Resources and Working Conditions

Coordination, teaching activities, staff work.

Experience level of lecturers on computer and programming tools should be high.

Good instructors are often in short supply.

Much training done on the job.

VI. PROGRAM IMPLEMENTATION

The process of producing programs from a set of program system specifications--that is, of implementing the program system design--is divided into three Phases: Program Development, the effort required to design programs that will perform a set of assigned operational functions; Program Coding, the translation of program specifications into program instructions; and Program Checkout, the running of the programs under test conditions to be sure that they are relatively error-free and will perform as specified.

Program Development repeats, on a smaller scale and a finer level of detail, much of the previous analysis and design process, but this process is now focused on the program system component in the information processing system. However, a thorough and accurate job of System Analysis and Design reduces the need to collect additional data in Program Development. To create the detailed designs for many programs during Program Development, the work is usually divided and so requires more people than for System Analysis and Design.

Program Coding, once detailed flow charts or other coding specifications are produced, is a straightforward task. However, even in the Coding Phase, many opportunities for improvement in the detailed design may be detected. In practice, design work does not cease with the coding specifications, but continues not only throughout code production but throughout checkout of the programs. Subject to many errors, coding needs thorough checking prior to program test to detect and remove illegal operators, misspelled and misplaced data references, and errors in logic.

Program Coding is usually done by dividing the programs into many small routines, each of which is coded, compiled, and checked out separately before being assembled into larger blocks and finally into a complete program. A great deal of the work associated with Program Checkout, then, is actually done during this gradual code checking process. No matter how thorough this code checking is, however, it does not entirely guarantee that the program will perform according to specifications either by itself or in combination with other programs. In fact, testing the performance of the individual programs, and of various program combinations, to insure the quantitative performance of the programs, is one of the lengthiest and most important aspects of Program Implementation.

A. OBJECTIVES

The mission of Program Implementation, in general, is to produce computer programs that perform, in a reliable and error-free manner, the data processing functions specified during the System Analysis and Design Phases. Program Development includes the detailed analysis and evaluation of the functions a program is to perform, the design of program logic and

a data structure that will perform those data processing functions efficiently, and the specification of that design in a form, such as detailed flow charts, that is readily amenable to coding.

Program Coding includes translation of program design specifications into error-free program code, and detecting and removing design deficiencies as the coding progresses.

Program Checkout includes the thorough evaluation of the code produced, to detect and remove all errors and to diagnose and remedy all operating deficiencies and failures to perform as specified.

B. TASKS

For Program Development, the tasks are:

1. Develop program system test plans
2. Design programs
3. Design program files
4. Establish system files

For Program Coding, the tasks are:

1. Code the programs
2. Desk check the programs

For Program Checkout, the tasks are:

1. Learn the test environment and test procedures
2. Compile and check the program code
3. Test individual programs
4. Test program subsystems
5. Test the program system

On small Projects, all tasks except Program Checkout, Tasks 4 and 5, may be performed by a single person assigned to each program. Project personnel who have designed and coded the individual programs usually have too much ego-involvement in the completed work to be sufficiently critical of its deficiencies in logic or design. To promote objectivity, the inspection, test, and certification of programs and associated documents should be performed by other individuals.

The division of large (10-30 thousand or more instructions) program systems into smaller parts for design and coding creates the need to integrate program parts produced by several people. This need may also arise when programs (designed to run as a system or under a common program monitor) are produced by several organizations. Since testing program systems and subassemblies involves the work of several people, a separate organizational entity for test work can be both efficient and objective. This test crew can design program subsystem and system tests, produce test materials, run tests, and evaluate the results.

In test design, all program paths should be exercised with representative values and some illegalities. This first level in the test hierarchy tests a small unit, e.g., 200 to 2000 machine language instructions that may be a routine or a program. At this level, as few as 25 instructions and as many as tens of thousands of instructions may be tested. For this test, the program is usually operated in a simulated environment and actual outputs are compared with expected results derived or calculated prior to the test. Test plans must specify the program environment, and test designs must specify the inputs and expected outputs based upon the program design. After each test run, the programmer analyzes the results and makes corrections to the code. All corrections must be verified by repeating the program test. The cycle of test, correct, and retest is usually repeated many times before a program operates satisfactorily.

The principal purpose of the program system test is to determine whether or not the computer program satisfies the requirements for operational information processing as described in the Preliminary Functional Description. In developing large systems, subsystem tests may be conducted in a similar manner prior to system test to check performance for only parts of the Preliminary Functional Description. Both simulated and "real" or actual data may be used. Simulated data are preferable for tests requiring close control of the test conditions; real data reflect the vagaries of actual operations and are preferable for testing system reliability and validity. System tests are not usually single, one-shot operations; normally, a battery of "system tests" is used to probe system operation under a variety of conditions. When several versions of the same basic program system exist for use in different operational environments, e.g., different equipment, data base, and functional requirements, many additional system or "adaptation" tests must be run as well as the basic system test. Further, system tests must be repeated each time a major change to the system is implemented or a new version or model of the system released. Hence, a set of well-designed and maintained documents to record system tests may be a permanent asset for continued program system development. The value of these records is further increased by detailing the procedures, techniques, and tools as well as feedback on them.

C. COMMUNICATION, COORDINATION, AND CONTROL

At the time of Preliminary Functional Description (PFD) concurrence, as the Coding Phase begins, the overall program system has been designed, the individual programs have been identified, and a set of documents specifies requirements, functions, and data structures in some detail. It would appear that the work can proceed on the design and coding of the specified programs independently. This is not so. Even with a detailed PFD, the opportunity usually exists for making further decisions about details in the designs of programs that interact with other programs or about the way in which the program and the data will be handled by the user. Therefore, the Project Leader and the Coding Supervisor must coordinate the detailed design work and further monitor and control it to insure that design compatibility among parts of the program system is maintained. To fulfill these responsibilities, the Project Leader must insure that all design decisions and changes are disseminated to all Project personnel.

Specifically, during Program Development, Coding, and Checkout, the Project Leader should coordinate:

- . Requirements and plans for use of "real" data for testing, or for joint testing with the user.
- . Input and output formats between interfacing programs.
- . Communication requirements of programs with the executive or control program.
- . Data designs with the central data base and central data file.
- . All data and program changes.
- . Data file requirements and work with the computing facility.
- . Portions of user documentation with all other programs.
- . Program changes and corrections with test personnel.
- . Requirements for interfaces with other systems--existing or in development, manual or ADP--such as data standards or timing requirements.

The Program Checkout Phase, like the Analysis and Design work, requires a high level of communication and coordination. In doing their jobs, checkout personnel must interact with:

- . The program analyst and designers to determine test requirements and to insure that system requirements are stated in a precise and testable way.
- . The personnel in the Coding Phase who are developing individual program requirements, designs and test plans.
- . The personnel of the EAM and computer operations, to set up procedures to make arrangements for running tests, and to reduce test data for evaluations.
- . Responsible programmers, to modify and correct programs during testing.
- . Users, to determine testing requirements, to coordinate the use of operational facilities for tests.

During Checkout, the Project personnel depend upon adequate and timely EAM services and computer support, and close cooperation between the machine room and test personnel is mandatory, because slight inefficiencies in procedures that increase turnaround time may seriously slow progress in the Project. Test personnel in the Project need test results as quickly as possible to initiate corrective action when program errors are detected. Interaction with the individual programmers during subsystem and system testing may become difficult and costly if the test facility is separated from the main programming activity by some distance. For example, this separation may slow the development of procedures for modifying and correcting programs quickly or for finding solutions to design problems. Also, arrangements should be made for a special supply of test tapes to record or store intermediate results and to accurately account for the results of several runs of the same programs and tests.

Interpretation of requirements, and arrangements for test data and live environment tests are only a few of the reasons for interaction with the user. To anticipate demonstration and turnover, there may be joint conduct of tests, use of user operators and facilities, and user aid in evaluating test results. Interaction with users and with equipment manufacturers may be needed for joint machine-program integration tests to test the appropriate functioning of both.

D. SUPERVISION

Again, during the Program Implementation activity, the supervisor's task is to monitor all activities and review products. Specific items to be evaluated in each phase are shown below.

Program Development Phase:

- . Program system design documentation for sufficiency of detail and quality of design.
- . Table designs and file structures for completeness, compatibility, sufficiency of detail and quality of design.
- . Individual program designs for completeness, compatibility, quality, and efficiency.
- . Storage allocation plans for feasibility, conflicts, potential timing problems, and efficient use of storage.
- . Program system test plans for adequacy and accuracy.

Program Coding Phase:

- . Program code for conformance to program designs and programming conventions, effective and efficient use of the programming language, adequate use of libraries, and adequacy of commentary.

Program Checkout Phase:

- . Test plans and test designs.
- . Test results.
- . Test documentation and reports.

The Project Leader should try to anticipate and avoid delays of various kinds during Program Implementation. Once under way, this activity is delayed by even a proposed change, e.g., by time spent to evaluate the implications of the change. Any change may result in a considerable amount of work being scrapped as detail designs for processing and data structures are redone and recoded. When the programs reach system testing, the Project Leader should try to defer changes to a later version of the system. To meet schedules, the work should not stop while decisions on changes are being made; therefore, quick decision-making will reduce the costs of changes that require rework.

During the Program Checkout period, many critical, unanticipated difficulties arise that require the supervisor to spend time either solving problems or expediting decision-making by other agencies.

E. COST FACTORS

Little experience data are available on the cost of designing and coding as distinct activities in producing a program. Some expert programmers can produce a detailed flow chart in one day, describe data tables in two days, and write the code in another two days for a relatively simple 1000-instruction program--a total of one man week. With design and code complete, the programmer must now check out the program; and this may require several times as much work. Generally, program design and coding progress by fits and starts as the programmer tries design approaches, sees some improvements, and then reworks his design and code.

The basic tasks such as designing programs, designing tables, and coding programs are most easily costed. The tasks that contribute less directly--planning tests, establishing the central data file, supervising, and documentation--are less readily costed. Some rough rules of thumb are:

Planning tests	One man month per 10,000 instructions *
Designing programs and data	One man month per 1,000-2,000 instructions-- more effort when the total system size exceeds 30,000 instructions
Establishing files	One man month per 10,000 items
Maintaining files	One man per month for each 40,000 items
Coding and desk checking	One man month per 5,000 instructions

For program testing, the size, complexity, and degree of innovation of the program system are primary determinants of cost. However, such factors as whether the test facility is conveniently located, the system specifications and test specifications precisely and unambiguously stated, and the test data voluminous and complex can seriously affect test costs and schedules. Tests are needed for each function, each subfunction, and each interaction as well as many joint effects. Since the number and size of tests are difficult to establish before the system has been thoroughly analyzed, prediction of costs is difficult. Computer runs are made to test programs that range between one hundred and several thousand instructions each, and under all sorts of conditions, so that the size and length of individual tests is more difficult to estimate than the number of tests. Some data that represent SDC experience with testing systems are shown below:

* Instructions refer to machine language instructions.

10 May 1965

116
(page 117 blank)

TM-2314/000/00

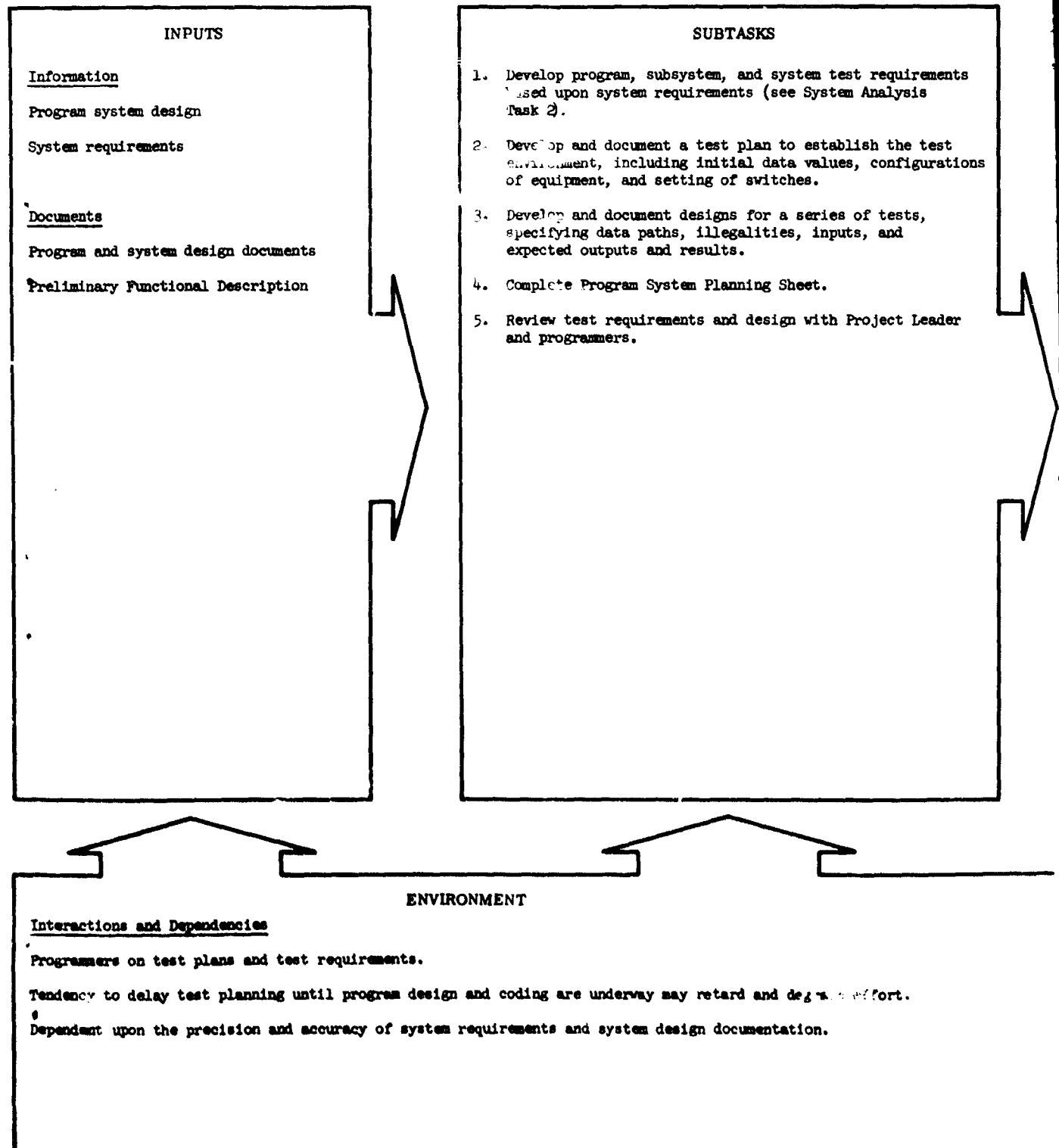
<u>Cost</u>	<u>Range</u>	<u>Average</u>
Computer hours per man month	1-6	4
Computer hours per 1,000 instructions		17
Computer hours per run		0.25
Runs per man month	3-11	5
Runs per 1,000 instructions	3-30	19

F. SCHEDULES

Since Checkout may take 50 percent or more of the total program development time, realistic and detailed scheduling is required. Short-term arrangements are equally as important to long-term schedules. For instance, in scheduling subsystem test runs, the test team should always try to have alternate run sequences laid out in case the planned sequence "hangs up" on a program fault early in the series of runs. If it can be avoided, computer runs should not be scheduled that depend upon the successful performance of any one program, but sets of independent programs and routines should be scheduled. Testing should not come to a halt while the results of a particular test are evaluated, nor should many runs of an apparently successfully operating program be accumulated before test results have been verified.

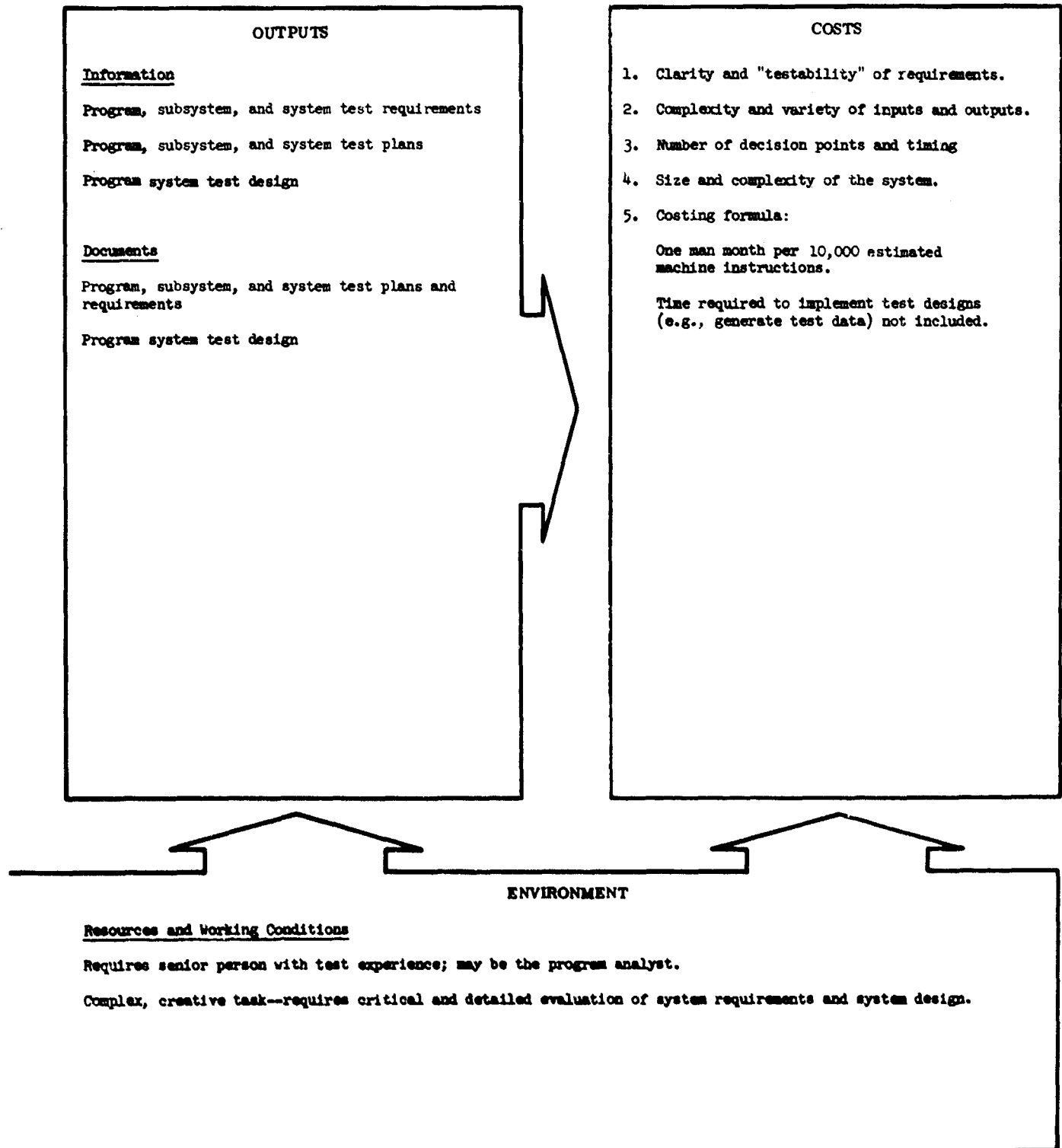
PROGRAM DEVELOPMENT TASK 1

DEVELOP PROGRAM SYSTEM TEST PLANS



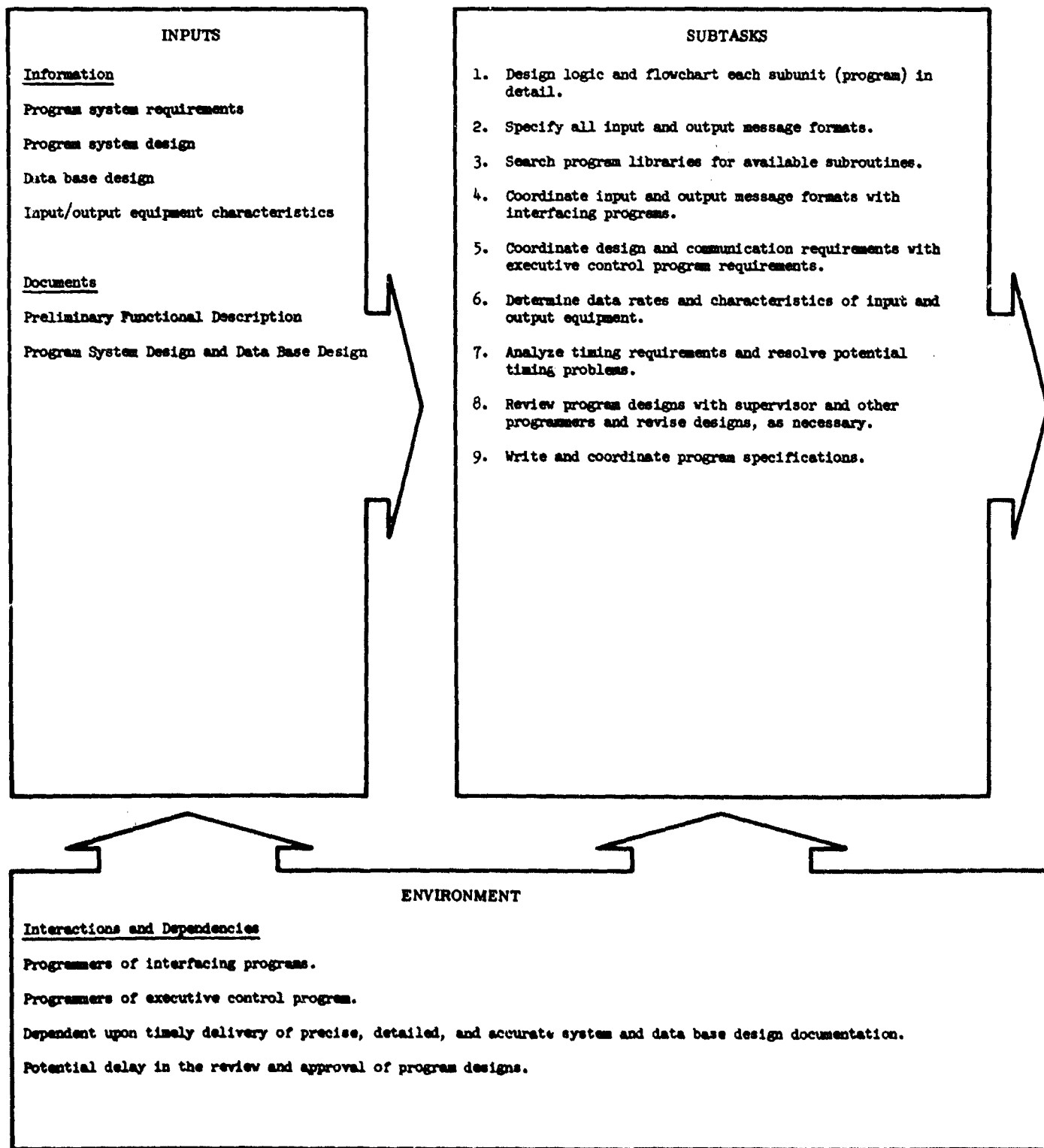
DESCRIPTION

Develop and document program system test requirements, test plans, and test designs to provide the specific plans and criteria for program and system evaluation.



PROGRAM DEVELOPMENT TASK 2

DESIGN PROGRAMS



DESCRIPTION

Design and document the individual programs and routines that have been specified (see System Design Task 2).

OUTPUTSInformation

Program designs

Input/output message formats

Program communication requirements

Timing analyses

Documents

Broad, detailed flow diagrams of each program

Program specifications

COSTS

1. Completeness, accuracy, and clarity of program system designs.
2. Designer's familiarity of area of application (i.e., experience with similar programs).
3. Degree of innovation required.
4. Size and complexity of programs and system.
5. Strictness of timing and storage limitations.
6. Number of logical blocks, dependencies, interfaces, program response requirements.
7. Number and range of inputs and outputs.
8. Mathematical vs. logical tasks.
9. Hardware constraints on program design.
10. Firmness of system design.
11. Experience level of programmers.
12. Costing formula:
 - One man month per 1000-2000 machine instructions.
 - One man month per 1000 instructions when the program is large (e.g., over 30,000 instructions).

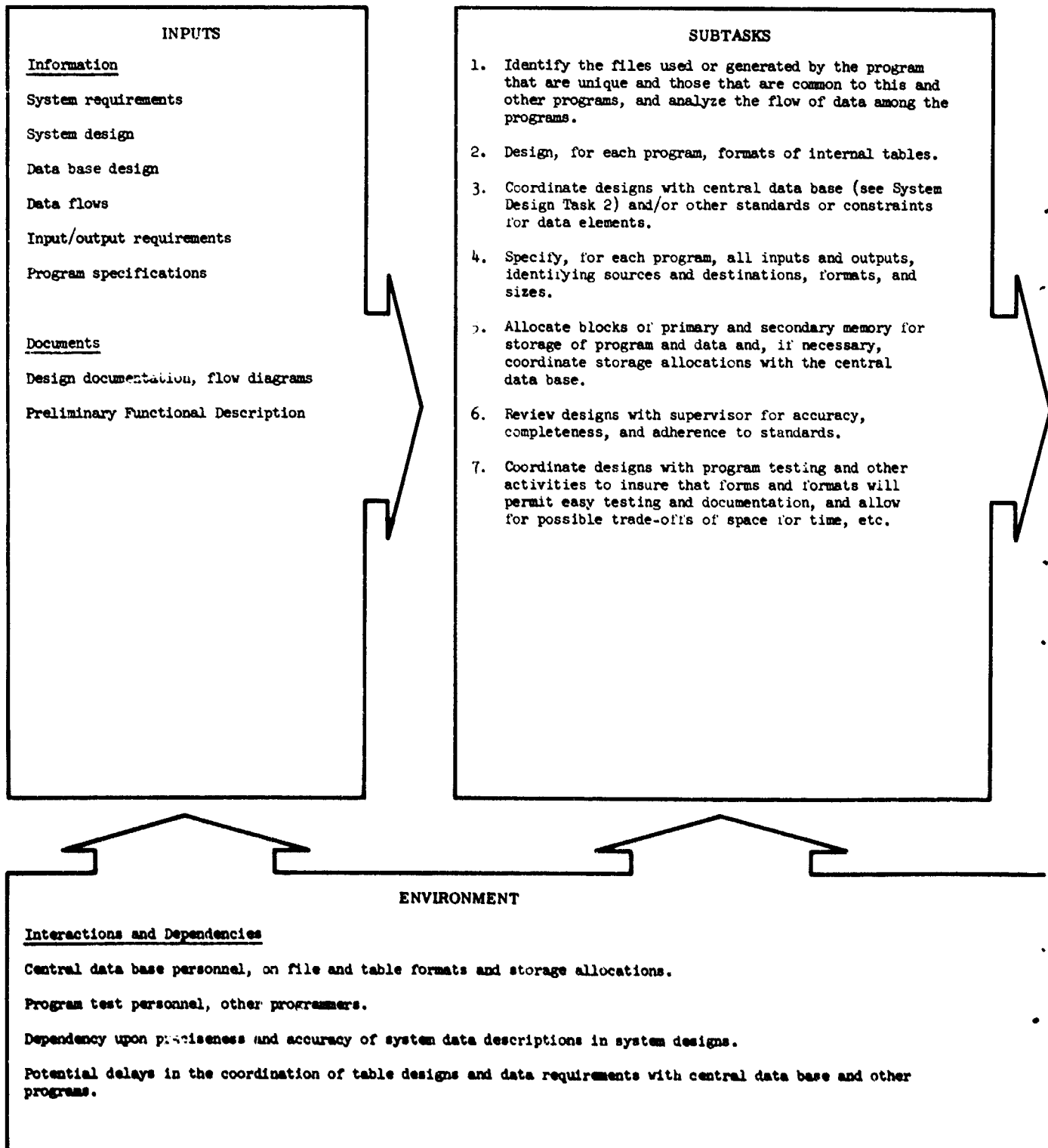
ENVIRONMENTResources and Working Conditions

Detailed, creative work. Few external contacts, little travel.

Requires close coordination with other programs.

PROGRAM DEVELOPMENT TASK 3

DESIGN PROGRAM FILES



DESCRIPTION

Develop and define the form of the data elements to be manipulated by each program, lay out storage allocations, and document program data structures.

OUTPUTSInformation

Formats of internal tables

Specifications for data storage allocations

Documents

Table layouts

Storage layouts

COSTS

1. Amount and variety of data handled by program.
2. Size and complexity of the system.
3. Amount of unique and independent data vs. amount of common, interdependent data.
4. Number of tables, items, files, classes.
5. State of organization, format, and validation of available data.
6. Rate of change of data.
7. Storage and/or timing constraints.
8. Security classification of the data.
9. Costing formula:
One man month per 10,000 items

ENVIRONMENTResources and Working Conditions

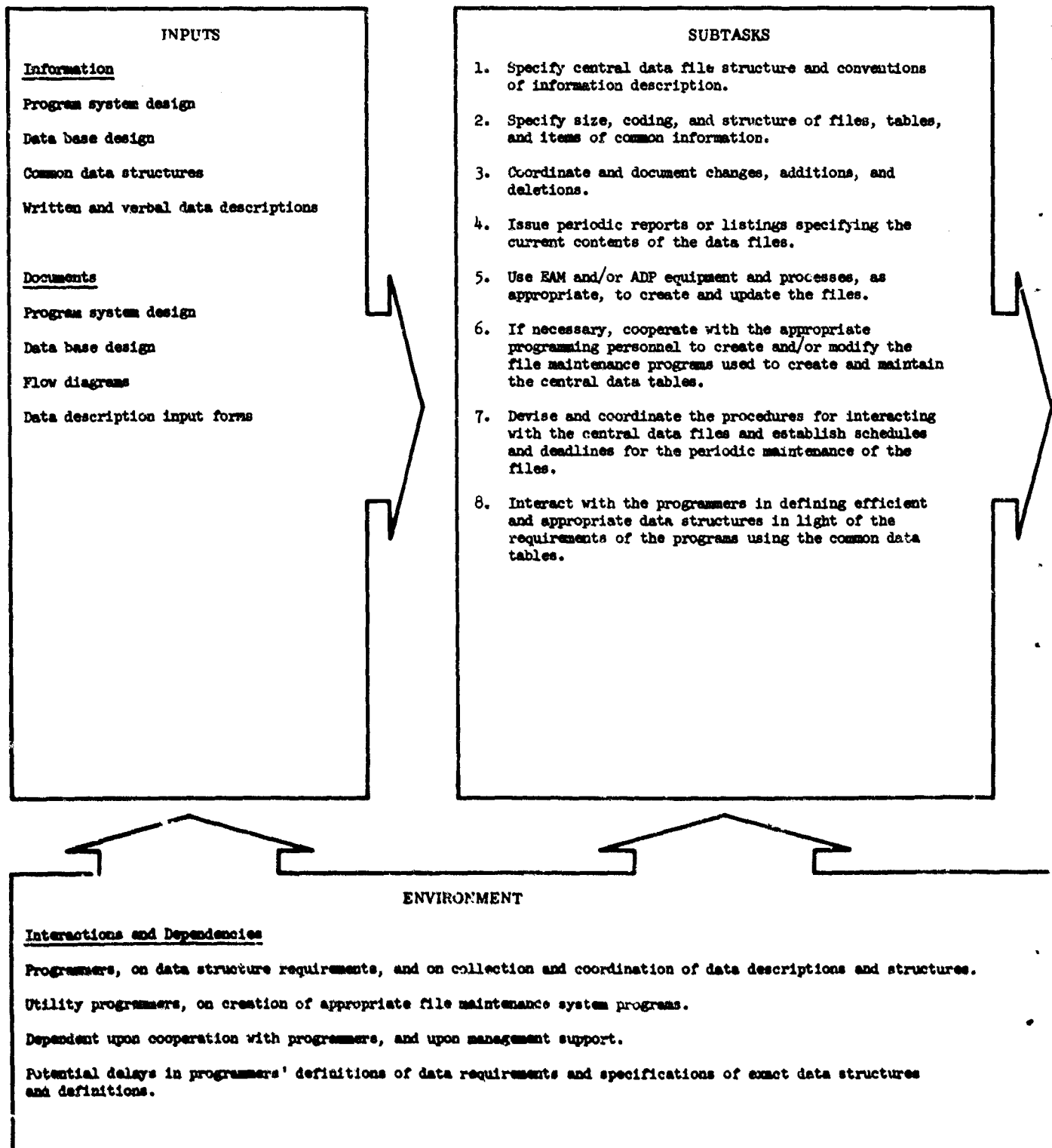
Knowledge of large data-base structures--their organization, maintenance, and associated machine storage characteristics.

Detailed, creative task.

Knowledge of standards that may exist for data elements.

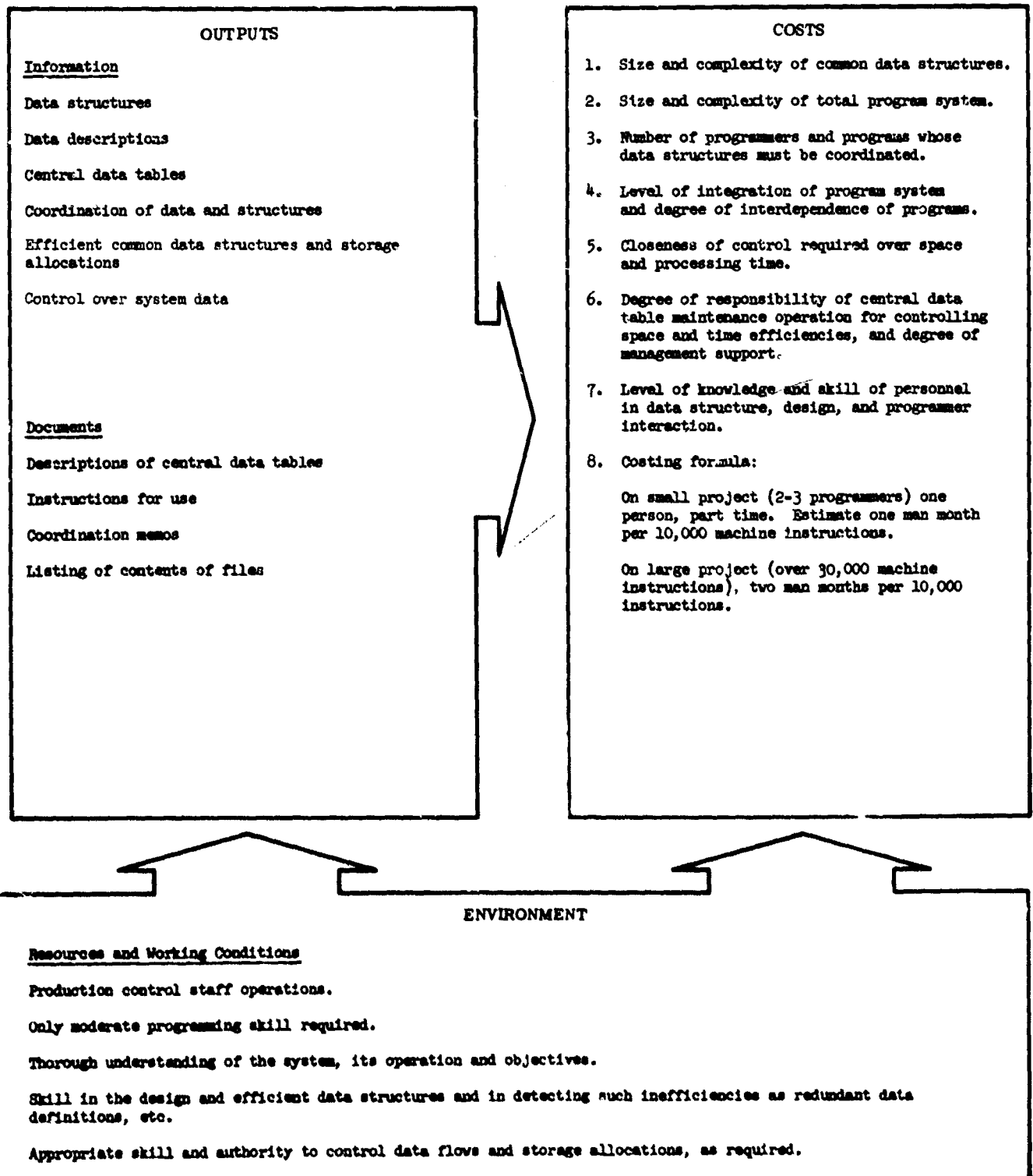
PROGRAM DEVELOPMENT TASK 4

ESTABLISH SYSTEM FILES



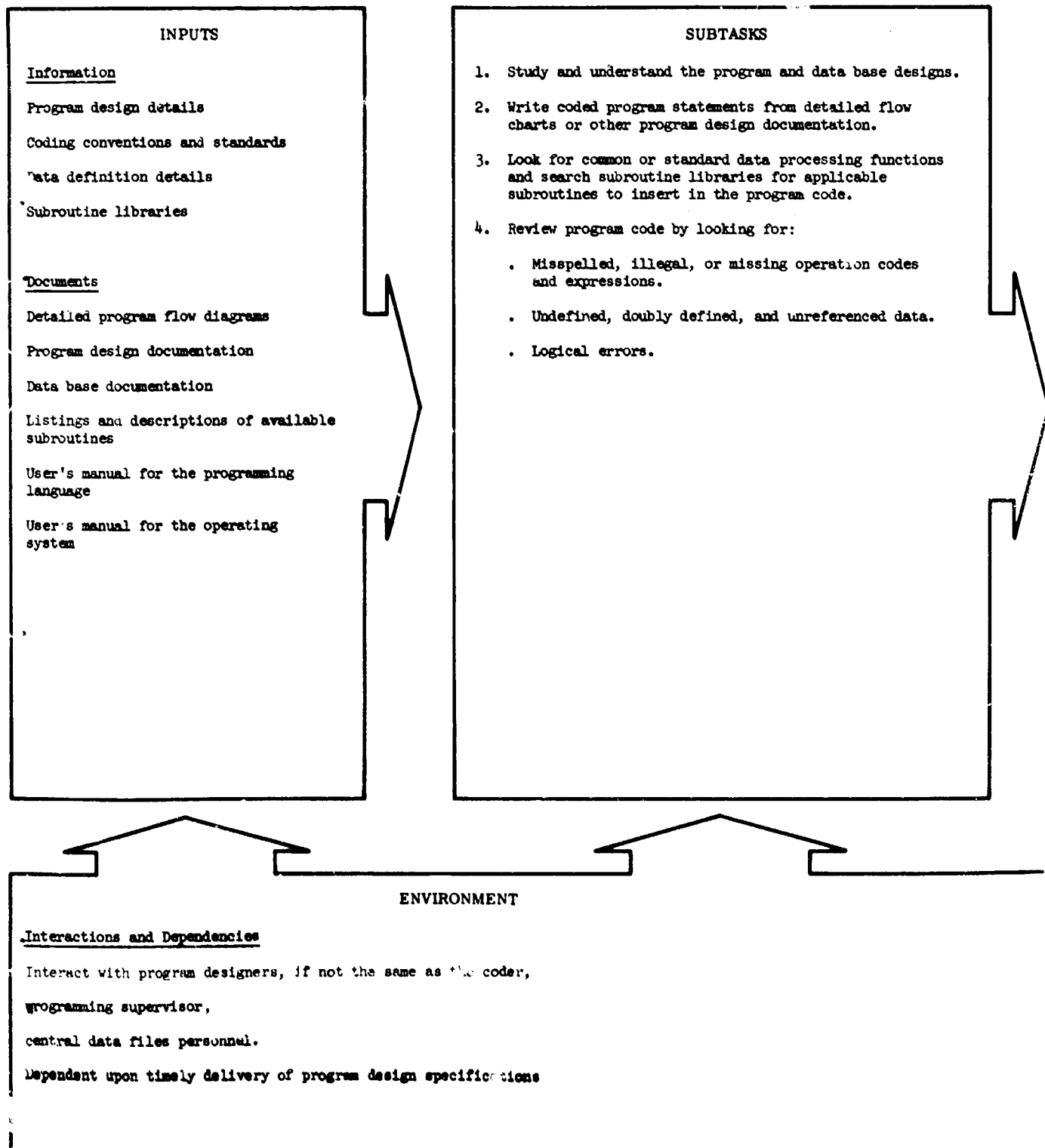
DESCRIPTION

Develop and maintain a central accounting system for information used by more than one program in the program system; document the central data file structure and the procedures for maintaining it; and periodically issue listings of the central file contents.



PROGRAM CODING TASK 1

CODE THE PROGRAMS



DESCRIPTION

Translate flow diagrams and other statements of program designs into coded instructions.

OUTPUTSInformation

Coded program statements

COSTS

1. Size and complexity of programs coded.
2. Proportion of new and reused code.
3. Familiarity of the programming language, computer, and operating system.
4. Procedure-oriented vs. machine-oriented language.
5. Experience level of coders.
6. Costing formula:

Gross estimate at one man month per 5,000 instructions.

Estimate that 2,000 procedure-oriented language statements are roughly equivalent to 10,000 machine-language instructions.

ENVIRONMENTResources and Working Conditions

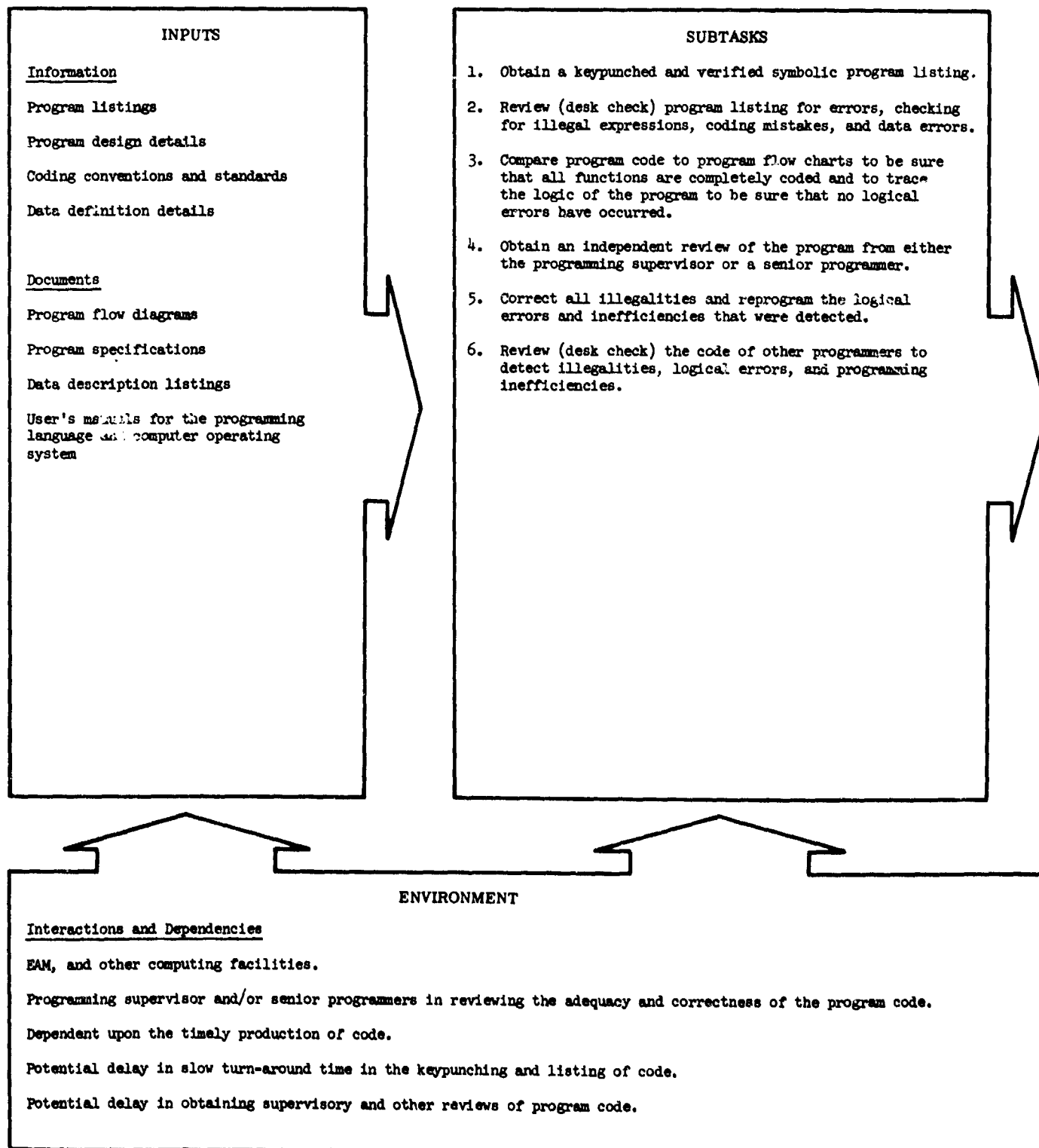
Less experience and skill required than for program design or analysis.

Usually performed by those who design the programs.

Relatively straightforward task, some room for creativity and cleverness.

PROGRAM CODING TASK 2

DESK CHECK THE PROGRAMS



10 May 1965

131
(pages 132 and 133 blank)

TM-2314/000/00

DESCRIPTION

Desk check program code by looking for illegal expressions, erroneous data reference, program logic errors, programming inefficiencies, and deviations from program specifications.

OUTPUTS

Information

Assurance of coding quality

Detection of program errors

Documents

Desk checked symbolic program deck and listing ready for compilation

COSTS

1. Size and complexity of programs checked.
2. Proportions of unfamiliar and difficult code.
3. Familiarity with the programming language, computer, and operating system.
4. Procedure- vs. machine-oriented language.
5. Experience level of checkers.
6. Excellence of code checked in terms of being well-organized, easily understood, free from errors, and simple.
7. Costing formula:

Cost included in the effort to produce the code (see Program Coding Task 1).

ENVIRONMENT

Resources and Working Conditions

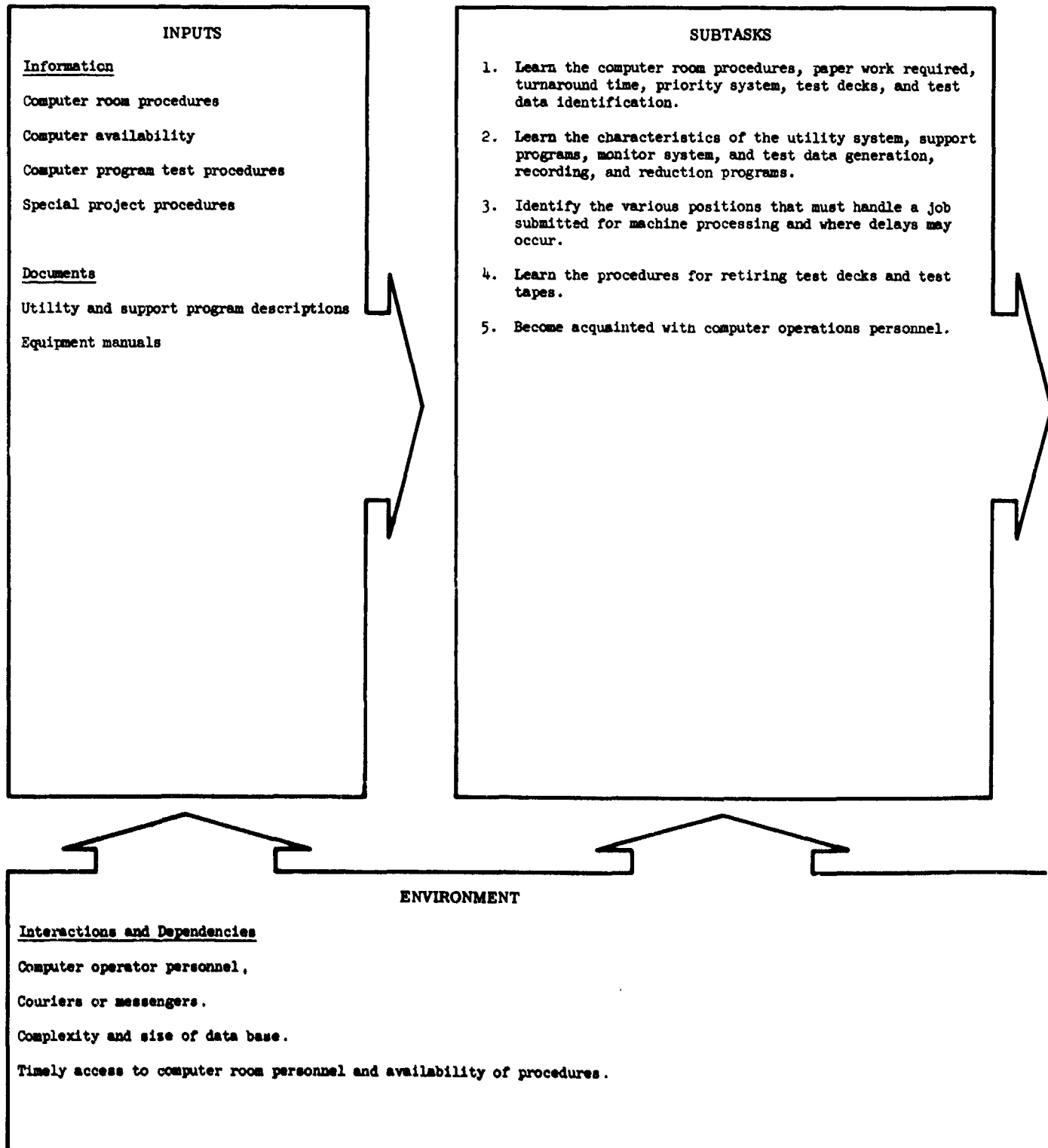
Usually performed by those who write the code and by the programming supervisor or leadman.

Requires considerable knowledge of programming techniques and of the coding conventions and standards established for the Project.

Tedious, mechanical task, calling for a sharp and critical eye in detecting errors and inefficiencies in program code.

PROGRAM CHECKOUT TASK 1

LEARN THE TEST ENVIRONMENT AND TEST PROCEDURES



DESCRIPTION

Using test requirements as a framework, learn the procedures for using the computer, the utility system, and other support systems.

OUTPUTSInformation

Ability to use procedures, compilers, computers, and programming support tools

COSTS

1. Adequacy of documentation of computer room procedures and test procedures.
2. Programmer experience with the computer facility.
3. Adequacy of documentation of equipment, and utility and support tools.
4. Stability (few changes) in the programming tools and the control programs, and rapid communication of any changes.
5. Costing Formula:
One man week per programmer.

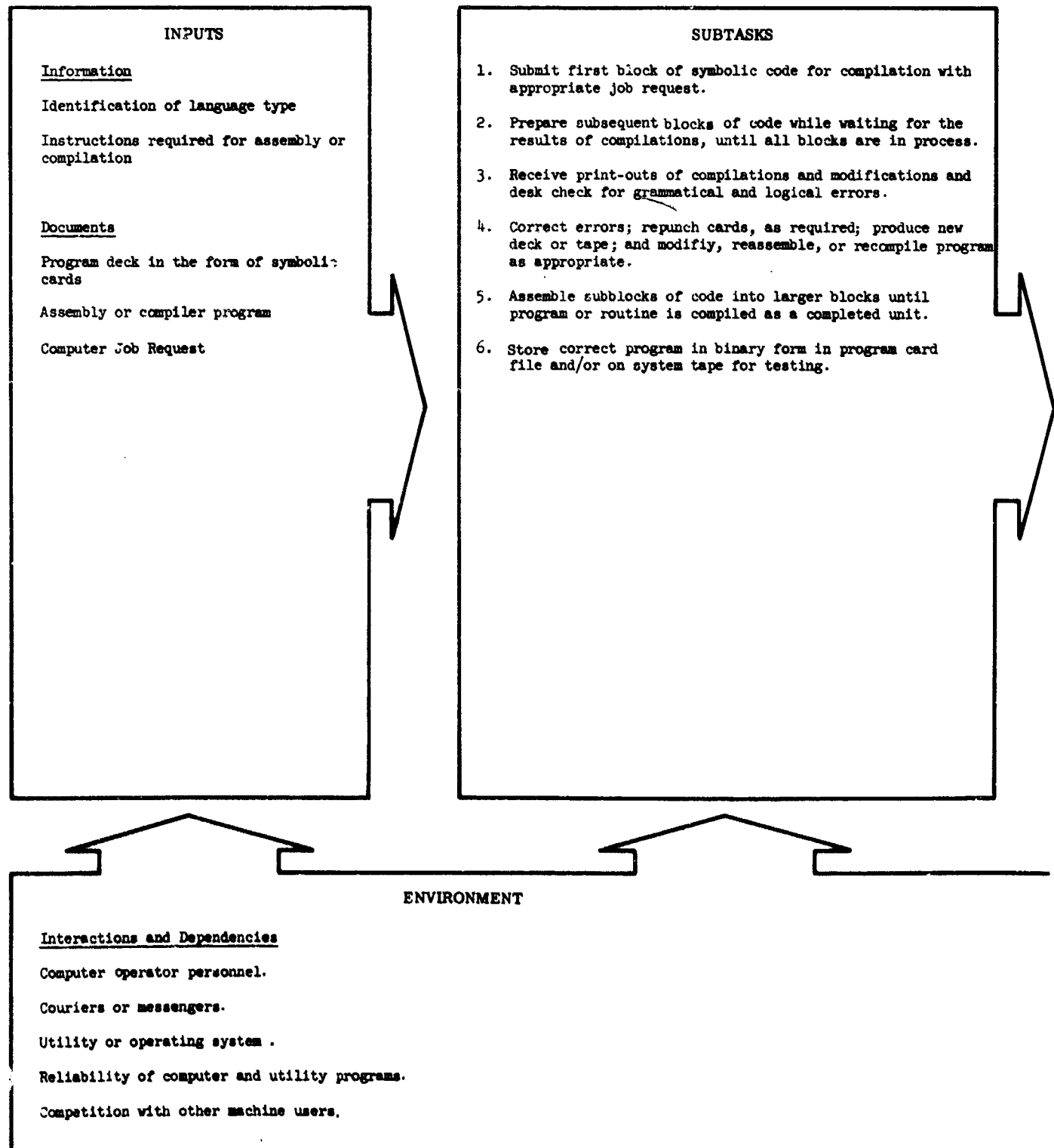
ENVIRONMENTResources and Working Conditions

Programmer test experience .

Mostly on-the-job training; little documentation of procedures; relatively straightforward task .

PROGRAM CHECKOUT TASK 2

COMPILE AND CHECK THE PROGRAM CODE



DESCRIPTION

As individual blocks of code are written in either symbolic assembly language or procedure-oriented language, assemble or compile each block into machine-readable (binary) form, check the listings for errors, correct the code and recompile, continuing this process until a satisfactorily compiled program or routine is obtained.

OUTPUTSDocuments

A program in binary, available on cards, tape, and print-out, ready for program testing

COSTS

1. Availability of reliable assembler or compiler and accurate documentation.
2. Availability of computer time and EAM support.
3. Programmer experience.
4. Costing formula:

Computer time rule of thumb for compiling with JOVIAL*
average 150 statements/minute
range 75-300 statements/minute**

*TM-WD-147/000/00 describes the operation and use of this JOVIAL compiler.

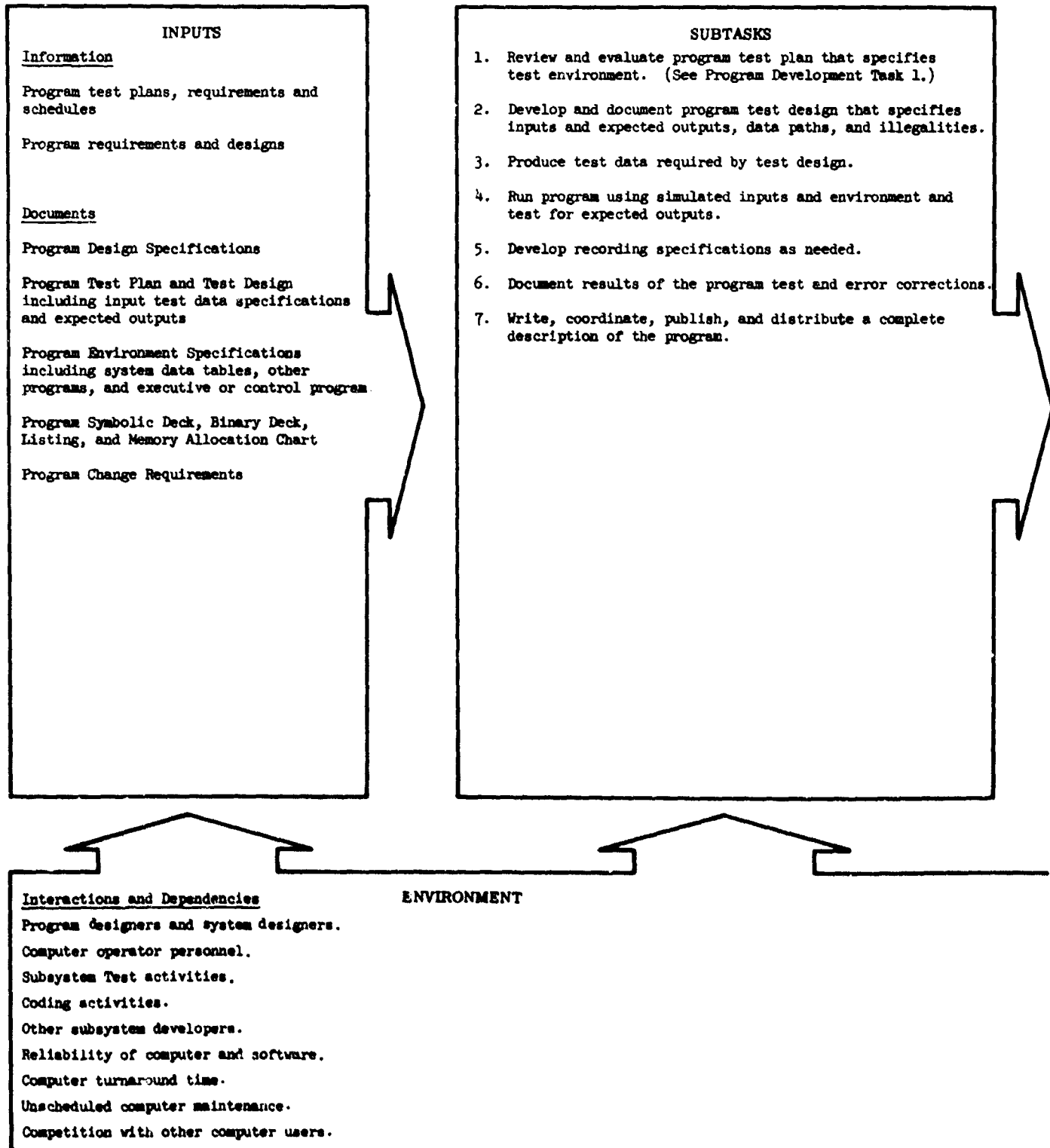
**Note that these statements generate both instructions and data.

ENVIRONMENTResources and Working Conditions

Adequate diagnostics and understanding of the compiler.
Computer time and operator skill.

PROGRAM CHECKOUT TASK 3

TEST INDIVIDUAL PROGRAMS



DESCRIPTION

Within the requirements set forth by general plans and requirements for program testing (see Program Development Task 1), plan, design, produce, and run performance tests of the individual programs to isolate and correct errors, rerunning the tests until all program requirements and design specifications are satisfactorily met.

OUTPUTSInformation

Program test designs, test data, and run results.

Recommendations for future program modifications

Recommendations for approval or deferral of requirements and design features

Assurance of program quality

Documents

Documentation of designs and tests

Reports of run results and actions taken

Memoranda on recommended changes

Program Descriptions

COSTS

1. Availability of computer: total computer time and turnaround time.
2. Availability of programming tools and descriptive documentation.
3. Adequacy of test planning and design.
4. Number of instructions in program to be tested.
5. Number of inputs and outputs.
6. Extent of innovation and complexity in the program design.
7. Number of program design changes to be implemented.
8. Extent of procedural documentation.
9. Costing formulas:

Anticipate one error per thirty instructions.

All testing requires between 40-50 percent of total development efforts.

Program test requires about 20 percent of the testing effort.

ENVIRONMENTResources and Working Conditions

Continual interaction with the computer and attendant delays require careful planning of programmers' time. Delays may be used for parallel activities such as documentation, replanning, and study.

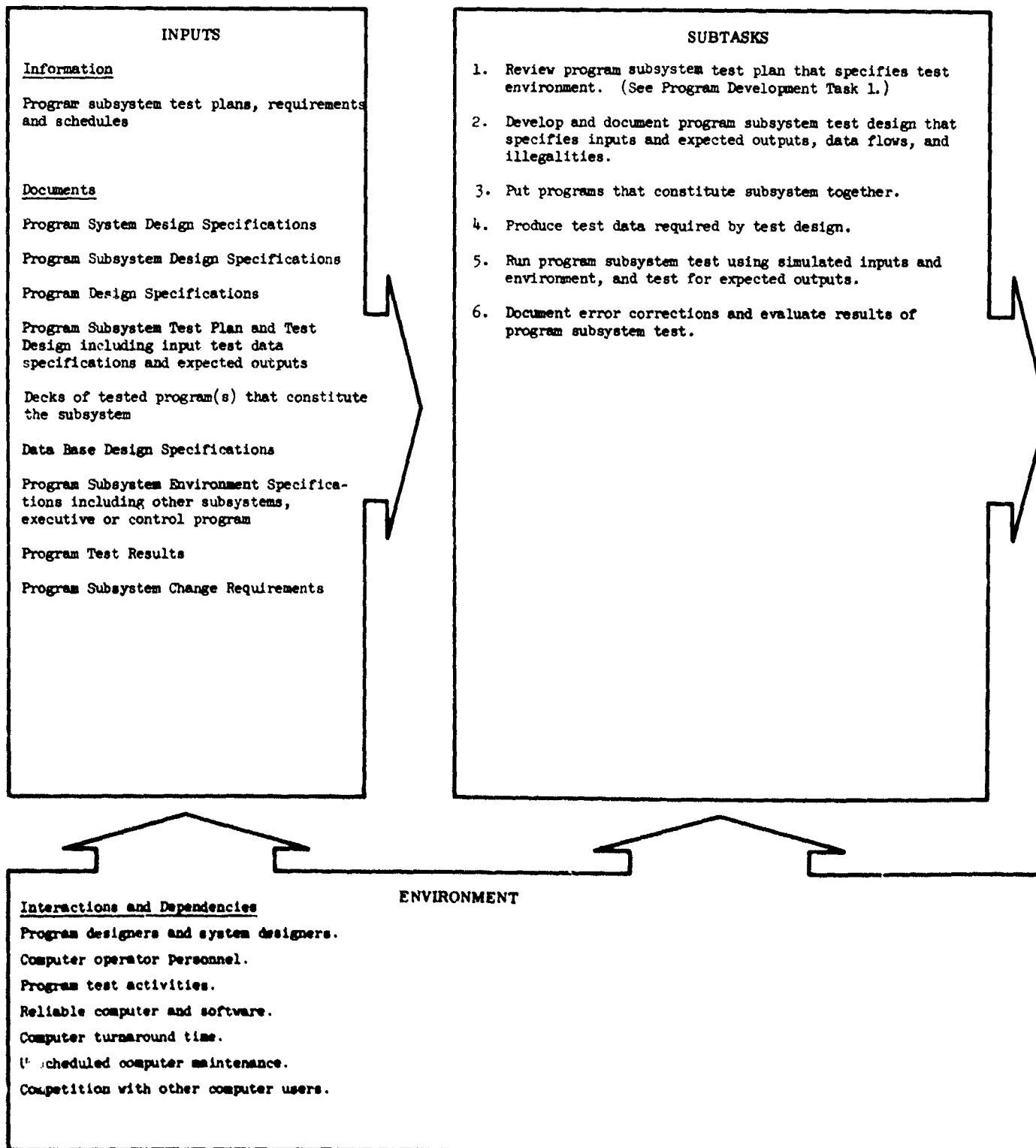
Programmer experience in testing, test planning, and designing.

Computer time.

Support programs (testing tools).

PROGRAM CHECKOUT TASK 4

TEST PROGRAM SUBSYSTEMS



DESCRIPTION

Within the context of the more general program system test plans, design, produce and run program subsystem tests for physical integration of functionally interdependent blocks of programs to isolate and correct failures of functional interactions and failures to meet program specifications.

OUTPUTSInformation

Subsystem test designs, test data, and run results

Recommendations for future subsystem modifications

Recommendations for cancellation, modification, or deferral of requirements and design features

Assurance of satisfactory program interactions in the subsystem

Documents

Documentation of designs and tests

Reports of run results and actions taken

Memoranda on recommended changes

COSTS

1. Availability of computer: total computer time and turnaround time.
2. Availability of programming tools and descriptive documentation.
3. Availability and accuracy of raw test data.
4. Adequacy of test planning and design.
5. Number of instructions in program to be tested.
6. Number of inputs and outputs.
7. Extent of innovation and complexity in the program design.
8. Number of program design changes to be implemented.
9. Extent of procedural documentation.
10. Number of displays in subsystems.
11. Costing formula:

Varies between zero and thirty percent of total testing effort depending on number of subsystems.

All testing requires between 40-50% of total development effort.

ENVIRONMENTResources and Working Conditions

"Subsystem," "task," "standing," or "assembly" testing usually involves considerable pressure, relatively long computer runs, late hours, and many conferences on catastrophic program failures and emergency changes to plans and schedules brought about by unexpected failures, delays, and changes.

Programmer experience in testing, test planning and designing.

Computer time.

Support programs (testing tools).

PROGRAM CHECKOUT TASK 5

TEST THE PROGRAM SYSTEM

INPUTS

Information

Program system test plans, designs, data, and schedules

Documents

Program System Design Specifications

Program System Test Plan and Test Design including input test data specifications and expected outputs

Program System Environment Specifications including system data tables, executive or control program

Program System Symbolic Deck, Binary Deck, Listing, and Memory Allocation Chart

Program Subsystem Test Results

Peripheral Equipment Operating Descriptions

SUBTASKS

1. Review program system test plan and design--revise as needed. (See Program Development Task 1.)
2. Integrate program subsystems for program system test.
3. Run system test with simulated and/or real inputs and environment and test for expected outputs.
4. Document the results of the system test and error corrections.
5. Rerun corrected program system and test for expected outputs.

ENVIRONMENT

Interactions and Dependencies

Program designers and system designers.

Computer operator personnel.

Subsystem test activities.

Turnover and Demonstration activities.

Other system developers.

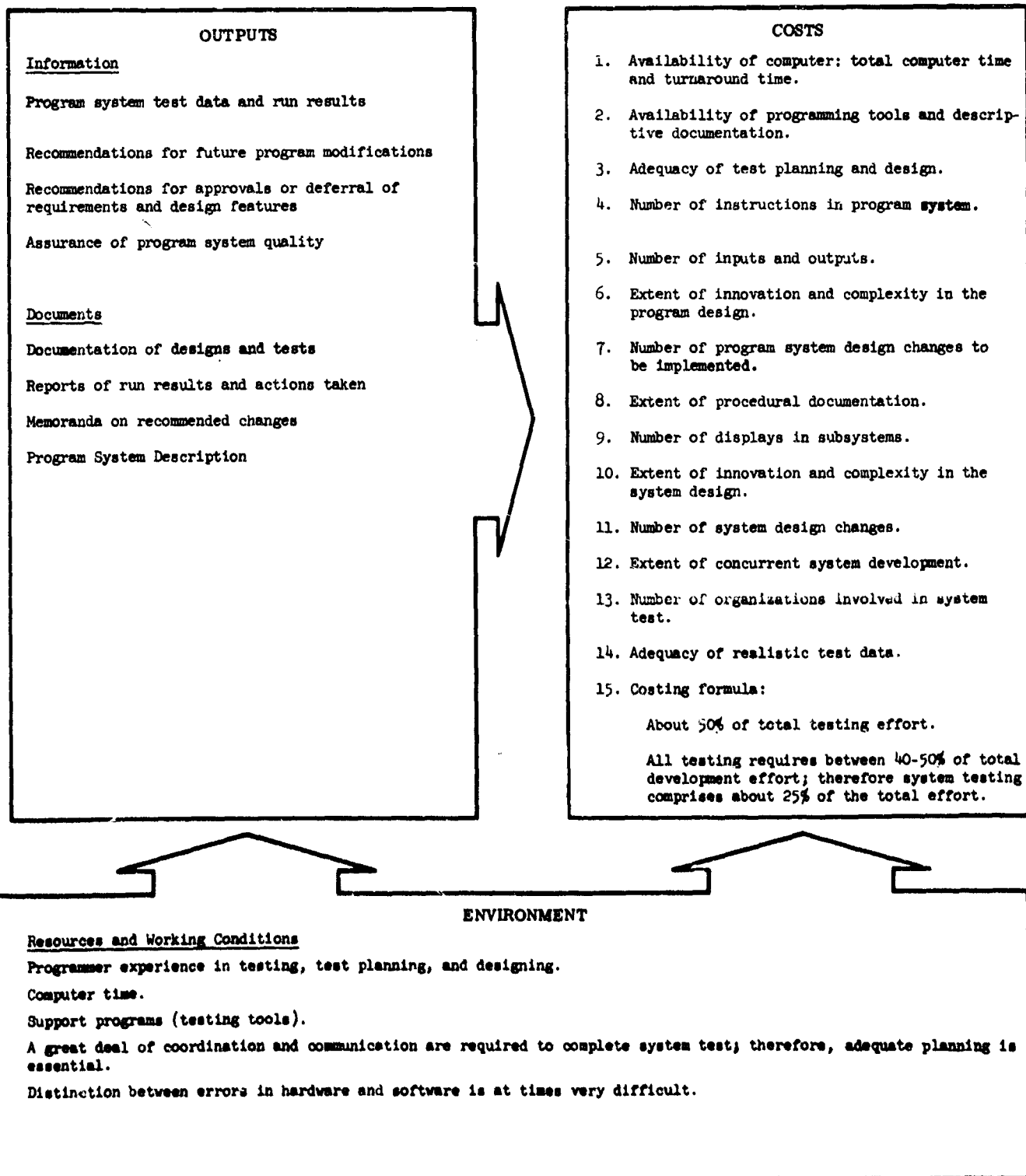
User personnel.

Peripheral equipment.

Dependency on reliable computer, software, and peripheral equipment.

DESCRIPTION

Within plans for the overall quality assurance of the program system, design, produce, and run tests (usually consisting of a series, or increasing size and complexity) of the total program system to isolate and correct system malfunctions.



VII. SUPPORT AND TURNOVER

The last set of Phases in the program system development process are those associated with preparing for and making the delivery of the system to the user. This includes preparing operating and maintenance manuals, helping the user prepare to operate the system through training and assistance, conducting the demonstration trials, and working closely with the user during the period of shakedown to detect and correct any remaining program errors.

The tasks associated with support of the user, although sometimes viewed as an imposition by those interested solely in the design and coding of computer programs, are an extremely important activity from the managerial point of view. The perceived success of the Project may rest upon how well the user understands his new system. Therefore, it is "good business" to provide the user with adequate information in the form of documentation, training (including practice in the use of the system), briefings, and other customer-relation services. Such services may include advice and consultation on procedures, organization, and operations.

A. OBJECTIVES

The mission of the Support and Turnover Activity is to prepare the user to receive and operate the system, and to change his mode of operation as needed to fit the new system.

In the User Documentation Phase, the objective is to develop and produce a set of instructional manuals that will best help the user to understand, operate, and maintain the program system.

In the User Training and Assistance Phase, the objective is to ease the user's transition into the new mode of operation. This is done by (1) building or redefining his data stores and associated data-handling procedures, (2) redefining work organization and procedures, (3) providing assistance to create understanding of the system's capabilities and (4) promoting acceptance of the new mode of operation.

In Turnover, the aim is to help the user demonstrate, to his own satisfaction, that the system will operate as specified, and to support the user with advice, guidance, and immediate trouble-shooting during the initial period of system operation.

B. TASKS

For User Documentation, the tasks are:

1. Verify the completeness and accuracy of the program system specifications.

2. Outline user documentation.
3. Produce user documentation.
4. Obtain concurrence upon user documentation.

For User Training and Assistance, the tasks are:

1. Advise user on data collection and conversion activities.
2. Develop a user training plan.
3. Conduct training program for user's staff, operator and maintenance personnel.

For Turnover, the tasks are:

1. Develop the Turnover Plan.
2. Conduct demonstrations.
3. Assist in the operational shakedown of the system.

Although the Project Leader must always plan on Project participation in the Support and Turnover activity, the amount of work required will depend greatly upon (1) the past experience of the user in applying ADP to his operations, and (2) the extent and complexity of the interaction between human operators and the ADP system. If the computer is used completely off-line, and the user is familiar with ADP operations, Project personnel will have to supply little support. However, if operation is completely on-line and dynamic, and involves a completely naive user, much training and support may be required.

User documentation is a particularly important deliverable product, since the operation of the system will be very difficult to learn without adequate descriptive material. To the extent possible, user documentation should be adapted to the needs of each user. For example, a standard set of documents for an off-line system may include using guides and procedures for operators but an on-line, real-time system requires much more documentation of this type such as instructions for each different operating position (sometimes called Positional Handbooks).

At NAVCOSSACT, the user is responsible for a very important task, i.e., the collection and conversion of data files to be used by the system, including the formulation of data collection and distribution procedures for use during system operation. If the user is unfamiliar with ADP, however, he may need considerable guidance in setting up files and establishing and implementing procedures. Also, the inexperienced user may be unaware of the stringent demands for precision, accuracy, and completeness in data to be used in computer systems. In such cases, the first compilation of data will probably not work without a great deal of editing and correction.

A breakdown of the data collection and conversion tasks expected of the user is:

1. Determine the types, forms, and structures for data needed as inputs for the ADP.
2. Identify the sources of the specified data.
3. Determine the volume and frequency of arrival of various data elements.
4. Determine or establish appropriate data collection procedures and forms, for both the initial and ongoing collection of data.
5. Collect data.
6. Verify data accuracy and completeness.
7. Manually process data by entering them on data forms (properly placed, scaled and arranged); by keypunching; by inspecting listings; and by manually controlling card files and listings.
8. Convert and store data in the specified storage medium, using the computer and computer programs provided for this purpose.
9. List stored data and inspect the results to verify the correctness of the data as stored.
10. Turn over the converted data base files to operational personnel.
11. Prepare, if required, simulated data base files and other test data.
12. Cooperate in testing by inspecting the results of tests to detect processing and data errors.
13. Specify requirements for data base load and maintenance programs.

Although the Turnover tasks stress the intangible activities such as coordination, consultation, training, indoctrination, and communication, the Turnover Plan may be accompanied by other tangible products, e.g., briefing documents, conference reports, test materials, and training materials. At NAVCOSSACT, the Turnover Plan must be completed not later than one month before the completion of Program Checkout.

On small Projects, the Project Leader or another Project member performs the Turnover tasks, particularly coordination, monitoring, and briefing. Checkout personnel do the planning, design, production, and actual conduct of demonstration tests. Usually analysts and programmers in the Project train and indoctrinate user personnel. On a large project, however, and especially one with many ADP centers requiring multiple installation of the program system, special Turnover or installation teams will be needed to coordinate with the user, to adapt the program system (e.g., introducing a new data base for each location), to indoctrinate and train the user, and to develop and conduct the demonstration tests.

11 May 1965

C. COMMUNICATION, COORDINATION, AND CONTROL

Much of the support activity insures that the plans made by the user and the developer are integrated, and that the schedules are met. During the User Documentation Phase, once documentation plans are firm, the main coordination task concerns changes to the system, i.e., to assure that the statements in the documents are current and accurate.

For the data base work, Project personnel must insure design and schedule compatibility for easy combination of the program system and the data base. Any decisions that extend or alter the basic data base design should be promptly transmitted to the user so that he can reflect them as necessary in the data base. Also, the responsibilities and work load of the data base team--e.g., ease of data collection and conversion and availability of time to respond to changes--must be considered in making any changes. Missing data may seriously affect the data base design and impose unexpected requirements on the programs. Again, equipment design that may differ from that expected constrains data base manipulation; e.g., precision, speed, and reliability of data handling may be reduced from that originally specified. Data may have to be estimated or interpolated to obtain the desired precision, or design requirements may have to be changed. These situations point up the need for close coordination between the data base and program design activities.

Turnover is a crucial phase because the user may readily find fault with the system unless he thoroughly understands it and accepts its limitations as well as its advantages. This is a period of transition for the user. From the familiar--the safe, secure, and stable, with introduction of a new program system, he enters into the unknown and uncertain. He may be anxious and ready to find fault with the system. To remove this uncertainty and insecurity, the Turnover team must try to continue to communicate freely and easily with him.

The Gantt Chart in Section 3 shows that the time interval for the Turnover Phase overlaps, to a considerable extent, the intervals for the entire Checkout Phase and even the final stages of the Coding Phase. If Turnover is handled by a separate team, their early work includes intense study of the details in both the design and planned operation of the system. The team uses this knowledge to plan the Demonstration test and to indoctrinate the user.

In addition, Turnover personnel must adopt and maintain a system orientation, i.e., they must understand and treat the system as a unit, and they must try to pass this orientation on to the user. While many of the minor operations of the system may go wrong or be found in error, these discrepancies must be viewed and weighed within the framework of the total system.

Although it is desirable to forbid or defer design changes at this stage, such changes may be needed and implemented, and consequently may disrupt Turnover. As an alternative to implementing such changes, a suggested technique is to set up the "two-tape" system for the user (that he will undoubtedly need after he assumes responsibility for program system maintenance). In such a system the basic tape is not changed, but all changes, e.g., changes in requirements or corrections of discrepancies, are introduced in a follow-on version of the "system tape." Then they will be ready for system testing as soon as the prior version is released for provisional use. This two-tape technique is also good for system maintenance and could be handed over to the customer with the system. Without the protection afforded by this technique, the inexperienced user may introduce changes into the operational version--changes that have not been thoroughly tested and so may foul up his system, possibly at a crucial time in operations.

D. SUPERVISION

The products that must be reviewed by the supervisor during the Support and Turnover Activity are:

- . Revisions of program specifications.
- . Plan for, outline of, and drafts of user documentation.
- . Training plans.
- . Turnover plans.
- . System changes for both programs and data base.

Support and Turnover activities demand frequent contact with the customer, e.g., in the form of briefings and conferences. To insure good customer relations, as well as good performance in other tasks in this phase, the Project Leader may personally make these customer contacts.

E. COST FACTORS

Although the costs of the mechanics, i.e., writing, editing, and reproducing a document, are fairly well known, the separation of these costs from the cost of doing the more creative part of documentation is not. That is, the cost rules, unrealistically, assume that this mechanical production of the document is done after or independently of the thinking and information-gathering work needed to generate the information going into the document. Since this is not usually the case, the Project Leader must estimate and account for these costs as well. If these costs were separable, the costs of the creative work would likely exceed those of production work.

The cost of Data Collection and Conversion itself depends largely upon the amount, i.e., the total number of items, complexity, number of different item types, the rate of change of data, and the number and accessibility of data sources. Another factor that adds to the cost is the handling of classified data. In a NAVCOSSACT Project, the cost of advising and monitoring the data base activity will vary between one and sixteen man weeks, depending upon the size and length of the data collection activity. Once communication channels and procedures have been established, junior personnel can usually be assigned.

"Assisting the user in preparing (and conducting) a demonstration test" can be a costly operation. Two factors have a heavy influence on costs; (1) the user's inexperience with ADP, and (2) the dispersion and variation of program system, e.g., in case it must be installed in several locations with differing equipment and program configurations, environmental conditions, and system interfaces. A successful demonstration depends upon thorough study of environmental differences with emphasis upon current, reliable, and detailed information. "Dry runs" prior to demonstrations help to expose and eliminate many problems that might occur during the Demonstration and also to expedite the education of users.

Costing such intangibles as coordination, consultation, and indoctrination is difficult. Experience shows that any funds allotted to this activity are profitably spent. Any lack of funds usually reduces the work on the "soft" tasks, the educational and consultation efforts, while funds for the "hard" tasks, actual Demonstration test development and running, remain adequate. Such poorly balanced funding for Turnover results in a more difficult self-indoctrination and shakedown period for the user. Therefore, rather than risk customer dissatisfaction, it is best to budget time and resources for coordination, consultation, and indoctrination.

F. SCHEDULES

In scheduling documentation, try to begin early enough to influence the final production of program specifications and pace it to match the development of the programs. During system testing, the availability of documents, even if only in rough draft form, is crucial. Polished, formal documents should be delivered at the time of system demonstration and sign-off as part of the delivered system. Of course, the documents will have been reviewed and approved before that date. On the other hand, during system shakedown, minor discrepancies and inefficiencies may be uncovered in both the programs and documents. Therefore, the format of the documents should permit changes to be made to the delivered final versions of the documents so that the system can begin operations with a "clean deck," in both documents and programs.

10 May 1965

151
(Pages 152 and 153 blank)

TM-2314/000/00

As implied earlier, close coordination of data base design and maintenance is essential for compatible schedules. For example, the Data Collection and Conversion task depends upon the data base design work. Also, if "real" data from the data base are to be used in program testing, delays in data gathering or difficulty in constructing the data files can slow testing. Realistic schedules are important because delays during testing can endanger the entire development cycle and little opportunity exists to add personnel, since they would require education on the entire Project work to date.

Turnover and Shakedown schedules may be endangered by all sorts of delays, particularly in tasks involving interaction with the user. Coordination itself takes time. For example, although the development of a training curriculum is not usually a problem, securing agreement with the user for a training schedule may be. The Project Leader should allow time for the user staff to review and modify proposed schedules and for coordination of all plans that involve user participation.

Best Available Copy

USER DOCUMENTATION TASK 1

VERIFY THE COMPLETENESS AND ACCURACY OF THE PROGRAM SYSTEM SPECIFICATIONS

INPUTS

Information

Details of the program and program system design specifications and requirements

Data base designs

Changes to the system

Documents

Preliminary Functional Description

Program system design specifications

Program design specifications

Data base design specifications

Flow charts

Design change documentation

SUBTASKS

1. Collect copies of all previous system documentation such as:
 - . Preliminary Functional Description
 - . Program system design specifications
 - . Program design specifications
 - . Program flow charts
 - . Data base design specifications
 - . Changes to the system
2. Review documentation to insure its accuracy and completeness and to evaluate its adequacy for use in the production of user documentation.
3. Check with system and program design personnel to be sure that documents are correct, complete, and up to date.
4. List missing information and interview programming and design personnel to obtain it.
5. Collate the documents and information collected and produce drafts of specifications and flow charts that reflect current information.
6. Coordinate the revised specifications with the appropriate programming personnel to verify the accuracy of the information.
7. Incorporate corrections and clarifications into the draft specifications.

ENVIRONMENT

Interactions and Dependencies

Programmers and analysts in collecting program specification information and verifying its accuracy and completeness.

Dependent upon the timely completion and quality of program documentation.

Potential delay in getting current information from individual programmers.

Potential delay in the review and coordination of the revised specifications.

Potential delay in resolving last-minute program changes.

DESCRIPTION

Verify the accuracy and completeness of program system documentation, and produce drafts of current, up-to-date specifications.

OUTPUTSInformation

Verification of the accuracy and completeness of program system documentation produced as working drafts

Documents

Drafts of updated specifications

COSTS

1. Amount and complexity of the previous system documentation.
2. Accuracy and completeness of the documentation.
3. Number of changes to be incorporated.
4. Number of reviews of coordination drafts.
5. Costing formula:

Technical review: 20 pages per man day
Revise: 10 pages per man day
Collect Information: 2 days per document
plus 2 hours per interview
Type: 20 pages per man day

At least two drafts of the revised specifications should be expected.

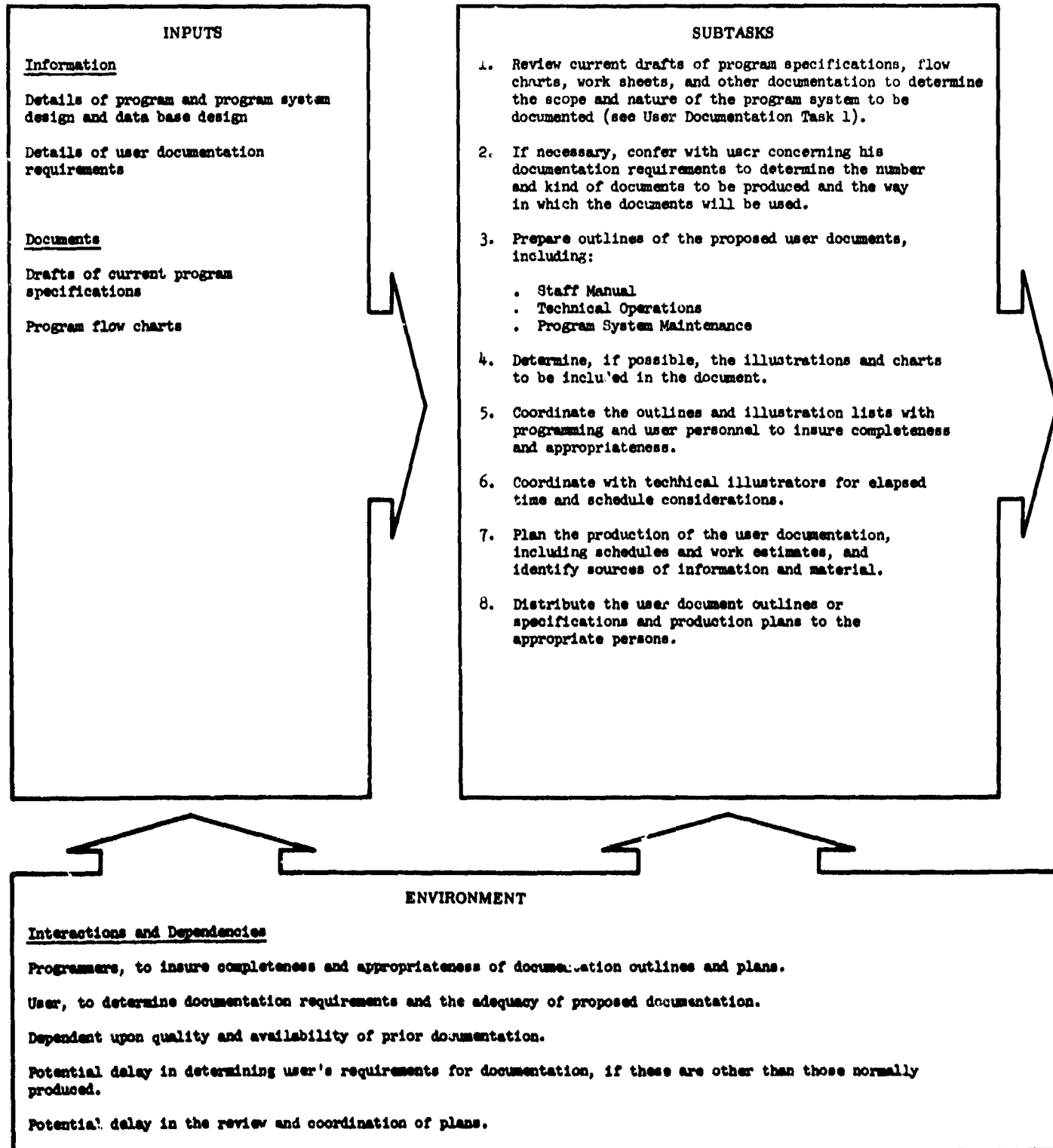
ENVIRONMENTResources and Working Conditions

Technical editing and writing personnel.

Programming specifications often tend to be vague and incomplete, and subject to many undocumented changes.

USER DOCUMENTATION TASK 2

OUTLINE USER DOCUMENTATION



DESCRIPTION

Determine user documentation requirements and the scope of the system, outline appropriate user documentation, plan its production, and issue user documentation specifications and production plans.

OUTPUTSInformation

Determination of user documentation requirements
Specification of user documentation
Coordination of specifications and plans
Identification of contributors

Documents

Outlines or specifications for user documentation
Schedules for production of user documentation

COSTS

1. Size and complexity of the system.
2. Quality of prior documentation.
3. Experience level of user in ADP applications and required user's manuals.
4. Number of coordination and review points for plans and outlines.
5. Costing formula:

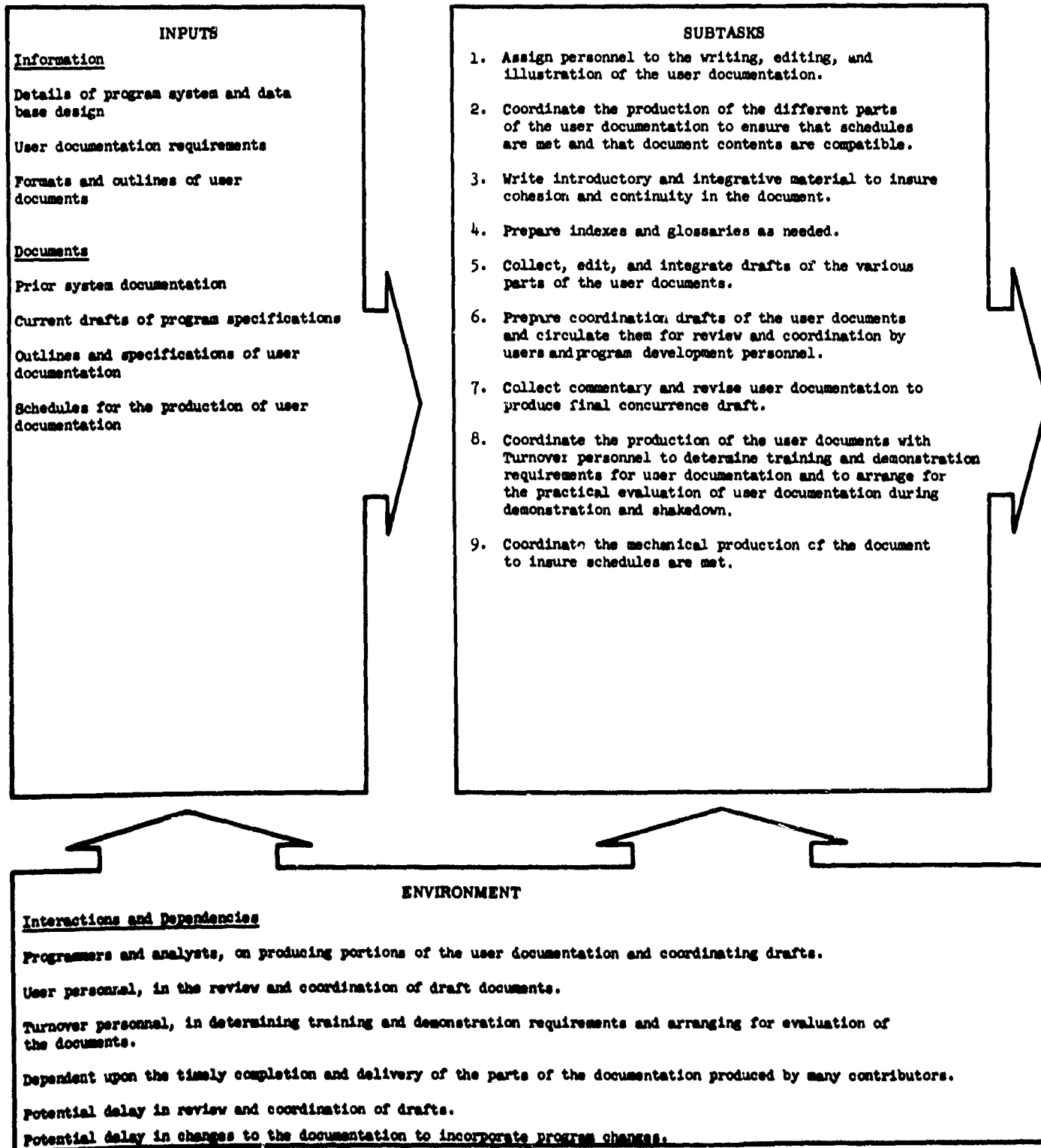
Approximately two man weeks per user's document, plus writing and editing costs of producing outlines and plans.
(Assumes each major function requiring one operator's attention results in one user document such as a guide or handbook)

ENVIRONMENTResources and Working Conditions

Technical writing and editing capability.
Programming documentation often inadequate and incomplete.

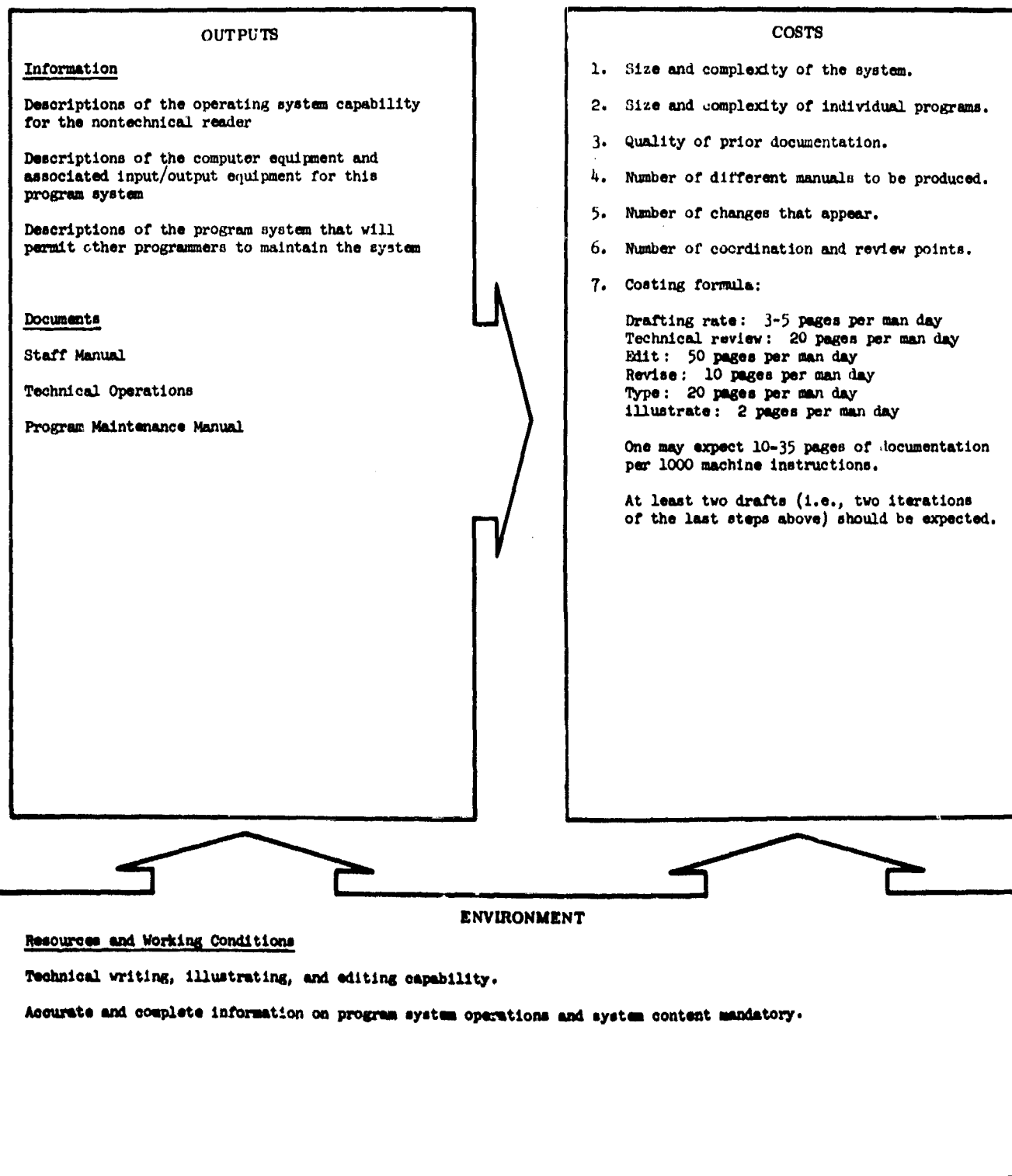
USER DOCUMENTATION TASK 3

PRODUCE USER DOCUMENTATION



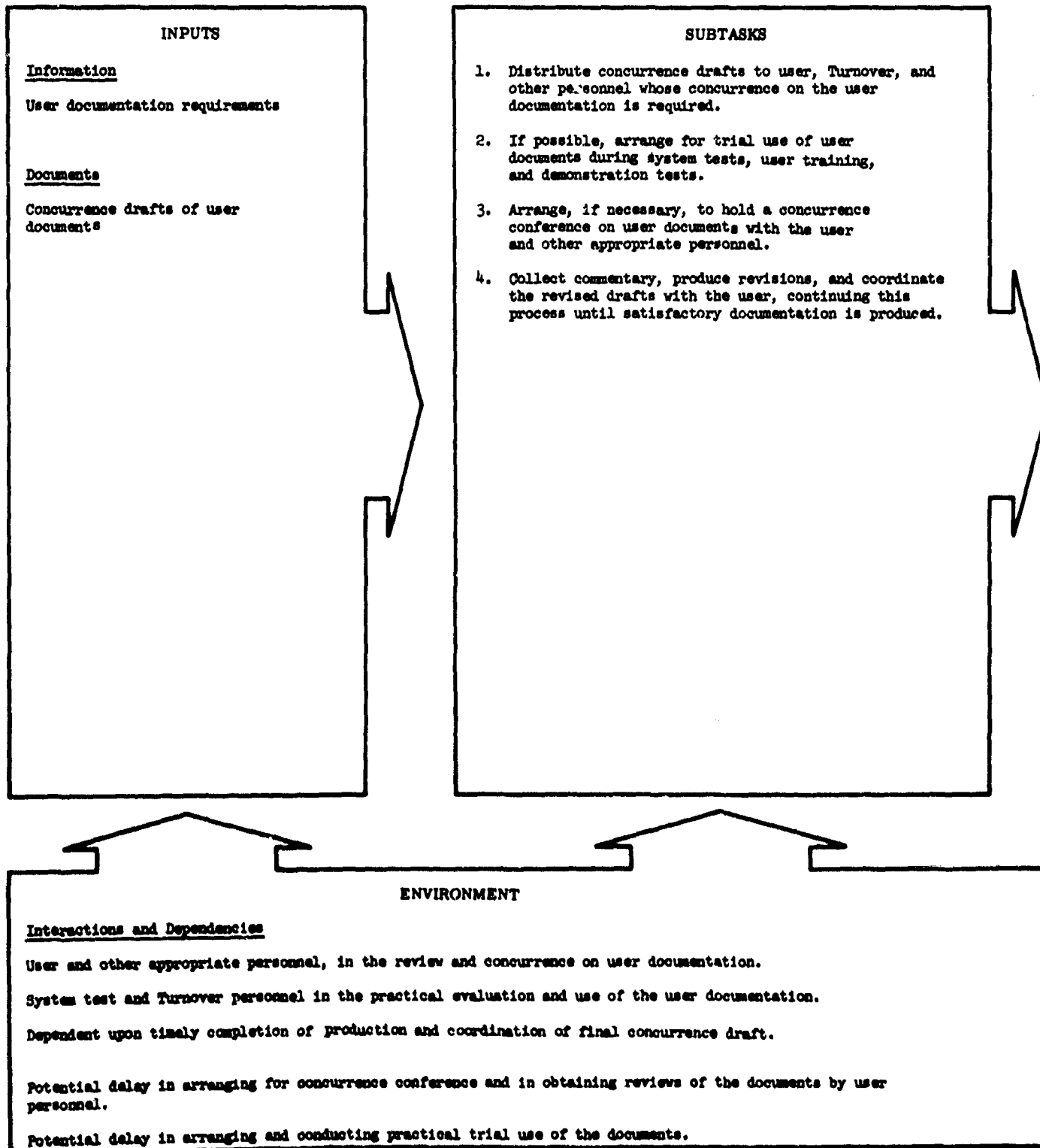
DESCRIPTION

Perform the technical writing and editing necessary to produce user documentation, coordinate the production of material by many contributors, and verify the adequacy and accuracy of the results.



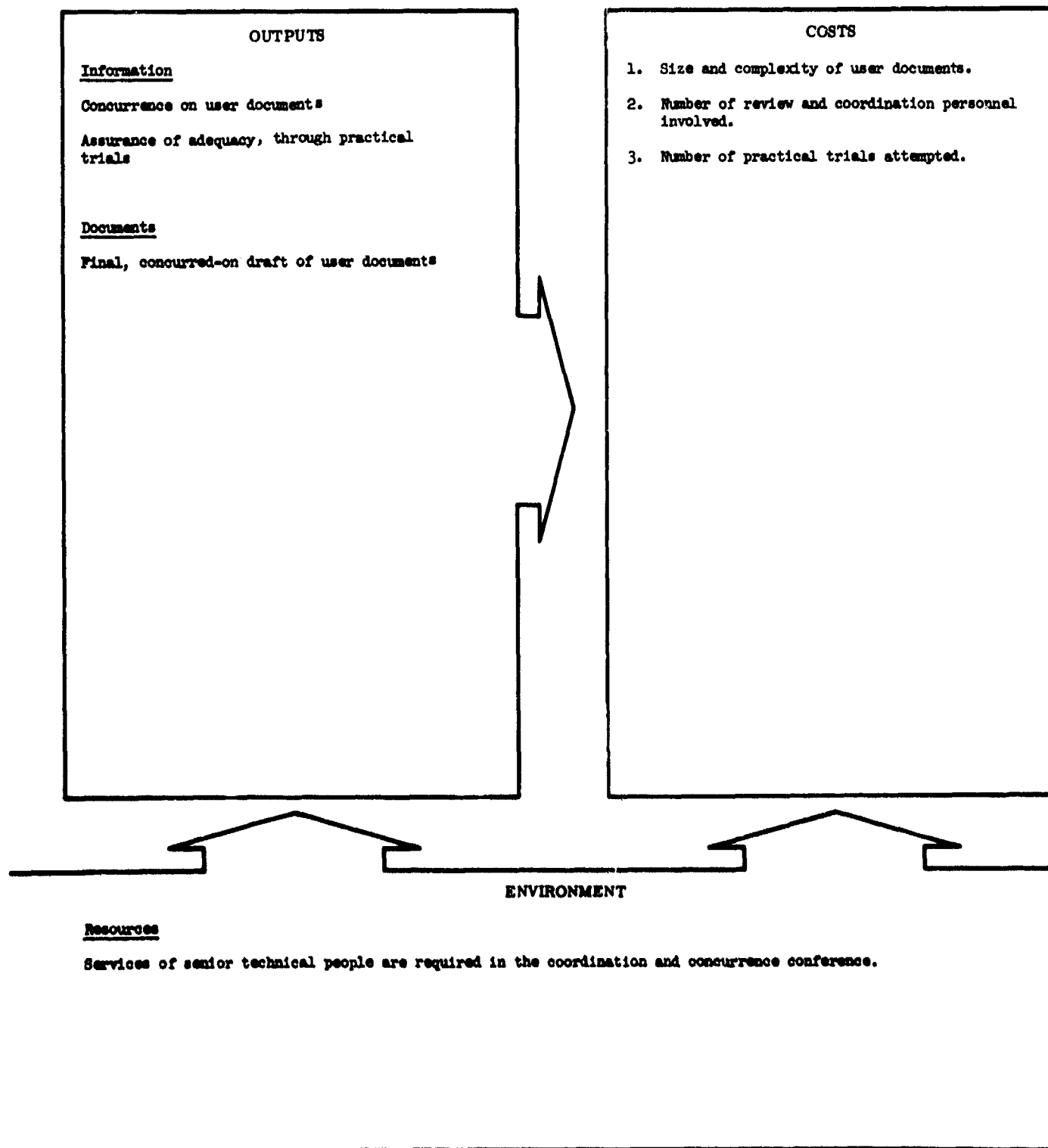
USER DOCUMENTATION TASK 4

OBTAIN CONCURRENCE ON USER DOCUMENTATION



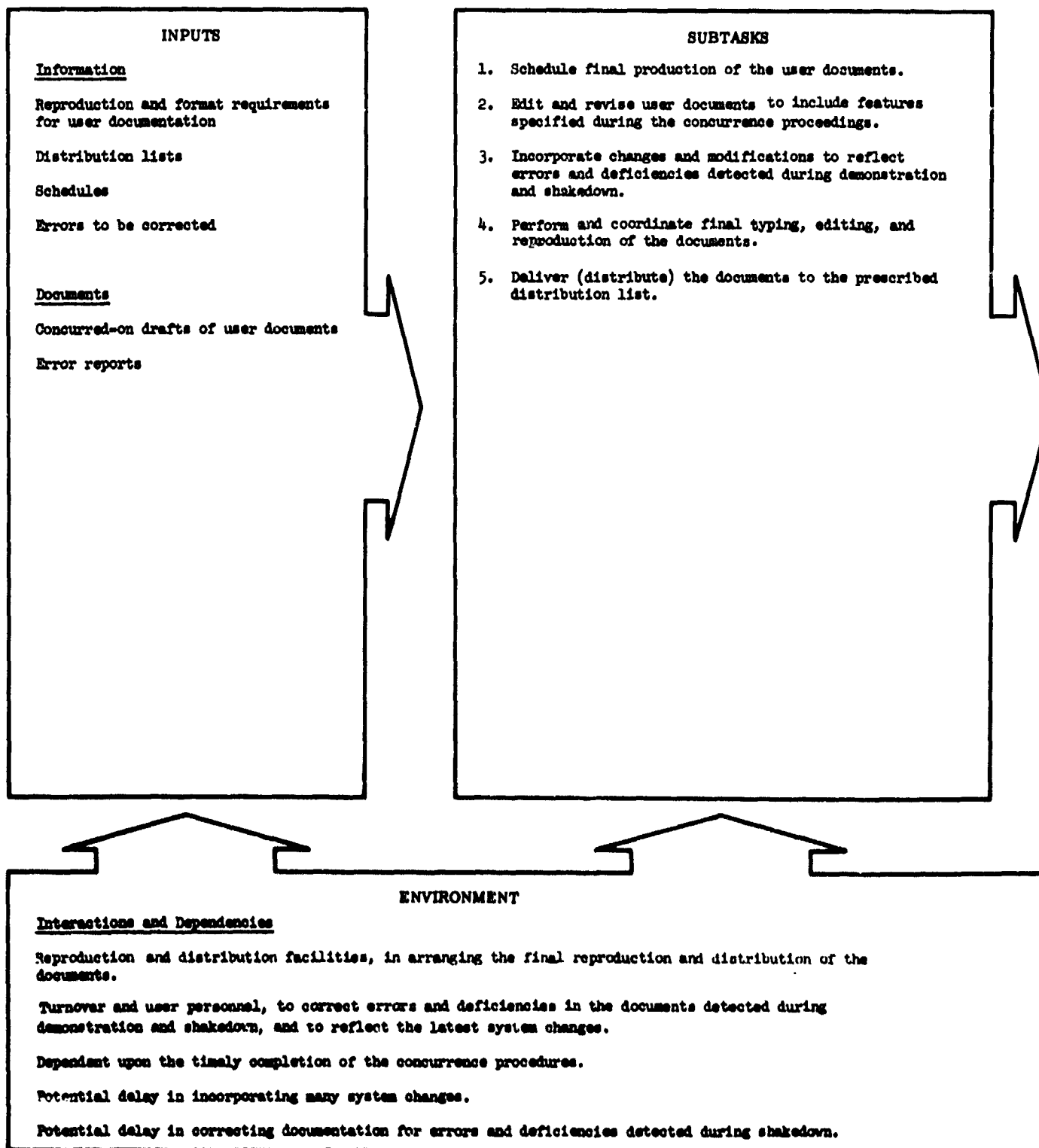
DESCRIPTION

Coordinate the final drafts of user documentation with the user and other appropriate personnel to obtain their approval and concurrence on the documentation, and, if possible, submit the documentation to a practical trial prior to final distribution.



USER DOCUMENTATION TASK 5

PUBLISH USER DOCUMENTATION



10 May 1965

163
(pages 164 and 165 blank)

TM-2314/000/00

DESCRIPTION

Produce and distribute the user documents, modified to reflect the latest changes and corrections in the system.

OUTPUTS

Documents

Final copies of user documents

COSTS

1. Number of document changes required to reflect error corrections and system changes.
2. Size of user documentation.
3. Number of copies.
4. Format and degree of polish.
5. Costing formula:

See Costing formula in User Documentation Task 3. These rates will be accelerated, influenced by the factors above.

ENVIRONMENT

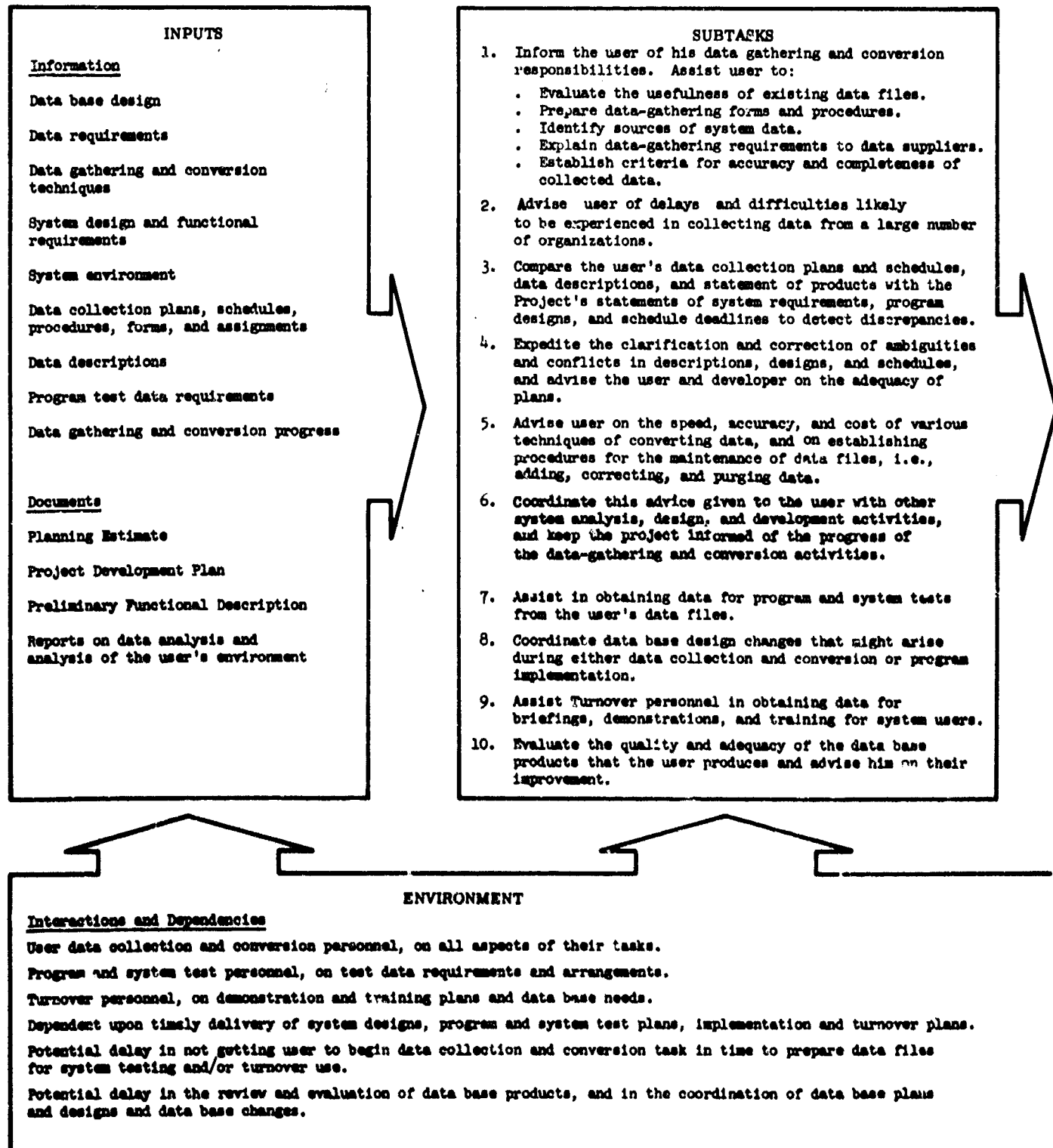
Resources and Working Conditions

Technical writing and editing capability.

Rigorous planning and scheduling to include all last-minute changes and to deliver documents on time.

USER ASSISTANCE TASK 1

ADVISE USER ON DATA COLLECTION AND CONVERSION ACTIVITIES



DESCRIPTION

Inform the user of his responsibilities for the collection and conversion of data, assist him in preparing to collect the data, advise him on the technical aspects of the task and the adequacy of his products, and coordinate and integrate the plans and designs of the system development activity and the user's data collection and conversion activity.

OUTPUTS

Information

Advice to the user on data-collection and conversion techniques

Coordination of data-collection activities with system implementation activities

Arrangements for use of data for program and system test and demonstration purposes

Coordination of data base changes

Assurance of quality in data base products

Arrangements for briefings and other transmissions of information

Documents

Progress reports

Technical memoranda to the user to provide technical advice and assistance

Technical memoranda to the Project to coordinate the details of data base structure and changes thereto

COSTS

1. Degree of knowledge of data base requirements.
2. Level of experience and skill of user in the collection and conversion of data.
3. Degree of close cooperation and rapport with the user.
4. Size and complexity of the system and the associated data base.
5. Degree of knowledge and understanding of user's terminology and environment.
6. Degree of dispersion of data sources, security classification of the data, and other conditions that might create problems of data accessibility.
7. Degree of dependence on sample data from the data base to be used in checkout and turnover.
8. Degree of user recognition of the need to communicate his plans, schedules, and state of progress.
9. Costing formula:

Part-time job for one analyst, not requiring senior personnel beyond the initial establishment of channels and evaluation of plans.

Gross estimate: One-quarter time for system analyst for the duration of the project.

ENVIRONMENT

Resources and Working Conditions

Requires the good will and cooperation of user data collection and conversion personnel.

Knowledge of user's terminology and environment.

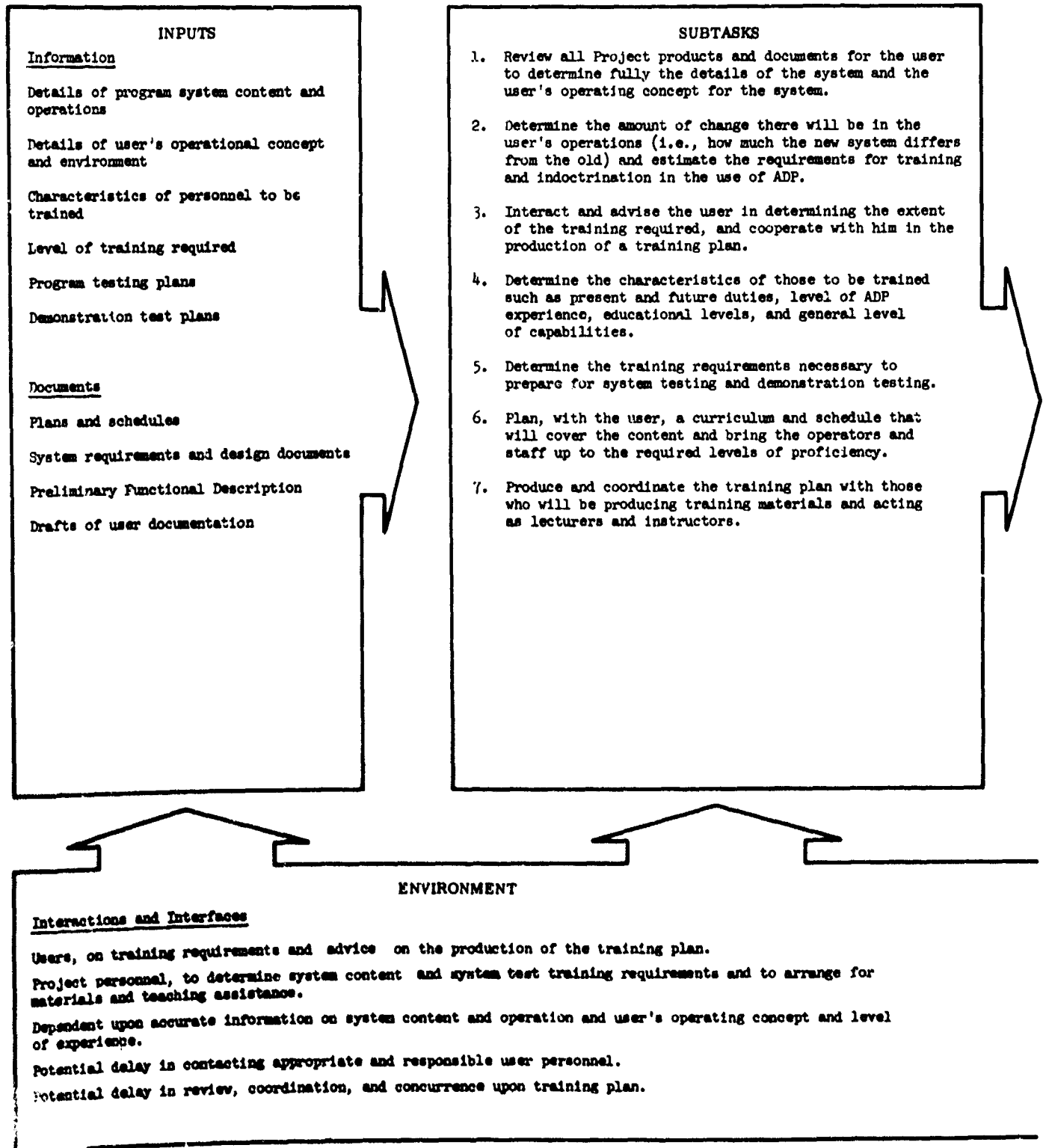
Knowledge of the system and the impact of data changes upon it.

One system analyst, part time.

Staff and advisory work requires many outside contacts and interactions with Project personnel.

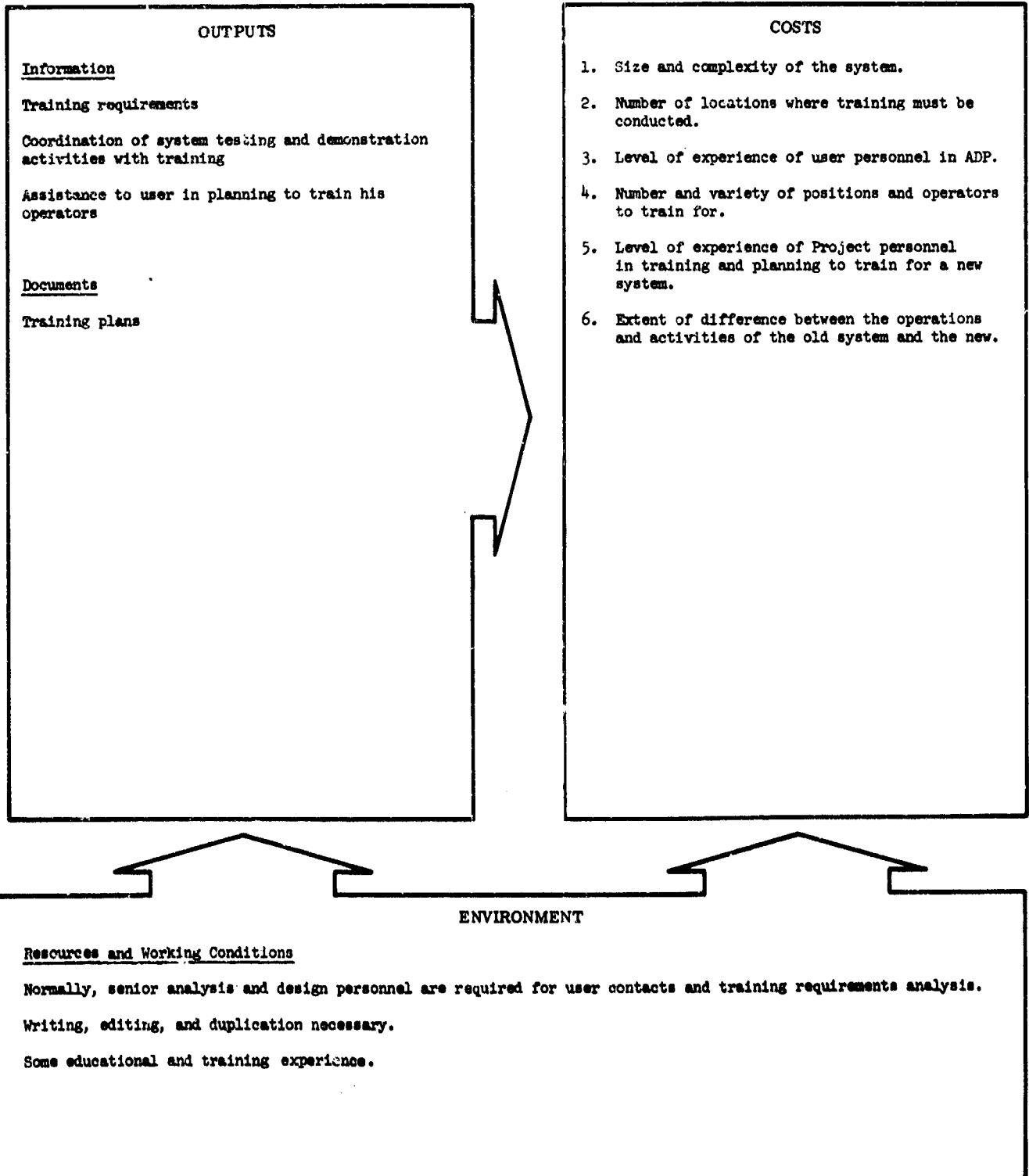
USER ASSISTANCE TASK 2

DEVELOP USER TRAINING PLAN



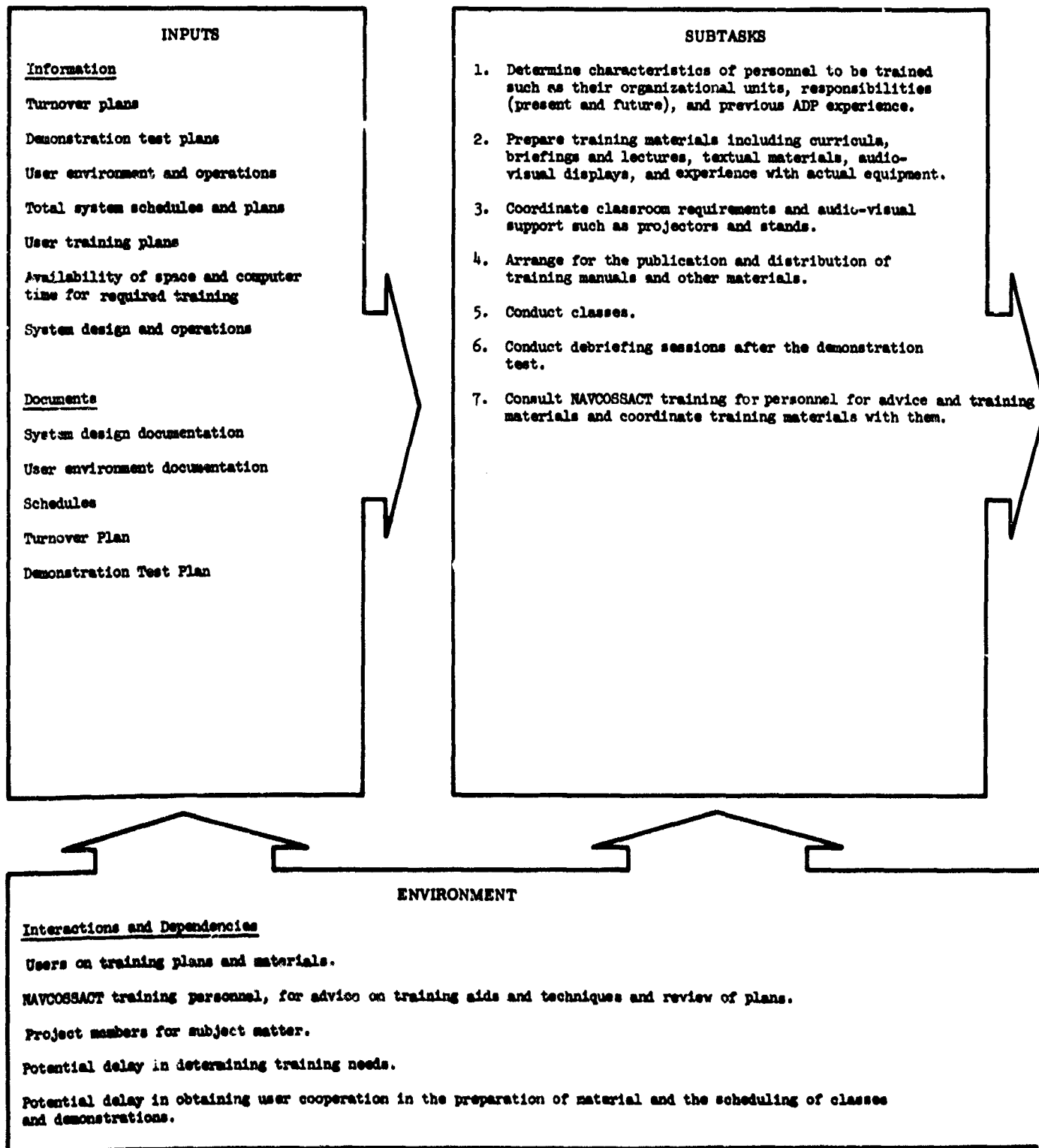
DESCRIPTION

Interact with the user to determine training requirements, and assist him in planning and scheduling training that will help his personnel to adjust to changes in operational concept, and also + prepare for system testing and demonstration testing activities.



USER ASSISTANCE TASK 3

CONDUCT TRAINING PROGRAM FOR USER'S STAFF, OPERATORS, AND MAINTENANCE PERSONNEL



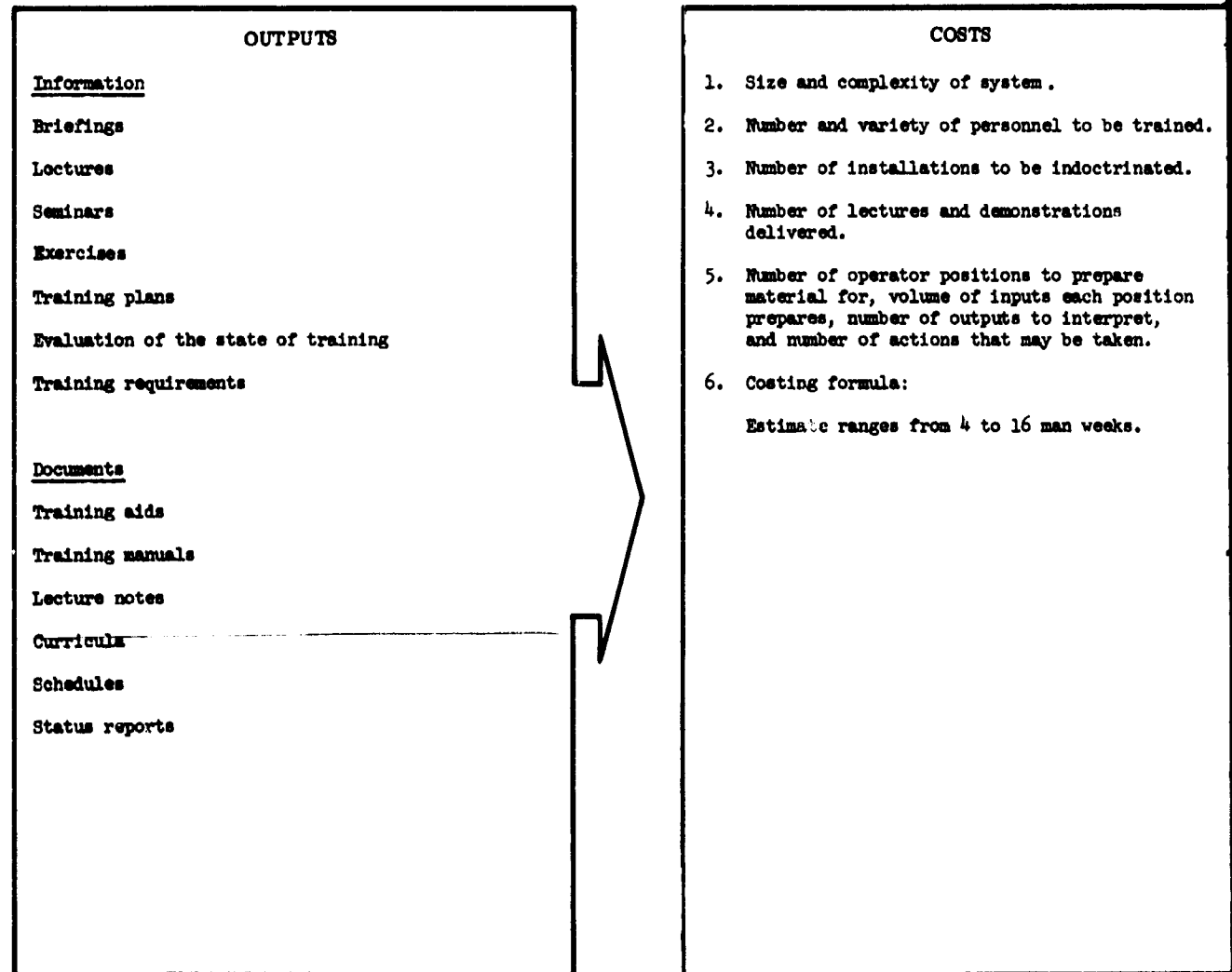
10 May 1965

171
(pages 172 and 173 blank)

TM-2314/000/00

DESCRIPTION

Train user personnel to interpret inputs and outputs, to prepare inputs, to control the computer and program system, and to maintain the program system.



ENVIRONMENT

Resources and Working Conditions

Knowledge of system and its operation.

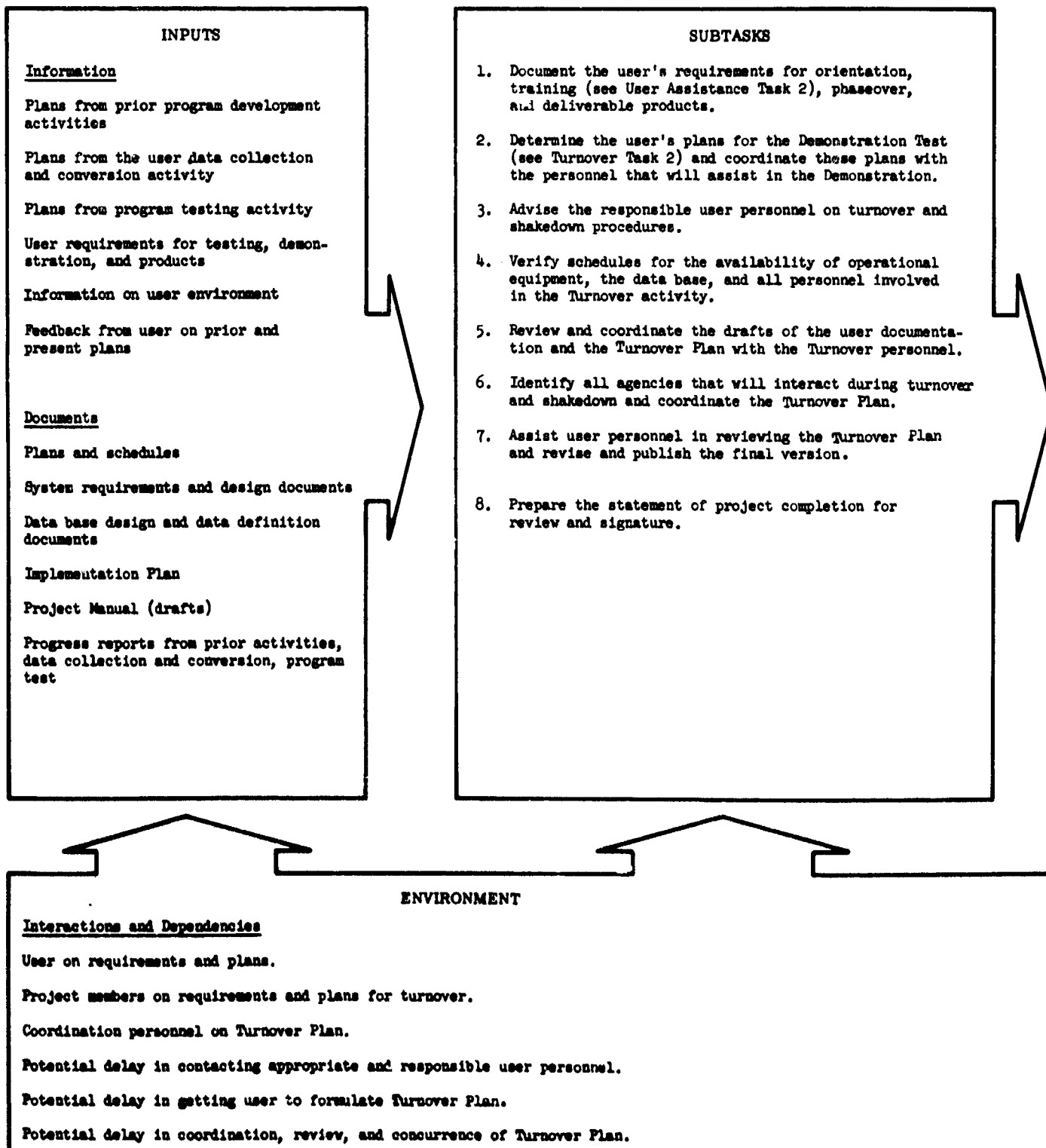
Knowledge of user and his operations.

Experience with curriculum development and teaching.

Duplication, graphic arts, technical writing, and other administrative support needed.

TURNOVER TASK 1

DEVELOP THE TURNOVER PLAN



DESCRIPTION In cooperation with the user, determine the phaseover requirements from current operations to the new system, including training, orientation, and demonstrations, and prepare plans to satisfy these requirements, including lists of turnover products, needed briefings, training and indoctrination efforts and procedures to demonstrate the successful operation of the system.

OUTPUTSInformation

Turnover requirements

Turnover plans

Assistance to user in turnover planning

Briefings

Documents

Turnover Plan, including delivery schedule for the completed program system and supporting documentation

Coordination memos and letters

COSTS

1. Size and complexity of system.
2. Number of locations to be installed.
3. Level of user experience in ADP.
4. Level of experience and skill of Project personnel for planning and doing turnover tasks.
5. Degree of rapport with user.
6. Number and variety of agencies involved in turnover plans and activities.

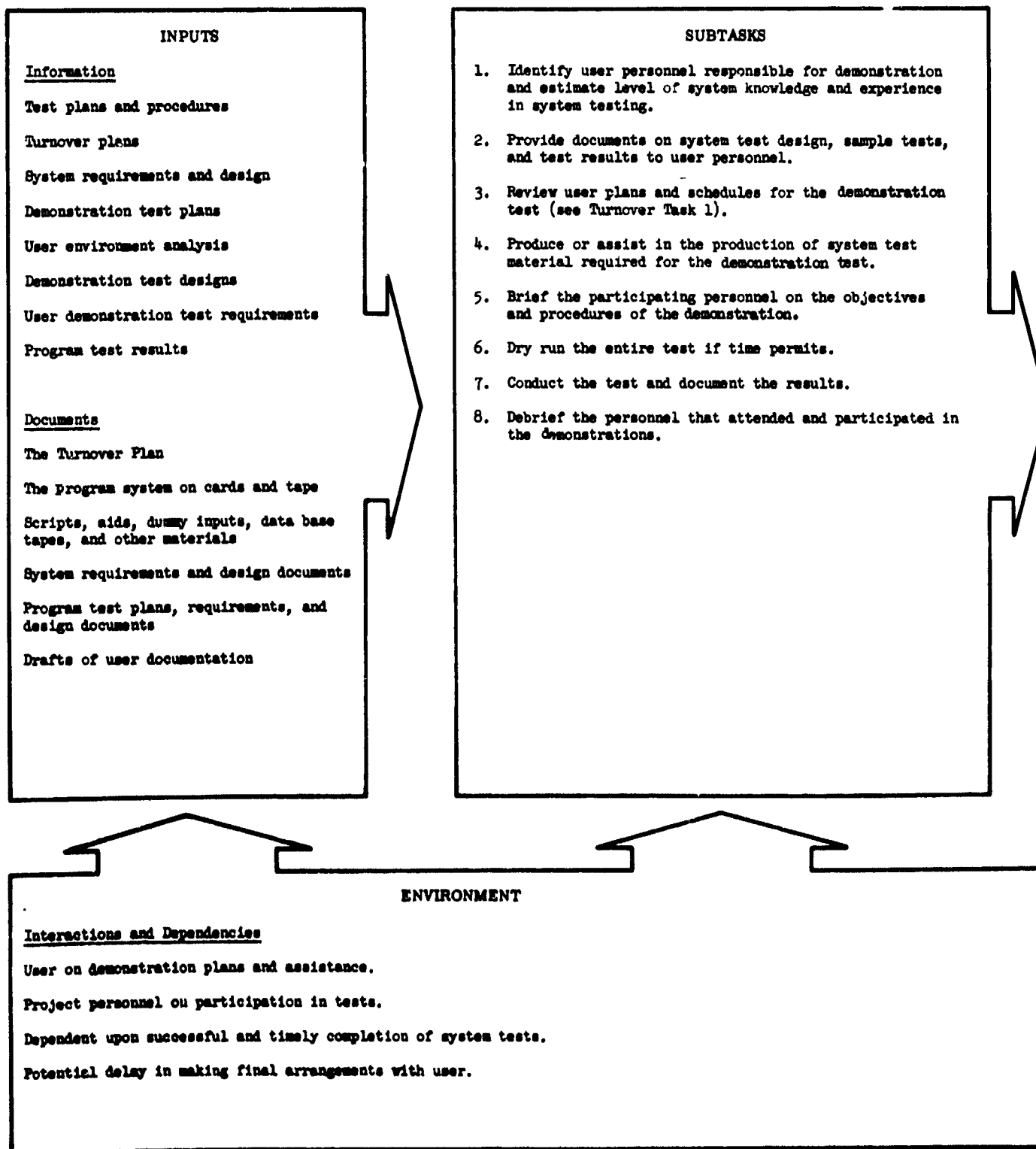
ENVIRONMENTResources and Working Conditions

Senior personnel required for user contacts.

Writing, editing, and duplication necessary.

Personal contact, analytic ability required.

Intensive contact with user on his plans and requirements.

TURNOVER TASK 2**CONDUCT DEMONSTRATION**

DESCRIPTION

Prepare for and conduct the demonstration test of the system jointly with user personnel.

OUTPUTSInformation

Briefings

Final preparations and plans

Debriefings

Documents

Detailed schedule

Reports on demonstration results

COSTS

1. Size and complexity of system to be tested.
2. Number and extent of demonstration tests to be run.
3. Level of experience of user personnel in conducting system tests and applying ADP to their operations.
4. Costing Formula:

Final preparation and actual conduct of the test for a system of moderate size should take an elapsed time of about one week. One or more dry runs, especially if associated with operator training, may add one or more weeks to the schedule.

Estimate about two man weeks for system of about 10,000 to 20,000 instructions.

ENVIRONMENTResources and Working Conditions

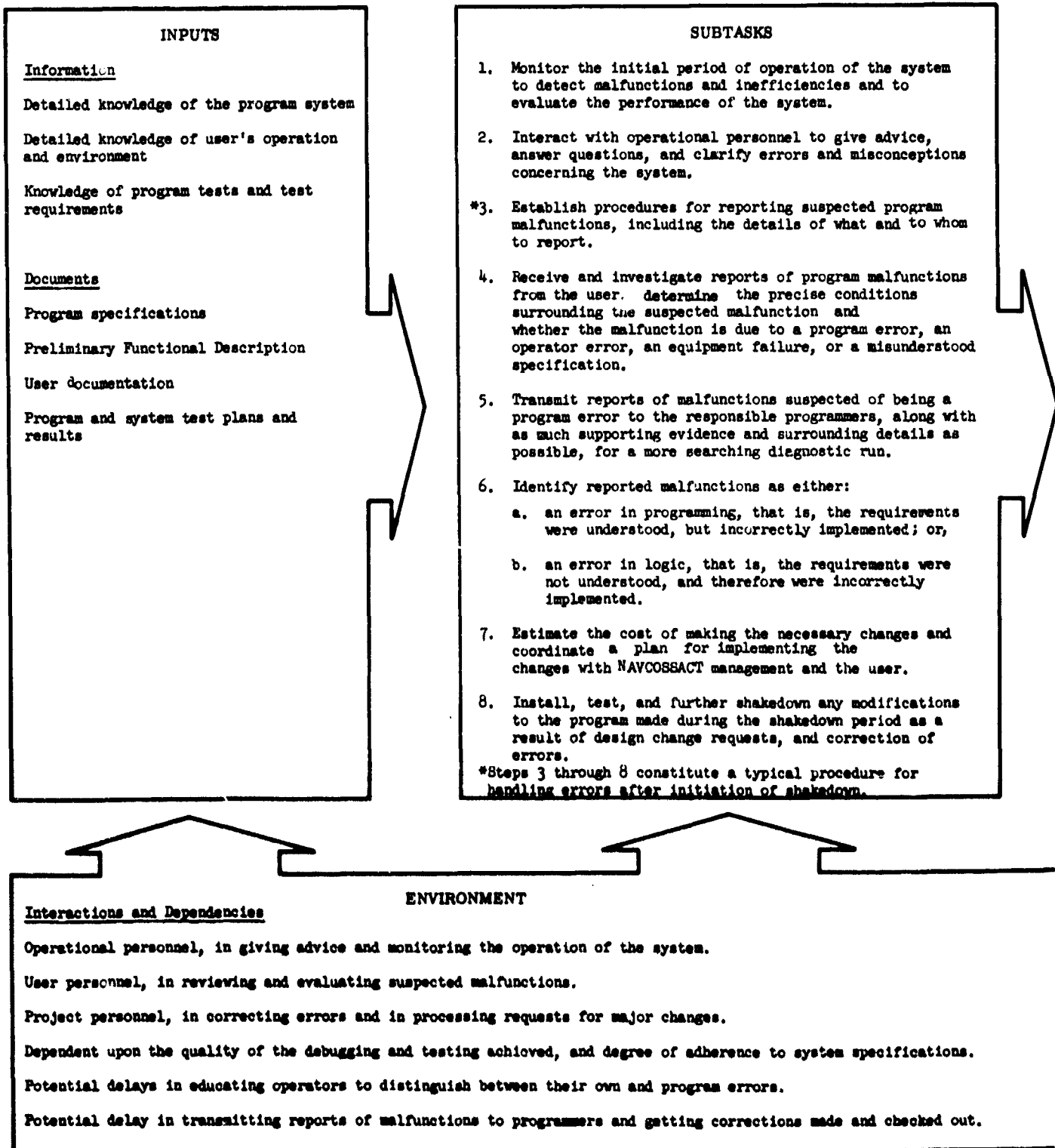
Requires skill in personal relations.

Knowledge of user's operation and ability to understand user's point of view.

Experience and skill in system test techniques.

TURNOVER TASK 3

ASSIST IN THE OPERATIONAL SHAKEDOWN OF THE SYSTEM



10 May 1965

179
(last page)

TM-2314/000/00

DESCRIPTION

Monitor the operation of the system during the shakedown period to detect and correct malfunctions in the programs, and to assist the user in learning to use and understand the system.

OUTPUTS

Information

Detected malfunctions

Operator training

Procedures for reporting and correcting malfunctions

Improved customer relations

Documents

Reports of suspected malfunctions

Design change requests

Agreements on the handling of system modifications

COSTS

1. Size and complexity of system.
2. Quality of delivered system and documentation.
3. Level of operator training.
4. Degree of change from prior operations.
5. Number of locations involved in shakedown.

ENVIRONMENT

Resources and Working Conditions

Detailed knowledge of system operations and user's environment.

Skill in personal relations.

Skill in detecting and correcting program malfunctions.

Many contacts with user and system operators.

Shakedown is a period of considerable anxiety and uncertainty as operators learn to use the system and as errors are detected, not only in the programs, but in procedures and equipment.

Period likely to be one of recurring emergencies and conferences to seek solutions to problems and misunderstandings.

PROJECT SUMMARY SHEET**PROJECT ID**

REQUEST CODE	DATE	REQUEST TITLE	REQUESTING
DIVISION	PROJECT LEADER	DATE ASS D.	CONTRACTOR

PROJECT D

MISSION AND OBJECTIVES

USING ORGANIZATIONS AND PLACEMENT

PRIMARY FUNCTIONS

SYSTEM EN

SYSTEM NAME	PROJECT NO	USING ORG
1. EMBEDDING SYSTEMS		
2. INTERFACING SYSTEMS		
3. SIMILAR SYSTEMS		

MANNING ASSUMPTIONS
KEY PERSONNEL**LEAD TIME A**

COMPUTI

COMPUTER	LOCATION	OPERATING SYSTEM	LANGUAGE	DATE OF FIRST USE	TOTAL EST

NUMBER OF COMPL**FIRST ESTIMATE**
(PE RELEASE)**SECOND ESTIMATE**
(POP RELEASE)**THIRD ESTIMATE**
(PFD RELEASE)

3E 2)

[illegible]

IDENTIFICATION

ORGANIZATION		REFERENCES	
		REQUESTING LETTER	PREPARED BY _____
	CONTRACT DATE	OTHER	REVIEWED BY _____
			APPROVED BY _____

DESCRIPTION

ENVIRONMENT

ORGANIZATION	NATURE OF APPLICATION

ASSUMPTIONS

CONTACTS

	USER
	CONTRACTOR
	EQUIPMENT
	OTHER

OPERATION USAGE

HOURS		MAXIMUM USAGE		TURNAROUND TIME	COMMENTS
ACT		HRS/DAY	HRS/WEEK		

INTER INSTRUCTIONS

ACTUAL (SOURCE LANGUAGE)	ACTUAL (MACHINE LANGUAGE)
-----------------------------	------------------------------

PROJECT SUMMARY (PARTIAL)

PROJECT PHASES & TASKS	NUMBER OF MAN MONTHS			START DATE			COMPLETION DATE		
	EST	REV EST	ACT	EST	REV EST	ACT	EST	REV EST	ACT
SYSTEM ANALYSIS									
1. PLAN THE PROJECT									
2. ANALYZE SYSTEM REQUIREMENTS									
3. ANALYZE USER ENVIRONMENT									
4. ANALYZE PRODUCTION REQUIREMENTS									
5. ANALYZE SIMILAR SYSTEMS									
6. EVALUATE CONTRACT PROPOSALS									
7. ANALYZE CHANGE REQUESTS									
SYSTEM DESIGN									
1. DESIGN TOTAL SYSTEM									
2. DESIGN PROGRAM SYSTEM									
3. OUTLINE PRELIMINARY FUNCTIONAL DESCR.									
4. PRODUCE PRELIMINARY FUNCTIONAL DESCR.									
5. FAMILIARIZE USER									
6. OBTAIN PFD CONCURRENCE									
7. INDOCTRINATE PRODUCTION PERSONNEL									
PROGRAM DEVELOPMENT									
1. DESIGN PROGRAM SYSTEM TEST									
2. DESIGN PROGRAMS									
3. DESIGN PROGRAM FILES									
4. ESTABLISH SYSTEM FILES									
PROGRAM CODING									
1. CODE PROGRAMS									
2. DESK CHECK PROGRAMS									
PROGRAM CHECKOUT									
1. LEARN TEST ENVIRONMENT									
2. COMPILE AND CHECK CODE									
3. TEST PROGRAMS									
4. TEST SUBSYSTEMS									
5. TEST SYSTEM									
USER DOCUMENTATION									
1. VERIFY SPECIFICATION DOCUMENTATION									
2. OUTLINE USER DOCUMENTATION									
3. PRODUCE USER DOCUMENTATION									
4. OBTAIN CONCURRENCE									
5. PUBLISH USER DOCUMENTATION									
USER TRAINING AND ASSISTANCE									
1. DATA COLLECTION AND CONVERSION									
2. DEVELOP USER TRAINING PLAN									
3. PROVIDE USER TRAINING AND ASSISTANCE									
TURNOVER									
1. DEVELOP TURNOVER PLAN									
2. CONDUCT DEMONSTRATION									
3. OPERATIONAL SHakedown									
TOTALS									

★ PHASES ABOVE THIS LINE WILL BE COMPLETED AFTER PLANNING ESTIMATE IS APPROVED.

PHASES BELOW THIS LINE WILL BE COMPLETED AFTER PRELIMINARY FUNCTIONAL DESCRIPTION IS APPROVED.

IDENTIFICATION

ORGANIZATION		REFERENCES	
		REQUESTING LETTER	PREPARED BY _____
	CONTRACT DATE	OTHER	REVIEWED BY _____
			APPROVED BY _____

DESCRIPTION

ENVIRONMENT

ANALYSIS	NATURE OF APPLICATION

ASSUMPTIONS

CONTACTS

	USER
	CONTRACTOR
	EQUIPMENT
	OTHER

USER USAGE

HOURS	MAXIMUM USAGE			TURNAROUND TIME	COMMENTS
	ACT	HRS/DAY	HRS/WEEK		

INTER INSTRUCTIONS

ACTUAL (SOURCE LANGUAGE)	ACTUAL (MACHINE LANGUAGE)

COMPUTER PROGRAM

PROGRAM IDENTIFICATION	PROGRAM NAME	PROJ
------------------------	--------------	------

PROGRAM DEVELOPMENT

PROGRAM DESIGN	MAN MONTHS			START DATE			END DATE	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST
LOGIC ANALYSIS & FLOW CHART								
TIMING ANALYSIS								
DESIGN REVIEW								
PROGRAM SPECIFICATIONS								
TOTAL AND ENTER ON SUMMARY SHEET PROGRAM DEVELOPMENT TASK 2						ELAPSED TIME		

DATA DESCRIPTION	NUMBER OF ITEMS			INPUT FORMATS			OUTPUT FORMATS	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST

DATA DESIGN	MAN MONTHS			START DATE			END DATE	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST
DATA ANALYSIS								
I/O FORMATS								
DATA DESIGN								
ALLOCATE STORAGE								
DATA REVIEW								
DOCUMENTATION								
TOTAL AND ENTER ON SUMMARY SHEET PROGRAM DEVELOPMENT TASK 3						ELAPSED TIME		

PROGRAM DESCRIPTION	NUMBER OF INSTRUCTIONS						NUMBER OF	
	NEW			REVISED			EST	REV EST
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST
ENTER TOTAL NUMBER OF INSTRUCTIONS ON SUMMARY SHEET								

PROGRAM CODING

CODING DESCRIPTION	NEW ROUTINES			REVISED ROUTINES			LIBRARY ROUTINES	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST

CODING	MAN MONTHS			KEY PUNCH & VERIFY MACH. HRS.			START DATE	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST
CODE								
DESK CHECK								
DOCUMENT								
ENTER TOTALS ON SUMMARY SHEET FOR PROGRAM CODING TASK 1 AND 2								

PROGRAM CHECKOUT

CHECKOUT DESCRIPTION	NUMBER COMPILE ASSY.			NUMBER CODE CHECKS			TEST	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST

	MAN MONTHS			START DATE			COMPLETION	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST
COMPILE AND CHECK CODE								
ENTER TOTALS ON SUMMARY SHEET FOR PROGRAM CHECKOUT TASK 2								
DESIGN PROGRAM TEST								
TEST DATA PRODUCTION								
TEST PROGRAM								
TOTAL AND ENTER ON SUMMARY SHEET FOR PROGRAM CHECKOUT TASK 3								

REMARKS	
PREPARED BY	REVIEWED BY

PLANNING SHEET

AM FUNCTION 6	TYPE JOB (NEW, CONVERSION, REV.)
------------------	----------------------------------

E	ASSIGNMENTS												
	ACTUAL	<table border="1"> <tr> <th>FIRST</th> <th>SECOND</th> </tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> </table>	FIRST	SECOND									
FIRST	SECOND												

IMATS	NUMBER OF TABLES			NO. OF CONSTANTS & PARAMETERS			NUMBER OF FILES		
ACTUAL	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL

E	ASSIGNMENTS												
	ACTUAL	<table border="1"> <tr> <th>FIRST</th> <th>SECOND</th> </tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> </table>	FIRST	SECOND									
FIRST	SECOND												

LOCKS	COMPLEXITY			NUMBER OF LIBRARY ROUTINES			NUMBER PAGES OF DOCUMENTS		
ACTUAL	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL

MTINES	NUMBER PAGES OF DOCUMENTS		
ACTUAL	EST	REV EST	ACTUAL

TE	END DATE			ASSIGNMENTS										
	ACTUAL	EST	REV EST	ACTUAL	<table border="1"> <tr> <th>FIRST</th> <th>SECOND</th> </tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> </table>	FIRST	SECOND							
FIRST	SECOND													
ELAPSED TIME														

NUMBER OF PROGRAM TESTS				NUMBER PAGES OF DOCUMENTS			
TRIALS							
ACTUAL	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	

DATE	NO. OF COMPUTER HRS.		ASSIGNMENTS	
ACTUAL	EST	REV EST		

		APPROVED BY
--	--	-------------

COMPUTER PROGRAM F

PROGRAM IDENTIFICATION	PROGRAM NAME	PROGRAM
------------------------	--------------	---------

PROGRAM DEVELOPMENT

PROGRAM DESIGN	MAN MONTHS			START DATE			END DATE	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST
LOGIC ANALYSIS & FLOW CHART								
TIMING ANALYSIS								
DESIGN REVIEW								
PROGRAM SPECIFICATIONS								
TOTAL AND ENTER ON SUMMARY SHEET PROGRAM DEVELOPMENT TASK 2						ELAPSED TIME		

DATA DESCRIPTION	NUMBER OF ITEMS			INPUT FORMATS			OUTPUT FORM	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST

DATA DESIGN	MAN MONTHS			START DATE			END DATE	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST
DATA ANALYSIS								
I/O FORMATS								
DATA DESIGN								
ALLOCATE STORAGE								
DATA REVIEW								
DOCUMENTATION								
TOTAL AND ENTER ON SUMMARY SHEET PROGRAM DEVELOPMENT TASK 3						ELAPSED TIME		

PROGRAM DESCRIPTION	NUMBER OF INSTRUCTIONS						NUMBER OF BLOC		
	EST	NEW	REV EST	ACTUAL	EST	REVISED	ACTUAL	EST	REV EST
ENTER TOTAL NUMBER OF INSTRUCTIONS ON SUMMARY SHEET									

PROGRAM CODING

CODING DESCRIPTION	NEW ROUTINES			REVISED ROUTINES			LIBRARY ROUT	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST

CODING	MAN MONTHS			KEY PUNCH & VERIFY MACH. HRS.			START DATE	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST
CODE								
DESK CHECK								
DOCUMENT								
ENTER TOTALS ON SUMMARY SHEET FOR PROGRAM CODING TASK 1 AND 2								

PROGRAM CHECKOUT

CHECKOUT DESCRIPTION	NUMBER COMPILE ASSY.			NUMBER CODE CHECKS			TESTS	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST

	MAN MONTHS			START DATE			COMPLETION D	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST
COMPILE AND CHECK CODE								
ENTER TOTALS ON SUMMARY SHEET FOR PROGRAM CHECKOUT TASK 3								
DESIGN PROGRAM TEST								
TEST DATA PRODUCTION								
TEST PROGRAM								
TOTAL AND ENTER ON SUMMARY SHEET FOR PROGRAM CHECKOUT TASK 3								

REMARKS

PREPARED BY

REVIEWED BY

PLANNING SHEET

FUNCTION	TYPE JOB (NEW, CONVERSION, REV.)
----------	----------------------------------

ASSIGNMENTS		
ACTUAL	FIRST	SECOND

NUMBER OF TABLES		NO. OF CONSTANTS & PARAMETERS		NUMBER OF FILES	
ACTUAL	EST	REV EST	ACTUAL	EST	REV EST

ASSIGNMENTS		
ACTUAL	FIRST	SECOND

COMPLEXITY		NUMBER OF LIBRARY ROUTINES		NUMBER PAGES OF DOCUMENTS	
ACTUAL	EST	REV EST	ACTUAL	EST	REV EST

NUMBER PAGES OF DOCUMENTS	
ACTUAL	EST

END DATE		ASSIGNMENTS	
ACTUAL	EST	FIRST	SECOND

NUMBER OF PROGRAM TESTS		NUMBER PAGES OF DOCUMENTS	
ACTUAL	EST	REV EST	ACTUAL

NO. OF COMPUTER HRS.		ASSIGNMENTS	
ACTUAL	EST	FIRST	SECOND

	APPROVED BY

COMPUTER PROGRAM

PROGRAM IDENTIFICATION	PROGRAM NAME	PROGRAM
------------------------	--------------	---------

PROGRAM DEVELOPMENT

PROGRAM DESIGN	MAN MONTHS			START DATE			END DATE	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST
LOGIC ANALYSIS & FLOW CHART								
TIMING ANALYSIS								
DESIGN REVIEW								
PROGRAM SPECIFICATIONS								
TOTAL AND ENTER ON SUMMARY SHEET PROGRAM DEVELOPMENT TASK 2						ELAPSED TIME		

DATA DESCRIPTION	NUMBER OF ITEMS			INPUT FORMATS			OUTPUT FORMATS	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST

DATA DESIGN	MAN MONTHS			START DATE			END DATE	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST
DATA ANALYSIS								
I/O FORMATS								
DATA DESIGN								
ALLOCATE STORAGE								
DATA REVIEW								
DOCUMENTATION								
TOTAL AND ENTER ON SUMMARY SHEET PROGRAM DEVELOPMENT TASK 3						ELAPSED TIME		

PROGRAM DESCRIPTION	NUMBER OF INSTRUCTIONS						NUMBER OF BLANKS	
	EST	NEW REV EST	ACTUAL	EST	REVISED REV EST	ACTUAL	EST	REV EST
ENTER TOTAL NUMBER OF INSTRUCTIONS ON SUMMARY SHEET								

PROGRAM CODING

CODING DESCRIPTION	NEW ROUTINES			REVISED ROUTINES			LIBRARY ROUTINES	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST

CODING	MAN MONTHS			KEY PUNCH & VERIFY MACH. HRS.			START DATE	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST
CODE								
DESK CHECK								
DOCUMENT								
ENTER TOTALS ON SUMMARY SHEET FOR PROGRAM CODING TASK 1 AND 2								

PROGRAM CHECKOUT

CHECKOUT DESCRIPTION	NUMBER COMPILE ASSY.			NUMBER CODE CHECKS			TESTS	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST

	MAN MONTHS			START DATE			COMPLETION DATE	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST
COMPILE AND CHECK CODE								
ENTER TOTALS ON SUMMARY SHEET FOR PROGRAM CHECKOUT TASK 2								
DESIGN PROGRAM TEST								
TEST DATA PRODUCTION								
TEST PROGRAM								
TOTAL AND ENTER ON SUMMARY SHEET FOR PROGRAM CHECKOUT TASK 3								

REMARKS

PREPARED BY

REVIEWED BY

PLANNING SHEET

FUNCTION 1	TYPE JOB (NEW, CONVERSION, REV.)
---------------	----------------------------------

ASSIGNMENTS		
ACTUAL	FIRST	SECOND

ATS		NUMBER OF TABLES		NO. OF CONSTANTS & PARAMETERS			NUMBER OF FILES		
ACTUAL	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL

ASSIGNMENTS		
ACTUAL	FIRST	SECOND

DOCKS		COMPLEXITY		NUMBER OF LIBRARY ROUTINES			NUMBER PAGES OF DOCUMENTS		
ACTUAL	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL

INES		NUMBER PAGES OF DOCUMENTS		
ACTUAL	EST	REV EST	ACTUAL	EST

E		END DATE		ASSIGNMENTS	
ACTUAL	EST	REV EST	ACTUAL	FIRST	SECOND

NUMBER OF PROGRAM TESTS				NUMBER PAGES OF DOCUMENTS		
TRIALS						
ACTUAL	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL

ATE		NO. OF COMPUTER HRS.		ASSIGNMENTS	
ACTUAL	EST	REV EST	ACTUAL	FIRST	SECOND

APPROVED BY	

COMPUTER PROGRAM

PROGRAM IDENTIFICATION	PROGRAM NAME	PROGR
------------------------	--------------	-------

PROGRAM DEVELOPMENT

PROGRAM DESIGN	MAN MONTHS			START DATE			END DATE	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST
LOGIC ANALYSIS & FLOW CHART								
TIMING ANALYSIS								
DESIGN REVIEW								
PROGRAM SPECIFICATIONS								
TOTAL AND ENTER ON SUMMARY SHEET PROGRAM DEVELOPMENT TASK 2						ELAPSED TIME		

DATA DESCRIPTION	NUMBER OF ITEMS			INPUT FORMATS			OUTPUT FOR	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST

DATA DESIGN	MAN MONTHS			START DATE			END DATE	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST
DATA ANALYSIS								
I/O FORMATS								
DATA DESIGN								
ALLOCATE STORAGE								
DATA REVIEW								
DOCUMENTATION								
TOTAL AND ENTER ON SUMMARY SHEET PROGRAM DEVELOPMENT TASK 3						ELAPSED TIME		

PROGRAM DESCRIPTION	NUMBER OF INSTRUCTIONS						NUMBER OF B	
	NEW		REVISED					
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST
ENTER TOTAL NUMBER OF INSTRUCTIONS ON SUMMARY SHEET								

PROGRAM CODING

CODING DESCRIPTION	NEW ROUTINES			REVISED ROUTINES			LIBRARY ROUT	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST

CODING	MAN MONTHS			KEY PUNCH & VERIFY MACH. HRS.			START DATE	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST
CODE								
DESK CHECK								
DOCUMENT								
ENTER TOTALS ON SUMMARY SHEET FOR PROGRAM CODING TASK 1 AND 2								

PROGRAM CHECKOUT

CHECKOUT DESCRIPTION	NUMBER COMPILE ASSY.			NUMBER CODE CHECKS			TESTS	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST

	MAN MONTHS			START DATE			COMPLETION	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV EST
COMPILE AND CHECK CODE								
ENTER TOTALS ON SUMMARY SHEET FOR PROGRAM CHECKOUT TASK 2								
DESIGN PROGRAM TEST								
TEST DATA PRODUCTION								
TEST PROGRAM								
TOTAL AND ENTER ON SUMMARY SHEET FOR PROGRAM CHECKOUT TASK 3								

REMARKS
PREPARED BY
REVIEWED BY

PLANNING SHEET

NAME	DATE	TIME	UNIT	DESCRIPTION	REMARKS	TYPE JOB (NEW, CONVERSION, REV.)
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31						
32						
33						
34						
35						
36						
37						
38						
39						
40						
41						
42						
43						
44						
45						
46						
47						
48						
49						
50						
51						
52						
53						
54						
55						
56						
57						
58						
59						
60						
61						
62						
63						
64						
65						
66						
67						
68						
69						
70						
71						
72						
73						
74						
75						
76						
77						
78						
79						
80						
81						
82						
83						
84						
85						
86						
87						
88						
89						
90						

[illegible][illegible][illegible][illegible]

TIMES		NUMBER PAGES OF DOCUMENTS		
ACTUAL		EST	REV EST	ACTUAL

RE		END DATE			ASSIGNMENTS	
	ACTUAL	EST	REV EST	ACTUAL	FIRST	SECOND
ELAPSED TIME						

NUMBER OF PROGRAM TESTS				NUMBER PAGES OF DOCUMENTS			
TRIALS							
ACTUAL	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	

[illegible]

1	APPROVED BY

PROGRAM SYSTEM P

SYSTEM CHECKOUT DESCRIPTION	NUMBER OF SUBSYSTEM TESTS							
	TESTS			TRIALS			TEST	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV E

PROGRAM SUBSYSTEM TESTS	MAN MONTHS			START DATE			COMPLETION	
	EST	REV EST	ACTUAL	EST	REV EST	ACTUAL	EST	REV ES
DESIGN SUBSYSTEM TEST								
TEST DATA PRODUCTION								
TEST SUBSYSTEM								
TOTAL AND ENTER ON SUMMARY SHEET, PROGRAM CHECKOUT, TASK 4						ELAPSED TIME		

PROGRAM SYSTEM TESTS								
DESIGN SYSTEM TEST								
TEST DATA PRODUCTION								
TEST PROGRAM SYSTEM								
TOTAL AND ENTER ON SYMMARY SHEET, PROGRAM CHECKOUT, TASK 5						ELAPSED TIME		

REMARKS

PREPARED BY	REVIEWED BY

ANNING SHEET

NUMBER OF SYSTEM TESTS			
	TRIALS		
ACTUAL	EST	REV EST	ACTUAL

[illegible]

	APPROVED BY
--	-------------

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R&D		
(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)		
1. ORIGINATING ACTIVITY (Org. use only)		2c. REPORT SECURITY CLASSIFICATION
System Development Corporation, Santa Monica, Calif.		Unclassified
		2d. GROUP
3. REPORT TITLE		
PLANNING GUIDE FOR COMPUTER PROGRAM DEVELOPMENT.		
4. DESCRIPTIVE NOTES (Type of report and inclusive dates)		
5. AUTHOR(S) (Last name, first name, initial)		
Larr, L., LaBolle, V., and Willmorth, N. E.		
6. REPORT DATE	7b. TOTAL NO. OF PAGES	7c. NO. OF REFS
10 May 1965	179p.	
8a. CONTRACT OR GRANT NO.	8b. ORIGINATOR'S REPORT NUMBER(S)	
Nonr 4543(00)	TM-2314	
a. PROJECT NO.		
c.	8c. OTHER REPORT NO(S) (Any other numbers that may be assigned to this report)	
d.		
10. AVAILABILITY/LIMITATION NOTICES		
11. SUPPLEMENTARY NOTES		
12. SPONSORING MILITARY ACTIVITY		
12. ABSTRACT This document offers a systematic approach for planning projects to develop computer-based information systems. The primary emphasis is placed on the computer program portion of such systems. A descriptive model of the development process forms the basis for a set of prescribed planning and management tasks. The model includes eight phases: (1) System Analysis, (2) System Design, (3) Program Development, (4) Program Coding, (5) Program Checkout, (6) User Documentation, (7) User Training and Assistance, and (8) Turnover. Each phase is further divided into tasks and subtasks for the purpose of more clearly understanding the elements of the development process. A detailed sequence of planning activities provides guidance for planning, scheduling and costing the tasks that comprise the development process, and forms are supplied to record the planning results and to serve as checklists for the required work. The forms and procedures also provide a basis for project control and for collection of data that may be used to improve estimates based upon experience. Although this Guide was prepared for use at the Naval Command Systems Support Activity, the material can easily be adapted to apply to programming in other organizations. (authors)		

DD FORM 1473
1 JAN 64

Unclassified

Security Classification

1A. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Systematic Model Planning Management Computer-Based Information Systems Design Analysis Program Development Coding Checkout						

INSTRUCTIONS

1. **ORIGINATING ACTIVITY:** Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization (corporate author) issuing the report.

2a. **REPORT SECURITY CLASSIFICATION:** Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. **GROUP:** Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. **REPORT TITLE:** Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parentheses immediately following the title.

4. **DESCRIPTIVE NOTES:** If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. **AUTHOR(S):** Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. **REPORT DATE:** Enter the date of the report as day, month, year, or month, year. If more than one date appears on the report, use date of publication.

7a. **TOTAL NUMBER OF PAGES:** The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. **NUMBER OF REFERENCES:** Enter the total number of references cited in the report.

8a. **CONTRACT OR GRANT NUMBER:** If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. **PROJECT NUMBER:** Enter the appropriate military department identification, such as project, subproject number, system numbers, task number, etc.

9a. **ORIGINATOR'S REPORT NUMBER(S):** Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. **OTHER REPORT NUMBER(S):** If the report has been assigned any other report numbers (either by the originator or by the sponsor), also enter this number(s).

10. **AVAILABILITY/LIMITATION NOTICES:** Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

- (1) "Qualified requesters may obtain copies of this report from DDC."
- (2) "Foreign announcement and dissemination of this report by DDC is not authorized."
- (3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through _____."
- (4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through _____."
- (5) "All distribution of this report is controlled. Qualified DDC users shall request through _____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. **SUPPLEMENTARY NOTES:** Use for additional explanatory notes.

12. **SPONSORING MILITARY ACTIVITY:** Enter the name of the departmental project office or laboratory sponsoring (paying for) the research and development. Include address.

13. **ABSTRACT:** Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as (TS), (S), (C), or (U).

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. **KEY WORDS:** Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.