

# PROCEDURES FOR MANAGEMENT CONTROL OF COMPUTER PROGRAMMING IN APOLLO

GPO PRICE \$ \_\_\_\_\_

CFSTI PRICE(S) \$ \_\_\_\_\_

Hard copy (HC) 6.00

Microfiche (MF) 1.25

**Bellcomm, Inc.**

ff 653 July 65

FACILITY FORM 602	N67 12167	
	(ACCESSION NUMBER)	(THRU)
	206	1
	(PAGES)	(CODE)
	CR-80132	08
	(NASA CR OR TMX OR AD NUMBER)	(CATEGORY)

PROCEDURES FOR MANAGEMENT CONTROL  
OF COMPUTER PROGRAMMING IN APOLLO

Sept. 28, 1966

B. H. Liebowitz  
C. S. Sherrerd  
E. B. Parker, III

Work performed for Manned Space Flight, National Aeronautics and  
Space Administration under Contract NASw-417.

# BELLCOMM, INC.

## TABLE OF CONTENTS

### ABSTRACT

#### 1.0 INTRODUCTION

#### 2.0 MANAGEMENT CONTROL - GENERAL BACKGROUND

##### 2.1 THE COMPUTER PROGRAM CONTRACT END ITEM (CPCEI)

##### 2.2 THE PROGRAMMING PROCESS

#### 3.0 MANAGEMENT CONTROL PROCEDURES AND TOOLS

##### 3.1 THE CONTRACT TECHNICAL STATEMENT OF WORK

##### 3.2 STATUS AND PROGRESS REPORTING

##### 3.3 THE CPCEI SPECIFICATION

##### 3.4 CONFIGURATION MANAGEMENT OF CPCEI'S

##### 3.5 TESTING OF COMPUTER PROGRAMS

##### 3.6 COMPUTER PROGRAMMING STANDARDS

##### 3.7 DOCUMENTATION REQUIREMENTS

#### 4.0 SUMMARY AND CONCLUSIONS

#### 5.0 ACKNOWLEDGEMENTS

### REFERENCES

### FIGURE I

#### APPENDIX I - GUIDELINES FOR THE PREPARATION OF TECHNICAL STATEMENTS OF WORK FOR COMPUTER PROGRAM PROCUREMENT CONTRACTS

#### APPENDIX II - MILESTONE AND PROGRESS REPORTING REQUIREMENTS OF COMPUTER PROGRAMMING

#### APPENDIX III - PREPARATION OF CONTRACT END ITEM DETAIL SPECIFICATION (COMPUTER PROGRAM)

**BELLCOMM, INC.**

APPENDIX IV - CONFIGURATION MANAGEMENT OF COMPUTER PROGRAM  
CONTRACT END ITEMS

APPENDIX V - TESTING REQUIREMENTS OF COMPUTER PROGRAMMING

APPENDIX VI - DESIGN AND IMPLEMENTATION PRACTICES OF COMPUTER  
PROGRAMMING

APPENDIX VII - GLOSSARY

TABLE OF ACRONYMS



ABSTRACT

Formal techniques for management control are required to ensure the production and delivery of working, usable computer programs. A set of techniques, which are general enough to be applied across the variety of computer programming tasks found in the Apollo Project, are set forth in this report.

Because most of the large, mission-critical computer programs in Apollo are developed by outside contractors, various aspects of the user-contractor relationship are given particular attention. However, in substance, the methods described herein are also applicable to in-house efforts.

The techniques are intended for use by an agency responsible for specifying the requirements for a program, and then monitoring and controlling the efforts of the contractor (or in-house group) who will produce the program. Specifically the report presents:

- a. A suggested format and content for the detailed computer program specification. The first part of the specification describes the performance requirements for the computer program. It serves as the basis for the contractor's design efforts, and as the basis for evaluating the adequacy of the completed program. The second part of the specification is a complete description of the finished computer program.
- b. Configuration management procedures by which change control can be effected, technical reviews of the contractor's design can be made, and program and specification status can be documented.
- c. Suggested milestones by which the progress of the task can be evaluated, and formats for periodic reports that a contractor must submit to document the status of the task.
- d. Recommended tests for verifying the design and implementation of computer programs. Included are requirements for test plan and test specification documents which are essential to the testing process.
- e. Guidelines for the development of programming standards to guide the activities of programmers so that the individually produced components in a large system will work together.

**BELLCOMM, INC.**

- f. For those efforts which are formally contracted, guidelines for producing a Statement of Work which will specify the technical and management control requirements for the task.

# BELLCOMM, INC.

## PROCEDURES FOR MANAGEMENT CONTROL OF COMPUTER

### PROGRAMMING IN APOLLO

#### 1.0 INTRODUCTION

Digital computers and their associated computer programs are used more in the Apollo Program than in previous NASA space efforts. This is particularly true in mission support applications, in which computers are used for equipment checkout, personnel training, mission control, and vehicle guidance and navigation.

The commitment to use computers in Apollo came at a time when formal and proven management techniques for the control of computer program production were either unknown or undocumented. Previous experience in government and industry had indicated that ineffective management of computer programming efforts could lead to slipped schedules, overrun costs, and unusable products.

The Apollo Program Office (APO), aware of these problems, initiated a study of management control tools which might reduce their occurrence in the Apollo project. The study was directed towards the production of tools to assist technical personnel within using agencies in specifying the requirements for, and controlling the development of, computer programs. In the course of the study, a survey was made of management control tools under development in using agencies within the NASA Centers, other government agencies, and industry.

Specific attention was given to the problems of controlling contractor efforts, since most computer programs critical to the support of Apollo missions are developed by outside contractors. To allow for uniform management procedures in systems which contain hardware and computer program elements, procedures already in use for hardware development were modified as appropriate for computer programming efforts. The tools developed in the study were made general enough to apply to the diverse applications of the many NASA agencies that use computer programs rather than being tailored to any single effort. Specific application of the techniques is left to the responsible personnel in the using agencies.

Because of this generality and despite the use of specific NASA terminology, the tools and procedures can be applied to most programming efforts large enough to require formal management procedures. They apply to situations in

which a user has the responsibility for setting requirements for a computer programming effort and then monitoring the efforts of a developmental organization in meeting those requirements. Thus, used selectively, the techniques apply to in-house efforts as well as those tasks for which a well defined contract exists.

This report presents under one cover the results of the study, some of which have been or will be incorporated in Apollo Program and Office of Manned Space Flight documents (2), (3). Topics covered include: change control, specification writing, program documentation, work statement writing, testing requirements, programming standards, and reporting requirements. The bulk of the material is presented in appendices, each of which contains a detailed description of a specific management control tool or procedure. The body of the report contains a brief summary of the management control tools, the way in which they interact, and defines certain key terms and concepts used in the appendices.

## 2.0 MANAGEMENT CONTROL - GENERAL BACKGROUND

Past experiences in computer programming tasks, especially those in which the associated computer is part of a much larger system with hardware and personnel interfaces, have indicated a need for effective management control procedures.\* In some efforts the ultimate users of a program have failed to adequately define requirements, control changes, insist on adequate documentation, establish formal acceptance procedures, or monitor progress. Evasion of these responsibilities has resulted in overrun costs, slipped schedules, unusable programs, and programs that only partially do the required job.

"Management control" is defined here as those actions taken by a using agency to insure timely delivery of a working, and usable product by a contractor.\*\* Some of these actions may be delegated to the contractor or to another organization as deemed necessary by the responsible agency. To be effective, these actions must be based on an understanding of the

---

\*See, for example, reference document 1, Section 1.0, 2.0.

\*\*The term "contractor" is used throughout this report to denote the organization which is given the responsibility of developing the computer program. This organization may be in-house, in which case most of the techniques discussed are still applicable. The term "using agency" or "user" denotes the group that has ultimate responsibility for the use of the program.

characteristics of a computer program and the process required to produce it. Procedures, policies, and documentation formats can then be developed to facilitate the specification of the task, inform the user of the task's progress, give him control points, and ensure the delivery of the product he wants.

The essential unit of management control is the end item, the "thing", the contractor is to produce. In this report, borrowing from NASA terminology, the end item of a computer programming effort is called the "computer program contract end item" (CPCEI).<sup>\*</sup> A system containing a digital computer (a "computer-based" system) may contain several CPCEI's as elements.

The following paragraphs describe the characteristics of the CPCEI and the process required to produce it. Specific management control tools and procedures developed with these characteristics in mind are given in the appendices and outlined in section 3.0

## 2.1 The Computer Program Contract End Item (CPCEI)

A computer program is the ordered set of instructions and data required to satisfy a particular objective through the use of a digital computer. The computer acts, under the control of the instructions, upon the data to perform well-defined and logically related functions. The end product of the process required to produce a program is usually a punched or magnetic deck of cards, tapes or other physical media containing the ordered set in a form suitable for insertion into a digital computer.

A computer program contract end item (CPCEI) is defined here as a computer programming end product which has been identified as a unit, and to which a prescribed set of management control procedures is to be applied. Each CPCEI is composed of subprograms which are called computer program components (CPC's).

A computer programming application (e.g., automatic checkout of a launch vehicle, on-board guidance and navigation) may require the development of:

Operational program(s) which perform(s) the job the user wants done (e.g., trajectory calculation, generation of test stimuli).

---

<sup>\*</sup>CPCEI is derived from the term "contract end item" (CEI) used in Apollo and Air Force configuration management (2), (4).

The data base which contains all the static and dynamic data supplied to the operational program (e.g., launch site coordinates, mathematical constants).

Support programs used for the test and evaluation of the operational program (e.g., simulation, data recording, data reduction programs).

Utility programs which include all tools necessary for the generation of the operational and support programs (e.g., compilers, assemblers, monitors, debugging aids).

The programs and data are interrelated in their use, forming a "programming system". The selection of which programs are to be considered CPCEI's and thereby subject to the full spectrum of management control procedures is at the discretion of the procuring agency. Guidelines for this selection appropriate to Apollo programming tasks are given in the "Apollo Configuration Management Manual"(2) and repeated in Appendix IV of this report. The determination of the CPC structure within a CPCEI is usually made by the contractor during his design effort.

## 2.2 The Programming Process

The "programming process" is the totality of activities required to produce a working, documented CPCEI. The number and type of distinct activities required for the development of a CPCEI depends on the size and complexity of the task. However, all computer programming tasks, large and small, involve at a minimum the following sequence of steps in some form:

- a. Specification of the system of which the computer program is a part.
- b. Definition of the functions to be performed by the computer program.
- c. Design of the computer program including allocation of functions to the computer program components.
- d. Documentation of the computer program design.
- e. Coding and documentation of individual computer program components.
- f. Manual check of the coding.
- g. Computer check of the coding.
- h. Checkout of the computer program in the system.

- i. Updating the computer program and its documents per system tests results.
- j. Final evaluation of adequacy of the computer program design.
- k. Formal qualification of the final computer program against the design requirements.
- l. Modifications or extensions of the computer program due to error corrections or in accordance with changing user requirements and/or new installations.

These activities can be considered to take place in three phases: the definition phase, the acquisition phase, and the operational phase.\*

In the definition phase the system specification is developed and from this the CPCEI functions are defined (a and b above). In the acquisition phase, the CPCEI is designed, documented, coded, and then debugged in a simulated environment (c, d, e, f, g). In the operation phase the CPCEI is checked in a system environment, integrated with the other end items that comprise the system, updated as required, formally qualified (accepted), and used (h, i, j, k, l).

### 3.0 MANAGEMENT CONTROL PROCEDURES AND TOOLS

Specific management control tools are required in each phase to assure that products of the phase are usable, timely, and produced within predicted costs. To establish effective management control, the user (responsible agency) must:

- a. Make an early identification of the desired end product including a detailed specification of what the end product is to do.
- b. Require the contractor to develop a management plan for his own use.
- c. Develop, or have developed by the contractor, design standards and test procedures to assure the quality of the product.

---

\*These titles are used in the NASA and Air Force definition of the life cycle of a large system, which typically may have CPCEI and hardware end items as elements (2), (4). See Appendix IV, section 2.2 for a more detailed description of how the programming process is managed within the phases. Also see Appendix II, section 3.1 for a description of significant computer programming milestones in each phase.

- d. Review the contractors design to see if it meets the requirements of the detailed specification.
- e. Require the contractor to report regularly on the status of the task.
- f. Review and control proposed changes to the design and configuration of the end product.
- g. Require the contractor to document all computer programs before approving their release.
- h. Inspect and formally qualify all products produced by the contractor to insure conformance to the detailed specifications.
- i. Write an effective statement of work which formally establishes the technical and management control requirements for the task.

The appendices of this report contain detailed descriptions of management control tools for computer programming. Figure I depicts the relationship between these tools and the phases of the programming process. A guide to the appendices is given below.

### 3.1 The Contract Technical Statement of Work (Appendix I)

The purpose of the Statement of Work (SOW) is to specify the tasks to be performed and the items to be produced by the contractor, and to outline practices the contractor is to follow in the performance of the tasks. These requirements must be specified in sufficient detail to clearly define the contractor's obligations and to enable evaluation of his performance by the user. The SOW as defined here does not contain financial and cost information or other material of a legal nature. This is contained in the contract itself.

The SOW is usually a product of the definition phase and should be produced when all required computer programs have been identified and specified. In many situations, however, the contract is let before a clear definition of end items is available; usually this is done to get contractor assistance in the definition phase. In such cases the SOW should be written in as much detail as possible, with requirements appended as they become available. If, however, the procurement takes place at the end of the definition phase, the resultant requirements specification should be included as an exhibit to the SOW.

Appendix I contains detailed guidelines for the writing of a SOW, a suggested format for the SOW and an example of a SOW written in accordance with the guidelines. The format allows for incorporation of the management control procedures discussed in this report, either by direct inclusion or by reference to supporting exhibits.



### 3.2 Status and Progress Reporting (Appendix II)

Monitoring of the contractor's progress during the programming process is required to obtain schedule visibility from which realistic appraisals of project status may be made. Effective monitoring requires:

- a. An early identification of key milestone events to be monitored;
- b. PERT and Companion Cost Reports to indicate anticipated completion dates, milestone completions, critical path(s), and anticipated costs;
- c. Periodic technical reviews to assure that the contractor is developing something that the user really wants; and
- d. Periodic progress reports provided by the contractor to document the project status.

Appendix II contains suggested milestones to serve as the basis for technical monitoring, formats for progress reports, and a description of technical reviews that are required in the programming process. The content of this appendix may be used, with appropriate modification, as that portion of a SOW which establishes contractor reporting requirements.

The appendix may also be used as a planning guide in that it describes the activities of the programming process. This description can be used as a checklist for those personnel responsible for planning the activities of a programming task and the allocation of resources within its life cycle.

### 3.3 The CPCEI Specification (Appendix III)

The basis for producing a computer program is a detailed description of the functions it is to perform. Without this written description it is impossible to be sure that the contractor will produce a program the user really wants. The requirements must be documented before committing a contractor's resources to extensive design, coding and debugging in order to avoid extensive redesign costs at a later time.

An urgent need also exists for a detailed written description of the CPCEI design and its finally produced configuration. Technical descriptions are used when designing, correcting, modifying, and updating a computer program. Moreover, a computer program is generally used over a period of time and by personnel who did not design and develop the program.

Since it is commonly the case that during program usage the original programmers are not available, potential users usually will have no information concerning the program other than that conveyed through the detailed descriptions.

These two essential documents can be considered as a single CPCEI specification with two distinct parts, Part I containing the "design to" requirements and Part II containing the description of the "as produced" CPCEI. A contractual requirement for the production and maintenance of and the adherence to the specification can then be conveniently made.

Part I is a product of the requirements analysis in the definition phase, and is the technical requirements document used to contract for design and development of the CPCEI. The contractor must comply to the requirements of the Part I Specification. Compliance is determined by evaluation of qualification and other test records (see Appendix V). The Part I Specification describes in mathematical, logical, and operational language the detailed information necessary to initiate and carry out the design of a required CPCEI. In addition to providing the primary "design to" guide for the design and development effort during the acquisition phase, this document is the design requirements "baseline" which provides: (a) the basis for approval by the using agencies of the performance of the computer programs to be developed, (b) the instrument which defines all essential interfaces with other computer programs, equipments, and communications links, (c) the direct basis for the development of support documentation associated with operation and use of the computer program, and (d) the basic vehicle for configuration control of the design of the CPCEI throughout the acquisition and operational phases of the system cycle.

The Part II Specification describes the exact configuration of the CPCEI as produced by the contractor. It is produced in increments during the acquisition phase. Initial portions describe the overall CPCEI design, including the data base. The programming specifications for, and the detailed descriptions of, the CPC's are included as they are completed. The total document provides a complete technical description of the CPCEI functions, structures, operating environment, constraints, data organization, diagrammatic/narrative flows, and source statement/machine language listings. The integrity of Part II is established at the First Article Configuration Inspection (FACI - see Appendix IV) prior to its acceptance by the procuring agency. Acceptance of Part II is dependent upon the accuracy and completeness with which it describes both the gross and detailed structure of the computer program.

Appendix III contains a description of and format for the two parts of the CPCEI Specification. This appendix, which is a replication of Exhibit XVIII of reference document 2, can be included in a SOW as the standard for producing a specification. If the Part I Specification has already been produced by the user and is on contract, only the requirement to produce the Part II Specification need be included within the SOW. If, however, the contractor is to produce both parts of the Specification, the entire Appendix III can be included.

### 3.4 Configuration Management of CPCEI's (Appendix IV)

"Configuration management" is the management of the technical requirements which define all end items (elements) in a system. It is implemented through uniform procedures of technical specification, change control, and status accounting for the elements. These procedures are required to insure that each end item will do its job, will work with all other end items, will have adequate spares where necessary, can be easily identified and traced, and will be properly documented. Some of these procedures are directed to the management of the individual end items while others pertain to the management of the system as a whole.

In a large system, end items may be mechanical assemblies, electronic equipments (including computers) and computer programs (CPCEI's). The CPCEI is subject to the configuration management requirements that pertain to end items, including:

- a. CPCEI Identification. Each CPCEI is given a name and number so that it may be identified and retrieved as necessary. Each of its component parts (major sub-routines) gets a part number as do all tapes, disc packs, card decks, etc., which contain the program. These numbers are marked on the end item and its containers.
- b. CPCEI Specification. Each end item must have a specification which describes what the item is to do and then how it does it. All specifications must be properly marked by identification numbers. The detailed requirements for the CPCEI Specification are given in Appendix III.
- c. Technical Reviews. Formal configuration management reviews are required to control the design and development activities of the contractor, and to maintain general awareness of and control over the technical status of the effort. These reviews consist of:

- (1) The Preliminary Design Review (PDR). The PDR is held early in the acquisition phase. The compatibility of the contractor's initial design approach with the requirements of the Part I Specification is reviewed. Any discrepancies uncovered must be corrected by the contractor until an approved design approach is achieved.
  - (2) The Critical Design Review (CDR). The CDR is held after the detailed design of the CPC's is completed. It is essentially a review of the programming specifications developed for each CPC and is held prior to and prerequisite to the coding and debugging of the CPC's. The primary purpose of the CDR is to provide a check-point for the user on the contractor's detailed design. Several CDR's may be required since the individual programming specifications for a large effort may be produced in increments over a long period of time.
  - (3) The First Article Configuration Inspection (FACI). The finished CPCEI is inspected to determine if it has been properly packaged and marked according to configuration management standards, and the Part II Specification is reviewed for completeness.
- d. Change Control and Baseline Management. Formal procedures and mechanisms for the control of changes to specifications and CPCEI's are required to keep the cost of changes within reason, to maintain the integrity of the schedule, and to identify to all concerned the up-to-date configuration of the CPCEI. These goals must be achieved without overly inhibiting the contractor's ability to function in an R&D type environment.

Effective change control can be accomplished only if some "things" exist to be controlled. These things are called "baselines", which are in actuality the current specification of the CPCEI. In the computer programming process three baselines are established (see Figure I). First the specification for the system in which the CPCEI operates is established as the system requirements baseline. From this the design requirements of the CPCEI are developed and documented in the Part I Specification. The Part I Specification is established at the end of the definition phase as the design requirements baseline for the activities of the acquisition phase. Once so established, changes to it must be carefully controlled and documented to insure that the contractor is designing according to the current version of what the user wants. The completed

Part II Specification becomes the product configuration baseline at the end of the acquisition phase. At this time, changes to the product - the CPCEI - become subject to formal control. The Part II Specification must be kept up-to-date with CPCEI changes to insure that a timely description of the CPCEI exists throughout the operation phase. Formal change control is established by the actions of a Change Control Board (CCB) composed of personnel from the using agency. The CCB monitors and evaluates changes proposed by the contractor, evaluates changes which arise from the requirements of other end items, and transmits changes which affect other end items. In general, there are two types of changes:

Class I changes - those which affect performance and safety as defined in the established baselines, the cost or delivery date of the CPCEI, or interfaces with other end items.

Class II changes - those which are not Class I (e.g., simple error corrections).

All Class I changes require CCB approval before they can be implemented. Class II changes do not require CCB approval.

Proposed changes are evaluated against the baseline documents formally established during the programming process. The initial determination of what is a Class I change is left to the contractor under the cognizance of a user representative. The CCB, however, reserves the right to make the final determination of all classifications. All changes must be accounted for by the contractor, who must update all pertinent documentation. It is important to note, however, that until a baseline is formally established by user-contractor concurrence, changes to it are not subject to formal change control. For instance, a contractor is free to make program changes to a CPCEI prior to FACI without CCB approval, as long as the design requirements baseline is not affected.

- e. Configuration Accounting. Accurate records must be developed, maintained, and disseminated throughout the acquisition and operation phase to assure that the exact status of the CPCEI and its specifications is well known. These records include:

- (1) Specification change notices (SCN) which record changes to all approved specifications.
- (2) A specification change log which formally records all approved SCN's.

- (3) An end item configuration chart which identifies approved engineering change proposals (ECP's) incorporated in a particular version of a specification.
- (4) The configuration index which provides an official listing of the CPCEI specification and significant support documents.
- (5) The change status report which is used to provide the contractor and the procuring agency with a summary of the status of all ECP's.
- (6) The version description document which identifies the elements (tapes, card decks, etc.) of a released computer program.

The detailed requirements for configuration management in the Apollo program are given in reference document (2), and for the Air Force in reference document (4) as amended by a supporting exhibit for computer programs (5). Appendix IV of this report summarizes these requirements (for CPCEI's), and can be used as the basis for developing a SOW exhibit for those cases in which reference documents (2) and (4) are not applicable or appropriate.

### 3.5 Testing of Computer Programs (Appendix V)

Computer program tests are required to evaluate the adequacy of program design, verify that the program as implemented satisfies its programming specifications, and serve as the basis for acceptance of the program.

Several stages of CPCEI testing are involved in the computer programming process: subprogram unit testing, subprogram package (consolidation) testing, program qualification testing, CPCEI hardware integration testing, simulated environment computer-based system testing, and live environment computer-based system testing.

Testing must be performed in a logical, well-thought out manner, building up in increments from the debugging of the individual subprograms to the final qualification tests of the entire CPCEI integrated into the computer-based system. The sequence of tests should be determined in the initial design effort and documented in a test plan which is reviewed by the using agency. The test plan outlines the time phasing of the tests, and the resources required for and the functions to be demonstrated in each test.

Once the test plan is approved, the contractor must prepare detailed test specifications for those tests so identified in the plan. These specifications describe the procedures for running each test and the success criteria for each test. The user may review each specification, but usually approval will be required only for the qualification test specifications.

From the point of view of the user-contractor relationship, the most important of these testing stages is the CPCEI qualification testing. The qualification tests verify that the requirements in the Part I Specification have been satisfied. Qualification testing can be divided into a number of preliminary tests at the program, subsystem, and system levels; in addition, a final system test is always necessary as a basis for formal CPCEI acceptance.

Descriptions of the types of tests required for computer programming efforts, and formats for the test plan and for test specifications, are given in Appendix V. This appendix can be used as a model for that part of a SOW which requires the contractor to develop the test plan and perform the actions required to carry out the plan.

### 3.6 Computer Programming Standards (Appendix VI)

Standards and procedures are required in large programming efforts to assure that the many complex subprograms will interact properly, that a consistent set of preferred programming techniques is used by the programmers, and that the products of the programmers will be understandable to others. These standards and procedures should be developed and documented early in the acquisition phase for easy reference by the programmers. Standards usually have to be tailored to each application, although widely used standards are beginning to emerge in certain areas (e.g., ASA flow chart standards).

Appendix VI contains a checklist of computer programming standards which may be applicable to specific CPCEI development efforts. This checklist includes such areas as overall computer program design, component and subprogram design, coding, data structuring, computer program documentation, computer program application and maintenance, and product inventory. This appendix can be used as a guide both for writing that portion of the SOW which requires the contractor to produce a standards document, and for those personnel responsible for producing the standards.

### 3.7 Documentation Requirements

Documentation is required in a programming effort to provide communication of pertinent information between interested parties, and to provide the basis for understanding, using, and

modifying the end product. The required documents, many of which have already been discussed in this report, must be clearly identified in a SOW as a contractual requirement. A summary list of the documents required to support the programming process is given below. This list includes, where appropriate, a reference to the appendix in which the document is described in detail.

- a. Computer Program Contract End Item (CPCEI) Specification Parts I and II (Appendix III).
- b. Test Plan and Specifications (Appendix V).
- c. Progress and Status Reports (Appendix II, section 2.1).
- d. Documents used in the support of formal reviews (Appendix II, section 2.2 and Appendix IV, section 3.2.3).
- e. A management plan by which the contractor details his approach to the management of the task (Appendix I, page 10).
- f. Programming standards and practices manual (Appendix VI).
- g. Technical memoranda describing the development of the CPCEI.
- h. Change control documentation (Appendix IV, sections 3.2.2 and 3.3).
- i. Configuration and Specification status documentation (Appendix IV, section 3.3).
- j. Manuals to document those actions required for the operation, maintenance, and modification of the delivered computer programming products. These include:
  - (1) Operator Position Handbooks - defining the man-machine interfaces and describing proper operator responses to specific occurrences.
  - (2) Training Manuals - the formalized procedures for training and testing men in and associated with the system.
  - (3) Diagnostic Manuals - specifying tests and their specific results to be used to verify the operational status of the system, and to identify the source of and cure for trouble when it exists.



- (4) User's Manuals - describing how to load and operate the program, and permissible operational configurations and variations within these configurations.

#### 4.0 SUMMARY AND CONCLUSIONS

Effective tools for the control of computer programming effects are described in this report. They have been developed from the experience of using organizations both in NASA and industry, and from the adaption of tools used in the control of hardware development. The tools are applicable to in-house as well as contracted efforts, although the emphasis in the appendices of this report is on the user-contractor relationship.

Although the nature of computer programming tasks varies from job to job, enough similarities exist to apply a generalized management approach across a broad spectrum of applications. In a large organization or in a large project like Apollo, this:

- a. Establishes a standard terminology which eases communications among the various agencies involved;
- b. Facilitates the reporting of meaningful milestones for the computer programming efforts which can be monitored by the various levels of management;
- c. Reduces duplication of effort by providing technical personnel with formats for work statements, specifications, etc.;
- d. Encourages the use of good management techniques by providing a checklist of good practices to technical personnel; and
- e. Eases the burden on contractors by establishing a common approach to management control.

Within a single using organization many of the same advantages also apply with particular emphasis on (c) and (d) above.

For all efforts, effective management control requires a clear understanding between the user and the contractor or developing agency of what the end product is to do, when it must be available, and what the allowable costs are. Several steps are required to facilitate this understanding. Detailed performance specifications must be produced by the user to serve as a basis for contractor efforts. The user must either produce or require the contractor to produce programming design standards

to assure that all components of the program will fit together as planned. The contractor must periodically report the progress of the task so that status and potential problems will be visible to the user. The user must review the contractor's design on a periodic basis to determine if he is getting what he really wants. Concurrence based on these reviews must be established before the contractor commits himself too heavily in the next phase of the activity. Change control by the user is required to prevent unwanted deviations in the end product, and to keep up with changes in the other end items of the system. The user must require the contractor to develop a well planned sequence of testing to validate the implementation of the computer program. A formal acceptance test based on the requirements of the performance specification is necessary to assure that the program is doing the job it was intended to do before it is actually used. Various documents are required to describe the end product so that it can be developed in a coordinated manner, used, and modified as required. Throughout the entire programming process detailed planning is required to assure that contingencies can be met. For those efforts which are formally contracted, a definitive Statement of Work is required to formalize the management control and technical requirements of the task.

Procedures, policies and suggested document formats to achieve these goals are included in the appendices of this report.

## 5.0 ACKNOWLEDGEMENTS

Significant contributions have been made by organizations outside of Bellcomm, Inc. Computer Sciences Corporation (Washington Division), under contract to Bellcomm, Inc., provided considerable assistance in defining the computer programming process and in developing initial versions of Appendices I through VI. Special acknowledgements are also due to: the Air Force Systems Command - Electronics System Division for considerable coordination in developing the final version of Appendix III and for contributing many of the concepts incorporated in Appendix IV; System Development Corporation for participation in this same coordination; IBM (Federal Systems Division) for constructive comments; various agencies within the Apollo Program Office - NASA, particularly the Configuration Management Office and the

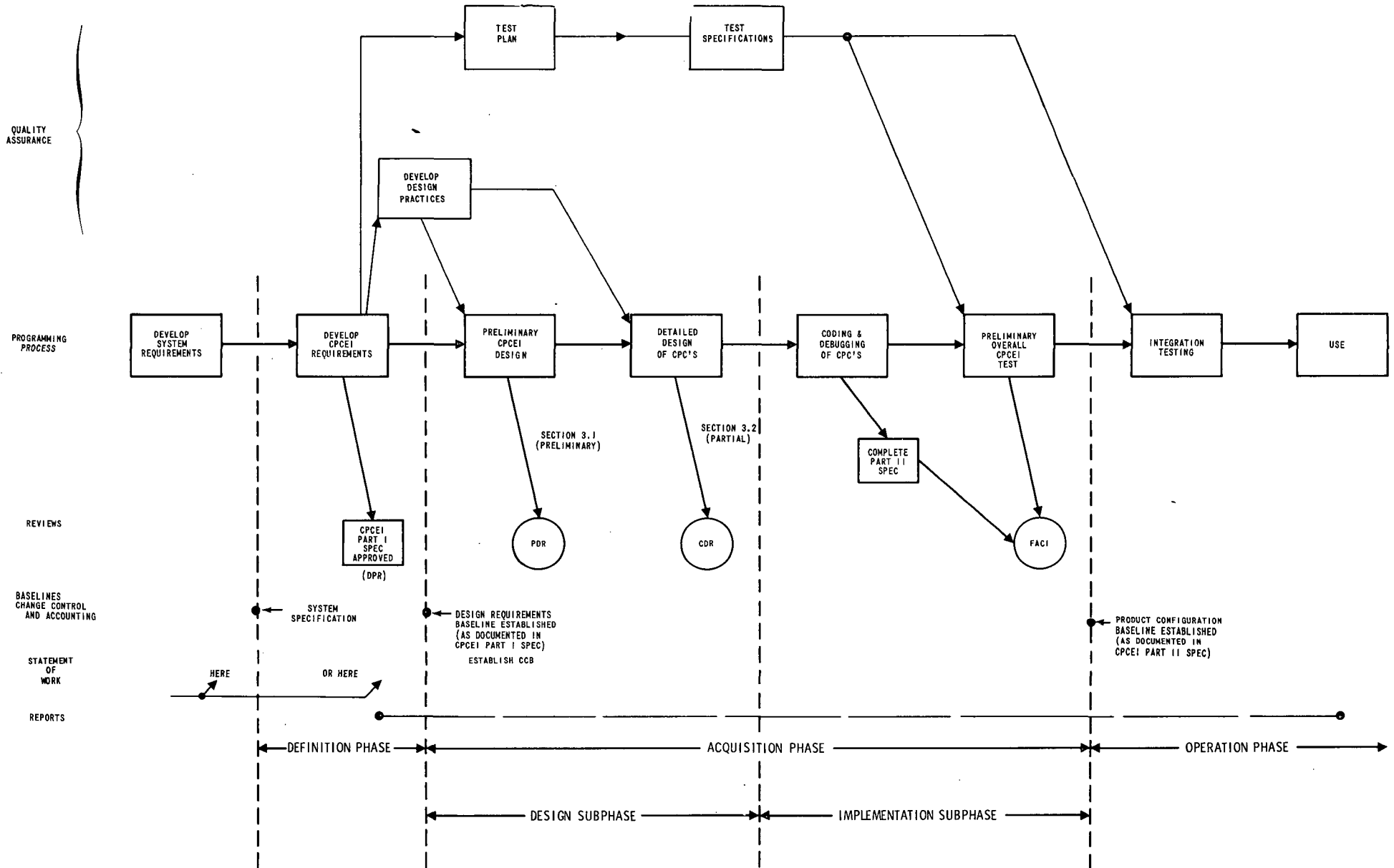


FIGURE 1 - MANAGEMENT CONTROL OF COMPUTER PROGRAMMING

**Page intentionally left blank**

**Page intentionally left blank**

Apollo Checkout Office; and numerous personnel from the using agencies at the Marshall Space Flight Center, Kennedy Space Center, and Manned Spacecraft Center for extensive comments on the content of the appendices.

*Burt H. Liebowitz*

B. H. Liebowitz

*Chris S. Sherrerd*

C. S. Sherrerd

*E. B. Parker III*

E. B. Parker, III

BHL  
1031-CSS-jab  
EBP

REFERENCES

1. Keese, W. M. et al, "Management Procedures in Computer Programming in Apollo - Interim Report", November 30, 1964, Bellcomm TR-64-222-1.
2. "Apollo Configuration Management Manual", NPC 500-1, Current Version.
3. "Statement of Work Handbook - Office of Manned Space Flight NASA" - NHB 5600, (to be published).
4. AFSCM 375-1, U. S. Air Force Systems Command - Current Version.
5. "Configuration Management Exhibit for Computer Programs" ESD Exhibit EST-1 Air Force Systems Command, L. G. Hanscom Field, Bedford, Massachusetts.

BELLCOMM, INC.

APPENDIX I

GUIDELINES FOR THE PREPARATION OF  
TECHNICAL STATEMENTS OF WORK FOR  
COMPUTER PROGRAM PROCUREMENT CONTRACTS

Note: The contents of the appendix were prepared for the Apollo Program Office specifically to be used in Apollo for the procurement of computer programs from outside contractors. Although NASA terminology is used and reference is made to NASA program control documents, the guidelines can be used in any computer program procurement.

TABLE OF CONTENTS

- 1.0 PURPOSE
- 2.0 SCOPE
- 3.0 APPLICABILITY
- 4.0 REFERENCE DOCUMENTS
- 5.0 PREPARATION OF THE SOW
  - SECTION 1 - "PURPOSE AND SCOPE"
  - SECTION 2 - "APPLICABLE DOCUMENTS"
  - SECTION 3 - "REQUIREMENTS"
    - PARAGRAPH 3.1 - "DELIVERABLE ITEMS"
      - PARAGRAPH 3.1.1 - "COMPUTER PROGRAM CONTRACT END ITEMS"
      - PARAGRAPH 3.1.2 - "DOCUMENTATION REQUIREMENTS"
      - PARAGRAPH 3.1.3 - "OTHER DELIVERABLE ITEMS"
    - PARAGRAPH 3.2 - "EXTENT OF WORK"
      - PARAGRAPH 3.2.1 - "CONTRACTOR TASKS"
      - PARAGRAPH 3.2.2 - "MAJOR MILESTONES"
      - PARAGRAPH 3.2.3 - "GOVERNMENT FURNISHED ITEMS"
  - SECTION 4 - "MANAGEMENT CONTROL"
    - PARAGRAPH 4.1 - "CONTRACTOR PROJECT MANAGEMENT"
    - PARAGRAPH 4.2 - "REPORTING REQUIREMENTS"
    - PARAGRAPH 4.3 - "CONFIGURATION MANAGEMENT"
  - SECTION 5 - "QUALITY ASSURANCE"
    - PARAGRAPH 5.1 - "TESTING REQUIREMENTS"
      - PARAGRAPH 5.1.1 - "GENERAL TESTING REQUIREMENTS"
      - PARAGRAPH 5.1.2 - "QUALIFICATION TESTING"
    - PARAGRAPH 5.2 - "METHODS AND PRACTICES"
  - SECTION 6 - "PREPARATION FOR DELIVERY"
  - SECTION 7 - "NOTES"
    - PARAGRAPH 7.1 - "SUPPLEMENTAL INFORMATION"
  - SECTION 8 - "EXHIBITS"
- ATTACHMENT I - SOW FORMAT
- ATTACHMENT II - EXAMPLE OF A SOW



GUIDELINES FOR THE PREPARATION OF TECHNICAL STATEMENTS  
OF WORK FOR COMPUTER PROGRAMMING PROCUREMENT CONTRACTS

1.0 PURPOSE

The purpose of this Appendix is to provide guidelines for preparing the Statement of Work (SOW) for computer programming efforts.

2.0 SCOPE

This appendix contains a description of, a format for, and an example of a Statement of Work for the procurement of computer program contract end items.

3.0 APPLICABILITY

These guidelines are applicable to NASA organizations responsible for the procurement of computer program contract end items (CPCEI). If the procurement is for a computer-based system, (both hardware and computer programs), these guidelines shall apply to the preparation of that part of the SOW that specifies the computer programming requirements.

4.0 REFERENCE DOCUMENTS

1. NPC 401, "NASA Policy and Procedures for Use of Contracts for Nonpersonal Services", April 21, 1964.
2. NPC 500-6, "Apollo Documentation Administration Instruction," August 1, 1964.
3. NPC 500-1, "Apollo Configuration Management Manual," May 18, 1964.
4. Appendix II of this report, "Milestone and Progress Reporting Requirements of Computer Programming."
5. Appendix V of this report, "Testing Requirements of Computer Programming."
6. Appendix IV of this report, "Configuration Management of Computer Program Contract End Items."
7. Appendix III of this report, "Preparation of Contract End Item Detail Specification (Computer Program)."
8. Appendix VI of this report, "Design and Implementation Practices of Computer Programming."

## 5.0 PREPARATION OF THE SOW

The purpose of the SOW is to specify the task to be performed and the items to be produced by the contractor, and to outline practices the contractor is to follow in the performance of the task. These requirements must be specified in sufficient detail to clearly define the contractor's obligations and to enable evaluation of his performance by NASA. The SOW is produced in the definition phase of the programming process (see section 2.2 of reference document No. 6 for a brief discussion of these phases).

Prior to preparation of the SOW, a search of the MSF program library shall be made to determine if any part or all of each CPCEI is currently available. If programs are available that will assist the contractor in his efforts, they shall be listed in Paragraph 3.2.3 of the SOW, "Government-Furnished Items."

The following sections of this appendix present guidelines for writing a CPCEI SOW. The section titles, given in quotation marks, refer to corresponding sections of a SOW (a sample format of which is given in Attachment I). An example of a SOW, written according to these guidelines for a hypothetical programming task, is given in Attachment II.

When detailed requirements documents are available they may be included as exhibits to the SOW. Such exhibits are binding upon the contractor. References are made in the guidelines as to when a supporting exhibit may be appropriate.

### Section 1 - "Purpose and Scope"

Section 1 shall describe the purpose and scope of the SOW, including a general description of the problem to be solved by the contractor's efforts.

### Section 2 - "Applicable Documents"

References shall be listed in this section. Each document will be identified and its role or usage described. The portions of a document that are applicable will be indicated in cases where the entire document is not applicable. Examples of documents which may appear on this list include the following:

1. CEI Specifications of the computer-based system to which this CPCEI is related.
2. Local computer programming standards specifications, if available and applicable.

3. NPC-500-1, "Apollo Configuration Management Manual".
4. Applicable progress reporting and testing procedures manuals.
5. NPC-500-6, "Apollo Documentation Administration Instruction".

### Section 3 - "Requirements"

This section shall identify and define the end and deliverable items to be produced by the contractor, the tasks to be performed by the contractor in conjunction with their production, and associated government-furnished items which are available for use or which must be used by the contractor.

The requirements described in this section should be as specific and quantitative as possible, so that the quality assurance procedures described in Section 5 may be used to verify that the requirements have been met.

#### Paragraph 3.1 - "Deliverable Items"

This paragraph identifies and specifies requirements for all end and deliverable items to be produced by the contractor.

##### Paragraph 3.1.1 - "Computer Program Contract End Items"

This paragraph shall specify the CPCEI's to be produced and delivered by the contractor.

A CPCEI is the end product of the programming process (e.g., a deck of cards, a magnetic tape, a permanently magnetized memory). Supporting documents for the CPCEI are not considered an integral part of the CPCEI but are separate deliverable items (see 3.1.2, below). Not all computer programs that are produced and delivered by the contractor are necessarily classified as CPCEI's. CPCEI's shall be those computer programs which are primary to the main purpose of a procurement contract, and other support computer programs which are considered sufficiently important by NASA to be separately identified as CPCEI's. Those programs which are not classified as CPCEI's will be designated as "Deliverable Computer Program Items (DCPI's)".

Each CPCEI shall be specified as follows:

#### a. CPCEI Identification

Each CPCEI shall be uniquely identified by a name and an identification number. (Exhibit X of reference document No. 3 provides for their identification).

b. CPCEI Definition

If at the time of preparation of the SOW the CPCEI Specification Part I\* exists, it shall be appended as an exhibit to the SOW. In this case, this paragraph shall contain a brief descriptive statement of the function of the CPCEI and will reference the aforementioned exhibit as the guiding specification.

If the CPCEI Specification Part I does not exist when the SOW is written, it will be prepared by the contractor as part of his work effort and, after approval by NASA, will define the CPCEI. In this case, a definition of the CPCEI, from which the contractor can prepare the CPCEI Specification Part I, must be included in the SOW. This definition should be accompanied by references to all requirements in existing higher-level specifications which the CPCEI must satisfy. The definition may be included in this paragraph or in a separate exhibit according to the volume of information presented.

Some important considerations in the CPCEI Definition are:

- (1) The functional requirements of the CPCEI.
- (2) The form of the CPCEI (e.g., card deck, magnetic tape, paper tape).
- (3) Constraints concerning the choice of the assembly/compiler language in which the CPCEI is to be written.
- (4) The interfaces to, and environment of, the CPCEI, such as:
  - (a) The computer(s) in which the CPCEI is intended to operate.
  - (b) I/O requirements.
  - (c) Timing considerations.
  - (d) Human engineering requirements.

The above list includes major elements that are required as part of the CPCEI Specifications.\* If requirements relating to critical aspects of the CPCEI are not available at the time of issuance of the SOW, this paragraph shall specify responsibility for

---

\*See reference document No. 7 for a complete description of the Part I Specification.

provision (e.g., if the computer has not been selected, the contractor may be assigned that responsibility, subject to NASA approval of the selection).

#### Paragraph 3.1.2 - "Documentation Requirements"

This paragraph identifies all documents required from the contractor during the contract period. All such documents shall be prepared in accordance with Document Requirements Descriptions to be specified by the NASA procuring agency (see reference document No. 2). These documents shall be identified in a Document Requirements List prepared by the NASA procuring agency and attached to the SOW as a supporting exhibit.

Deliverable documents generally can be classified as follows:

- a. CPCEI Specifications - describing the design of test requirements for, and use of an end item. Formats and descriptions of these documents are given in reference document No. 7.
- b. Technical Descriptions of DCPI's - describing the design of, test plans for, and use of deliverable computer program items (computer programs that are not CPCEI's - see paragraph 3.1.3). The configuration management and documentation requirements set forth in reference document No. 6 may be invoked for DCPI's in part or in full, at the discretion of the procuring agency.
- c. Test Specifications - describing the detailed testing to be performed on the CPCEI's and DCPI's. Standards for these documents are given in reference document No. 5.
- d. Reports - describing the status and progress of the contractor's activities, including fiscal and technical progress. Formats and contents of reports are given or referenced in reference document No. 4.
- e. Operator Position Handbooks and User's Manual - describing the human requirements and actions for operation of the computer programs.
- f. Management Plans - describing requirements to be imposed on the contractor management, timing, manpower, and administrative schedules.

#### Paragraph 3.1.3 - "Other Deliverable Items"

This paragraph lists deliverable items which are neither documents nor CPCEI's. These items, which are not primary objects

of the procurement, are essential to the use and/or maintenance of the CPCEI(s). Some examples are compilers produced for a CPCEI, special loaders, and simulation programs required for the testing of a CPCEI.

A deliverable computer program is defined in the SOW in the same manner as a CPCEI (see paragraph 3.1.1). Each deliverable item is identified, with available specifications referenced as supporting exhibits.

Determination of the extent of the contractor's obligation, for each DCPI, to meet the provisions set forth in Section 5.0 ("Quality Assurance") of the SOW is at the discretion of the procuring agency. Any relaxation of standards and requirements for a DCPI shall be clearly indicated in this paragraph.

#### Paragraph 3.2 - "Extent of Work"

This paragraph describes in detail the extent of the work to be done. This description will be in terms of:

1. Contractor Tasks
2. Major Milestones
3. Government Furnished Items

#### Paragraph 3.2.1 - "Contractor Tasks"

This paragraph shall identify the various tasks that the contractor must accomplish. Each task will be defined in terms of its resultant CPCEI and/or DCPI.

All technical contractor functions will be listed and described. A task number shall be assigned to each separately identifiable task. This paragraph is used to describe such tasks as:

- a. Production of the CPCEI's and DCPI's specified in the SOW.
- b. Technical liaison with NASA and personnel of other contractors.
- c. Maintenance of CPCEI's and DCPI's for a specified time period.
- d. Special studies, such as mathematical analyses, specifications of future efforts, computer hardware analyses, system improvement recommendations, and information flow analyses.

Paragraph 3.2.2 - "Major Milestones"

This paragraph identifies the major milestones of the contracted effort. These major milestones are related to the various tasks, CPCEI's, and/or DCPI's as appropriate. Key events during software development such as item deliveries, design reviews, equipment availabilities, and delivery of documents, are included.\* Each milestone is specified by:

- a. Identification - Each milestone shall be identified by a name and a number.
- b. Definition - Each milestone shall be defined by a brief description of the activity with which it is associated.

The required completion date for each milestone shall be given in a supporting exhibit. These dates are given relative to the starting date of the contract period or other key milestones (e.g., "30 days after contract start," "60 days before launch date"). At a minimum, the delivery date for each end item is specified. For milestones which cannot be so specified, this paragraph indicates the time and manner in which such specification shall take place.

Paragraph 3.2.3 - "Government Furnished Items"

This paragraph lists NASA-furnished items available for contractor use, such as:

- a. Equipment: computers, display sources, communication lines, and accounting equipment.
- b. Computer Time: time on NASA-owned or furnished computers.
- c. Data and Specifications: NASA-provided data and systems specifications documents.
- d. Personnel: analysts, programmers, computer operators, and operational personnel.
- e. Facilities and Supplies: buildings, desks, office space, magnetic tapes, cards, etc.

---

\*A list of suggested key milestones, applicable to the programming process, is given in Reference document No. 4.

- f. Documents: key documents required during the programming process, including baseline documentation and system and mission descriptions, which already exist or are being produced by other contractors or NASA agencies.
- g. Computer Programs: compilers, assemblers, simulators, scientific programs, existing subroutines, etc.

Each NASA furnished item shall be identified by a name and number and a brief description. An availability date, a statement of NASA obligation (e.g., amount of computer time to be furnished, the location of the computer, the work shift), and a reference to descriptive documentation, shall be given when applicable. A clear distinction shall be made between items which are optionally available for use and those which must be used by the contractor.

#### Section 4 - "Management Control"

This section specifies requirements on contractor activities in order to permit effective management control of the computer programming procurement.

##### Paragraph 4.1 - "Contractor Project Management"

This paragraph shall:

- 1. Require the contractor to establish a management organization which will be responsible for the tasks and functions described in the SOW.
- 2. Specify the participation and location of key personnel.
- 3. Specify the location of the contractor's work force.
- 4. Require the contractor to submit, for NASA approval, a management plan. This plan shall include but not be limited to:
  - a. An identification and description of all subtasks required to satisfy the requirements of the SOW. Each subtask shall be directly correlated with a task specified in paragraph 3.2.1 of the SOW. Each description shall include criteria for demonstrating the degree of task accomplishment to permit management control of progress.
  - b. The assignment of responsibility for each subtask to the contractor's organizational elements.



- c. The allocation of resources, including manpower, equipment, and facilities, to each subtask.
- d. The policies, methods, and procedures to be employed in the accomplishment of each subtask.
- e. The number and level of technical competence of both professional and non-professional persons to be applied to the subtasks.
- f. A master phasing schedule setting forth major decision dates, beginning and ending points of major activities, and dates of the milestones listed in paragraph 3.2.2.
- g. An organizational chart showing the complete program management organization and its relationship to higher levels of corporate management and to the functional departments of the contractor company.

The submission date (relative to the start of activities) for this plan shall be specified in this paragraph.

Paragraph 4.2 - "Reporting Requirements"

This paragraph establishes a uniform and comprehensive system of progress and status reporting by the contractor. The paragraph:

- 1. Establishes the requirements for contractor reporting. A reference shall be made to a supporting exhibit for detailed reporting requirements.
- 2. Refers to the Document Requirement List (see paragraph 3.1.2) for an identification of the reports required from the contractor.
- 3. Defines major milestone reviews and NASA concurrence procedures. These reviews are referenced to the major milestones listed in paragraph 3.2.2.
- 4. Specifies PERT requirements.
- 5. Identifies special reports uniquely associated with a particular effort.
- 6. Requires the contractor to include in his overall management plan a reporting plan showing internal monitoring methods.

Detailed reporting requirements are given in a SOW exhibit. A basis for this exhibit is given in reference document No. 4, which contains reporting and review requirements applicable to computer programming contracts, and guidelines for the establishment of milestones to be monitored by NASA. The SOW reporting requirements exhibit shall contain these requirements plus other requirements deemed necessary by NASA for the particular effort.

#### Paragraph 4.3 - "Configuration Management"

This paragraph establishes configuration management of CPCEI's as a requirement on the contractor.

The contractor shall be required to conform to the requirements of configuration management as set forth in reference document No. 3.

In addition, this paragraph:

1. Identifies a preliminary baseline document for each of the CPCEI's.
2. Designates a cognizant NASA Configuration Management Office (CMO) and specifies contractor responsibilities in support of this office.
3. Specifies configuration management procedures, beyond those given in reference document No. 3, that may be required for the development of the CPCEI's of the particular programming effort (e.g., reviews, change control procedures, change accounting procedures, etc.)
4. Specifies any NASA configuration management procedures which shall apply to the development of each DCPI of the particular programming effort.

#### Section 5 - "Quality Assurance"

The provisions specified herein shall pertain to the qualification of CPCEI's and DCPI's. Included are testing requirements and the specification of acceptable design and implementation procedures and standards.

#### Paragraph 5.1 - "Testing Requirements"

Requirements for the verification of the design, performance, and programming of the CPCEI's and DCPI's are specified in this paragraph.

Paragraph 5.1.1 - "General Testing Requirements"

This paragraph establishes requirements for contractor testing of the CPCEI's and DCPI's exclusive of the formal qualification tests. The contractor shall be required to:

- a. Produce a test plan for each CPCEI and DCPI listed. The requirements for the CPCEI test plan are given in reference document No. 5.
- b. Produce test specifications for each test so identified in the test plan. The requirements for CPCEI test specifications are given in reference document No. 5. These test specifications shall completely describe at least those tests directly related to qualification of the CPCEI's and DCPI's.
- c. Perform, and report the results of, these tests.

Levels and hierarchies of testing appropriate to computer programming efforts, and descriptions of the test plan and the specification documents, are given in reference document No. 5. This document is a basis for supporting SOW exhibits on CPCEI and DCPI testing. It must be modified for each programming effort due to the job sensitivity of computer program testing. The overall purpose of testing shall be a demonstration that the completed computer program satisfies the requirements and specifications to which it was produced.

Paragraph 5.1.2 - "Qualification Testing"

This paragraph identifies all computer programs which will be subject to formal qualification test procedures. At a minimum this shall include all CPCEI's.

Specific management requirements for qualification testing shall be given. All responsibilities for the design and operation of the tests shall be specified (e.g., "NASA will provide qualification test specifications 30 days before the scheduled test date.", or "The contractor shall be responsible for implementing the tests.").

Detailed technical requirements for qualification testing shall be given in the Part I Specification which is included as a supporting exhibit to the SOW. A discussion of qualification testing is given in reference document No. 5.

The formal acceptance of a computer program shall be dependent on the fulfillment of all quality assurance requirements specified within the SOW, as well as on the appropriate administrative and budgetary aspects of end item procurement.

Paragraph 5.2 - "Methods and Practices"

This paragraph specifies standards and procedures to be followed in the design and implementation of computer programs, concerning such aspects as:

- a. Modularity in design of the logic and packaging structures of the computer programs.
- b. Data structures and table look-up techniques.
- c. Subprogram interfaces and subroutine hierarchies.
- d. Executive control portions of computer programs.
- e. Coding language selection and/or development.
- f. Automatic hardware maintenance of computer programs.
- g. Input-output and man-machine interfaces.
- h. Installation adaptation techniques.
- i. End item maintenance.
- j. Flow chart formats.
- k. Change control procedures (contractor's internal mechanism).
- l. Programmer documentation procedures.

The standards utilized in these areas are job sensitive. If these standards are not specified completely in the SOW, this paragraph shall specify contractor responsibility in doing so. Any standards produced by the contractor shall be subject to NASA approval. An attempt should be made to provide in the SOW a minimal set of standards plus guidelines for the completion of the set. Details for these standards may be given in an exhibit supporting the SOW. See reference document No. 8 for a further discussion and checklist of such standards.

Section 6 - "Preparation for Delivery"

This section shall give the necessary packaging and marking instructions for each CPCEI, DCPI, and deliverable document. At a minimum the contractor shall be required to reserve, release, or deliver to NASA one master and one duplicate of each deliverable item.

Section 7 - "Notes"

This section shall include information which is stated for convenience only, and is not a part of the SOW in the contractual sense, i.e., it shall not include requirements which constrain design, development, and qualification of the CPCEI and require compliance by a contractor.

Paragraph 7.1 - "Supplemental Information"

Background information or rationales which will be of assistance in understanding the SOW itself may be included herein, (e.g., technical data, overall system descriptions, mission descriptions).

Section 8 - "Exhibits"

Requirements specified as exhibits are contractually a part of the SOW to the extent that they impose requirements on design, development and testing. This section shall include, but not be limited to, requirements which are:

- a. Bound separately for convenience, as in the case of a classified appendix or a large body of statistical data.
- b. Of a temporary nature, as in the case of an interim performance requirement for early test models of the CPCEI or a minimal set of programming standards.
- c. Sufficiently detailed to require separate publication as a self-sufficient document, e.g., reporting requirements and formats, Documents Requirements List.

The following exhibits are called for in various paragraphs above:

- a. Document Requirements List
- b. Milestone Completion Dates
- c. Reporting Requirements
- d. Testing Requirements
- e. Design and Implementation Practices
- f. Part I End Item Specification (for each CPCEI)

ATTACHMENT I

SOW FORMAT

- 1.0 PURPOSE AND SCOPE
- 2.0 APPLICABLE DOCUMENTS
- 3.0 REQUIREMENTS
  - 3.1 DELIVERABLE ITEMS
    - 3.1.1 COMPUTER PROGRAM CONTRACT END ITEMS
    - 3.1.2 DOCUMENTATION REQUIREMENTS
    - 3.1.3 OTHER DELIVERABLE ITEMS
  - 3.2 EXTENT OF WORK
    - 3.2.1 CONTRACTOR TASKS
    - 3.2.2 MAJOR MILESTONES
    - 3.2.3 GOVERNMENT FURNISHED ITEMS
- 4.0 MANAGEMENT CONTROL
  - 4.1 CONTRACTOR PROJECT MANAGEMENT
  - 4.2 REPORTING REQUIREMENTS
  - 4.3 CONFIGURATION MANAGEMENT
- 5.0 QUALITY ASSURANCE
  - 5.1 TESTING REQUIREMENTS
    - 5.1.1 GENERAL TESTING REQUIREMENTS
    - 5.1.2 QUALIFICATION TESTING
  - 5.2 METHODS AND PRACTICES

Table of Contents Contd.

6.0 PREPARATION FOR DELIVERY

7.0 NOTES

7.1 SUPPLEMENTAL INFORMATION

8.0 EXHIBITS

ATTACHMENT II

EXAMPLE OF A STATEMENT OF WORK

The following is an example of a Statement of Work produced according to the guidelines given in this appendix. It illustrates the manner in which these guidelines can be used; many of the clauses may be used in other Statements of Work with slight modifications.

The OLO project discussed in this example is completely hypothetical. It is not to be associated in any manner with the Lunar Orbiter program currently in development under control of NASA-LRC. However, it is representative of projects in which pre-contract computer program analysis has been accomplished, and the CEI Specification Part I has been completed.

The exhibits referenced in this example are not included in this appendix (except for the Document Requirements List).



Document No. 1735

Revision No. 0

Release Date June 15, 1965

STATEMENT OF WORK

GROUND SUPPORT COMPUTER PROGRAM

FOR

ORBITAL LUNAR OBSERVATORY

———

APOLLO PROGRAM

Approved By B. H. Liebestod

NASA

Approval Date June 15, 1965

TABLE OF CONTENTS

- 1.0 PURPOSE AND SCOPE
- 2.0 APPLICABLE DOCUMENTS
- 3.0 REQUIREMENTS
  - 3.1 DELIVERABLE ITEMS
    - 3.1.1 COMPUTER PROGRAM CONTRACT END ITEMS
    - 3.1.2 DOCUMENTATION REQUIREMENTS
    - 3.1.3 OTHER DELIVERABLE ITEMS
  - 3.2 EXTENT OF WORK
    - 3.2.1 CONTRACTOR TASKS
    - 3.2.2 MAJOR MILESTONES
    - 3.2.3 GOVERNMENT FURNISHED ITEMS
- 4.0 MANAGEMENT CONTROL
  - 4.1 CONTRACTOR PROJECT MANAGEMENT
  - 4.2 REPORTING REQUIREMENTS
  - 4.3 CONFIGURATION MANAGEMENT
- 5.0 QUALITY ASSURANCE
  - 5.1 TESTING REQUIREMENTS
    - 5.1.1 GENERAL REQUIREMENTS
    - 5.1.2 QUALIFICATION TESTING
  - 5.2 METHODS AND PRACTICES
    - 5.2.1 DESIGN AND IMPLEMENTATION PRACTICES
- 6.0 PREPARATION FOR DELIVERY
- 7.0 NOTES
  - 7.1 SUPPLEMENTAL INFORMATION

Table of Contents Contd.

8.0 EXHIBITS

- I. SYSTEM PERFORMANCE/DESIGN REQUIREMENTS SPECIFICATION, ORBITAL LUNAR OBSERVATORY (OLO), SS 107600C, DECEMBER 1, 1964.
- II. CPCEI SPECIFICATION PART I, OLO OPERATIONAL SUPPORT COMPUTER PROGRAM, CG 107652A, FEBRUARY 1, 1965.
- III. DCPI PERFORMANCE SPECIFICATION, OLO SIMULATOR, FEBRUARY 1, 1965.
- IV. REPORTING REQUIREMENTS FOR OLO COMPUTER PROGRAMS.
- V. TESTING REQUIREMENTS FOR OLO COMPUTER PROGRAMS.
- VI. DESIGN AND IMPLEMENTATION PRACTICES FOR OLO COMPUTER PROGRAMS.
- VII. EQUIPMENT CONFIGURATION DESCRIPTION, MSC IBM 7094 COMPUTER.
- VIII. DOCUMENT REQUIREMENTS LIST (DRL).
- IX. DOCUMENT REQUIREMENTS DESCRIPTIONS FOR DOCUMENT REQUIREMENT LIST.

## 1.0 PURPOSE AND SCOPE

The purpose of this Statement of Work (SOW) is to specify the tasks required of the contractor in the production of the computer programs for the Orbital Lunar Observatory (OLO) and to present the management and technical framework within which these tasks shall be performed. A brief description of the Orbital Lunar Observatory (OLO) is given in Section 7.1.

## 2.0 APPLICABLE DOCUMENTS

The following documents, of exact issue shown, form a part of this SOW to the extent specified herein. In the event of conflict between the requirements given in the documents referenced here and this SOW, this SOW shall supersede.

SS 107600C, "System Performance/Design Requirements Specification," Orbital Lunar Observatory (OLO), December 1, 1964

NPC 500-1, "Apollo Configuration Management Manual," May 18, 1964

NPC 500-6, "Apollo Documentation Administration Instruction," August 1, 1964

MIL-STD-129, "Marking for Shipment and Storage"

## 3.0 REQUIREMENTS

The contractor shall perform the following tasks and produce the following end items according to the provisions stated herein and in the applicable documents listed in Section 2.

### 3.1 Deliverable Items

The contractor shall produce and deliver to NASA the items described in the following subparagraphs. The collection of these items represents the minimum requirements for products to be delivered to NASA by the contractor. Other items produced by the contractor as a part of this contractual effort shall be considered NASA property. Delivery of items not specified in the following subparagraphs shall be made only upon request by NASA.

## 3.1.1 Computer Program Contract End Items (CPCEI's)

The following programs are designated as CPCEI's and are thereby subject to the NASA configuration management requirements given in NPC 500-1, "Apollo Configuration Management Manual" (see paragraph 4.3 of this SOW):

a. Orbital Lunar Observatory Operational Support Computer Program (OLOOS) - CPCEI No. 107652A

The performance design requirements for the OLOOS computer program are provided in CPCEI Specification Part I, #CG 107652A, dated February 1, 1965, and are attached as Exhibit II. A brief description of the OLOOS is provided herein for information purposes and does not supersede the requirements in the CPCEI Specification.

The OLOOS will perform real-time operational support to the OLO spacecraft. It will generate all command messages that are to be transmitted to the OLO and store and analyze all OLO status data that are transmitted to the ground stations by the OLO. It will compute the OLO trajectory and orbit and maintain the status of OLO on-board instruments. It will diagnose malfunctions that are detectable from transmitted data and will provide corrective commands where feasible. The OLOOS will provide the primary ground support to the OLO during its entire mission.

The OLOOS will operate within the IBM 7094 configuration described in Exhibit VII. It will be coded in FORTRAN IV language and will be permanently stored on magnetic tape.

b. OLO Data Reduction Computer Program (OLODR) - CPCEI No. 107653

The performance and design requirements for the OLO Data Reduction Computer Program shall be prepared by the contractor and submitted for NASA approval. These requirements shall be documented in a CPCEI Specification Part I in accordance with NPC 500-1. The System Specification for the OLO,

#55 107600C (attached as Exhibit I), and the following paragraph will, when combined, constitute the baseline for the establishment of the performance and design requirements.

The OLODR shall process, in non-real-time, the recordings of data transmitted from the OLO. The primary input shall be magnetic tape output of OLOOS. It shall provide hard-copy printout for detailed analysis and graphic displays for observation at selected consoles. The OLODR shall operate on the MCC IBM 7094 computer. It shall provide all operational personnel with the processed information required for the fulfillment of their duties as specified in the system specification. The OLODR shall be written in FORTRAN IV, operating autonomously under control of its own executive. It shall be fully contained on one magnetic tape and shall utilize a maximum of four tape drives during its operation.

### 3.1.2 Documentation Requirements

All deliverable documents for the contract are specified in the document Requirements List (DRL) provided in Exhibit VIII of this SOW. The Document Requirement Descriptions (DRD's) for each document on the DRL are compiled and are attached as Exhibit IX of this SOW. The administration of the documentation program for this contract shall be in accordance with NPC 500-6, "Apollo Documentation Administration Instruction," dated August 1, 1964.

### 3.1.3 Other Deliverable Items

The contractor shall deliver the following items to the procuring agency at the times specified in the master schedule:

#### a. Deliverable Computer Program Items (DCPI's)

DCPI's are defined as those deliverable computer programs that are not classified as CPCEI's:

- 1) Orbital Lunar Observatory Simulator (OLOS).  
The performance and design requirements for the Orbital Lunar Observatory Simulator

Computer Program are provided in the "DCPI Performance Specification," dated February 1, 1965, attached as Exhibit III. A brief description of the OLOS is provided below for information purposes. This information does not supersede the requirements in Exhibit III.

The OLOS will simulate, in real-time, the performance of the OLO spacecraft, including maneuver and processing functions. It will operate in conjunction with the OLOOS and will enable the complete simulated checkout of the OLOOS and related ground support operations prior to launch. The OLOOS will also be used as the basis for the development of a future extended OLOOS system and will operate within the IBM 7094 configuration referenced in Exhibit VII. It will be coded in FORTRAN IV language and permanently stored on magnetic tape.

- 2) Other Deliverable Computer Program Items (DCPI's). The contractor shall identify those additional computer programs not stated as requisites in this SOW that are required for the completion of tasks specified in this SOW (e.g., special one-shot debugging tools, subprogram data reduction programs, executive program for subprogram testing, data generators). The procuring agency will approve the development of these computer programs or will provide alternative solutions that are deemed to be in the best interest of the government. All computer programs of this category that are approved shall be designated as Deliverable Computer Program Items (DCPI's).

For the DCPI's, the contractor shall use his internal documentation control or establish a method with the procuring agency. The method used shall be based upon the principles of configuration management as set forth in "Apollo Configuration Management Manual," NPC 500-1, dated May 18, 1964. Specifically, the documentation control shall provide for the identification, control and accounting of the configuration of each DCPI.

## b. Simulation Tapes

The contractor shall deliver to the procuring agency the simulated data tapes called for in test specifications and used in actual tests of the OLOOS. These tapes shall be clearly labeled and referenced to the appropriate test specifications. A symbolic listing of the contents of each tape shall be included.

3.2 Extent of Work

The contractor shall perform the tasks that are necessary to provide the procuring agency with the deliverable items stated in 3.1. In conjunction with this requirement, the following specific efforts are identified.

## 3.2.1 Contractor Tasks

The contractor shall perform the following tasks:

## a. Computer Program Development

The contractor shall design, code, and test the CPCEI's and DCPI's required by this SOW. This shall include the preparation of the documents listed in the DRL, Exhibit VIII. The milestones in 3.2.2 present a list of events associated with this task.

## b. Computer Program Maintenance

The contractor shall provide a level-of-support effort to maintain and modify the accepted computer programs for a period of one year after CPCEI acceptance. This support shall allow for minimal retrofit to the CPCEI's. During this time period, the contractor shall conduct a training program for NASA personnel in the use and maintenance of the delivered computer programs.

## c. Special Study for OLO II

The contractor shall conduct a detailed study and shall make recommendations in a study report concerning future extended OLO computer program requirements. This study shall be initiated after the acceptance of all CPCEI's and will be based upon a statement of requirements to be provided by NASA.



### 3.2.2 Major Milestones

As a minimum, the contractor shall identify the following milestones in the preparation of a master schedule and PERT network. The DRL (Section 3.1.2) provides schedule constraints upon the delivery of specifications. In addition, all CPCEI's shall be delivered for qualification testing within eighteen (18) months from start of contract. The contractor shall comply with these constraints in the generation of the master schedule in the Contractor's Management Plan.

#### a. Computer Program Performance/Design Requirements

- 1) CPCEI Specification Part I OLODR--complete
- 2) Performance Specifications--complete for each DCPI
- 3) OLOOS Definition Phase Review (DPR)--complete
- 4) OLOOS and OLODR Test Plan--complete

#### b. Computer Program Design

- 1) CPCEI Specification Part II, Section 3.1 OLOOS--complete
- 2) CPCEI Specification Part II, Section 3.1 OLODR--complete
- 3) Computer Program Design Specifications--complete for each DCPI
- 4) OLOOS Preliminary Design Review (PDR)--complete
- 5) OLODR PDR--complete

#### c. Computer Subprogram Design

- 1) CPCEI Specification Part II, Section 3.2, OLOOS--complete
- 2) CPCEI Specification Part II, Section 3.2, OLODR--complete

- 3) Computer Program Component Design Specifications--complete for each CPC
- 4) Critical Design Review--complete for each CPC

d. Computer Program Test Specification

- 1) OLOOS Computer Program Integration Test Specifications--complete
- 2) OLODR Computer Program Integration Test Specifications--complete
- 3) OLOOS Qualification Test Specifications--complete
- 4) OLODR Qualification Test Specifications--complete
- 5) DCPI Qualification Test Specifications--complete for each DCPI

e. Computer Subprogram Coding/Testing

- 1) OLOOS Computer Program Component Coding and Debugging--complete for each CPC
- 2) OLODR Computer Program Component Coding and Debugging--complete for each CPC
- 3) DCPI Computer Program Component Coding and Debugging--complete for each CPC

f. Computer Program Testing

- 1) OLOOS Computer Program Preliminary Qualification Testing--complete
- 2) OLODR Computer Program Preliminary Qualification Testing--complete
- 3) DCPI Computer Program Preliminary Qualification Testing--complete for each DCPI
- 4) OLOOS FACI--complete
- 5) OLODR FACI--complete

- 6) OLOOS Integration Testing--complete
  - 7) OLODR Integration Testing--complete
  - 8) DCPI Integration Testing--complete for each DCPI
  - 9) OLOOS Computer Program Final Qualification Testing--complete
- g. Special Study for OLO II
- 1) Statement of Requirements--available
  - 2) Coordination Draft of Study Report--complete
  - 3) Final Study Report--complete
- h. Computer Program Maintenance
- 1) Training Curriculum and Schedule--established and approved
  - 2) Training Program Classes--initiated
  - 3) Training Program--complete

### 3.2.3 Government Furnished Items

NASA will furnish the following:

- a. Twenty-five (25) hours per week of prime shift computer time on the MCC IBM 7094 computer. The contractor may utilize time up to this maximum allocation according to his needs during the period of the contract.
- b. EAM and keypunching services for the production of end items required in this SOW.
- c. Applicable documents referenced in Section 2. of this SOW.
- d. Office space and supplies for the housing of all contractor personnel. The contractor shall utilize these on-site facilities in the performance of the tasks specified in this SOW.

- e. FORTRAN IV compiler for the IBM 7094. The contractor shall code his CPCEI subprograms using this compiler.
- f. Computer programs that are contained in the MSC computer program library. NASA will not be responsible for maintenance of these programs.

#### 4.0 MANAGEMENT CONTROL

Management controls are set forth herein to ensure that the deliverable items are produced efficiently, work progress and problem areas are adequately monitored, and technical changes are adequately controlled. Management control requirements for this contractual effort are described below.

#### 4.1 Contractor Project Management

##### 4.1.1 Management Organization

The contractor shall establish a management organization which will be responsible for the tasks and functions described in this SOW. The contractor's project organization shall be responsible to a resident project manager assigned exclusively to this project. The project manager shall have the necessary authority to be responsive to the technical direction of the Manned Spacecraft Center (MSC). The project manager shall designate persons within his organization who will serve as points of contact for all matters relating to particular functions or operations. The project organization shall assure that cost control, quality assurance, and technical and managerial coordination of the contract are adequately performed. The project manager and all project personnel will be located in the on-site office space provided by NASA at MSC.

##### 4.1.2 Management Plan

The contractor shall prepare and submit a management plan for NASA approval (see the DRL, Exhibit VIII). As a minimum, this plan shall contain the following:

- a. An identification and description of all sub-tasks required to satisfy the requirements of this SOW. Each subtask shall be directly correlated with a

task given in Section 3.2.2 of the SOW. Each description shall include criteria for demonstrating the degree of task accomplishment to permit visibility of progress.

- b. The time schedules for these subtasks.
- c. The assignment of responsibility for each subtask to the contractor's organizational elements.
- d. The allocation of resources, manpower, equipment, and facilities to each subtask and end item.
- e. By reference, the policies, methods, and procedures to be employed in the accomplishment of each subtask.
- f. The number and level of persons in professional positions and non-professional positions applied to the subtasks.
- g. A master phasing schedule setting forth major decision dates, beginning and ending points of major activities, and dates of the key events (including the milestones listed in 3.2.2).
- h. PERT networks for the subtasks and tasks, prepared in accordance with "NASA PERT and Companion Cost Handbook."
- i. An organizational chart showing the complete program management organization, and its relationship to higher levels of corporate management and to the functional departments of the contractor company.

#### 4.1.3 Contractor Working Relationships

To achieve the overall objectives of this contract, it is necessary for the contractor to work effectively with a number of other organizations which have major responsibilities in providing the ground support system for the OLO space flight. The contractor is required to provide for liaison with the integration contractor who is responsible for the OLO integration plan. MSC will take the necessary steps to ensure that the integration contractor is responsive to the contractor's needs in carrying out the

tasks herein defined. The contractor may be required to work with other organizations such as Goddard Space Flight Center, Jet Propulsion Laboratory, etc. MSC will make initial arrangements for working relationships with these parties. The contractor may then be required to deal directly on detailed technical matters with these organizations.

#### 4.2 Reporting Requirements

To ensure that both NASA and the contractor maintain a mutual awareness of the progress being made in the development of the CPCEI's and DCPI's, formal reporting and reviewing of contractor performance and contract status will be required. The contractor shall comply with the requirements of "Reporting Requirements for OLO Computer Programs" in Exhibit IV. The Document Requirements List (DRL) enumerates those reports which the contractor shall deliver to NASA. All delivered documents will be subject to approval by NASA. Monthly technical progress reviews and technical design reviews will be held for all CPCEI's and DCPI's in accordance with requirements in Exhibit IV.

The contractor shall produce and deliver to NASA a Reporting Plan (as part of his Management Plan) in accordance with the DRL. This plan shall describe procedures and responsibilities for internal reporting. It shall also include provisions for progress reports and technical reviews.

#### 4.3 Configuration Management

The CPCEI's identified in paragraph 3.1.1 shall be subject to formal configuration management procedures as specified in NPC 500-1 "Apollo Configuration Management Manual."

##### 4.3.1 Designation of Configuration Management Office

The cognizant NASA configuration management office for OLO will be the MSC Configuration Management Office (CMO). In accordance with NPC 500-1, NASA will establish a Configuration Control Board (CCB) for OLO within the CMO which will process engineering change proposals for the CPCEI's. The contractor shall review and assess the impact of all engineering change proposals which affect the computer program baselines.

##### 4.3.2 Baseline Establishment

The following baseline documents are hereby established for each of the CPCEI's listed in paragraph 3.1.1:

- a. OLOOS - CEI Specification Part I #CG 107652A, February 1, 1965 (see Exhibit II).
- b. OLODR - Section 3.1.1.b of this SOW and Exhibit I.

Each of these baseline documents represents an initial basis for the processing of engineering changes as specified in NPC 500-1. These baselines will change as the programming process develops.

#### 4.3.3 Change Control

The contractor shall establish an internal CMO and a CCB contained therein. The functions and makeup of the internal CCB shall be specified in the contractor produced Management Plan.

The contractor CCB shall operate according to the requirements of NPC 500-1 and shall contain one (1) representative designated by NASA. The CCB shall be responsible for processing all engineering change proposals that affect any of the CPCEI's identified in 3.1.1.

Change control procedures for DCPI's shall be processed by the contractor according to the DCPI Configuration Management Plan (see the DRL, Exhibit VIII).

#### 4.3.4 Configuration Status

As specified in the DRL, the contractor shall maintain and deliver to NASA periodically a configuration index which contains an up-to-date summary of the latest approved configuration for each CPCEI. The contractor shall also deliver to NASA periodically a Configuration Management Status Report which contains complete information relating to the current status of all baseline changes being processed by the contractor.

### 5.0 QUALITY ASSURANCE

The contractor shall develop and follow procedures that will assure high quality delivered items. Minimum requirements are described below.

## 5.1 Testing Requirements

The contractor shall test the CPCEI's and DCPI's at all stages of their development in order to increase reliability and ensure compliance with design requirements.

### 5.1.1 General Requirements

#### a. Computer Program Test Specifications

1. The contractor shall prepare test specifications for the CPCEI's of this SOW. They shall be prepared for the integration and qualification tests as defined in Exhibit V, "Testing Requirements for OLO Computer Programs," and shall be based on the requirements of Section 4 of each CPCEI Specification Part I. The test specifications shall be submitted to NASA for approval.
2. The contractor shall prepare (for internal use) test specifications for qualification tests for each DCPI. These qualification test specifications will set forth the criteria for qualification of the DCPI's.

#### b. Computer Program Testing

The contractor shall conduct tests of each CPCEI and DCPI in accordance with appropriate computer program test specifications.

### 5.1.2 Qualification Testing

The contractor will assist and support formal qualification testing of the OLOOS, OLODR, and DCPI programs in accordance with the appropriate qualification test specifications prepared by the contractor and approved by NASA. Successful completion of qualification testing is a prerequisite for the acceptance of each program.

## 5.2 Methods and Practices

### 5.2.1 Design and Implementation Practices

The contractor shall adhere to the requirements set forth in Exhibit VI with respect to design and implementation practices used in the production of



the CPCEI's and DCPI's identified in Sections 3.1.1 and 3.1.3. These requirements represent a minimal set identifiable at the beginning of contract activity. A set of complete standards shall be produced by the contractor for each CPCEI and DCPI. These standards shall represent good programming practices and serve the purpose of facilitating coordination of the efforts of the many programmers required for production of the CPCEI's and DCPI's. The standards document shall be approved by NASA before coding begins. It shall contain the minimal set specified in Exhibit VI and standards covering the following areas:

- a. Modularity in design of the computer programs.
- b. Data structures and table look-up techniques.
- c. Subprogram interfaces and subroutine hierarchies.
- d. Executive control portions of computer programs.
- e. Coding conventions and constraints.
- f. Automatic hardware maintenance computer programs.
- g. Input/output format design considerations.
- h. Installation adaptation design considerations.
- i. Design techniques for ease of maintenance.
- j. Flow chart formats.
- k. Programmer documentation procedures.
- l. Tape format design considerations.

## 6.0 PREPARATION FOR DELIVERY

The contractor shall package magnetic tapes in dustproof, waterproof, antimagnetic containers with sufficient thermal and vibration insulation material to ensure that the tapes will be undamaged and usable when received by NASA. Each tape will be clearly marked with the name and identification number of the computer program it contains.

Each report which is delivered shall be clearly and completely identified. Reports shall be packaged with sufficient insulating material to ensure that they will be undamaged when received by NASA.

Packages and containers for all deliverable items shall be marked in accordance with the requirements of MIL-STD-129.

#### 7.0 NOTES

The following paragraph is provided for information purposes only. It does not modify requirements and is not contractually binding.

#### 7.1 Supplemental Information - Orbital Lunar Observatory (OLO)

The OLO Spacecraft is a lunar reconnaissance vehicle, designed to orbit the moon at a distance of 100 miles and operate in this orbit for a period of six months. Information transmitted from this spacecraft will be used to determine an optimum landing site for the first APOLLO mission. The OLO will be launched in late 1966.

The OLO will contain two experimental packages. Experimental Package One consists of a 36-inch reflecting telescope with an assortment of filters. Telescopic images of the surface of the Moon will be photographed, and these photographs will be transmitted back to Earth periodically.

The OLO contains a command computer with a 500-word memory. A message of a maximum of 40 commands can be transmitted to the spacecraft and stored in the memory. These commands will then be executed by the spacecraft at the time specified in the command.

The command message will be generated by a ground-based support computer. This message can contain both spacecraft and experiment commands. In general, the message will be generated on the basis of data received from the OLO during the previous contacts.

The optical or experimental axis of the OLO will be the principal axis. Once the OLO is in orbit about the Moon, its optical axis will be pointed at the Moon's surface. A detailed description of the OLO is given in Exhibit I.

BELLCOMM, INC.

APPENDIX II

MILESTONE AND PROGRESS REPORTING REQUIREMENTS  
OF COMPUTER PROGRAMMING

TABLE OF CONTENTS

1.0 INTRODUCTION

2.0 FORMAL REPORTS AND REVIEWS

2.1 REPORTS

2.1.1 PERIODIC TECHNICAL PROGRESS REPORTS

2.1.2 PERT AND COMPANION COST REPORTS

2.2 TECHNICAL REVIEWS

2.2.1 PERIODIC REVIEWS

2.2.2 CONFIGURATION REVIEWS

2.2.3 TEST SPECIFICATION REVIEWS

3.0 COMPUTER PROGRAM DEVELOPMENT MILESTONES

3.1 SOME SUGGESTED MILESTONES

3.2 CRITERIA FOR REPORTING MILESTONE ACHIEVEMENT

REFERENCE DOCUMENTS

ATTACHMENT I

ATTACHMENT II

FIGURE I

MILESTONE AND PROGRESS REPORTING REQUIREMENTS  
OF COMPUTER PROGRAMMING

1.0 INTRODUCTION

Close monitoring is an essential aspect of management control of the computer programming process. The procuring agency must be aware of the status of the technical, financial and schedule aspects of the task, in order to solve potential problems.

To perform the monitoring function, the procuring agency must first develop a standard of expected progress. This standard is actually a complex and non-homogeneous set of information. It contains technical requirements (e.g., the expected performance of the computer program), estimates of expected costs and manpower utilization, and milestones indicating the expected completion dates for important activities. The contractor's actual progress, as determined from information collected in reports, reviews and informal communications, is measured against this standard. Any deviation from the standard will point out a potential problem area which should be investigated immediately.

This appendix presents suggested reviews and report formats for collecting this technical, fiscal, and schedule information from a contractor during the programming process. Also included are guidelines for establishing a set of milestones that can be used, by the procuring agency, as the basis for detailed monitoring of the task status.

2.0 FORMAL REPORTS AND REVIEWS

Formal reports and reviews should be specifically and explicitly defined in the contract Statement of Work (SOW). (See reference document 5). This should include as a minimum, periodic reports describing significant occurrences during the period covered, and reviews to evaluate the technical aspects of the contractor's activities.

2.1 Reports

Written reports provide information concerning significant achievements and problem areas. The format, content and frequency of publishing such reports should be specified in the SOW. Three types of reports are recommended: periodic technical progress reports, PERT and Companion Cost reports, and an end of contract report.

### 2.1.1 Periodic Technical Progress Reports

A periodic progress report contains a record of contractor activity during the period covered. It lists the major computer programming milestones which have been reached during the period, identifies and describes any current problem areas, and predicts progress for the next reporting period.

The frequency of these reports is dependent upon the size and criticality of the task. If many diverse individuals have immediate interest in the progress of the task, frequent reports (e.g., monthly) may be used to achieve coordination of effort. If interest is less immediate and/or less coordination is required, quarterly reports may be sufficient.

In lieu of the final periodic report the contractor should submit an end of contract report. This report is an overall record of accomplishments and furnishes a historical record of the computer program development effort.

Attachment I contains a suggested format for the content of periodic technical progress reports. Attachment II discusses the content of an end of contract technical report.

### 2.1.2 PERT and Companion Cost Reports

PERT charts depict the expected completion times of activities, the relationships between milestones, and show which activities are critical to meeting the required schedules. PERT can be used by the contractor as a planning tool, and by both the contractor and the procuring agency as a monitoring tool. An up-to-date PERT chart provides, on essentially a continuous basis, a snapshot of the task status and a prediction of future progress. The level of detail in a PERT chart is a function of the size and nature of the programming effort. This level is usually determined by the contractor; the procuring agency may, however, specify certain milestones that must be included.

The contractor should be required to submit periodic reports depicting milestone achievements, critical paths, expended costs and anticipated costs. For NASA these reports are prepared in accordance with the prevailing NASA-PERT and Companion Cost system (see reference document 2) which requires that:

- a. Schedule and milestone progress be reported bi-weekly on NASA Form 577.
- b. Cost data be reported either monthly or quarterly, as specified by the procuring agency, on NASA Form 533.

## 2.2 Technical Reviews

Technical reviews provide an opportunity for the procuring agency and the contractor to identify and solve technical problems. These reviews also provide a check that the computer program contract end item (CPCEI) being produced will both satisfy the user's requirements, and be compatible with the other elements of the computer-based system. Three types of technical reviews should be contractually required: periodic technical progress reviews, technical configuration reviews, and test specification reviews. Reviews of these three types should be combined when practicable.

### 2.2.1 Periodic Reviews

The purpose of technical progress reviews is to ensure regular periodic meetings between the procuring agency and the contractor. Progress and accomplishments are discussed, and discrepancies and technical problems that require procuring agency action are identified and resolved. These reviews are also used to respond to procuring agency requests for briefings on specific topics. Unless specifically delegated, the conduct of these reviews is the responsibility of the procuring agency.

### 2.2.2 Configuration Reviews

The purpose of technical configuration reviews is to verify, by inspection or demonstration, that the design and development of the CPCEI is being accomplished satisfactorily. The primary input to these reviews is the technical documentation produced by the contractor. Unless specifically delegated, the procuring agency is responsible for the conduct of these reviews.

At a minimum, four formal reviews are held (see below). The last three of these reviews are formal configuration management procedures and are described in detail in reference document 1. Additional reviews may be scheduled at the discretion of the procuring agency.

a. Definition Phase Review (DPR)

The DPR results in establishing the CPCEI Specification Part I as the design requirements baseline. It is held when all functions to be performed by the CPCEI have been delineated, and the preliminary CPCEI design requirements are available and documented in the Part I Specification. The DPR certifies that these design requirements are adequate in accordance with the computer-based system specification.

It should be noted that the DPR is not necessarily a formal review between the contractor and the procuring agency, since the Part I Specification is often produced within the procuring agency prior to engaging a contractor. However, the DPR is always held to certify the Part I Specification.

b. Preliminary Design Review (PDR)

The PDR is a formal technical review of the CPCEI overall design approach as formulated from the CPCEI design requirements baseline. It is held when the functional flow diagrams have been developed, the computer program components (CPC's) of the CPCEI have been identified, and an allocation of functions to the CPC's has been made. The PDR certifies that all functional interfaces between the CPCEI and the other components of its computer-based system have been considered and properly effected in the CPCEI design, and that the design approach fulfills the requirements of the Part I Specification.

c. Critical Design Review (CDR)

The CDR's are formal technical reviews of the completed detailed design of the individual CPC's as documented by block diagrams and detailed narrative descriptions. It is the purpose of a CDR to examine the design of a CPC or group of CPC's for reasonableness, compatibility, and completeness, and to assure that inter-CPC interfaces have been properly considered.



#### d. First Article Configuration Inspection (FACI)

FACI is a formal auditing of the CPCEI Specification Part II (see reference document 4) and the results of CPCEI preliminary qualification tests to assure that:

1. the Part II Specification adequately and correctly describes the finally implemented CPCEI;
2. preliminary qualification testing has been successfully completed; and
3. the CPCEI has been properly marked and identified.

The results of FACI are the formal establishment of the Part II Specification as the product configuration baseline and, if appropriate, the formal release of the implemented CPCEI for shipment to the user's facility for integration into the system of which it is a part.

#### 2.2.3 Test Specification Reviews (TSR)

The TSR's are reviews of test specifications prepared by the contractor for those tests identified as "critical" by the procuring agency. A specification describes the required resources for, and the goals, procedures, and success criteria of, a critical test. The reviews are held to certify the adequacy of the proposed tests in fulfilling their goals.

### 3.0 COMPUTER PROGRAM DEVELOPMENT MILESTONES

This section defines certain programming milestones which are appropriate for the monitoring of a contractor's activities. It also discusses criteria for determining completion of these milestones.

#### 3.1 Some Suggested Milestones

Milestones serve as checkpoints in the control of the programming process. A set of milestones which can be used is shown in Figure 1 and described below. Figure 1 presents a PERT-like description of a general model of the programming process. It depicts the relationship of the milestones

to each other and the three phases of the system life cycle as discussed in reference document 1, section 2.2. The milestones actually used in a programming effort must be tailored to the specific task.

The list given below is at the level of detail that should be monitored by a procuring agency. The planning for and monitoring of a programming task by the contractor will generally have to be done at a finer level of detail. In the following discussion no attempt is made to indicate the iterative nature of some of the milestones, although this is an important aspect of the programming process.

1. Computer-Based System Performance Requirements Document Available

This document or set of documents is the starting point for the programming process. It defines the functions of the computer-based system in which the CPCEI is to be used. It describes the operational capability, functional objectives, physical limitations, and other idiosyncracies of the system. It distinguishes between which computer-based system functions shall be implemented by hardware, which by software, and which manually. This document must be available prior to defining the performance and design requirements of the CPCEI.

2. CPCEI Performance and Design Requirements Document Complete

This milestone denotes the availability of the overall approved CPCEI design requirements, produced as the CPCEI Part I Specification. Part I describes the functional requirements of the CPCEI, including design constraints, interface requirements, and test requirements. It is produced in increments in the definition phase, and when approved (see 3 below), it becomes the design requirements baseline. (See reference document 1). This milestone is prerequisite to formal design activity.

3. Definition Phase Review

(See paragraph 2.2.2(a), above.)

4. Preliminary Design Complete

This milestone is reached when the contractor has formulated and documented a design approach which is complete enough to be reviewed by the procuring agency. This

approach includes at a minimum an identification of the structure of the CPCEI - i.e., the components of the CPCEI and the data flow between them - and an allocation of functions to each of these components. This information serves as the basis for the Preliminary Design Review and eventually becomes part (i.e., Section 3.1) of the Part II Specification.

5. Preliminary Design Review

(See paragraph 2.2.2(b), above.)

6. Test Plan Document Complete

This document defines the scope of testing to be performed on the CPCEI. It identifies the several tests to be performed, their purposes and the resources required for their development. The test plan designates which tests are to be considered critical and, therefore, require detailed test specifications. It is prepared concurrently with or immediately upon completion of the CPCEI design effort. (See reference document 3). This document is prerequisite to writing detailed test specifications.

7. Computer Hardware Characteristics Frozen

This milestone denotes the point in time when the computer hardware characteristics are specified, fixed, and available to the CPCEI development contractor. This generally must occur prior to detailed design of the computer program components.

8. Programming Standards Defined

Programming standards are the rules and procedures followed by programmers during the detailed design and implementation of the CPC's. They cover such areas as modularity in the logic and physical structure of the program, coding and flow chart conventions, data formats, etc. The definition of standards is considered complete when a standards manual is available for programmer use. (See reference document 6). A set of standards is usually prerequisite to the start of detailed design and coding.

9. Individual CPC Programming Specifications Complete

The detailed design of each CPC is documented in a programming specification that is used as the basis for coding the CPC. This documentation includes flow diagrams; a narrative description of the logic of the CPC, and a description of all CPC interfaces. Eventually it becomes a part (i.e., Section 3.2) of the Part II Specification.

10. Critical Design Reviews

(See paragraph 2.2.2(c), above.)

11. CPC Coding and Debugging Completed

This milestone reflects the completion of a CPC to the point at which it is ready to be tested with other CPC's. This includes completion of the code, successful assembly or compilation, and preliminary checkout by the programmer.

12. Utility and Support Programs Available

These milestones reflect the completion of the utility and support programs that must be especially developed for use with the CPCEI. They are achieved when these programs have been tested and are available for use. The number of such milestones depends on the nature of the programming task as well as the number of such programs required. If the computer manufacturer supplies all utility and support programs from an existing library, such milestones are not needed.

13. Test Specification Documents Complete

Each test designated as critical in the test plan is documented with a specification that describes the procedures, resources, and success criteria needed for the running and evaluation of the test. (See reference document 3). The document is considered complete when it is made available for review by the procuring agency. This review is a prerequisite to performing these critical tests.

14. Test Specification Review

(See paragraph 2.2.3, above.)

15. Preliminary Qualification Tests Completed

This milestone represents completion of the preliminary qualification tests as evidenced by the availability of documented test results. Preliminary qualification tests are usually performed in a simulated environment to determine if the CPCEI is ready to be incorporated into the computer-based system in which it is to be used. Where appropriate, these tests are run prior to transporting a program from a development facility to the user's facility.

16. First Article Configuration Inspection

(See paragraph 2.2.2(d), above.)

17. Personnel Position Handbooks and User's Manual Complete

The personnel position handbooks, if appropriate, describe the actions to be taken by personnel stationed at operator consoles during the operational use of the CPCEI. They are used primarily for training these personnel and as references during a mission. The user's manual describes how the CPCEI is to be loaded into the computer and operationally used.

18. Computer-Based System Integration Tests Completed

This milestone represents completion of the computer-based system integration tests and the availability of complete documentation of the test results. Integration tests are performed on the CPCEI after its incorporation into the computer-based system to determine if this incorporation is complete, and to determine if all interfaces between the CPCEI and other portions of the computer-based systems have been properly implemented.

19. Final Qualification Tests Completed

This milestone represents completion of the final qualification tests, and the availability of complete documentation of the test results. These are tests performed on the fully integrated computer-based system, for the purpose of formally assuring that the CPCEI and the other system elements are ready for mission use.

20. Post FACI Review

This review is held to verify that the formal qualification tests have been successfully completed, and to determine if all approved engineering change proposals (see reference document 1, section 3.2.2) have been incorporated into the CPCEI. Successful completion of this review is required before the CPCEI can be formally accepted.

21. Final Insertion of System Parameters

This milestone reflects the final updating of the data base to reflect the needs of a particular mission. For example, a guidance and navigation computer program may be updated just prior to launch with relevant trajectory parameters. This updating may occur, for certain efforts, earlier in the programming process than shown in Figure 1 (for example, prior to integration testing). Usually, a special test is run to determine if the data has been properly loaded.

3.2 Criteria for Reporting Milestone Achievement

Once milestones have been identified, it is essential to define criteria for measuring their completion. The following definitions are suggested:

A computer program document is considered complete when it has been reproduced and copies are available to the procuring agency.

A set of computer program documents is considered complete only when the complete list identifying the documents which compose the set has been agreed upon by the procuring agency and each document of that list is complete according to the above criterion.

A computer program component is considered to have been completely coded when it has been keypunched and satisfactorily assembled.

A computer program component is considered to have been completely debugged when, in the judgment of the contractor, it is ready to be combined and tested with other CPC's.

A CPCEI is considered completely checked out when, after all of the CPC's of the CPCEI have been completely debugged, the procuring agency and the contractor concur that the CPCEI is available for preliminary qualification testing.

A CPCEI and all documents pertaining to it is considered completely updated when all change orders pertaining to errors and design oversights discovered during testing have been completed in accordance with established configuration management procedures.

REFERENCE DOCUMENTS

1. Appendix IV of this report, "Configuration Management of Computer Program Contract End Items."
2. NASA PERT and Companion Cost System Handbook, latest issue.
3. Appendix V of this report, "Testing Requirements of Computer Programming."
4. Appendix III of this report, "Preparation of Contract End Item Detail Specification (Computer Program)."
5. Appendix I of this report, "Guidelines for the Preparation of Technical Statements of Work for Computer Program Procurement Contracts."
6. Appendix VI of this report, "Design and Implementation Practices of Computer Programming."



ATTACHMENT I

SUGGESTED FORMAT FOR CONTENT OF  
PERIODIC TECHNICAL PROGRESS REPORT

Introductory Material

1. Front Cover. The front cover contains the following information:

- Report Number
- Report Title and Project Name
- Contract Number
- Contractor's Name and Address
- Full Name and Address of the Procuring Agency
- Security Classification and other Related Notices when required
- Date of Report

2. Title Page. The title page (unnumbered) contains the same information as the cover. Approval signature blocks are also included.

3. Abstract or Summary. The abstract or summary is provided to enable a reader to quickly and easily obtain the main points of the report. Examples and comparisons should be avoided except where necessary to establish a basis for the work. The following information should be included:

- Object of the Report
- Scope of Work Covered
- Conclusions based on Findings
- Summary of Recommendations

4. Table of Contents

5. List of Illustrations

6. List of Tables

Main Body

The main body of the report consists of the following elements:

1. Introduction: The introduction contains the scope and purpose of the report, a general background of the project, and a breakdown of the associated functional subsystems. Also included in the introduction is a brief summary of work performed during the reporting period.
2. Discussion: The discussion section should be subdivided as required into applicable subjects. Each subdivision includes a detailed narrative and analysis of the work performed in that area. Also included is a summary of computer programming statistics (e.g., total number of coded instructions, and hours of computer time used for debugging), schematic diagrams, drawings, charts, tables, formulae, measurement procedures, test methods, and other supporting data relevant to the project not specifically required to be located in other sections of the report. It should be noted that negative aspects may be as relatively significant as positive aspects. Identification of new innovations and general advances in programming technology by contractors is of special interest, and is to be included.
3. Program for Next Reporting Interval:

This section describes the work planned for the next period in as much detail as possible. Tests to be performed and milestones to be met should be described, along with the plans for ensuring their timely completion.
4. Conclusions and Recommendations:

Conclusions and recommendations that can be logically drawn from an analysis of the work performed during the reporting period are presented. This includes appropriate statements concerning the effect of the results described in the "Discussion" section on the overall objective of the contract. It is equally important that both favorable and unfavorable conclusions be pointed out.

5. Bibliography: This section contains full citations to work referenced throughout the main body of the report, and references to closely related work, background material, and publications which offer complete information on all aspects of the work being reported. Specific reference should be given to papers published in scientific journals and periodicals which have been written as a result of work accomplished or findings relative to the contract.
6. Glossary: This section consists of a list of uncommon or new technical terms, and the definition of these terms.
7. Appendices: Related or additional material too bulky or detailed to be included in its entirety under the "Discussion" section of the report, but directly related to a complete understanding of the report, is included as appendices.

ATTACHMENT II

SUGGESTED FORMAT FOR CONTENT OF  
END OF CONTRACT TECHNICAL REPORT

Introductory Material

1. Front Cover. The front cover contains the following information:

- Report Number
- Report Title and Project Name
- Contract Number
- Contractor's Name and Address
- Full Name and Address of the Procuring Agency
- Security Classification and Other Related Notices when required
- Date of Report

2. Title Page. The title page (unnumbered) contains the same information as the cover. Approval signature blocks are also included.

3. Abstract or Summary. The abstract or summary is provided to enable a reader to quickly and easily obtain the main points of the report. Examples and comparisons should be avoided except where necessary to establish a basis for the work. The following information should be included:

- Object of the Report
- Scope of Work Covered
- Conclusions based on Findings
- Summary of Recommendations

4. Table of Contents

5. List of Illustrations

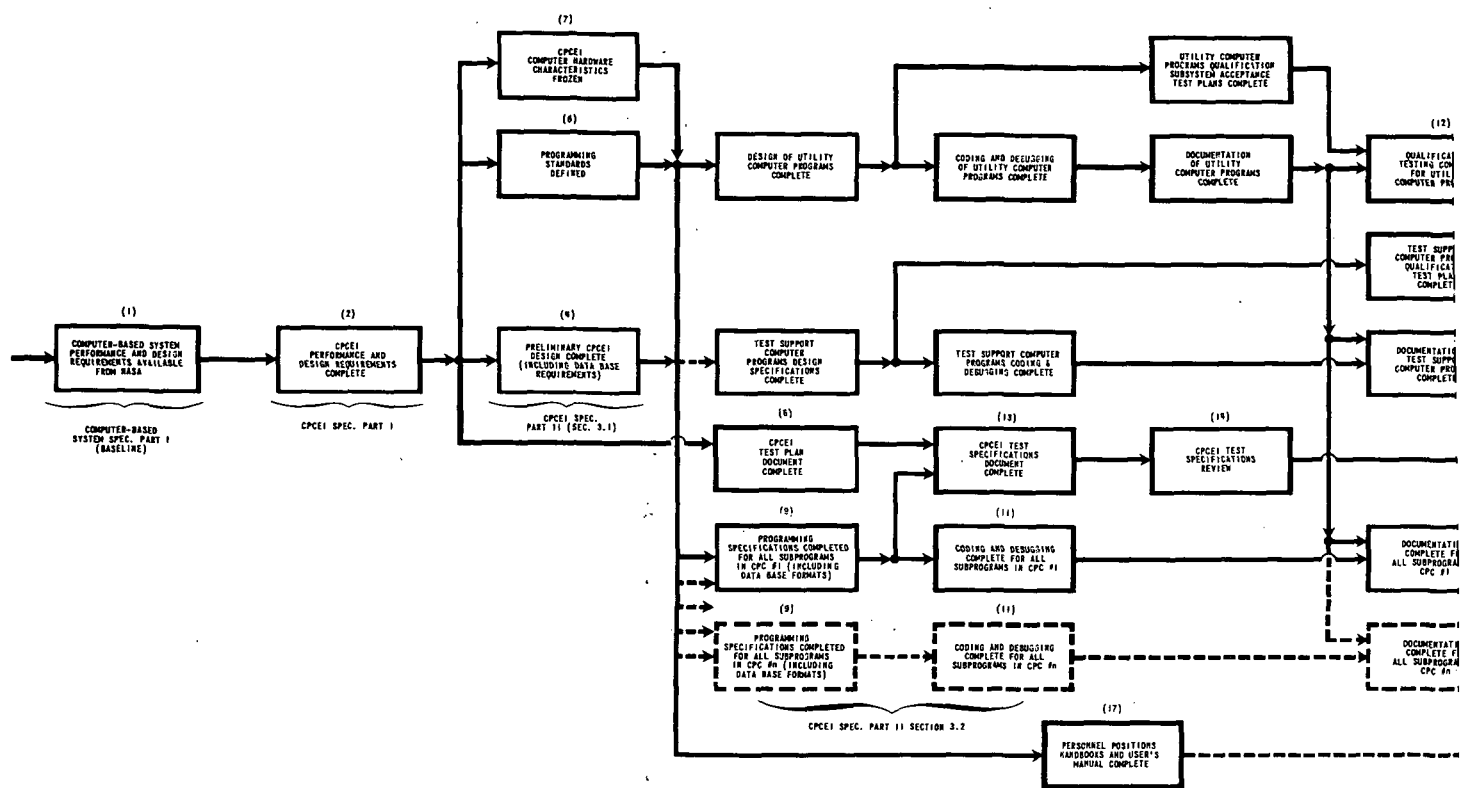
6. List of Tables

Main Body

The main body of the report consists of the same elements as the Periodic Technical Report (see Attachment I) with the exception of section 3, "Program for Next Reporting Interval", which is not applicable. While the same elements are covered, the scope of the items in each section must be expanded to cover the entire contract period. In addition, one appendix of the End of Contract Report must be a complete compilation of the related statistics as outlined below:

Statistical Summary Outline

1. Computer Program Statistics:
  - a. Total number of statements/instructions in both source and object language of each CPCEI.
  - b. Total number of pages of supporting documentation for each CPCEI (Initial copy only).
  - c. Total computer machine time in hours used for each CPCEI by computer type.
  - d. Total manpower by job classification used for each CPCEI.
  - e. Other statistics as deemed necessary by the procuring agency.



## PHASES OF THE PROGRAMMING PROCESS

## DEFINITION PHASE

## ACQUISITION PHASE

## DESIGN SUBPHASE

## APPLICABLE BASELINE

COMPUTER BASED SYSTEM  
CEI SPEC. PART I  
(ESTABLISHED EARLY IN  
THE DEFINITION PHASE)

ESTABLISH CPCEI SPEC. PART I (DESIGN REQUIREMENTS BASELINE)

(3) DPR

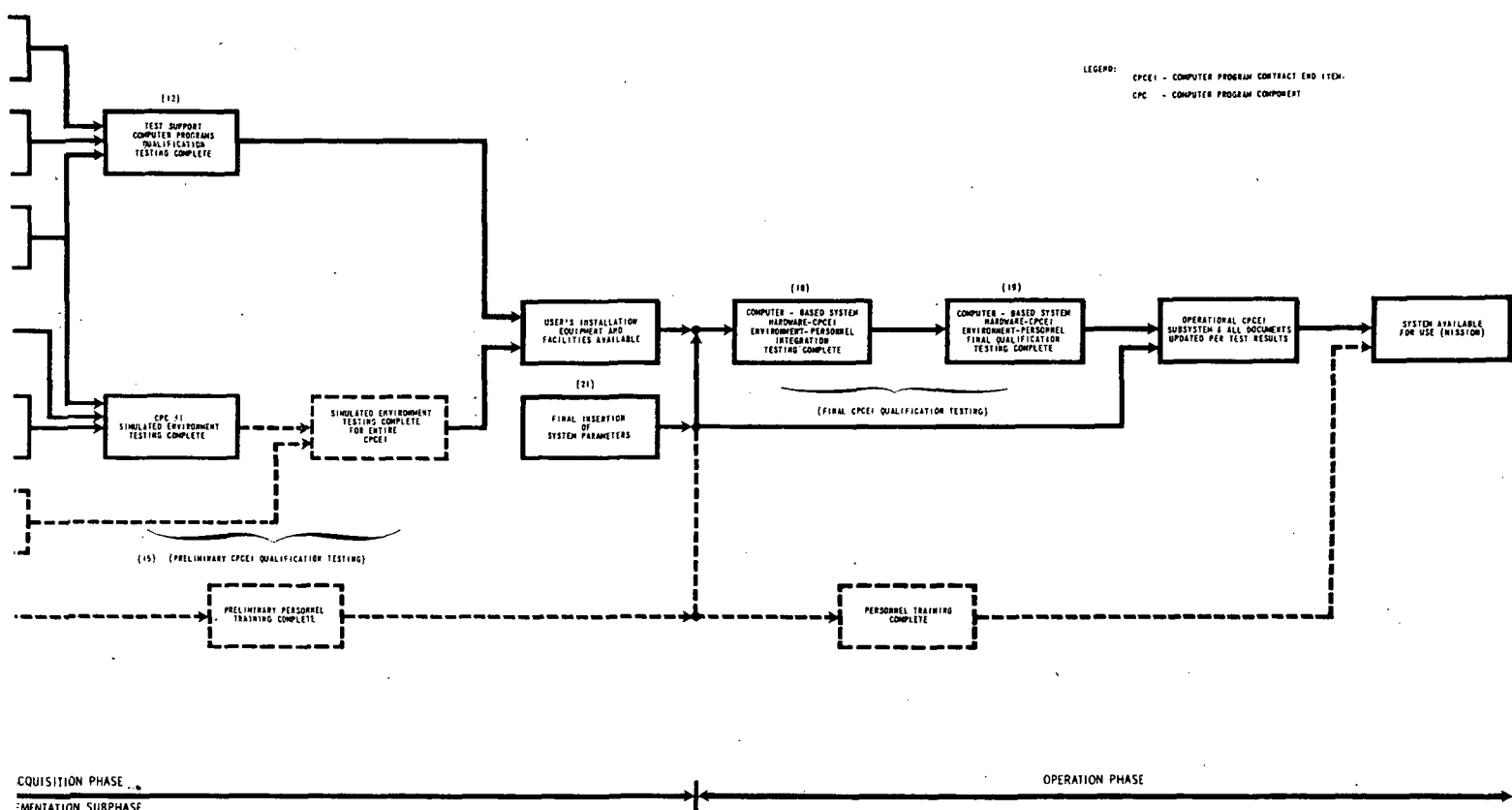
(5) PDR  
(PRELIMINARY DESIGN REVIEW)(10) CDR  
(CRITICAL DESIGN REVIEW)CONFIGURATION  
MANAGEMENT REVIEWS

[DEFINITION PHASE REVIEW]  
• REVIEW OF CPCEI PERFORMANCE  
AND DESIGN REQUIREMENTS  
• ESTABLISHMENT OF CPCEI PART I  
AS DESIGN REQUIREMENTS  
BASELINE

• CHECK OF COMPATIBILITY OF DESIGN EXCEPT AS  
DOCUMENTED IN CPCEI SPEC. PART II  
(SEC. 3.1) WITH CPCEI SPEC. PART I  
• REVIEW OF OVERALL PROGRAM FLOW CHART  
• VERIFICATION OF INTERFACE DEFINITIONS  
(COMPARE INTERFACE FORMATS IN CPCEI SPEC.  
WITH INTERFACE CONTROL DOCUMENTS FOR  
COMPUTER-BASED SYSTEM)

(ONE FOR EACH CPC OR GROUP OF CPC'S)  
• REVIEW OF ANALYTICAL DATA DEVELOPED DURING DESIGN PHASE  
(a) DERIVED EQUATIONS AND ALGORITHMS  
(b) TIMING AND THROUGHPUT STUDIES  
(c) SIMULATION AND TEST DATA  
REVIEW OF CPC INTERFACES  
APPROVAL OF DETAILED DESIGN OF CPC'S

COMPUTER PROGRAMMING PROCESS



ACQUISITION PHASE  
IMPLEMENTATION SUBPHASE

OPERATION PHASE

ESTABLISH CPCEI SPEC. PART II (CPCEI PRODUCT CONFIGURATION BASELINE)

(16) FACI

(20) POST - FACI REVIEW

(FIRST ARTICLE CONFIGURATION INSPECTION)

- VERIFICATION OF ACCOUNTING OF EVERY CPCEI COMPONENT (SUBPROGRAM AND PART (TAPE, CARD DECK, ETC.))
- AUDITING OF THE RESULTS OF PRELIMINARY QUALIFICATION TESTS AND THE COMPLETION OF ALL CHANGES RESULTING THEREFROM
- VERIFICATION OF COMPLETION OF ENTIRE PART II SPEC. AND ITS COMPATIBILITY WITH PART I
- ESTABLISHES CPCEI SPEC. PART II AS THE PRODUCT CONFIGURATION BASELINE
- VERIFICATION OF VALIDITY OF FINAL QUALIFICATION TEST PROCEDURES

- VERIFICATION OF COMPLETION OF FINAL QUALIFICATION
- VERIFICATION OF INCORPORATION OF CHANGES RESULTING THEREFROM
- FORMAL ACCEPTANCE OF THE FINAL IMPLEMENTED CPCEI

NOTE: TIME OF FACI

- IF CPCEI IS TO BE SHIPPED FROM DEVELOPER FACILITY TO USER FACILITY, FACI SHOULD BE HELD SUBSEQUENT TO PRELIMINARY QUALIFICATION TESTS AND PRIOR TO SHIPMENT
- IF CPCEI IS DEVELOPED AT USER FACILITY, FACI SHOULD BE HELD WHEN CPCEI SPEC. PART II IS COMPLETE AND PRIOR TO COMPUTER-BASED SYSTEM INTEGRATION AND FINAL QUALIFICATION TESTS

21. ①

②

21

BELLCOMM, INC.

APPENDIX III

PREPARATION OF  
CONTRACT END ITEM DETAIL SPECIFICATION  
(COMPUTER PROGRAM)

Note: This Appendix is a replication of Exhibit XVIII of NPC 500-1, May 1966. Although this was originally prepared for NASA usage, it is applicable to all computer programming efforts. A detailed description of the contents of a Part I Specification is given in pages 8-21; a detailed description of the contents of the Part II Specification is given in pages 22-33.



TABLE OF CONTENTS

- 1.0 PURPOSE
- 2.0 SCOPE
  - 2.1 CPCEI SPECIFICATION
  - 2.2 SPECIFICATION ADDENDA
- 3.0 APPLICABILITY
- 4.0 APPLICABLE DOCUMENTS
- 5.0 EXPLANATION OF TERMS
- 6.0 PROCEDURAL REQUIREMENTS
  - 6.1 COMPUTER PROGRAM CONTRACT END ITEM DETAIL SPECIFICATION
  - 6.2 ADDENDA TO COMPUTER PROGRAM CONTRACT END ITEM DETAIL SPECIFICATION
  - 6.3 DETAILED INSTRUCTIONS FOR THE PREPARATION OF THE CPCEI SPECIFICATION
    - 6.3.1 CPCEI PART I SPECIFICATION
      - SAMPLE FORMAT "A" (TITLE PAGE OF CPCEI SPECIFICATION)
      - SAMPLE FORMAT "B" (INTRODUCTORY PAGE OF PART I)
      - SAMPLE FORMAT "C" (GENERAL OUTLINE OF PART I)
    - 6.3.2 CPCEI PART II SPECIFICATION
      - SAMPLE FORMAT "D" (GENERAL OUTLINE OF PART II)
      - SAMPLE FORMAT "E" (INTRODUCTORY PAGE OF PART II)
    - 6.3.3 CPCEI SPECIFICATION ADDENDA
      - SAMPLE FORMAT "F" (ADDENDUM NOTICE)

PREPARATION OF  
CONTRACT END ITEM DETAIL SPECIFICATION  
(COMPUTER PROGRAM)

1.0 PURPOSE

The purpose of this Appendix is to provide instructions for the preparation of the detailed specification for each computer program contract end item (CPCEI) and for any necessary addenda thereto.

2.0 SCOPE

These instructions are applicable to the specification for end items categorized as "computer program". The instructions contained herein define the content and format for each of the two parts of the CPCEI detail specification. Also included are instructions for the preparation of addenda to the CPCEI Specification.

2.1 CPCEI Specification

The computer program contract end item (CPCEI) Specification is the detailed specification of computer programs which have been designated as end items by the procuring agency. The two parts to the CPCEI Specification are:

Part I - Performance and Design Requirements:

This part is used to specify requirements peculiar to the design, development, test, and qualification of the CPCEI. Part I is usually a product of the definition phase and is used to contract for the design and development of the CPCEI.

Part II - Computer Program Technical Description:

The Part II Specification is the detailed technical description of a CPCEI as actually implemented by the contractor. The Part II Specification is a product of a program acquisition phase and is one of the instruments used for CPCEI acceptance. The Part II Specification defines the product configuration

baseline. It also serves as an instrument for subsequent use by in service and/or contractor personnel in diagnosing trouble, making adaptation changes, and designing modifications to the CPCEI.

## 2.2 Specification Addenda

An addendum to a CPCEI Specification establishes requirements for a new end item which, in terms of performance and design, is so much like an existing CPCEI that it is desirable to refer to an existing specification on an "add" and "delete" basis.

## 3.0 APPLICABILITY

This Appendix is applicable to NASA agencies and NASA contractors responsible for the development of computer programs which are classified as end items. Guidelines for the selection of end items are given in Appendix IV. Each contractor to the Government shall be responsible for compliance by his subcontractors, vendors, and suppliers to the extent that his subcontractors, vendors, and suppliers participate in the preparation of this type of specification.

## 4.0 APPLICABLE DOCUMENTS

NPC 500-1 - "Apollo Configuration Management Manual", Exhibits I, II, VII, VIII, X, XI, and XVII.

ANA Bulletin 445

M200-A - "Standardization Policies, Procedures, and Instructions."

## 5.0 EXPLANATION OF TERMS

(See Appendix VII)

## 6.0 PROCEDURAL REQUIREMENTS

These procedures are based on the policies and practices of the procuring agency for configuration management. It is recommended that contractors review these policies and practices as part of implementing the procedural requirements in this exhibit.

### 6.1 Computer Program Contract End Item Detail Specification

The CPCEI Specification defines requirements for a computer program that has been identified as a CPCEI. The CPCEI Specification is composed of two distinct and different uses in the contractual control of CPCEI acquisition. Each of the two parts, when prepared to comply with these instructions, is complete in content and format with respect to its intended use. The CPCEI Specification is controlled and accounted as an entity, using a single end item configuration chart, and a single specification change log.

Part I is a product of a program definition phase or requirements analysis and is the technical requirements document used to contract for design and development of the CPCEI. Contractor compliance with Part I is determined by evaluation of qualification and other test records. The Part I Specification describes in mathematical, logical, and operational language all of the detail necessary to initiate and carry out the design of a required computer program contract end item. In addition to providing the primary "design to" guide for the computer programming design and development effort during the acquisition phase, this document provides (a) the basis for approval by the procuring and using agencies of the fine detail of the performance of the computer programs to be developed, (b) the instrument which defines all essential interfaces with other computer programs, equipment, and communications links, (c) the direct basis for the development of support documentation associated with operation and use of the computer program, and (d) the basic vehicle for configuration control of the design of the CPCEI throughout acquisition and operational phases of the system cycle.

Part II of the CPCEI Specification is a product of the acquisition phase. It provides a complete technical description of the CPCEI functions, structures, operating environment, and constraints, data organization, diagrammatic/narrative flows, and source statement/machine language listings. Part II must be identical to the actual CPCEI which results from the contractor's developmental work during the acquisition phase, and which is qualified (or to be qualified) under the terms and conditions of the contract as meeting the detailed performance requirements initially specified in Part I of the

CPCEI Detail Specification. The integrity of Part II is established by First Article Configuration Inspection (FACI) prior to its acceptance by the procuring agency. Acceptance is dependent upon the accuracy and completeness with which it describes both the gross and detailed structure and functioning of the computer program.

6.2 Addenda to Computer Program Contract End Item Detail Specification

Origination of an addendum to a CPCEI Detail Specification creates a new specification. It is used when an item to be designed and developed is so like an existing CPCEI that it is desirable to restrict design activity to "engineer to the same criteria, with these additions/deletions." An addendum changes a basic CPCEI Detail Specification by adding or deleting requirements on a paragraph-by-paragraph basis. Addenda shall be identified with a specified issue of a basic specification. The specification so created (basic specification plus addenda) then becomes controlled and maintained as a separate and distinct specification, to be updated and revised as necessary, independent of changes to the basic specification from which it was created.

6.3 Detail Instructions for the Preparation of the CPCEI Specification

The contents of the CPCEI Specification shall be arranged in accordance with the format and paragraph headings given in 6.3.1 (Part I) and 6.3.2 (Part II). General deviations from the requirements of this exhibit require prior approval of the procuring agency. Each CPCEI Detail Specification which deviates from the general requirements of this exhibit shall cite, in Section 10 "Appendix", the procuring agency instrument authorizing the deviation. For convenience in describing the minimum essential content, the paragraphs outlined in 6.3.1 and 6.3.2 are arranged in a format which shall apply if the specification is to be issued as a single document. However, each part of the specification required for a large computer program is typically too complex and bulky to be published and distributed physically in one bound volume. In this case, each part may be arranged into separate volumes corresponding to individual functional elements, or as determined by mutual agreement between the contractor and the procuring agency to meet the requirements of a particular system.

NOTE: In the descriptions to follow, the term "computer program component" (CPC) is used to denote the separately produced subprograms which compose a CPCEI. These subprograms are usually separately compilable, perform a specified function or set of functions that are related, and are usually produced by a single programmer. The Part II Specification contains a detailed description of each CPC in a CPCEI and denotes the relationship between them.

#### 6.3.1 CPCEI Part I Specification

This subparagraph contains a description of the paragraph headings of a CPCEI Part I Specification.

"Title Page" - (Computer Program Contract End Item Detail Specification) - The title page shall conform to the format of Sample Format "A" and include the following information referenced to Sample Format "A":

"Specification Number"	An identifier unique, within system, to this CPCEI. (See Exhibit X of NPC 500-1). For CPCEI's, the first two (2) characters (the prefix code) shall be "CG".
"Revision Identification"	Sequentially assigned character(s) to uniquely identify each revision of the specification.

Note: The specification number and revision identification may be composed of either, or both, alpha and/or numeric characters. Under no circumstances shall the combination of specification number and revision identification exceed 15 characters. (See Exhibit X of NPC 500-1)

"CEI"	Contract End Item Number. (See Exhibit X of NPC 500-1)
"Approved Nomenclature"	In accordance with Exhibit X and standard practice.
"System Identification"	List the system or systems which the CPCEI is designed to support. For CPCEI's which cannot be identified with specific systems, enter the phrase "Not System Computer Programs".

The title page shall be followed by introductory material as appropriate, including the End Item Configuration Chart and the Specification Change Log. Such introductory material shall be followed immediately by Part I of the specification. The introductory material between the title page for the CEI specification and the introductory page to Part I shall be numbered per standard practice for introductory material, e.g., i, ii, iii, etc.; or I, II, III, IV, etc. Part I and Part II shall each be numbered within themselves as required by the sample format, e.g., for Part I, I-1 of \_\_\_\_ (enter the last page number of Part I), I-2 of \_\_\_\_ (enter last page number of Part I, etc.; for Part II, II-1 of \_\_\_\_ (enter last page number of Part II), II-2 of \_\_\_\_ (enter last page number of Part II), etc.

"Introductory Page" - (Part I of CPCEI Specification) - The introductory page shall conform to the format of Sample Format "B". The identifying information appearing on this page shall be identical to the respective element of information appearing on the CEI specification title page. The preparing activity and the NASA office with computer program responsibility for the CPCEI shall validate the basic Part I in the approval blocks. A general outline of the following paragraphs is given in Sample Format "C".

Section 1, "Scope" - This section of the CPCEI specification shall begin with the following opening phrase: "This part of this specification establishes the requirements for performance, design, test, and qualification of a computer program identified as (insert nomenclature and contract end item number). This CPCEI is used to (provide)(accomplish).... This CPCEI requires ....." Subsequent sentence-paragraph(s) shall contain a summary of the purpose of the specification, a brief description of the functions to be performed, and a brief summary of the content and composition of the specification.

Section 2, "Applicable Documents" - This section of the CPCEI Specification shall begin with the following lead phrase: "The following documents, of exact issue shown, form a part of this specification to the extent specified herein. In the event of conflict between documents referenced here and the detailed content of Sections 3, 4, and 10, the detailed requirements in Sections 3, 4, and 10 shall be considered superseding requirements." Those documents (specifications, standards, drawings, bulletins, manuals, etc.) which are applicable to paragraphs within other sections of the specifications shall be listed. Within the body of the specification,

reference to those documents shall be by reference to their basic document number and to the title, method number, specifically identified requirements, or other definitive designation. The format in listing the applicable documents shall comply with Defense Standard Manual M200A, Chapter V.

Section 3, "Requirements" - This section shall contain performance and design requirements for the CPCEI. This section shall include the functional requirements for the CPCEI and establish those requirements which will be used for verification during test. This section shall define the CPCEI and specify design constraints and standards necessary to assure compatibility of the CPCEI with other computer programs and equipments. Performance and design requirements included herein are allocated from, identical with, or in recognition of, requirements established by the system specification. Requirements included in the system specification, which are directly related to requirements specified herein, may be incorporated by reference. Requirements shall be specified to the level of detail necessary to establish limits for design. Quantitative requirements shall be specified within the three principal subparagraphs included herein. General and descriptive material may be included in the basic Section 3.

Paragraph 3.1, "Performance" - This paragraph shall specify the functional requirements of the CPCEI. Quantitative requirements shall be specified within the principal subparagraphs included herein. General and descriptive material may be included in basic paragraph 3.1.

Paragraph 3.1.1, "System Requirements" - This paragraph shall specify the limits and/or capacities of the CPCEI performance. Requirements specified herein are the product of analysis, as well as those contained in the system specification. These characteristics are the performance parameters which must be specified to constrain design within requirements established by primary mission/use of the CPCEI, e.g., for a space tracking system; this could include track capacity, number and type of inputs processed, etc. Requirements included herein shall be stated in quantitative terms, with tolerances where applicable.

Paragraph 3.1.2, "Operational Requirements" - This paragraph shall specify, in subparagraphs defined below, the operational requirements of the CPCEI. Requirements shall be stated in quantitative terms, with tolerances where applicable. General and descriptive material may be included in basic paragraph 3.1.2, which shall incorporate, either directly or by reference,



a functional block diagram or equivalent representation of the CPCEI. The graphic portrayal shall be accomplished to the level of detail necessary to illustrate the functional operation of the CPCEI, the relationships between these functions, and the relationships between the functions and other identified system functions. This diagram is not intended to be restrictive on computer program design in any way.

Requirements for separately identified CPCEI functions shall be described in subsequent paragraphs as appropriate. A subparagraph shall be included for each operational function, plus special functions such as sequencing control, displays, error detection and recovery, input and output control, real-time diagnostics, operational data recording, etc. The descriptions of these CPCEI functional requirements shall include the relative sequencing, periodicities options, and other important relationships of each as appropriate.

Paragraph 3.1.2.1, "Function 1" - The basic paragraph shall begin with descriptive and introductory material which defines the function and its relationship to other functions. Then, the following three subparagraphs shall specify the quantitative requirements concerning the function.

Paragraph 3.1.2.1.1, "Source and Type of Inputs" - This paragraph shall specify the source(s) and type(s) of input information associated with this information of the CPCEI. This shall include a description of the information, its source(s) and, in quantitative terms; units of measure, limits and/or ranges of units of measures, accuracy/precision requirements, frequency of arrival information, etc., where applicable.

Paragraph 3.1.2.1.2, "Destination and Types of Outputs" - This paragraph shall specify the destination(s) and type(s) of output information associated with this function of the CPCEI as a result of the processing described in paragraph 3.1.2.1.3. This shall include a description of the information, its destination(s) and, in quantitative terms; units of measure, accuracy/precision requirements, frequency of output information, etc., where applicable.

Paragraph 3.1.2.1.3, "Information Processing" - This paragraph shall specify the information processing associated with this function. The paragraph shall incorporate a detailed prose and mathematical description, including necessary logical concepts, timing requirements, mathematical

techniques, required accuracies with tolerances, data manipulation considerations, and options. A graphic portrayal of this function shall be included for clarity as appropriate.

Paragraph 3.1.2.n, "Function n" -

Paragraph 3.1.2.n.1, "Source and Type of Inputs" -

Paragraph 3.1.2.n.2, "Destination and Types of Outputs" -

Paragraph 3.1.2.n.3, "Information Processing" -

Paragraph 3.1.3, "Data Base Requirements" - This paragraph shall specify, in descriptive and quantitative terms, the requirements for all parameters which affect the design of the CPCEI. The detailed definition of parameters shall include a description of the data and quantitative definitions of units of measure, ranges of units of measure, and accuracy/precision requirements where applicable. In addition, wherever applicable, this paragraph shall specify the methods necessary to convert these parameters into a form suitable for use by the computer program. In the case of a multi-site system in which the actual data values of certain parameters will vary among site installations, the complete set of such site adaptation parameters shall be identified either directly in a separate subparagraph or by reference.

Paragraph 3.1.4, "Human Performance" - Human performance/human engineering requirements for the CPCEI shall be specified in this paragraph; for example, minimum times for human decision making, maximum time for program responses, maximum display densities of information, clarity requirements for displays, etc. For CPCEI's which directly support a system(s); this paragraph shall cite the appropriate paragraph(s) of the system specification which establish the human performance/human engineering requirements for all system equipment, and incorporate requirements peculiar to this CPCEI on an add and/or delete basis.

Paragraph 3.2, "CEI Definition" - This paragraph shall, in subparagraphs include herein, specify the functional relationship of the CPCEI to other equipment/computer programs and identify government furnished computer programs incorporated in the CPCEI. General and/or descriptive material may be included in basic paragraph 3.2.

Paragraph 3.2.1, "Interface Requirements" - This paragraph shall specify, either directly or by reference, requirements imposed on the design of the CPCEI because of its relationship to other equipment/computer programs. It also includes detailed interface definitions resulting from contractor analysis and requirements contained in the system specification. General and/or descriptive material may be included in basic paragraph 3.2.1. Quantitative requirements shall be included in the subparagraphs included herein.

Note: Interfaces defined in this section shall include, at a minimum, all relevant characteristics of the computer, such as memory size, word size, access and operation times, interrupt capabilities, and special hardware capabilities. If the compiler/assembler is another, or part of another, CEI, the computer program language(s) to be employed shall be specified as one of the interfaces in subparagraph 3.2.1.2. If the compiler/assembler is a Government-furnished component to be incorporated into this CPCEI, it shall be referenced in subparagraph 3.2.2. If this compiler/assembler is to be constructed as part of the development of this CPCEI, the language characteristics are to be defined under paragraph 3.1, "Operational Requirements".

Paragraph 3.2.1.1, "Interface Block Diagram" - The relationship of the CPCEI to other equipment/computer programs with which it must interface shall be graphically portrayed in this paragraph. This paragraph shall incorporate, in subparagraphs as appropriate, either directly or by reference, a functional block diagram or equivalent representation of the interface requirements of the CPCEI. The graphic portrayal of the CPCEI shall be accomplished to the level of detail necessary to identify the functional interfaces between the CPCEI and other identified equipment/computer programs.

Paragraph 3.2.1.2, "Detailed Interface Definition" - This paragraph shall specify, in subparagraphs as appropriate, the functional relationship of the CPCEI to interfacing equipment and computer programs. This information shall be given in quantitative terms, with tolerances where applicable, to the level of detail necessary to permit design of the CPCEI. Functional interfaces shall specify the input/output requirements of the CPCEI in terms of data rate, message format, etc. In addition, this paragraph shall specify design requirements imposed upon other equipment/computer programs as a result of the design of this CPCEI,

e.g., card formats, operator console equipment, display formats, etc. This paragraph shall incorporate, either directly or by reference, interface drawings and/or other documentation necessary to specify the functional interfaces of this CPCEI with other equipment/computer programs.

Paragraph 3.2.2, "Government-Furnished Property List" -

This paragraph shall list the government-furnished computer programs which the CPCEI must be designed to incorporate. The listing shall identify the program by nomenclature, specification number, model number if appropriate, and associated documentation.

Paragraph 3.3, "Design Requirements" - This paragraph shall specify, in appropriate subparagraphs, requirements which affect the design of the CPCEI and are distinguishable from the performance requirements of paragraph 3.1. These requirements result from general considerations of CPCEI useability. These may include but are not limited to requirements for:

- a. The use of programming standards to assure compatibility among computer program components (CPC - subprograms or groups of functionally related subprograms).
- b. Program organization, such as overall program segmentation.
- c. Program design resulting from consideration of modifications to the CPCEI during operation; for example, on-site modification requirements and the permissible amount of operational degradation allowed during installation of modifications may be specified.
- d. Special features, to facilitate the testing of the CPCEI. For example, special procedures for the design of computer program component interfaces, requirements for intermediate printouts, and commentary on the program listing may be required.
- e. Expandability, to facilitate the growth of the CPCEI.

Section 4, "Quality Assurance Provisions" - Requirements for formal verification of the performance of the CPCEI in accordance with the requirements for Section 3 of this specification shall be specified in this paragraph. Formal verification of performance of the CPCEI shall determine acceptance of the CPCEI. This paragraph shall specify formal verification requirements to a level of detail which:

- a. Designates verification requirements and methods in Section 4 for performance and design requirements in Section 3. The methods of verification to be specified herein may include inspection of the CPCEI, review of analytical data, demonstration tests, and review of test data.
- b. Specifies requirements for verification to the level of detail necessary to clearly establish the scope and accuracy of the test method.
- c. Permits ready identification of each verification requirement specified in Section 4 with the appropriate performance/design requirement paragraph in Section 3.
- d. Allocates verification requirements to the subparagraphs included herein.

Note: This section shall not incorporate, either directly or by reference, detailed test planning documentation and operating instructions. Requirements specified herein shall be the basis for preparation and validation of such documents. All test/verification requirements shall be specified within the subparagraphs included herein.

Paragraph 4.1, "Implementation Test Requirements" - The term "Implementation Test" is defined to include all tests of the CPCEI other than that accomplished during the integration tests (see paragraph 4.2). The implementation test requirements shall be specified in the following subparagraphs.

Subparagraph 4.1.1, "Design and Development Testing" - This subparagraph shall specify the requirements for computer program component and program tests conducted in the acquisition phase prior to the preliminary qualification tests. These are tests performed on the individual computer program components and groups of components to establish that the

CPCEI is ready for preliminary qualifications testing. These requirements shall be specified in sufficient detail to permit design of such tests and to serve as a guide to the programmers in computer program component debugging.

Note: Verification requirements included in a higher level specification (e.g., system specification) may be incorporated by reference to avoid duplication.

Subparagraph 4.1.2, "Preliminary Qualification Testing" - This subparagraph shall identify the requirements for preliminary qualification testing. These tests verify each requirement of Section 3, which can be tested in a simulated environment. They are typically held at the contractor's development facility and may serve as the basis for transfer of the CPCEI to the user's facility.

This subparagraph shall specify only those preliminary qualification requirements which require formal recognition by the procuring agency. It shall specify, to whatever extent is possible, the location and relative scheduling of the tests, the computer-based system components which will not be available (and hence must be simulated or ignored), the objectives of the tests, and the program functions to be tested.

Subparagraph 4.1.3, "Special Test Requirements" - In this subparagraph all special verification test requirements, if any, not included in the above two subparagraphs shall be specified.

Paragraph 4.2, "Integration Test Requirements" - This paragraph shall identify requirements specified in Section 3 which cannot be verified until the CPCEI is assembled into or used with the computer-based system environment and other CPCEI's. Final verification that all requirements in Section 3 have been satisfied shall be considered here.

Integration test requirements shall be specified in the following subparagraphs.

Subparagraph 4.2.1, "General" - This subparagraph shall specify CPCEI tests which are required in direct support of system integration, i.e., integration of the CPCEI, hardware, external environment (communication lines, etc.) and operating personnel into an operative computer-based system. Consideration should be given to the impact of

training on CPCEI test design. Such tests, prior to formal acceptance tests, usually involve attempting to run parts of the CPCEI functions on a limited basis as components of the computer-based system are installed sequentially. This subparagraph shall identify each performance/design requirement to be verified in each test.

Subparagraph 4.2.2, "Acceptance/Qualification Test" - This subparagraph shall specify requirements for formal qualification of the integrated CPCEI to demonstrate and/or verify that the requirements established in Section 3 have been satisfied. This paragraph shall, in subparagraphs as appropriate, specify the requirement and method of verification for the requirements specified in Section 3.

Verification of the requirements may be accomplished by inspection, or review of analytical data, or by demonstration, or test and review of test data, or combinations of these.

Section 6, "Notes" - This section shall include information which is stated here for administrative convenience only and is not a part of the specification for the CPCEI in the contractual sense, i.e., it shall not include requirements which constrain design, development, and qualification of the CPCEI and require compliance by the contractor.

This section of the specification shall include information of particular importance to the procuring agency in using this particular specification as a contractual instrument for acquisition of the CPCEI, either initially or for follow-on procurement.

Background information or rationale which will be of assistance in understanding the specification itself or using the CPCEI it specifies may be included herein, e.g., technical data ordering instructions.

Section 10, "Appendix" - Requirements specified in the appendix are contractually a part of the specification, and to the extent they impose requirements on design, development, and qualification of the CPCEI, they must be satisfied. This section may include, but not be limited to, requirements which are:

- a. Bound separately for convenience, as in the case of a classified appendix or a large body of statistical data.

- b. Of a temporary nature, as in the case of an interim performance requirement peculiar to early test models of the CPCEI. Requirements peculiar to early test articles of the CPCEI shall be specified in an appendix which adds to, deletes, changes, or establishes new requirements applicable to Section 3 or 4 of Part I.

Instrumentation requirements for test articles of the CPCEI shall be specified only to the level of detail necessary to establish the type and total capacity of the instrumentation. Requirements specified herein (with the exception of instrumentation) shall be specified to the level of detail required by the paragraphs in Sections 3 and 4 of Part I to which they relate.

If any terms, symbols, and abbreviations are used in the body of this specification whose meanings are not widely known, they may be defined in a Glossary in this section.



SAMPLE FORMAT "A"

Specification No. \_\_\_\_\_

Revision No. \_\_\_\_\_

Release Date \_\_\_\_\_

CONTRACT END ITEM DETAIL SPECIFICATION  
(COMPUTER PROGRAM)

PERFORMANCE/DESIGN  
AND  
PRODUCT CONFIGURATION  
REQUIREMENTS

(CEI Number)

(APPROVED NOMENCLATURE)

FOR

(PROJECT OR SYSTEM NAME)

(PROJECT OR SYSTEM)

Approved by \_\_\_\_\_  
(Preparing Activity)

Approved by \_\_\_\_\_  
(NASA Office)

Date \_\_\_\_\_

Approval Date \_\_\_\_\_

Contract Number \_\_\_\_\_

SAMPLE FORMAT "B"

Specification No. \_\_\_\_\_

Revision No. \_\_\_\_\_

Release Date \_\_\_\_\_

CONTRACT END ITEM DETAIL SPECIFICATION  
(COMPUTER PROGRAM)

PART I

PERFORMANCE/DESIGN  
REQUIREMENTS

(CEI Number)

(APPROVED NOMENCLATURE)

FOR

(PROJECT OR SYSTEM NAME)

(PROJECT OR SYSTEM)

Approved by \_\_\_\_\_  
(Preparing Activity)

Approved by \_\_\_\_\_  
(NASA Office)

Date \_\_\_\_\_

Approval Date \_\_\_\_\_

Contract Number \_\_\_\_\_

SAMPLE FORMAT "C"

GENERAL OUTLINE OF CPCEI SPECIFICATION PART I

TITLE PAGE (OVERALL CPCEI SPECIFICATION)

INTRODUCTORY PAGE (PART I)

1. SCOPE

2. APPLICABLE DOCUMENTS

3. REQUIREMENTS

3.1 PERFORMANCE

3.1.1 SYSTEM REQUIREMENTS

3.1.2 OPERATIONAL REQUIREMENTS

3.1.2.1 FUNCTION 1

3.1.2.1.1 SOURCE AND TYPE OF INPUTS

3.1.2.1.2 DESTINATION AND TYPES OF OUTPUTS

3.1.2.1.3 INFORMATION PROCESSING

3.1.2.2 FUNCTION 2

. .

. .

. .

3.1.2.n FUNCTION n

3.1.3 DATA BASE REQUIREMENTS

3.1.4 HUMAN PERFORMANCE

3.2 CPCEI DEFINITION

3.2.1 INTERFACE REQUIREMENTS

3.2.1.1 INTERFACE BLOCK DIAGRAM

3.2.1.2 DETAILED INTERFACE DEFINITION

3.2.2 GOVERNMENT-FURNISHED PROPERTY LIST

3.3 DESIGN REQUIREMENTS

4. QUALITY ASSURANCE

4.1 IMPLEMENTATION TEST REQUIREMENTS

4.1.1 DESIGN AND DEVELOPMENT TESTING

4.1.2 PRELIMINARY QUALIFICATION TEST

4.1.3 SPECIAL TEST REQUIREMENTS

4.2 INTEGRATION TEST REQUIREMENTS

4.2.1 GENERAL

4.2.2 ACCEPTANCE/QUALIFICATION TEST

6. NOTES

10. APPENDIX

### 6.3.2 CPCEI Part II Specification

This subparagraph describes the paragraph headings of a CPCEI Part II Specification.

The overall outline of the CPCEI Specification Part II is contained in Sample Format "D", with all major paragraphs and subparagraph headings listed.

Throughout the Part II Specification, the phrase "Computer Program Component" is used to refer to subprograms (separately compilable groups of computer program instructions and data) or groups ("packages") of functionally related subprograms.

A discussion of the contents of these major paragraphs and subparagraphs follows.

"Introductory Page" - (Part II of CPCEI Specification) - The introductory page shall conform to the format of Sample Format "E". The identification appearing on this page shall be identical to the respective elements of information appearing on the CEI specification title page. The procuring agency shall validate the basic Part II in the approval blocks.

Section 1, "Scope" - This section shall contain the following lead phrase: "This specification establishes the requirements for complete identification and acceptance of (insert contract end item number and nomenclature) to be formally accepted by the procuring agency, subsequent to establishment of the product configuration baseline. The product configuration baseline shall be established by First Article Configuration Inspection (FACI) for serial number (insert CPCEI serial number)."

Section 2, "Applicable Documents" - This section of the specification shall begin with the following lead phrase--"The documents, of exact issue shown, form a part of this specification to the extent specified herein. In the event of conflict between documents referenced here and the detail content of Sections 3, 4, 5, and 10, the detail contents of Sections 3, 4, 5 and 10 shall be considered a superseding requirement." List those documents (specifications, standards, bulletins, manuals, etc.) which are applicable to paragraphs within other sections of the specification. Within the body of the specification, reference to these documents shall be by reference to their basic document number and to the title, specifically identified requirements, or other definitive designation.

Section 3, "Requirements" - This section shall specify the detailed configuration of the CPCEI. This section shall contain a complete technical description of the CPCEI structure and functions, the data base, and the individual CPC's. General and/or descriptive material may be included in basic Section 3 lead paragraph.

Paragraph 3.1, "CPCEI Characteristics" - This paragraph shall contain a description of the overall structure and functions of the CPCEI. This description shall include: the allocation of functions to the CPC's that compose the CPCEI; and a graphic portrayal of storage allocation. This description shall be given in the following paragraphs.

Paragraph 3.1.1, "Functional Allocation" - The relationship of each computer program component (CPC) to the performance and design requirements of the Part I Detail Specification shall be specified to the level of detail necessary to identify how the computer program components are associated with the requirements of the functions specified in the Part I Specification. If the CPC's are grouped into functional entities ("packages") for separate stages of development and checkout, this grouping shall be delineated.

Paragraph 3.1.2, "CPCEI Flow Chart" - This paragraph shall graphically portray the operations performed by the CPCEI. This shall be done by a (series of) flow chart(s) which depict(s) the processing being performed, the sequence of operations, and decision points. A "top-level" flow chart shall be used to depict in a single figure the overall information flow of the CPCEI. This diagram shall reference lower level flow charts included in this paragraph, as appropriate, to provide more detailed information. The lowest level flow charts shall be those which identify as functional entities the computer program components described in paragraph 3.2 below. All symbology used in the flow chart shall be defined either directly in this paragraph, or on the individual flow chart sheets, or by reference to a documented set of standards.

Paragraph 3.1.3, "CPCEI Timing and Sequencing" - This paragraph shall describe the timing and sequencing of operations of the computer program components relative to each other. If the sequencing is dynamically controlled during the CPCEI's operations, this description shall include the method for sequence control and the logic and input conditions of that method. Such factors as timing variations, plus such internal operations as data transfers in and out of core, disc, drum, or tape memory, sensing of discrete input signals, and the timing relationships between interrupt operations within the CPCEI, shall be included.

Paragraph 3.1.4, "Storage Allocation" - The relationship of the CPCEI storage requirements to the total computer equipment storage capability shall be graphically portrayed in this paragraph. This paragraph shall incorporate, in subparagraphs as appropriate, either directly or by reference, a schematic diagram, or equivalent representation. This graphic portrayal of the CPCEI shall be accomplished to the level of detail necessary to identify such requirements as: data base allocation, computer program allocation, computer allocation, and spare storage allocation. If allocations cannot be specified precisely or portrayed graphically in a manner meaningful for program design, the algorithms used to allocate storage will be described.

Paragraph 3.1.5, "Data Base Characteristics" - This paragraph shall include a detailed definition of the content and storage location of each file, table, and item within each table that is incorporated in the CPCEI data base, as well as the storage location of each computer program component contained in the CPCEI. This paragraph will contain the following:

1. "File Description." A list of files that have been incorporated in the computer program data base shall be included. This shall include a descriptive title for each file, length of file, and format, etc.
2. "Table Description." A list of tables that have been incorporated in the computer program data base shall be included. This shall include a descriptive title for each table, method of indexing the table, length of table, and block format for items and itemless tables, etc.
3. "Item Description." A list of all items contained in the computer program data base shall be included. This shall include, for each item, a descriptive title, most significant bit, number of bits, coding type, scaling factor, and, if appropriate, units and item value, etc.
4. "Graphic Table Description." The relationship of the items specified in 3 above "Item Description" to the tables listed in 2 above to the files listed in 1 above will be graphically portrayed. This shall incorporate, in subparagraphs as appropriate either directly or by reference, a diagram or equivalent representation. The graphic

portrayal of each table shall be accomplished to the level of detail necessary to identify words per block, untagged items, bits/items, bit allocation, number of blocks, and type of table construction.

5. "Data Organization." A definition of the relationship of the items, tables, and files contained within the data base and the computer program components described in paragraph 3.2, to locations in computer storage, shall be included. This paragraph will incorporate such information as the following:
  - a. A list of files, specifying for each file the address in storage and number of tables contained in the file, etc.
  - b. A list of tables, specifying for each table the location within the file and number of words contained in the table, etc.
  - c. A list of items, identifying item location within the table, number of bits, item type, scaling, etc.
  - d. A list of computer program components, specifying for each the storage address and number of words allocated.
6. "CPCEI Constants." A list of all constants (e.g., fixed values assigned to the parameters defined in the Part I CPCEI Specification) contained in the CPCEI, other than those which are defined as "Adaptation Data" below, shall be included. The list contained herein shall include, as a minimum, a description of each constant and its actual numerical or coded value.
7. "Adaptation Data." For multi-site computer based systems, the actual data required to adapt the CPCEI to the environment associated with each site shall be listed. For convenience, this information may be contained in Section 10, "Appendix."
8. "Relationship of Computer Program Components to Data Base." The relationship of the various computer program components to the various tables and items contained in the CPCEI data base shall be graphically portrayed. This paragraph shall incorporate, in subparagraphs as appropriate, either directly or by reference, a diagram or



equivalent representation. This graphic portrayal of the relationship of CPC's to the data base shall be accomplished to the level of detail necessary to identify the tables and items within the tables required by each computer program component to the relationship of the CPC to each table/item (e.g., sets the item, uses the item, etc.). The use of such tables and items by each CPC shall be fully described in the appropriate subparagraph under paragraph 3.2.

Paragraph 3.2, "Computer Program Component Characteristics" - The individual computer program components (CPC's) shall be described in separate paragraphs as required. This description shall be given at a level of detail that will define the design and configuration of the CPC sufficiently to allow for CPC modification and adaption in the operation phase. Each CPC shall be described in words, flow charts, and with a listing of the instructions used. The basic paragraph 3.2 shall contain the following lead phrase: "This paragraph contains the detailed technical descriptions of the computer program components identified in Paragraph 3.1 of this specification. The instruction listings contained herein by inclusion or reference specify the exact configuration of the (name of CPCEI) . . ." The following paragraphs will be repeated for each CPC.

Note: The contents of paragraphs 3.2.1-3.2.n to follow represent information that corresponds in general to "programming specifications" normally produced by analysts, and "coding specifications" normally produced by programmers, during the acquisition phase. The format described below is intended to allow for use of such technical documentation directly as portions of the Part II CPCEI Specification, thus eliminating the need for redocumentation. Other formats may be allowed at the discretion of the procuring agency if they meet this intent.

Paragraph 3.2.1, "Computer Program Component 1" - The basic paragraph shall identify the CPC by including as a minimum the title, tag (symbolic code), and component identification number. It shall also include a brief abstract of the functions of the CPC, the language(s) in which it is written, and its major functional interfaces. The CPC shall then be described in detail in subparagraphs as follows:

Subparagraph 3.2.1.1, "CPC No. 1 Description" - This subparagraph shall describe in words, figures, equations, and references to the flow chart(s) of subparagraph 3.2.1.2, the functions and design of the CPC.

This subparagraph shall contain, as appropriate, a description of: the program logic and data flow; equations to be solved; algorithms used to solve these equations; timing and accuracy characteristics; and any special conditions for operation of the CPC. The description shall be sufficiently detailed to facilitate understanding of, and modification to, the listing given in subparagraph 3.2.1. Equation derivations and numerical analysis shall not be included herein, but may be included in Section 6, "Notes".

Subparagraph 3.2.1.2, "CPC #1 Flow Chart" - This subparagraph shall graphically portray the operations performed by the CPC. This shall be done by a (series of) flow chart(s) which depict the processing described in subparagraph 3.2.1.1, including the sequence of operations and decisions points, in the CPC. The "highest level" flow chart shall depict on a single sheet the overall information flow of the CPC and shall reference the flow chart(s) in paragraph 3.1.2 that identifies the CPC. In general, the lowest level flow chart identifies all decision points in the CPC and references higher level charts as appropriate. All flow charts shall use descriptive symbology and shall reference the program listing of the CPC by use of statement labels or tags. All symbology used in the flow chart shall be defined in this subparagraph, or by reference to paragraph 3.1.2 above, or by reference to a documented set of standards, or on the individual flow chart sheets.

Subparagraph 3.2.1.3, "Interfaces" - This subparagraph shall describe the relationship of the CPC to other CPC's, to the part of the data base external to the CPC, and, to other CPCEI's where applicable. Appropriate subparagraphs shall include:

- (a) The exact format, content and source of all input data.
- (b) The exact format, content and destination of all output data.
- (c) A list of the subroutines called by the CPC.
- (d) A list of other CPC's which call the CPC.
- (e) A list of external (to the CPC) tables, buffers, constants, and control registers used by the CPC.

Subparagraph 3.2.1.4, "CPC Data Organization" - This subparagraph shall contain, or refer to a portion of subparagraph 3.2.1.6 if appropriate, a list and description of all data items and tables which are unique to and included

within the CPC, and shall describe the areas of memory available for temporary storage. This list shall include all internally defined symbols and their equivalence and meaning.

Subparagraph 3.2.1.5, "Limitations" - This subparagraph shall summarize any known or anticipated limitations of the CPC. A listing of all restrictions and constraints which apply to the CPC shall be provided, including timing requirements, limitations of algorithms and formulas used, limits of input and output data, associated error correction sensing, and the error checks programmed into the routines.

Subparagraph 3.2.1.6, "Listing" - This subparagraph shall contain a complete listing of instructions contained in the computer program component(s). The type of listing provided in this paragraph shall be established between the contractor and the procuring agency. The listing will show the relationship to the flow diagrams above by appropriate use of statement labels or tags. For convenience, the listing may be included in Section 10, "Appendix."

Paragraph 3.2.2, "Computer Program Component #2"

.

.

.

Paragraph 3.2.n, "Computer Program Component #n"

Section 4, "Quality Assurance" - This section shall specify the tests which must be accomplished to demonstrate that the CPCEI performance and configuration is as specified in Section 3 of the Part II Specification.

Paragraph 4.1, "Test Specifications Cross Reference Index" - In this paragraph, all Test Plan and Test Specification documents shall be referenced, with cross indexing to the CPCEI functions being tested and the CPCEI components and other computer programs required for the tests. This paragraph shall also describe or reference special simulation capabilities required for test/verification of the CPCEI.

Paragraph 4.2, "Other Quality Assurance Provisions" - This paragraph shall reference applicable standards and/or specify the test/verification requirements, methods and procedures which apply to duplication of the computer program (e.g., tapes and/or card decks) which are covered by the specification.

Section 5, "Preparation for Delivery" - This section shall specify, in subparagraphs as appropriate, the requirements for packaging, marking and otherwise preparing the CPCEI for shipment. Where approved existing Government Specifications are adequate to satisfy current requirements for the item and components thereof, these shall be incorporated by reference in lieu of providing duplicate detailed requirements in this section. Where suitable specifications do not exist, requirements peculiar to the CPCEI shall be specified in appropriate subparagraphs included herein.

Paragraph 5.1, "Delivery Procedures" - This paragraph shall describe the delivery procedures of the completed CPCEI. Delivery procedures include a description of the product packaging (card decks, tapes, manuals) involved, where they shall be stored or delivered to, and the contractors responsibilities for, during, and after delivery.

Paragraph 5.2, "Markings" - This section shall contain subparagraphs specifying in detail the identification markings to appear on every separable portion of the CPCEI to be delivered by the contractor and formally accepted by the procuring agency.

Section 6, "Notes" - This section of the specification is not contractually binding. It shall include information of particular importance to the procuring agency in using Part II of this specification as a contractual instrument, or administrative or background information, e.g., ordering instructions for technical data pertaining to the computer program, or specific information related to the use of the program in future assembly and integration testing. It shall not include requirements which constrain design, development, or qualification of the CPCEI. It shall reference the technical manuals which can be singularly and peculiarly identified with the CPCEI, and which are necessary to its operation and maintenance.

For each CPC this paragraph shall include any pertinent information not included in the above subparagraphs, such as rejected alternative CPC designs, the rationale behind the design, reference material in support of the algorithms used, and suggestions for future modifications to the CPC if changes in requirements should materialize. It shall also describe as appropriate the pertinent tests which were performed to verify the final implementation of the CPC, with key test results included or referenced.

Section 10, "Appendix" - This section of the specification may contain requirements which are part of Section 3 and/or of Section 4, but are bound separately for comments. Examples are computer-produced listings, multi-site adaptation requirements, etc.

SAMPLE FORMAT "D"

GENERAL OUTLINE OF CPCEI SPECIFICATION PART II

INTRODUCTORY PAGE (PART II)

1. SCOPE
2. APPLICABLE DOCUMENTS
3. REQUIREMENTS
  - 3.1 CPCEI CHARACTERISTICS
    - 3.1.1 FUNCTIONAL ALLOCATION
    - 3.1.2 CPCEI FLOW CHART
    - 3.1.3 CPCEI TIMING AND SEQUENCING
    - 3.1.4 STORAGE ALLOCATION
    - 3.1.5 DATA BASE CHARACTERISTICS
  - 3.2 COMPUTER PROGRAM COMPONENT CHARACTERISTICS
    - 3.2.1 COMPUTER PROGRAM COMPONENT #1
      - 3.2.1.1 DESCRIPTION
      - 3.2.1.2 FLOW CHARTS
      - 3.2.1.3 INTERFACES
      - 3.2.1.4 CPC #1 DATA ORGANIZATION
      - 3.2.1.5 LIMITATIONS
      - 3.2.1.6 LISTINGS
    - 3.2.2 COMPUTER PROGRAM COMPONENT #2
      - .
      - .
      - .
    - 3.2.n COMPUTER PROGRAM COMPONENT #n

4. QUALITY ASSURANCE

4.1 TEST SPECIFICATIONS CROSS REFERENCE INDEX

4.2 OTHER QUALITY ASSURANCE PROVISIONS

5. PREPARATION FOR DELIVERY

5.1 DELIVERY PROCEDURES

5.2 MARKINGS

6. NOTES

10. APPENDIX

SAMPLE FORMAT "E"

Specification No. \_\_\_\_\_

Revision No. \_\_\_\_\_

Release Date \_\_\_\_\_

CONTRACT END ITEM DETAIL SPECIFICATION

(COMPUTER PROGRAM)

PART II

COMPUTER PROGRAM

DESIGN

(CEI NUMBER)

(APPROVED NOMENCLATURE)

### 6.3.3 CPCEI Specification Addenda

Frequently, a requirement develops for a CPCEI which is very similar to an existing CPCEI. When this occurs, it is desirable to create the new CPCEI by accomplishing minimum re-design of the existing CPCEI. To accomplish this, it is necessary to maintain visibility, throughout the design and development cycle, of differences in performance, design, and configuration requirements between the two CPCEI's. This visibility can often be acquired and maintained by creating the specification for the new CPCEI as an addendum to the specification for the existing CPCEI.

The use of a specification addendum presents a formal means of writing a specification for a new CPCEI, by changing the specification for an existing CPCEI in a manner which permits ready comparison of the exact relationship between the two CPCEI's. This is accomplished by writing the new specification by direct reference to the existing specification on a paragraph-by-paragraph basis, recording in the new specification specific references to each paragraph in the existing specification and noting each addition, deletion, or change. Where no change is necessary, the phrase "no change" shall be used. The paragraph numbering between the two documents shall be identical, with the exception of paragraphs added to the new document which do not have an exact counterpart in the existing specification.

A specification created in this manner is a new and complete specification in every sense. The method of preparing a specification for a new CPCEI by creating an addendum to an existing specification shall be used when the following conditions are satisfied:

- (a) There is sufficient reason to establish a direct relationship between the new CPCEI and an existing CPCEI as a basis for design and development, e.g., progressing from one type, model, series of a CPCEI to another; or when minor changes must be accomplished to a very limited number of components of a CPCEI for a specific mission.
- (b) The basic specification, to which the addendum is prepared, complies with the requirements of this exhibit with respect to format and content.



The specification created by use of an addendum must be identified and maintained as a separate specification. Both the specification created by use of an addendum, and the basic specification to which the addendum is prepared, shall have independent change cycles. A specification change notice to either is not automatically a change to both. Each change to either must be reviewed, and if it is desirable to change both the basic specification and the specification prepared as an addendum, two separate specification change notices must be prepared.

When a new specification is created by the preparation of an addendum to an existing specification, an Addendum Notice shall be prepared which conforms to the format and includes the content required by sample format F. The Addendum Notice shall be the first entry in Section 2, "Applicable Documents." All of the entries in the Addendum Notice refer to the original CPCEI specification as the basic document for preparation of the addenda CPCEI. Each entry shall be transcribed from the title page and specification change log of the basic specification.

SAMPLE FORMAT "F"

--A D D E N D U M   N O T I C E--

This Specification has been prepared as an Addendum to:

Specification No. \_\_\_\_\_

Revision \_\_\_\_\_

Release Date \_\_\_\_\_

CEI No. \_\_\_\_\_

FOR

(Approved Nomenclature)

Used with

(PROJECT OR SYSTEM NAME)

(PROJECT OR SYSTEM)

The exact content of specification (insert same number as above) used as the basic document for this addendum is the revision referenced above plus the following specification change notices to specification (insert same number as above).

BELLCOMM, INC.

APPENDIX IV

CONFIGURATION MANAGEMENT OF  
COMPUTER PROGRAM CONTRACT END ITEMS

NOTE: The information contained in this appendix is based on the requirements for the configuration management of computer programming as given in NPC 500-1, "The Apollo Configuration Management Manual." The requirements ~~are~~ defined in NPC 500-1 apply to computer programs which are elements of large systems. They can, however, be applied toward the management of any computer programming task.

TABLE OF CONTENTS

- 1.0 INTRODUCTION
- 2.0 CONFIGURATION MANAGEMENT OF COMPUTER PROGRAMS - A GENERAL DISCUSSION
  - 2.1 THE COMPUTER PROGRAM CONTRACT END ITEM
  - 2.2 CONFIGURATION MANAGEMENT IN THE PROGRAMMING PROCESS
  - 2.3 SELECTION OF CPCEI'S
- 3.0 CONFIGURATION MANAGEMENT REQUIREMENTS
  - 3.1 CONFIGURATION IDENTIFICATION
    - 3.1.1 CPCEI SPECIFICATION
    - 3.1.2 IDENTIFICATION NUMBERS AND MARKINGS
  - 3.2 CONFIGURATION CONTROL
    - 3.2.1 BASELINE MANAGEMENT
    - 3.2.2 CHANGE CONTROL
    - 3.2.3 TECHNICAL REVIEWS AND INSPECTIONS
  - 3.3 CONFIGURATION ACCOUNTING
    - 3.3.1 SPECIFICATION CHANGE NOTICE (SCN)
    - 3.3.2 SPECIFICATION CHANGE LOG (SCL)
    - 3.3.3 END ITEM CONFIGURATION CHART
    - 3.3.4 CONFIGURATION INDEX
    - 3.3.5 CHANGE STATUS REPORT
    - 3.3.6 VERSION DESCRIPTION DOCUMENT
    - 3.3.7 CPCEI MAINTENANCE

REFERENCE DOCUMENTS

FIGURE I

CONFIGURATION MANAGEMENT OF  
COMPUTER PROGRAM CONTRACT END ITEMS

1.0 INTRODUCTION

"Configuration Management" is a method for managing the technical requirements that define the elements of a system. It is implemented through uniform procedures of specification, change control, and status recording.

Configuration management was developed to help solve severe problems which had plagued early missile system projects--problems which are characteristic of large scale system development. For example, a lack of precisely defined performance requirements early in some projects led to excessive costs, because many system elements had to be redesigned at a later time to satisfy the users' needs. Another problem--the loss or lack of drawings and technical documentation describing equipment under production and in the field--resulted in systems that could not be changed or maintained. Even where documentation existed, lack of adequate change control and accounting procedures resulted in equipment that did not match specifications, and in loss of interface compatibility between equipments. For some systems spare parts were lacking, and components could not be traced to determine their availability and location.

Formal techniques for the generation and transmittal of critical information were found to be essential in alleviating these problems. Configuration management is one such technique. It provides procedures for generating, controlling, and updating the information contained in technical specifications, and for documenting the status of such information.

This appendix describes configuration management procedures for those elements of a system which are computer programs. Included is a discussion of configuration management within the development cycle of a computer program and, either directly or by reference:

- a. a standard for the format and content of a technical specification for a computer program;
- b. a description of a numbering system for the identification of computer programs and specifications;
- c. procedures for the control, documentation, and dissemination of engineering changes to technical specifications;

- d. procedures for technical reviews and inspections during the development cycle;
- e. procedures for documenting the status of a computer program and its specifications.

## 2.0 CONFIGURATION MANAGEMENT OF COMPUTER PROGRAMS - A GENERAL DISCUSSION

The key concept of configuration management is the generation, as information becomes available, of increasingly more detailed specifications describing the system. Before configuration management can be applied, it is necessary to develop a specification that defines the functions of the system. Based on an analysis of the requirements within the "system specification", detailed functional requirements for each system element are specified. The fundamental element so specified is called an "end item" or more officially a contract end item (CEI). It is the basic element for the procurement of contractor services and subsequently for control of contractor activities by the procuring agency.

A large system may consist of many types of end items; e.g., mechanical assemblies, facilities, electronic equipments (including computers), and computer programs. Originally configuration management procedures were concerned mainly with "hardware" end items. However, the use of "computer based" systems--those with digital computers as controlling elements--has increased to the point where management control of computer programming end items has become a prime concern in itself.

### 2.1 The Computer Program Contract End Item

A computer program is the ordered set of instructions and data required to control the operation of a digital computer. The end product of the process required to produce a program is usually a punched deck of cards, magnetic tapes, or other physical media containing the ordered set in a form suitable for insertion into a digital computer. Under control of the instructions, the computer acts upon the data to perform a set of well-defined and logically related functions. A computer program contract end item (CPCEI) is defined in this report as a computer programming end product whose development, as designated by the procuring agency, is subject to configuration management.

A computer programming effort usually requires the development of several programs and the data essential to their use. A particular application (e.g., automatic checkout of a launch vehicle, on-board guidance and navigation) may require the development of:

Operational program(s) which perform(s) the job the user wants done (e.g., generation of test-stimuli, trajectory calculation).

A data base which contains all the static and dynamic data supplied to the operational program (e.g., launch site coordinates, mathematical constants).

Support programs used for the test and evaluation of the operational program (e.g., simulation data recording, data reduction programs).

Utility programs which include all tools necessary for the generation of the operational and support programs (e.g., compilers, assemblers, monitors, debugging aids).

The programs and data form a "programming system." For both technical and managerial reasons it is usually desirable to partition the total programming system into several CPCEI's. For example, the operational program combined with the data base may be one CPCEI. A support program used to test this CPCEI may itself be a CPCEI. The set of utility programs may also be designated as a CPCEI. This selection is done by the procuring agency or definition phase contractor in the definition phase. It results in the allocation of functions to separately identified computer programs, each of which can then be conveniently produced and managed as a unit.

## 2.2 Configuration Management in the Programming Process

The totality of activities required for the production of a working documented computer program is called the "programming process." Configuration management starts early in, and continues throughout, this process.

For convenience the process can be described in terms of the life cycle of the system in which the program operates. This life cycle can be considered to take place in four phases:\*

---

\*This concept conforms to NASA requirements for phased planning prior to the commitment of significant resources in a project (see reference document 8). The description of phases used in Apollo configuration management (1) predates the description of phases as defined in (8); hence, they differ slightly. The intent, however, is the same.

The conceptual phase which begins with the realization that a problem exists--a problem that can be solved or alleviated by a new system. The requirements for the system are defined and a systems concept is developed.

The definition phase in which all required end items are identified, performance requirements for each end item are generated, and contractors are selected to produce the end items.

The acquisition phase in which the end items are designed, developed, produced, and documented.

The operational phase in which the end items are integrated to form the system, and during which the system is used to perform its operational mission.

The programming process for a CPCEI within the system starts in the definition phase and continues throughout the acquisition and operational phases. A generalized description of this process follows, showing the relationship between configuration management and the other activities of the programming process (see Figure 1).<sup>\*</sup> In this description it is assumed that a higher level specification, describing the computer-based system, has already been produced. Specific configuration management activities are underlined.

- a. Definition Phase - In this phase a detailed analysis is made of system requirements contained in the computer-based system specification, resulting in tradeoffs between what is to be done by hardware, computer programs, and personnel. The required operational, support, and utility programs are identified. For each identified CPCEI a requirements document is produced which defines in detail the functions it is to perform, its interfaces with other end items, the design constraints and practices to be followed in its development, and requirements for its qualification. This document is called the CPCEI Part I Specification.

If the Part I Specification is produced by a contractor, it is reviewed by the procuring agency. Upon satisfactory completion of this review, the Part I Specification is established as the design requirement baseline for use in the acquisition phase. Its initial function is to govern the

---

<sup>\*</sup>See also reference document 4, section 3.1 and Figure 1 for a more detailed description of milestones in the programming process.



design, development and testing of computer programs throughout the acquisition phase. Its continuing function throughout the system life cycle is to serve as a reference against which the impact of each proposed design change is assessed.

- b. The Acquisition Phase - The activities of the acquisition phase result in the development of a fully assembled CPCEI that has been qualified in a simulated environment prior to integration into the computer-based system. It can be conveniently divided into two subphases:

1. The Design Subphase

The design process begins with a definition of the structure of the computer program as a whole, in terms of functions allocated to individual subprograms, storage allocation, real-time apportionment, computer program operating sequences, and the format of the data base. The preliminary design approach is reviewed at the Preliminary Design Review (PDR) to determine if the requirements of the Part I Specification are met by the design. Concurrence in this design is required before the contractor can proceed to the detailed design of the computer program components (CPC's) which compose the CPCEI. The design of each CPC or convenient groups of CPC's is then documented in a programming specification which is sufficiently detailed to enable a programmer to code and debug the CPC. Each CPC design is then reviewed by the procuring agency, in a Critical Design Review (CDR), to determine if the CPC will function as required. Concurrence in this design is required before the contractor can proceed to code the CPC. Other activities in the design subphase include the development of a test plan based on the quality assurance requirements of the Part I Specification, development of a programming standards manual based in part on the design requirements of the Part I Specification, and the initiation of formal change control and accounting of the Part I Specification.

2. The Implementation Subphase

During this subphase the contractor codes and debugs all CPC's, produces detailed test specifications, tests the CPC's in groups until all CPC's have been assembled together, demonstrates the validity of the

CPCEI by means of a preliminary qualification test in a simulated environment, and completes the documentation of the CPCEI. The completed technical description is documented in the Part II Specification which contains the description of the overall design, the programming specifications, flow charts, and listings. Other documents such as operator handbooks, progress reports, and PERT reports are produced as appropriate in the subphase. The terminating milestone for the implementation subphase is First Article Configuration Inspection (FACI). The primary objectives of FACI are to determine if the CPCEI is properly marked and documented, and to establish the completed Part II Specification as the product configuration baseline. The Part II Specification then serves in the operation phase as an instrument for use by government or contractor personnel in diagnosing troubles, adapting the computer program to the environmental and operating requirements of specific site locations, and designing minor changes to the CPCEI.

- c. Operation Phase - The operation phase commences after FACI and continues for the life of the CPCEI. In this phase the CPCEI is integrated with the hardware and personnel of the computer-based system, qualified and/or formally accepted in the system, and maintained between uses. The CPCEI may be modified for use in several installations or for use on a mission-by-mission basis. Change control and accounting for both the Part I and Part II Specification is affected throughout the operational phase.

The activities of the operation phase can often be described in three subphases:

1. Subphase 1 - System Integration, in which system tests are made for integrating the hardware, CPCEI, system environment, and operating personnel into a working total system. CPCEI corrections and modifications, adaptation of the data base to operational installations, and updating CPCEI documents, are done as appropriate. The CPCEI is formally accepted after demonstration that it fulfills the requirements of the Part I Specification. Finalized mission data are loaded into the program.

2. Subphase 2 - Use of the Complete Program in Its Intended Mission.
3. Subphase 3 - Maintenance of the CPCEI after the system becomes operational to meet changing user requirements.

### 2.3 Selection of CPCEI's

The term "CPCEI" has been used synonymously in this appendix and throughout this report with computer program. In the Apollo program there is, however, a further connotation in the use of "CPCEI." In the narrow sense a CPCEI is a computer program whose development is subject to the procedures of configuration management as defined in the "Apollo Configuration Management Manual" (reference document 1). In this sense the CPCEI is typically a computer program which is one of many elements in a large system. In a project such as Apollo, however, many of the required computer programs do not fall into this category. For example, a trajectory simulation program may work only within a computer and not with other equipment used during a mission. There is then some leeway in selection of which programs are to be identified as CPCEI's. The designation of which computer programs are to be subjected to the formal requirements of reference document 1 is left to the discretion of the procuring agency.\* The designation is made on the basis of criticality, cost, and need for formal control, subject to the following guidelines:

- a. At a minimum, operational programs used in direct real-time support of a mission (including training and pre-launch checkout) should be classified as CPCEI's.\*\*
- b. Certain non-real-time or offline programs, such as support and utility programs associated with a CPCEI, or data processing programs used for telemetry reduction, may be designated as CPCEI's if:
  1. The status of the program directly affects mission schedules;

---

\*It should be emphasized here that the elements of configuration management (e.g., technical specification, change control) are applicable to all computer program development efforts, whether the programs are identified as CPCEI's or not.

\*\*In this context, "direct real-time support" means that the results of computer processing are used to monitor or control the course of the mission or training exercise.

2. Changes to the configuration of the program directly affect the configuration of designated end items; or
3. The programming effort requires a large expenditure of resources. In this case it may be deemed desirable to achieve a high level of management control through the use of configuration management procedures.

Consideration must also be given in the identification process to the multimission use of some computer programs. Many systems rely on the flexibility of programmable digital computers to support a rapid succession of diverse missions. The degree to which a program, performing the same general function on a mission-to-mission basis, changes for each mission use may vary greatly. In some cases only data may change; in other cases slight program revisions may be required; in still other cases extensive modification may result. Hence, it is difficult to establish rigid rules to determine if a program used in a particular mission should be classified as a new CPCEI or should be considered as a change to an existing CPCEI. The following guidelines, however, should be followed in determining whether or not a program is to be re-identified as a new end item (i.e., given a new identification number and defined in a new specification):

- a. A program originally used in a particular mission may be modified and used without re-identification for a succeeding mission if:
  1. changes to its design are minor enough so that new design reviews are not required; and
  2. the existing specification can be conveniently modified by the use of specification change notices (SCN's);\* and
  3. the original program will not be used again.
- b. A program should be re-identified for use in succeeding missions if:
  1. new design reviews are required, or
  2. new subfunctions requiring additional CPC's are required, or

---

\*See page 25.

3. a change of computing equipment causes a significant reprogramming effort, or
4. a new specification is required to incorporate extensive modifications made over a period of time, or
5. slight changes are required, but the original version of the program will be used again.

Since some part of the program may be similar from mission to mission, it may not be necessary to produce a completely new specification for each mission. A specification addendum referenced to some previous specification may be sufficient for a particular mission (see reference document 3, section 6.3.3).

In some instances it may be desirable to identify different portions of a program as separate end items. An executive routine, for example, may be mission independent, while the application programs it controls may change greatly from mission to mission. A single specification may, therefore, be adequate for the development of the executive, while a series of specifications may be needed for the test programs. The identification of the executive and test programs as separate end items would facilitate management control of this type of effort.

In all cases, it is the procuring agency's responsibility to identify which programs are new end items. The requirements of configuration management are such that some degree of flexibility is allowed in this process to account for differences in the nature of computer programming tasks.

### 3.0 CONFIGURATION MANAGEMENT REQUIREMENTS

The procedures for configuration management can be classified in three areas: configuration identification, configuration control and configuration accounting.

#### 3.1 Configuration Identification

A CPCEI is identified by a detailed specification and by a set of identification numbers.

### 3.1.1 CPCEI Specification

The format and content for a CPCEI Specification are given in reference document 3.\* The specification is divided into two parts to reflect the dichotomy of its purpose.\*\* Part I is produced as a result of the requirements analysis in the definition phase; it defines the requirements which the contractor's design must satisfy. Part II describes the product that is produced in the acquisition phase to fulfill the requirements in the Part I Specification; it serves as the detailed technical description of the CPCEI.

The specification is a collection of documents that are (or should be) produced separately and in increments during the programming process. The Part I Specification contains performance requirements, design constraints, interface specifications, and test requirements, all of which are essential inputs to the contractors design efforts. The Part II Specification contains the overall design approach, programming specifications, flow charts and program listings--information which is needed for the control of the design process and eventually to describe the completed CPCEI.

The use of a single specification with two parts is one of several ways in which this information could be published. Separate technical memorandums and reports could be used, as is done in many efforts. As a single document, however, the information can be produced, maintained, updated, and revised according to well-defined procedures. The issuance of a single specification ensures that the vital information contained within will always be correlated and not lost, as is the potential with separate documents.

---

\*The specification standard given in reference 3 is derived from the standard used in NASA configuration management (reference document 1 - Exhibit XVIII). Except for some modification in terminology, it is also similar to the standard used in U. S. Air Force configuration management (reference document 2 - Exhibit XX). This standard has been made general enough to be applied to any computer programming effort.

\*\*This division of a specification also exists for hardware CEI's. It is interesting to note that while the role of the Part I Specification is similar for both computer program and hardware CEI's (i.e., they both define requirements for end item performance) the roles of the respective Part II Specifications differ. For hardware CEI's the Part II Specification is used as a "build to" specification to control the production of succeeding copies of original items. For CPCEI's it is a description of the product as built and is used for the purpose of maintaining the CPCEI in the operation phase.

It should be noted that the use of a single specification is a management concept rather than a physical concept. The specification may be issued as several physical volumes if its content is too extensive to be included between a single set of covers.

### 3.1.2 Identification Numbers and Markings

Numbers must be assigned to an end item, its parts, and its specification to assure that these items may be properly identified. The selection of a numbering system is somewhat arbitrary except that it must be consistent and provide unique identifiers for all items. The detailed requirements for the numbering system used in NASA are given in reference document 1, Exhibit X.

The required numbers include:

- a. The specification number consisting of a prefix code, a document identifier number, and a suffix code indicating the latest approved version of the document.
- b. An end item number. This is the unique reference number by which the end item can always be identified. It is assigned in the definition phase as soon as the end item is identified. The end item number consists of a unique end item identifier plus a series number. The series number is used to distinguish different versions of a CPCEI which perform the same general function in several missions (see page 10).
- c. A number for each identified "part" within the end item. For a CPCEI, a "part" is defined as the card decks, magnetic tapes, etc., in which the program is contained. A physical part as defined above may in many cases correspond to a logical part, a computer program component (CPC). To facilitate identification:
  1. Each card deck containing a CPC is given a header card and a trailer card. Each header and trailer card contains the name and number of the CPCEI and the name and part number of the subprogram. These cards are treated as comment cards by the assembler or compiler; that is, they have no program functions other than identifying a particular card deck.

2. Each card within a card deck is marked by a printed CPCEI number and a punched sequence number identifying its location within the deck. In a large enough project it is recommended that the CPCEI numbers be preprinted on the cards for convenience. The punched sequence number not only serves as a unique card identifier (when coupled with the CPCEI number) but also allows the computer or ADP equipment to be used in ordering a "shuffled" deck of cards.
3. Each CPC card deck is color-coded for easy identification. The band or case of each sub-program deck is marked by the CPCEI number and part number.
4. Each tape, reel tape cannister, disc-pack, or other storage medium used is marked with the CPCEI number and part number. Where appropriate, the information is included in machine readable form, for example, as a tape header block.

### 3.2 Configuration Control

Configuration control is the establishment of technical control points called "baselines" and the systematic evaluation, coordination, and disposition of all proposed changes to these baselines.

#### 3.2.1 Baseline Management

A baseline is a collection of information, as known up to a particular instant of time, describing the technical characteristics--performance, design, configuration, testing--of a system or system element. Baselines are established at discrete points in the life of a system as references for the evaluation of proposed changes to the technical characteristics. The totality of baselines plus approved changes provides an up-to-date description of the system.

- a. The Design Requirements Baseline - The Part I Specification is established by the procuring agency at the beginning of the acquisition phase as the design requirements baseline (DRB). Once so established, it becomes the controlling document for the design and testing activities of the



contractor. Any change or addition to it must be submitted as a design requirements change and must be formally approved before the change can be implemented by the contractor. The DRB functions throughout both the acquisition and operation phase, although usually the volume of requirements changes will diminish in the operation phase. All approved changes to the DRB must be reflected in supporting technical documentation and in the end item itself. All proposed changes are formally submitted as engineering change proposals (ECP's).\*

- b. The Product Configuration Baseline - The Part II Specification is established as the product configuration baseline after successful completion of the First Article Configuration Inspection (FACI). At this time, Part II is audited to determine if it adequately describes the fully-assembled CPCEI. From this point on, all changes to the CPCEI as reflected in the Part II Specification must be formally accounted for and controlled.

### 3.2.2 Change Control

The term "change" refers to any alteration to an established baseline. It is implied that any change will be reflected in the CPCEI and its relevant documentation. Change control by the procuring agency is established by identifying in advance the types of changes which are considered critical, requiring the contractor to submit a formal engineering change proposal (ECP) for each critical change, establishing a Change Control Board (CCB) which evaluates each critical change, and disseminating the results of each CCB decision.

- a. Change Classification - Changes to an established baseline are considered to be either of two types, Class I or Class II.

---

\*See Page 18.

Class I changes are those which, because of their criticality, require formal procuring agency approval before a contractor can effect them. Changes are designated as Class I whenever one or more of the following is affected:

1. Operational capability as specified in the base-lined Part I CPCEI Specification.
2. Contract price or schedule.
3. Systems equipment, computer programs, or facilities produced by other contractor(s) to the extent that the affected other contractor(s) must accomplish a change to maintain compatibility at the interface(s).

A Class II change is one which the contractor may affect without prior approval by the procuring agency. Such changes may include:

1. Changes to correct editorial errors.
2. Changes to correct computer program errors.
3. Other changes of a minor nature within categories specifically defined by the procuring agency, e.g., adaptation data or recompiling within specified limits.

The criteria given above are derived mainly from change processing requirements given in ANA Bulletin 445 (5). The major difference is in the classification of computer program error changes as Class II changes. Under strict interpretation of ANA Bulletin 445, a computer program change of any type after FACI would be classified as a Type I change, because the corresponding product configuration baseline would also have to be changed. In a computer programming effort, the number of these changes can be large even after FACI. The contractor must be given leeway in making such changes without having to wait for the approval of the procuring agency. To avoid unnecessary delays, these changes may be considered as Class II even though the product configuration baseline must also be changed. The contractor must update the Part II Specification to reflect the change and notify the procuring agency of the change.

- b. Change Processing - The contractor initially classifies all changes that he proposes. The initial determination is made according to guidelines given above and is subject to review by a representative of the procuring agency.

Class I changes are documented in an engineering change proposal (ECP) and submitted to a Change Control Board (CCB) composed of personnel from the procuring agency. The content of an ECP is based on necessary design study, determination of total impact, and coordination with the appropriate other participating agencies. Following review, a Change Control Board Directive (CCBD) will be issued to notify affected agencies of the CCB approval or disapproval of the proposed action. The CCB must take action within some reasonable time, say five days, on each change. In general, no Class I change can be made by a contractor unless an approving CCBD is issued. In some cases, expedited approval may be required. This occurs when a problem, identified during system functional checks or installation at a field site, necessitates an immediate change to the computer program. These cases are reported by the most expedient means (teletype, telephone, personal contact, etc.) to the procuring agency. Implementation of the computer program change by the contractor can be initiated immediately following receipt of an expedited approval from the procuring agency. Following this initial approval, a confirming CCBD will be issued. In response to the official directive, the contractor will prepare an "after-the-fact" ECP for submission to the CCB.

Class II changes may be reported on the contractor's own form, providing it includes, as a minimum, a reference to the baseline specification(s) affected, a description of the change, and a justification of the change. The change report is submitted to the designated Government representative for formal review of the proposed classification. The review must be performed expeditiously to allow the contractor to continue his work. A Class II change can be implemented immediately upon approval of its classification. Any disapproved Class II classification may be followed by preparation and submission of a Class I ECP, either at the discretion of the contractor or as directed by the procuring agency.

An exception to the above procedure must be made in the case of Class II changes which are error corrections to the CPCEI. To prevent undue delay, all error corrections may be implemented prior to the approval of the procuring agency. The procuring agency, at its discretion, can then review these changes on an "after-the-fact" basis to determine if any of the error corrections are in actuality Class I changes.

- c. CCB Functions - The procuring agency CCB is responsible for approving/disapproving contractor-submitted changes, for disseminating change information to all concerned parties, and for transmitting CCB's to the contractor, based on approved changes received from other end item contractors. The CCB is a technically-oriented organization whose personnel must have sufficient knowledge of both the system and the computer programs. It is not a voting board, i.e., the chairman has final say over all actions and the rest of the board serves as advisors.

In a large system there may be several CCB's arranged in a hierarchical order. A single CCB may have responsibility for several end items. The arrangement of CCB's must be determined by the agency responsible for the entire system, prior to entering the acquisition phase for the system. Although many CCB's may exist in a system, the CPCEI contractor should be "aware" of only one, the one that has cognizance in his area.

- d. Engineering Change Proposal - The engineering change proposal (ECP) is used for transmitting information essential to the evaluation of proposed Class I changes. An ECP form is prepared by the contractor and submitted to the procuring agency CCB. The nature and impact of the proposed change must be defined within the ECP to the extent necessary to evaluate the change. This information is based on design study and, as appropriate, coordination with other agencies.

The ECP may, in many situations, have to be submitted in two steps, preliminary and formal. A preliminary ECP provides information required for obtaining authorization to perform the work necessary to make the change and to test the change as appropriate. A formal ECP provides the information required to obtain approval to put the change in the system. A formal ECP is accompanied by a specification change notice (SCN)\* for the affected part(s) of the CPCEI Specification.

The minimum content of the ECP should contain:

1. Change identification information including contractor's name, the affected specifications, change identification number, priority classification, and date of submittal.
2. Descriptive information including a description of the change, justification for the change, reference as appropriate to other change documentation, further analysis effort required, and allowable tradeoffs in making the change.
3. Status and action requirements including estimated completion dates, estimated costs, date by which action is needed, and effects of delays in approval.
4. Impact on supporting items, interfaces with other end items, and impact on the structure and operation of the CPCEI.

A detailed description of the format, content, and use of an ECP is given in section 6.4 of the Addendum to Exhibit IX of reference document 7.

### 3.2.3 Technical Reviews and Inspections

Technical reviews provide checkpoints on the contractor's design and implementation efforts. Reviews are required to determine if the contractor's design efforts will satisfy the design requirements baseline and to provide the basis for

---

\*See page 25.

establishing the Part II Specification as the product configuration baseline. Formal reviews serve as control points, since procuring agency approval of the reviewed material must be obtained before the contractor can proceed to his next major activity. Although reviews may be held at any time, three specific reviews are formally required for the purposes of configuration management: the Preliminary Design Review (PDR), the Critical Design Review (CDR), and the First Article Configuration Inspection (FACI). In addition, a fourth (Post-FACI) review may be necessary.

- a. The Preliminary Design Review (PDR) - The PDR gives the procuring agency an opportunity to determine early in the programming process if the contractor is designing a product that actually satisfies the requirements of the Part I Specification. The output of the review is either concurrence by the procuring agency of the design approach or a set of action items to be acted on by the contractor.

The PDR is held when the contractor's design activity has progressed to the point where functions have been allocated to individual computer program components (CPC's) and flow charts showing the data flow between all CPC's have been produced. This occurs early in the design subphase of the acquisition phase. The following is accomplished in the PDR:

1. The compatibility of the selected design approach with Part I of the CPCEI Specification is established. This is accomplished by review of flow charts, storage allocation charts, timing estimates, descriptions of significant algorithms, and other engineering documents as required by the procuring agency. An initial presentation of the information required in section 3.1 of the Part II Specification may be used as the basis for this review (see reference document 3).

2. The integrity of the design approach is established. This is accomplished by review of algorithms, proposed programming techniques, design simulation results, estimated storage requirements, etc.
  3. The compatibility of the CPCEI with other computer programs, or equipment external to the computer in which the CPCEI is used, is established. This is accomplished by review of proposed data formats, timing constraints, interface drawings, and other systems engineering documents as determined by the procuring agency.
- b. Critical Design Review (CDR) - The CDR provides a control point for evaluating the detailed design of the CPC's identified in the approved design approach. For a large program, several CDR's may be necessary since the individual CPC's may be designed over a period of time. It is the responsibility of the procuring agency to schedule the CDR's so that the contractor's design activities can be expedited; to this end several CPC's may be reviewed at a single CDR.

A particular CDR is held when the inputs, outputs and computational methods have been determined for a (group of) CPC(s). The prime goal of the CDR is to establish concurrence between the procuring agency and the contractor on the design of the CPC(s); this is to be done before significant resources are committed to their development. The following is accomplished as part of a CDR:

1. The compatibility of the CPC design with those functions allocated to it in the approved design approach is investigated; the integrity (i.e., feasibility) of the design is evaluated. Initial versions of material that will eventually be incorporated into section 3.2 of the Part II Specification may be used as the basis for evaluation (reference document 3).\* Other information

---

\*This information (data formats, computational methods, etc.) may be initially published by the contractor as a "programming specification" to be used by a programmer after the CDR as the basis for flow charting, coding, and testing the CPC. Updated versions of this material plus the resultant program listings are incorporated into the Part II Specification when it is published.

may be required, including test results for selected algorithms and results from simulations of the proposed design.

2. The system compatibility of the CPC design is established by comparison of applicable interface control documents.

The result of a CDR is either approval by the procuring agency of the CPC design or a set of action items requiring redesign effort by the contractor.

- c. First Article Configuration Inspection (FACI) - The term "FACI" is a carry over from configuration management of hardware production items. The FACI in hardware development is the inspection of the first of a series of articles to determine if its configuration agrees with the description in the Part II Specification, if it is properly marked, and if all supporting documentation is up-to-date. A primary result of a successful FACI is establishment of the Part II Specification as the product configuration baseline, which is then used to control the production of succeeding articles.

Although the reproduction of copies of a CPCEI is usually not a significant activity in the programming process, and despite the somewhat awkward terminology, a FACI is an appropriate and necessary milestone in the programming process. A well run FACI can help eliminate many of the problems associated with the transfer of a computer program from a developmental environment to an operational environment, particularly if this involves a change in geographical location.

It provides a check that the program is adequately documented and has been successfully tested before it can be transferred and integrated with the computer based system.

In general, FACI is held upon completion of the Part II CPCEI Specification and preliminary qualification testing of the CPCEI in a simulated environment. If the CPCEI is developed at the contractor's facility for subsequent shipment to the user's facility, FACI should be held prior



(and prerequisite) to this shipment. If the CPCEI is developed at the user's facility, FACI should be held prior to final qualification testing of the computer based system (see reference document 6 for a discussion of qualification testing).

The specific documents and computer program item(s) to be reviewed, and the detailed actions to be accomplished at a given FACI, are determined by the contractor and procuring agency at the time the agenda for the review is established. The following is accomplished as a part of the FACI for a CPCEI:

1. The configuration of the CPCEI, as evidenced by identification markings on the involved tapes and card decks, is compared directly to the Part II CPCEI Specification. The existence of each program component (i.e., subprograms) and part (tape, card deck, etc.) of the implemented CPCEI is verified.
2. The completeness of the Part II CPCEI Specification is verified (i.e., Part II must be complete and must describe the CPCEI to the procuring agency's satisfaction). All changes resulting from FACI must be incorporated before Part II is approved. The Part II Specification when approved becomes the product configuration baseline.
3. The results of preliminary acceptance testing (tests of the entire CPCEI in a simulated environment) are audited to determine:
  - i. the status of the CPCEI as a working system element,
  - ii. the validity of the qualification test results,
  - iii. if all modifications resulting therefrom have been incorporated in the Part II CPCEI Specification.

4. Where the procuring agency has ordered delivery of specific configuration management records of the CPCEI, such data is audited by direct comparison to the information contained in the Part II CPCEI Specification.
5. The contractor's computer program release system and change control procedures are reviewed and evaluated.

The result of FACI is either the approval of the Part II Specification as the product configuration baseline and approval to transfer the CPCEI to an operational environment for integration, or a set of action items that must be acted on by the contractor to obtain this approval.

- d. Post-FACI Reviews - the procuring agency may require a series of post-FACI reviews. These reviews are held to:
  1. ensure that all changes to the CPCEI since FACI have been properly incorporated and accounted for;
  2. determine if the CPCEI can be certified as ready for use in a mission. This is accomplished by review of any major design changes since FACI and by review of final qualification test results.

The manner, type and number of these reviews is left to the discretion of the procuring agency; as a minimum, one review shall be held prior to use in a mission. The end result of this review is acceptance of the CPCEI for mission use. This review is usually held as part of an overall review for the system in which the CPCEI operates.

### 3.3 Configuration Accounting

Configuration accounting, as defined in reference document 1, is the reporting and documenting of all proposed changes to an established baseline configuration. Accounting procedures insure that the baseline is maintained in an accurate and timely manner. In the computer programming process these procedures

apply to the maintenance of the specification, the CPCEI itself, and are extended to cover the maintenance of significant supporting documents (handbooks, user manuals, the test plan, etc.).

In order to maintain the above mentioned items, the contractor must prepare and update change documentation, keep an index of all maintained items and a description of the components of the CPCEI.

The detailed procedures for configuration accounting must be specified by the procuring agency before the contractor initiates his effort. These procedures are left to the discretion of the procuring agency. The procedures used in NASA configuration accounting are given in reference document 1; those used in Air Force configuration accounting are given in reference documents 2 and 7. The accounting procedures as stated in these references are summarized below.

#### 3.3.1 Specification Change Notice (SCN)

The specification change notice (SCN) is used to record approved changes to established baselines. The SCN identifies changes to the baselines, e.g., add/delete a page; the actual changes are attached to the SCN in the form of change pages. An SCN is required on all Class I changes and must accompany a formal ECP.

The SCN is also used to transmit changes such as misspellings, typographical errors, etc. However, changes of such a minor nature are usually transmitted as an incidental part of the next occurring SCN. A suggested SCN format and detailed description for its use is given on page 3 of Exhibit XXI in reference document 7.

#### 3.3.2 Specification Change Log (SCL)

A specification change log (SCL) lists all SCN's by date and identification number. A SCL is usually attached as a cover sheet to each newly generated SCN to provide a means of verifying to the recipient that his specification contains all approved changes. A suggested format and directions for use of the SCL are given on page 5 of Exhibit XXI in reference document 7.

### 3.3.3 End Item Configuration Chart

A complete revision of an approved specification may be issued when the volume of changes to the previous version becomes too great to conveniently handle. The end item configuration chart is a summary record which identifies all approved ECP's which are incorporated in a specific revision of a CPCEI Specification. It is issued by the contractor only as part of a revision of the specification. (See Page 5 of Exhibit XXI in reference document 7.)

### 3.3.4 Configuration Index

The Configuration Index contains an official listing of the CPCEI Specifications and all significant support documents which require maintenance as a result of approved changes to the CPCEI. The initial issue contains one section listing all volumes of the first issue of the Part I Specification. Subsequent issues of the index show revisions of the Part I Specification, and in additional sections show the major documents dependent on the Part I Specification which must be maintained to reflect all approved changes. Each succeeding issue of the Index is expanded to include the latest document revision and all ECP's which have been incorporated or have been approved but not incorporated in a published version. Index sections are deleted when the document it describes has outlived its usefulness. A suggested format and instructions for its use is given on page 6 of Exhibit XXI in reference document 7.

### 3.3.5 Change Status Report

The change status report details the status of all proposed changes to a CPCEI. It includes contractor identification and a list of each ECP with a status summary. The change status listing is used to periodically inform the procuring agency and the contractor of the current status of all proposed ECP's. It and the configuration index are supplemental and taken together contain the current status of the CPCEI and its related changes. A suggested format of a change status listing and instructions for its use are given on page 9 of Exhibit XXI in reference document 7.

### 3.3.6 Version Description Document

A version description document is used to accompany the release of a CPCEI or any part thereof. It contains the identification of the computer program element delivered and includes pertinent additional information relating to status and usage. This includes:

- a. element identification (by name and number);
- b. inventory of material contained (e.g., tapes, card decks of the CPCEI, utility programs, support programs, documentation);
- c. a list of all Class I and II changes;
- d. interface capability and adaptation instructions;
- e. operational description; and
- f. installation instructions.

A suggested format and instructions for its use are given on page 11 of Exhibit XXI in reference document 7.

### 3.3.7 CPCEI Maintenance

In addition to generating change paper for each error correction (after FACI), the following accounting procedures are required to protect the CPCEI and ensure that the Part II Specification adequately reflects the configuration of the CPCEI.

1. Each time a part (subprogram card deck, master program tape, permanent memory, etc.) is modified and successfully reassembled or recompiled, the suffix of its part number must be increased by one. In this manner the suffix serves as a version number for the part. A modification may encompass several program changes and may be reported at periodic intervals to minimize paper work.

2. The appropriate documents (including the listings) must be updated to reflect the number and configuration of the modified part.
3. Where appropriate, one master card deck and two copies of the latest version of the CPCEI master tapes shall be maintained. In addition, the two previous versions of the master tape(s) shall be maintained with appropriate listings.

REFERENCE DOCUMENTS

1. NPC 500-1, "Apollo Configuration Management Manual," current version.
2. AFSCM 375-1, Air Force Systems Command, current revision.
3. Appendix III of this report, "Preparation of Contract End Item Detail Specification (Computer Program)."
4. Appendix II of this report, "Milestone and Progress Reporting Requirements of Computer Programming."
5. ANA Bulletin 445.
6. Appendix V of this report, "Testing Requirements of Computer Programming."
7. ESD Exhibit EST-1, "Configuration Management Exhibit for Computer Programs," U. S. Air Force Systems Command, L. G. Hanscom Field, Bedford, Massachusetts.
8. NASA Directive NPD 71-21.1, October 28, 1965.

## MAJOR MILESTONES

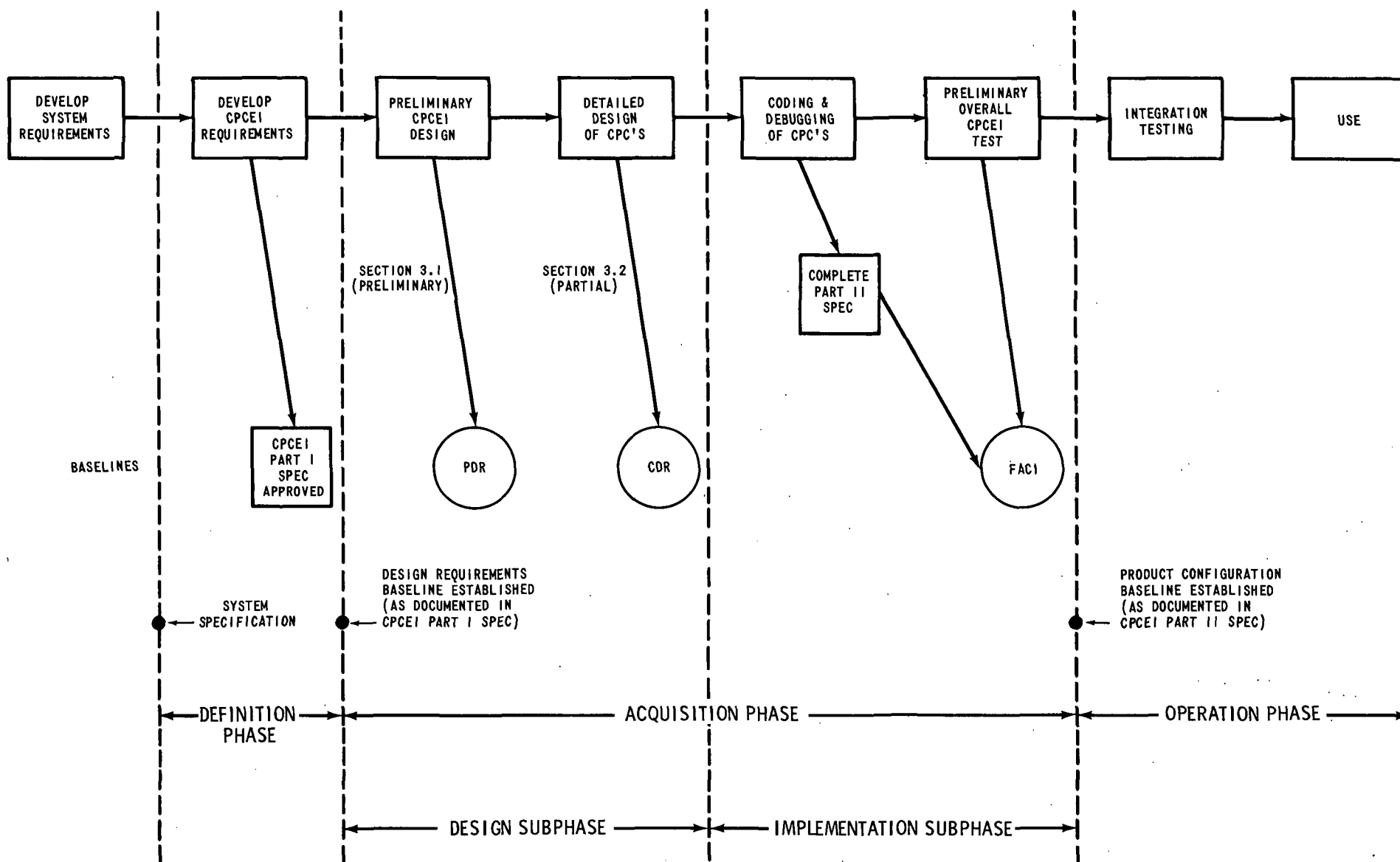


FIGURE 1 MAJOR MILESTONES COMPUTER PROGRAM CONFIGURATION MANAGEMENT



BELLCOMM, INC.

APPENDIX V

TESTING REQUIREMENTS OF  
COMPUTER PROGRAMMING

TABLE OF CONTENTS

- 1.0 INTRODUCTION
- 2.0 TYPES OF TESTS IN THE PROGRAMMING PROCESS
  - 2.1 VALIDATION AND VERIFICATION TESTS
  - 2.2 QUALIFICATION TESTS
  - 2.3 CRITICAL TESTS
- 3.0 TEST MANAGEMENT
  - 3.1 ESTABLISHMENT OF TEST REQUIREMENTS
  - 3.2 TEST PLANNING
  - 3.3 TEST SPECIFICATION
  - 3.4 TEST REPORTING
- REFERENCE DOCUMENTS
- ATTACHMENT I
- ATTACHMENT II
- ATTACHMENT III

## TESTING REQUIREMENTS OF COMPUTER PROGRAMMING

### 1.0 INTRODUCTION

Testing of a computer program contract end item (CPCEI) is a critical function within the programming process. Tests are used to: 1) determine if the components of a computer program contract end item (CPCEI) are implemented correctly; 2) determine if the CPCEI satisfies the requirements of its Part I Specification; 3) obtain test results that are used to determine if certain milestones have been achieved; and 4) formally qualify the completed computer program for operational use.

Certain tests require extensive preparation and resources. If they are not planned concurrent with the design effort itself, they may cause an impact on schedules and costs. Although the activities of planning, documenting and implementing tests are usually the responsibility of the contractor, the procuring agency can do much to assure that they are successfully accomplished. To this end, this appendix describes test requirements that can be imposed on a contractor in a programming effort. Specifically contained herein are descriptions of: tools for establishing test requirements, the types of computer program tests that can provide the basis for test planning, the test plan and test specification documents, and guidelines for planning testing in the programming process.

### 2.0 TYPES OF TESTS IN THE PROGRAMMING PROCESS

Tests are applicable at many levels and to several stages in the programming process. The exact tests that are required by the nature of a particular task should be determined by the contractor, with direction from the procuring agency.

This section describes tests that yield status information and occur at significant points in the programming process. These tests can be considered in two categories; those used in support of program development, and those used for formally qualifying the program for operational use. Several of the tests within each category may be identified for close management control by the procuring agency. Therefore, a third category - critical tests - consisting of tests from the other two categories is defined.

## 2.1 Validation and Verification Tests

Several stages of testing are required to validate the contractors design (i.e., determine if it meets the requirements of the CPCEI Part I Specification) and to verify that the implementation of the design is correct.

### (1) Pre-Implementation Design Tests

If the program is sufficiently complex, tests may be run on trial designs prior to establishing an initial design approach. Typically these tests indicate real-time performance characteristics, computational accuracies, storage limitations, etc. Tests of this type may continue throughout the design process.

### (2) Subprogram Checkout

As each individual subprogram of the CPCEI is coded, it may be checked for errors. First, the programmer desk checks the code by visual inspection and by hand manipulation with selected data. This is followed by assembling the subprogram on the computer. Each assembled subprogram is then tested by using controlled data inputs. The goal of this stage is to identify and remove errors prior to combining subprograms into "packages".

### (3) Subprogram "Packages" Checkout

Once the individual subprograms have been verified, they are consolidated into functionally related subprogram packages. Packages are first debugged as a whole, with no inputs, to establish that they can cycle. Then, controlled inputs are introduced to test the packages' ability to respond properly. The purpose of this stage is to remove coding errors from the interfaces between subprograms. There may be several levels of testing within this stage, corresponding to the levels of subprograms within the CPCEI.

### (4) CPCEI Simulated Environment Tests

The CPCEI may be tested in a simulated environment prior to its integration into the total computer-

based system. This may be done for several reasons:

- a. to obtain a more controlled test of the CPCEI than could be accomplished in the total system;
- b. to determine the safety of the CPCEI in the system without exposing the system to unnecessary hazard;
- c. the system may not be available for use;
- d. to serve as the basis for preliminary qualification of the CPCEI prior to transferring it from a development facility to a using facility;
- e. to provide preliminary training in the use of the CPCEI and to evaluate proposed operating procedures for the CPCEI.

Such tests are contingent upon the availability of a system environment simulator and associated test support tools. The development of these items may be quite costly and time consuming; such development must be planned as carefully as the development of the CPCEI itself.

#### (5) System Integration Tests

Once the CPCEI has been separately checked out to a reasonable degree, it is incorporated into the computer-based system. Integration of the hardware, CPCEI, external environment, and personnel into a complete system includes overall system tests to:

- a. assure completion of the integration;
- b. validate any portions of the CPCEI that could not be adequately tested in a simulated environment;
- c. verify that the system, as implemented and integrated, satisfies the system performance and design requirements;
- d. provide the basis for training of personnel and developing operating procedures for the system; and
- e. provide the basis for formal qualification of the CPCEI before it is used in the system.

These tests directly involve the CPCEI as a controlling system element. They range in complexity from simple tests to determine if the program is correctly loaded into the computer's memory, to full environmental system load tests.

#### (6) Pre-Mission System Test

The last stage of CPCEI testing is done shortly before the system is used in a mission. These tests verify that the integrated system is operational and all components are working. This is not primarily a test of the computer program, but involves the CPCEI as one of the system components to be checked. However, if any changes have been made in the computer program as a result of errors and/or design oversights uncovered during system integration tests, or if a separate loading of the computer program is involved prior to mission launch, separate tests of the CPCEI may be required.

### 2.2 Qualification Tests

Qualification tests are tests used to formally demonstrate to the procuring agency that the computer program performs according to the CPCEI Specification and is ready to use in its intended mission. From the viewpoint of the procuring agency-contractor relationship, these are the most important tests in the testing process. For many programming efforts, qualification tests can be considered at two levels: preliminary, and final.

Preliminary qualification tests are performed on the CPCEI as a complete, separate entity for the purpose of assuring that the CPCEI is ready for shipment to the user's facilities, if appropriate, to be incorporated into the total system. These tests demonstrate, to the degree possible in a simulated environment, that the CPCEI satisfies the requirements of Section 3.0 of the Part I Specification.

Final qualification tests establish that the CPCEI, as implemented in the total computer-based system, satisfies the performance and design requirements given in Section 3.0 of the Part I Specification. This may require a series of tests to demonstrate the operation of the total CPCEI in different mission modes. At least one test, however, should involve the total CPCEI in a nominal mission. The results of the tests are one of the factors by which the acceptability of

the CPCEI for operational use is determined. Final CPCEI qualification tests usually follow system integration tests, which may be part of the final qualification tests of the total computer-based system or may be run prior to the system qualification tests. The latter timing might occur when the operation of the program could affect the safety of the system. It might then be necessary to fully qualify the CPCEI in a simulated or partial environment.

### 2.3 Critical Tests

The set of tests described above can be partitioned, for the purposes of management control, into critical tests and non-critical tests.

"Critical tests" are those which are controlled by the procuring agency in the sense that their procedures are reviewed and their results are audited. A test is designated as critical if its completion is a significant milestone in the programming process. For example, qualification tests are critical since their successful completion is prerequisite to program use in its intended mission. Certain other tests may also be deemed critical. For example, a "package" test of an executive routine may be considered critical if the availability of the routine is prerequisite to the testing of other packages in the CPCEI.

A non-critical test is one which is required for program development, but is not controlled by the procuring agency. Usually design tests, debugging tests, early phases of package tests, and early phases of integration tests are considered non-critical.

The selection of which tests are critical is made by the procuring agency prior to the design of tests by the contractor. For each critical test the contractor must produce a detailed test specification as described in Section 3.3 of this appendix. This specification is reviewed for approval by the procuring agency. The contractor may produce specifications for his own use for non-critical tests, but they are neither reviewed nor controlled by the procuring agency.

In some programming efforts it may be desirable to define more than two classes of tests (critical and non-critical). Each class could then be subject to a different degree of control by the procuring agency. However, such classifications must be clearly defined by the procuring agency prior to test design by the contractor.

### 3.0 TEST MANAGEMENT

The responsibility for test management - the planning, control and evaluation of tests - lies with both the contractor and the procuring agency. The contractor must plan and specify the tests in detail, run the tests, and report test results. The procuring agency must require the contractor to perform these tasks, and review significant test milestones.

Test management can be considered to take place in three major stages: establishment of test requirements by the procuring agency, planning of tests by the contractor to meet these requirements, and control of the specifications produced by the contractor for each critical test.

#### 3.1 Establishment of Test Requirements

Test requirements are established by the use of the contract statement of work (SOW) and the Part I Specification.\*

One section of the SOW is used to specify the tasks the contractor is to perform in support of testing during the programming process. The SOW should encourage the concurrent planning and design of both tests and the CPCEI by the contractor. The testing section of the SOW should as a minimum:

- a. require the contractor to satisfy the testing requirements as given in Section 4.0 of the Part I Specification;
- b. require the contractor to produce a test plan early in the design process that shows how the contractor intends to satisfy the testing requirements of the Part I Specification;
- c. identify critical tests for which the contractor must produce detailed procedures. These procedures should be documented in a test specification and submitted to the procuring agency for approval;
- d. require the contractor to perform each test, document the results of critical tests, and submit these results to the procuring agency at specified times in the programming process (e.g., in support of reviews and inspections).

---

\*Although the Part I Specification may be an exhibit to a SOW, the two documents will be considered as separate in this discussion (see reference documents 3, 4).



The Part I Specification provides the technical requirements that enable the contractor to carry out the testing tasks specified in the SOW. Section 3.0 defines the performance and design requirements for the CPCEI. Section 4.0 specifies testing methods to be used in verifying that the CPCEI satisfies the requirements given in Section 3.0. Section 4.0 contains a description of which types of tests are to be run in the acquisition phase and which in the operation phase, and which are to be run in support of technical development and which in support of formal qualification. For each test type Section 4.0 indicates the CPCEI functions that must be verified. As such it is the starting point for test planning and design by the contractor. It is the basis for determining the frequency and scheduling of tests in the programming process.

### 3.2 Test Planning

Significant resources (e.g., equipment, computers, computer programs and personnel) are required for testing in many programming efforts. The identification of these resources together with the identification and scheduling of the tests themselves is an essential element of the contractor's overall plan for developing the CPCEI.

To encourage adequate consideration of testing in the planning process, the procuring agency should require the contractor to produce a test plan either as a separate document or as part of an overall planning document. The contractor, within the plan, indicates the manner in which the testing requirements given in Section 4.0 of the Part I Specification will be satisfied.

The test plan includes a description of test philosophy, a definition of the stages of testing to be used, an identification and brief description of each test within a stage, an estimated start and completion date for each test, an estimate of the resources required, a description of methods for reporting test results, and an identification of all critical tests.

Certain tests covered in the test plan may not be the responsibility of the CPCEI contractor. For instance, system integration tests may be the responsibility of a prime contractor for that effort, and the qualification tests may be designed by the procuring agency or a separate contractor. In such cases, the CPCEI contractor is usually responsible for supplying inputs, requirements, and assistance to the responsible contractor or agency.

An initial version of the plan should be available at the time of the Preliminary Design Review (PDR - when the initial design approach is complete). The procuring agency, can then review the plan to determine its adequacy; this may be done at the PDR or at a separate review. Once approved the plan becomes the basis for implementing test activities in the programming process. It may be modified because of changes to the Part I Specification. It may also be modified by the contractor to reflect changes in the planning process. Changes caused by the former are controlled by configuration management procedures (see reference document 2). The degree to which changes of the latter kind are controlled is left to the discretion of the procuring agency.

Guidelines, based on the testing stages defined in Section 2.0, for preparing a test plan are given in Attachment II. Some of the factors which should be considered by a contractor when planning tests, and which should be considered by the procuring agency when evaluating test plans, are given in Attachment I.

### 3.3 Test Specification

The procedures for critical tests are documented by the contractor in a test specification which must be reviewed by the procuring agency.\* The test specification is the basis for preparing, conducting, and evaluating a test. It describes the goals of the test, the necessary resources, the detailed test procedures, the responsible organizations and the success criteria for the test.

The contractor prepares test specifications concurrently with the design of the CPCEI portions that are to be tested. The test specification, for package tests which are designated as critical, should be available at the last Critical Design Review (CDR) of a component of that package. Higher level test specifications should be available for approval in draft form prior to running the test.

Each test specification is reviewed by the procuring agency to determine if the test satisfies some portion of the test requirements, to check if the test is redundant, and to determine if the procedures are adequate for fulfilling the goals of the test. Once approved the specification becomes the basis for running the test. Any changes to it must be approved by the procuring agency prior to the running of the test.

---

\*In certain cases a test specification may be produced by the procuring agency, or another contractor. For example, the final qualification test specification may be developed by a systems integration contractor.

Guidelines for the preparation of a test specification document including a sample format are given in Attachment III.

### 3.4 Test Reporting

The results of all critical tests, and the completion of all non-critical tests must be reported to the procuring agency. This reporting is done in the following ways:

- (a) the results of preliminary qualification testing are dated at First Article Configuration Inspection (FACI) as discussed in reference document 2;
- (b) the results of final qualification test are audited prior to acceptance of the CPCEI for use in a mission;
- (c) the results of all critical tests are documented in addenda to the appropriate test specification. These addenda are submitted to the procuring agency upon completion of the test;
- (d) all testing activities, including milestone completions, are covered in periodic progress reports (see reference document 1).

REFERENCE DOCUMENTS

1. Appendix II of this report, "Milestones and Progress Reporting Requirements of Computer Programming."
2. Appendix IV of this report, "Configuration Management of Computer Program Contract End Items."
3. Appendix I of this report, "Guidelines for the Preparation of Technical Statements of Work for Computer Program Procurement Contracts."
4. Appendix III of this report, "Preparation of Contract End Item Detail Specification (Computer Program)."

ATTACHMENT I

GENERAL GUIDELINES FOR CPCEI TESTS PLANNING

This attachment contains a checklist of questions that should be answered in the various stages of computer program testing, a discussion of the sensitivity of test design to the nature of the programming effort, and a discussion of factors which determine how thorough the testing process must be.

## 1.0 TESTING OF COMPUTER PROGRAMS

Experience has indicated that a step-by-step approach to CPCEI testing is optimum for large-scale programming efforts. The assumption should be made that no program component is error-free until so proven. The approach should be to start with debugging of subprograms, then debug packages of subprograms, then larger groups, and finally the entire computer program. Hence, CPCEI testing is highly sequential, with tests at increasing levels of complexity during the programming process.

The contractor's test planning should result in a delineation of the several stages of computer program testing and the sequence and functions of tests at each stage. The following sections present a checklist of questions that should be answered at each stage. This list is generally applicable to all computer programming efforts, subject to the job sensitivities as discussed in this attachment.

### 1.1 Individual Subprograms Desk Checking and Debugging

- (a) Does the subprogram coding appear to be correct to the coder, i.e., does it appear that it can be compiled without fatal diagnostics? (Desk Checking)
- (b) Does the subprogram appear to be correct to the coder insofar as the coder understands the subprogram's functions and interfaces with other subprograms? (Debugging)

### 1.2 Subprogram "Packages" Checkout

- (a) Do all interfaces between the subprograms of the package appear to be properly designed and free of coding errors?
- (b) Will this package of subprograms cycle without fatal loops or stops when no input data (or only nominal input data) are inserted?
- (c) Have all functions which this package is designed to provide been included and properly implemented?

### 1.3 CPCEI Simulated Environment Tests

- (a) Are all interfaces between the CPCEI and hardware portions of the computer-based system free of incompatibilities?
- (b) Does the computer program appear to be capable of accommodating expected throughput levels of system operation?

#### 1.4 Hardware - CPCEI - Environment - Personnel System Integration Tests

- (a) Will the entire system actually work? Have the system performance requirements been fully satisfied?
- (b) Are all interfaces between the operating personnel and the CPCEI-hardware portions of the system free of incompatibilities?
- (c) Are estimates of operating conditions (physical model parameters, system traffic loads, etc.) in the system design sufficiently accurate?
- (d) Is the system design free of oversights?
- (e) Will the system accommodate actual throughput levels of system operation?
- (f) Can the CPCEI continue to operate at an acceptable level of performance under extreme operating conditions?

#### 1.5 Pre-Mission System Checkout

- (a) Is the computer-based system still in operating condition?
- (b) Have all modifications to the CPCEI been properly implemented?

#### 2.0 TASK SENSITIVITIES OF TEST DESIGN

Computer program testing is highly job-sensitive. Both the types of tests performed and the thoroughness of their performance at each of the stages listed above are dependent upon certain overall system design factors. Such dependencies must be considered in computer program test planning.

Six of the most significant overall system design factors which influence computer program test types and design of the levels of testing listed above are discussed as follows:

##### (1) System External Communications Environment

Computer-based systems with extensive communications interfaces warrant considerable testing effort in the overall system environment (simulated and/or actual) for both verification and design evaluation objectives.

(2) Nonlinearities in Hardware Components

If a system has narrow dynamic operating ranges for any of the analog hardware components with which the CPCEI has interfaces, careful verification and design evaluation testing is important at both the simulated environment and integrated system test stages to assure continued system operation when the levels of operation deviate from the nominal levels for which the system was designed.

(3) Remoteness of CPCEI in System

The requirements for and design of the various stages of verification testing are influenced by how remote the CPCEI will be in the final integrated system. If the computer design lacks extensive program debugging facilities (i.e., for trap transfers, data planting, start-stop control, etc.), a great deal of program testing may be required, using a computer simulator (operating in another computer); or sophisticated testing procedures and techniques may be necessary.

(4) Complexity of Interfaces Among the Computer Program Components (Subprograms and Data Tables)

In general, the more complex the logical structure of a computer program, the more the emphasis should be placed upon the subprogram debugging, program packages checkout, and simulated environmental testing stages. Indirect addressing, dynamic job lists, recursive subroutine calling and nesting, phantom and real interrupts, and dynamic memory allocation for both program and data table segments are examples of computer program logic design complexities which strongly influence the planning and designing of these early verification testing stages.

(5) Memory vs. Operating-Time Tradeoffs in CPCEI Design

Computer programs with sophistication and "cleverness" of design, for minimization of memory requirements or of operating time, require special emphasis on the early stages of verification testing. Furthermore, where operating time is critical, design evaluation tests in the final integrated system are important to assure that the CPCEI design is adequate for real-time operating with throughput levels.



(6) Cost vs. Perfection Tradeoffs in Implementation

In some cases, the design and implementation of a CPCEI may deliberately fall short of full requirements in order that an imperfect but working CPCEI may be implemented and become available for use as soon as possible. When this is done, a later programming effort is usually anticipated for extension of the CPCEI to satisfy the remaining requirements. Such cost versus perfection tradeoffs in computer program implementation affect the later stages of verification and design evaluation testing. Tests for adequacy of the CPCEI for user's needs should not be concerned with CPCEI requirements which were temporarily, but purposely, omitted from the CPCEI until such postponed requirements are implemented.

3.0 TASK SENSITIVITIES OF TEST THOROUGHNESS

Four of the more significant overall computer-based system design factors which influence computer program test thoroughness are:

(1) Relative Cost of a System Malfunction

The relative cost of a system malfunction during use (mission) affects the thoroughness of CPCEI verification and design evaluation testing. If a system malfunction could be very costly (i.e., if human life could be jeopardized and/or if large sums of money are involved), it is important to have a high degree of assurance of operational success, which in turn requires very thorough testing. On the other hand, if a malfunction would not be catastrophic or if the system design includes redundancy or manual control backup provisions so that malfunctions in one area would probably not affect to any great extent the mission success, then test thoroughness might to some degree be compromised to reduce the costs and time of CPCEI implementation.

(2) Hardware Availability

If the scheduled delivery date of the computer and associated hardware is late in the programming process, more thorough testing at the early stages (program package and simulated environmental tests) should be done so that the period of integration testing after hardware delivery can be minimized.

(3) Computer-Based System Development Schedule Considerations

Schedule considerations often influence the relative thoroughness of each of the several levels of testing and lead to tradeoffs among them. If development schedules are tight, then the earlier stages of testing should be quite thorough. If, on the other hand, schedules permit some testing after CPCEI "delivery" or acceptance, then the thoroughness of the earlier stages of testing might be somewhat compromised.

(4) Evolution of CPCEI Use (Mission Sequence, etc.)

When a CPCEI's use evolves with time, design evaluation testing can likewise evolve, such testing being concerned each time only with the next mission or stage of CPCEI operation.

ATTACHMENT II

GUIDELINES FOR THE PREPARATION OF  
A CPCEI TEST PLAN DOCUMENT

The suggested content for a test plan is discussed below.

A. Purpose and Scope

These sections define the purpose, coverage and limitations of the test plan document.

B. Applicability

This section identifies the organizations and personnel to whom the requirements of the test plan apply.

C. Reference Documents

This section contains a list of all existing documents which relate to the design and testing requirements for the particular CPCEI(s).

D. Test Philosophy

This section defines test philosophy by stating: the purposes of the tests covered by the plan; the modularity structure of the CPCEI as far as testing is concerned, i.e., the succession of "packages" which are to undergo tests as a unit; and criteria for accepting the results of tests as satisfactory.

E. Testing Requirements

This section identifies each stage of testing applicable to the particular programming effort, and specifies for each stage of testing:

- (1) an identification of the individual tests and their time sequence, including an estimated start and completion time for each test, within the stage.

- (2) an identification of the functions to be exercised and checked by each test. These functions should be related to the requirements of the Part I Specification.
- (3) a specification of the testing environment (equipment, operator personnel, communication channels, computer system configuration, etc.).
- (4) an identification of the required test input data generators.
- (5) an identification of the resources required, including:
  - a. Support computer programs (i.e., test support and utility programs such as simulation, recording, data reduction, operational tape maintenance, trap and trace programs, core dump programs, etc.).
  - b. Personnel requirements, including: (1) the individual, department, or agency responsible for coordinating and conducting the test, (2) the personnel required to prepare test inputs and to evaluate the outputs, and (3) the contractor personnel, other agencies, or suppliers who are required to participate in the individual test activity.
  - c. Each type of equipment (including the computer) that is required during the test activity, plus the required operating time and duration of availability for each equipment. If an equipment has associated requirements for other supporting equipment, all such related equipment must be identified.

F. Resources Required for Testing

This section has five subsections which summarize the resources required for all tests:

- (1) Support Computers and Computer Programs, a summarization of support computer program requirements as identified in the various paragraphs above, including identification of how these support programs shall be obtained and on what computer installations they shall be run. Note: Such items which are Government-furnished equipment shall be listed in section (4) below.

- (2) Personnel Requirements for Testing, a summarization of personnel requirements as identified in the various paragraphs above, plus requirements for system operating personnel training. Requirements for both contractor and procuring agency personnel shall be given, with a description of the necessary levels of skills and competence of each.
- (3) Equipment/Facilities Requirements for Testing, a summarization of equipment and facilities required for testing as identified in the various paragraphs above.
- (4) Government-Furnished Items, an identification of the items, systems, computer programs, data specifications, etc., to be provided by the procuring agency, and arrangements for their procurement.
- (5) Requirements for Liaison with Other Agencies, an identification of all items, systems, computer programs, data specifications, etc., required from contractors, other government agencies, etc., or participation in tests required by other agencies, and the liaison arrangements required.

G. Test Results Reporting Requirements

This section shall delineate the procedures for the reporting of test results. It shall define those tests whose completions are major milestones of the computer programming process and those tests which serve as a basis for CPCEI acceptance.

H. Test Specifications Required

This section shall contain a list of all critical tests for which detailed test specification documents are required.

Given below are suggested formats for a title page and a test plan document. These formats are intended to be used as a checklist rather than as an absolute guideline.

TITLE PAGE FOR COMPUTER PROGRAM TEST PLAN

The title page of the computer program test plan should contain the following minimum information:

"Document Number" - The document number is a unique number assigned to distinguish this document from all others.

"Volume Number" - This number is used to distinguish individual volumes of the test plan.

"Revision Number" - This is a sequentially assigned character which uniquely identifies each revision to the test plan.

"Date" - Date of latest revision; date of initial publication.

"Contract Number" - This identifies the contract under which the test plan was prepared.

"Approved Nomenclature" - This identifies the CPCEI involved as well as the computer-based system of which it is a part.

"Prepared by" - This identifies the contractor or agency who prepared the document.

"Approvals" - The appropriate signatures and dates identify the responsible individuals and the titles of their office.

"Security Classification" - The appropriate security classification should be stamped or printed at the top and bottom of the title page and each additional page in accordance with current security regulations, guides, and directives.

SAMPLE OUTLINE FOR COMPUTER PROGRAM TEST PLANS

- 1.0 PURPOSE
- 2.0 SCOPE
- 3.0 APPLICABILITY
- 4.0 REFERENCE DOCUMENTS
- 5.0 TEST PHILOSOPHY REQUIREMENTS
  - 5.1 PURPOSES OF TESTS
  - 5.2 TESTING MODULARITY
  - 5.3 TEST SUCCESS CRITERIA
- 6.0 TESTING REQUIREMENTS
  - 6.1 OVERALL SEQUENCE OF TESTS
  - 6.2 REQUIREMENTS FOR CPCEI DESIGN EVALUATION CHECKS
  - 6.3 INDIVIDUAL SUBPROGRAMS DECK CHECKING AND DEBUGGING
  - 6.4 SUBPROGRAM "PACKAGES" CHECKOUT
    - 6.4.1 SEQUENCE OF PACKAGES
    - 6.4.2 FUNCTIONS TO BE TESTED
    - 6.4.3 TESTING ENVIRONMENT
    - 6.4.4 SUPPORT COMPUTER PROGRAMS REQUIRED
    - 6.4.5 PERSONNEL REQUIRED
    - 6.4.6 EQUIPMENT REQUIRED
  - 6.5 SIMULATED ENVIRONMENT TESTS
    - 6.5.1 SEQUENCE OF TESTS
    - 6.5.2 FUNCTIONS TO BE TESTED
    - 6.5.3 TESTING ENVIRONMENT
    - 6.5.4 SUPPORT COMPUTER PROGRAMS REQUIRED

- 6.5.5 PERSONNEL REQUIRED
- 6.5.6 EQUIPMENT REQUIRED
- 6.6 OVERALL SYSTEM INTEGRATION TEST REQUIREMENTS
  - 6.6.1 SEQUENCE OF TESTS
  - 6.6.2 FUNCTIONS TO BE TESTED
  - 6.6.3 TESTING ENVIRONMENT
  - 6.6.4 SUPPORT COMPUTER PROGRAMS REQUIRED
  - 6.6.5 PERSONNEL REQUIRED
  - 6.6.6 EQUIPMENT REQUIRED
- 6.7 QUALIFICATION TEST REQUIREMENT
  - 6.7.1 SEQUENCE OF TESTS
  - 6.7.2 FUNCTIONS TO BE TESTED
  - 6.7.3 TESTING ENVIRONMENT
  - 6.7.4 SUPPORT COMPUTER PROGRAMS REQUIRED
  - 6.7.5 PERSONNEL REQUIRED
  - 6.7.6 EQUIPMENT REQUIRED
- 6.8 PRE-MISSION SYSTEM CHECKOUT REQUIREMENTS
  - 6.8.1 SEQUENCE OF TESTS
  - 6.8.2 FUNCTIONS TO BE TESTED
  - 6.8.3 TESTING ENVIRONMENT
  - 6.8.4 SUPPORT COMPUTER PROGRAMS REQUIRED
  - 6.8.5 PERSONNEL REQUIRED
  - 6.8.6 EQUIPMENT REQUIRED



7.0 RESOURCES REQUIRED FOR TESTING

7.1 SUPPORT COMPUTERS AND COMPUTER PROGRAMS

7.2 PERSONNEL REQUIREMENTS FOR TESTING

7.3 EQUIPMENT/FACILITIES REQUIREMENTS FOR TESTING

7.4 GOVERNMENT-FURNISHED ITEMS

7.5 REQUIREMENTS FOR LIAISON WITH OTHER AGENCIES

8.0 TESTS RESULTS REPORTING REQUIREMENTS

9.0 TEST SPECIFICATIONS REQUIRED

ATTACHMENT IIIGUIDELINES FOR THE PREPARATION OF A  
CPCEI TEST SPECIFICATIONS DOCUMENT

The major sections which should be included in a CPCEI test specifications document are discussed below. A sample outline is included.

A. Introduction, Scope, Applicability

A description of the purpose, scope and applicability of the specification.

B. Reference Documents

A list of documents referenced in the test specification, including the test plan document for the CPCEI, the Part I Specification, and any applicable test standards documents.

C. Test Requirements

- (1) Test Prerequisites, a specification of those tests which have to be completed and those equipments which must be available before this test can be run.
- (2) Test Management, a definition of the responsibilities for conducting and coordinating the given test activity, distinguishing between (1) contractor responsibilities, (2) procuring agency responsibilities, and (3) other agency responsibilities.
- (3) Personnel Requirements, an identification of all personnel to be involved in the given test activity, including the individual job classifications, number of personnel required, reporting periods for each type of personnel and their duties.
- (4) Hardware Requirements, a specification of the hardware requirements necessary for conducting the test, including (1) the computer complex and associated peripheral equipment, and (2) other supporting equipment or contract end items which interface with the computer complex. For each item listed the quantity required, the model designation, the duration of use, and the source or location

shall be specified. If limited operational capability is satisfactory, the extent of reduced operating capability that will allow successful test performance should be defined.

- (5) Supporting Computer Program Requirements, an identification of additional computer program support that is required to perform the specific test. Examples of non-operational support programs that are specified under this section are: trap, trace, and core dump routines; simulation, data reduction, data analysis and recording programs; and any other necessary support or utility program.
- (6) Functions Under Test, a specification of the areas or aspects of CPCEI performance (functions) to be tested. For each function the following information shall be given:
  - a. Initialization Values - identifies all static values for data items which are not part of the dynamic input data used to test the program.
  - b. Range of Parameters - identifies the upper and lower limits of numerical values which serve as dynamic inputs to the function being tested. For non-numerical data items, the proper symbology shall be indicated.
  - c. Required Accuracies - specifies the required accuracies of all inputs and the associated output data values.
  - d. Data Collection Methods - defines each type of recording, (either analog, digital, or observer) and specifies the frequency and duration of data collection.
  - e. Analysis Techniques - specifies the procedure to be followed in determining if the outputs collected are acceptable.
  - f. Test Input Values - lists the precise input values to be used or, if necessary, specifies the method to be used in generating the precise inputs.
  - g. Expected Output Values - lists the expected output values which are considered to be indicative of a successful test. In cases where a given range, frequency, or level of output is to be considered successful, this information shall be included.

- (7) Test Procedure, a description in time sequence of the required test inputs, operator actions, environmental changes and conditions. The role of each actively participating test observer shall be specified. Documentation required to record each significant action in the procedure shall be specified.
- (8) Test Results, the actual results of the tests as documented during test execution.

TITLE PAGE FOR COMPUTER PROGRAM TEST SPECIFICATIONS

The title page of each Test Specification should contain the same minimum information as described in Attachment I for the test plan.

SAMPLE OUTLINE FOR COMPUTER PROGRAM TEST SPECIFICATION

1.0 INTRODUCTION

2.0 SCOPE

3.0 APPLICABILITY

4.0 REFERENCE DOCUMENTS

COMPUTER PROGRAM TEST PLAN

CPCEI SPECIFICATION PART I

STANDARDS

OTHER PUBLICATIONS

5.0 TEST REQUIREMENTS

5.1 TEST PREREQUISITES

5.2 TEST MANAGEMENT

5.3 PERSONNEL REQUIREMENTS

5.4 HARDWARE REQUIREMENTS

5.5 SUPPORTING COMPUTER PROGRAM REQUIREMENTS

5.6 FUNCTIONS UNDER TEST

5.6.1 FUNCTION 1

a) INITIALIZATION VALUES

b) RANGE OF PARAMETERS

c) REQUIRED ACCURACIES

d) DATA COLLECTION METHODS

e) ANALYSIS TECHNIQUES

f) TEST INPUT VALUES

g) EXPECTED OUTPUT VALUES

5.6.2 FUNCTION 2

.

.

.

5.6.n FUNCTION "n"

.

.

.

5.7 TEST PROCEDURE

6.0 TEST RESULTS

BELLCOMM, INC.

APPENDIX VI

DESIGN AND IMPLEMENTATION PRACTICES  
OF COMPUTER PROGRAMMING

TABLE OF CONTENTS

- 1.0 INTRODUCTION
- 2.0 PROGRAMMING PRACTICES - GENERAL DISCUSSION
  - 2.1 DEFINITION OF "DESIGN AND IMPLEMENTATION PRACTICES"
  - 2.2 THE ROLE OF PRACTICES IN THE PROGRAMMING PROCESS
  - 2.3 ESTABLISHMENT OF PRACTICES
- 3.0 CHECKLIST OF AREAS SUBJECT TO PROGRAMMING PRACTICES
  - 3.1 OVERALL COMPUTER PROGRAM DESIGN
    - 3.1.1 PROGRAM ORGANIZATION
    - 3.1.2 FUNCTIONAL ALLOCATION TO SUBPROGRAMS
    - 3.1.3 MODES OF OPERATION
    - 3.1.4 OVERALL DATA ORGANIZATION
    - 3.1.5 TESTABILITY
  - 3.2 SUBPROGRAM DESIGN
    - 3.2.1 MODULARITY OF DESIGN
    - 3.2.2 INPUT-OUTPUT ISOLATION
    - 3.2.3 TRANSFER OF CONTROL
    - 3.2.4 COMMUNICATION CONVENTIONS
  - 3.3 CODING
    - 3.3.1 LANGUAGE
    - 3.3.2 SPECIAL-PURPOSE MACROS
    - 3.3.3 SUBPROGRAM ELEMENT NAMES
    - 3.3.4 CODING TECHNIQUES
    - 3.3.5 SPECIAL PROCEDURES AND LIMITATIONS



Table of Contents Contd.

- 3.4 DATA STRUCTURING
  - 3.4.1 DATA CATEGORIZATION
  - 3.4.2 STORAGE ALLOCATION
  - 3.4.3 DATA IDENTIFICATION
  - 3.4.4 TABLE AND BLOCK STRUCTURES
  - 3.4.5 TABLE LOOK-UP TECHNIQUES
  - 3.4.6 DATA CONVERSION, PACKING, AND SEPARATION
  - 3.4.7 DATA CODES AND FORMATS
  - 3.4.8 PROTECTED STORAGE REQUIREMENTS
- 3.5 COMPUTER PROGRAM DOCUMENTATION
  - 3.5.1 FLOW CHART CONVENTIONS
  - 3.5.2 PROGRAM LISTINGS
  - 3.5.3 CPC SPECIFICATIONS
- 3.6 COMPUTER PROGRAM APPLICATION AND MAINTENANCE
  - 3.6.1 GENERIC PROGRAMMING
  - 3.6.2 ADAPTATION TO SPECIFIC INSTALLATIONS
  - 3.6.3 UPDATING AND DELETING PROCEDURES
  - 3.6.4 STATUS ACCOUNTING AND REPORTING
- 3.7 PRODUCT INVENTORY
  - 3.7.1 SUBPROGRAM IDENTIFICATION
  - 3.7.2 DUPLICATION REQUIREMENTS

Table of Contents Contd.

3.8 SPECIAL PROGRAMMING TECHNIQUES

3.8.1 REAL-TIME BREAK

3.8.2 INTERRUPT CONTROL

3.8.3 ROLLBACK

3.8.4 SUBPROGRAM HIERARCHY CONTROL

3.8.5 HARDWARE ERROR DETECTION/DIAGNOSIS BY SUBPROGRAMS

3.8.6 AUTONOMOUS MULTI-PROCESSOR CONTROL

3.9 OTHER PROGRAMMING PRACTICES

3.9.1 PROGRAMMING ACTIVITY COORDINATION

3.9.2 USE OF AVAILABLE LIBRARY SUBPROGRAMS

3.9.3 PHYSICAL AND MATHEMATICAL CONVENTIONS

3.9.4 TERMINOLOGY AND SYMBOLOGY

REFERENCE DOCUMENTS

DESIGN AND IMPLEMENTATION PRACTICES  
OF COMPUTER PROGRAMMING

1.0 INTRODUCTION

Design and implementation practices, usually documented in a programming standards manual, are used to guide the activities of programmers to assure that the products they produce are compatible with each other and the intended design of the computer program. Practices tell a programmer "how" to write a program; together with programming specifications which tell "what" the program is to do and schedules which tell "when" the program is to be written, they are the basis for management control of the programmer's activities.

This appendix contains a discussion of the role of practices in the programming process, a discussion of how practices are established by both the contractor and procuring agency, and a checklist of practices that can be used as a guide in determining which practices should be used in a programming effort.

Note: The practices discussed in this appendix apply to the development of a computer program contract end item (CPCEI) at various levels of its structure. In some instances the practices may apply only at a certain level and not at others. For convenience in making this distinction the following terminology is used in this appendix:

1. The CPCEI - highest level of computer program, the entire set of instructions and data designated as a single end item.
2. Computer Program Component (CPC) - a segment of the CPCEI, containing a functionally related set of instructions and data, whose development is specifically identified for monitoring by the procuring agency. CPC's may exist at several levels of CPCEI structure; that is, a CPC may be part of another CPC. The highest level of CPC is the first level of CPCEI structure. The lowest level CPC contains no other CPC; it is the lowest level of the CPCEI structure monitored by the procuring agency.
3. Subprogram - a segment of the lowest level CPC containing a functionally related set of instructions and data. The division of a CPC into subprograms is usually based on technical considerations, such as modularity and testability, as determined by the contractor. A CPC may have several levels of subprograms.

4. Subprogram element - a subprogram element is a segment of the lowest level of subprogram; it can be a group of computer words (e.g., MACRO, data table) or a single computer word (e.g., an instruction or data word).

## 2.0 PROGRAMMING PRACTICES - GENERAL DISCUSSION

The "quality" of a computer program is determined by its ability to do the job for which it is intended, to be modified as required, to be maintained, and to be understood by others than those who produced it. Although quality is difficult to measure, it is evident that adherence to effective standards and practices by programmers during the programming process can increase the chances of producing a working, high quality end item.

### 2.1 Definition of "Design and Implementation Practices"

Programming practices are the rules and formats imposed on the production of a computer program in order to achieve standardization in the physical outputs of the programming process. This includes the rules and procedures for human performance during program design and implementation. Specifically, practices are of three types: standards of formats (for punched cards, flow diagrams, etc.), standards of techniques (used in program design), and standards of procedures (regulating programming personnel during the design, documentation, coding and checkout of programs). In this appendix the phrase "design and implementation practices", or simply "practices", is used to refer to all three types of standards and procedures.

### 2.2 The Role of Practices in the Programming Process

Practices are used to control and guide the efforts of programmers to ensure that a uniform self-consistent computer program is developed. Practices, when properly established and administered, assist in:

- a. coordinating the efforts of the many programmers involved in a large programming effort. This coordination is necessary to assure that individual CPC's are produced in an orderly and timely manner and are compatible with each other;
- b. making the computer program understandable, so that if necessary it may be used and maintained by personnel who were not involved in its design and implementation. This also facilitates the modification of the program if the user's requirements change, and the sharing of the program or parts thereof by other users;

- c. disseminating information concerning preferred programming techniques. This an important aspect of programming personnel training;
- d. preventing, or at least minimizing, the delays in program development due to human error (e.g., oversights in the design of subprogram interfaces, incomplete implementation of necessary changes).

### 2.3 Establishment of Practices

The set of practices required for any given computer programming activity depends upon the nature of that activity. Therefore, a single detailed set of practices cannot be established for universal application. Furthermore, since practices do not pertain to what a computer program must do but how the computer program must be produced, the programming contractor should have considerable control over the details of the particular practices by which he is to abide.

The procuring agency's primary interest is that an effective set of practices is established and used. The specific contents of a particular set of practices, however, may be left to the discretion of the contractor, except for special standards which have general applicability and are stated as contractually binding.

The procedure for establishing adequate practices involves:

- a. the procuring agency stating, in the contract Statement of Work (SOW), the requirement for the production of a set of practices by the contractor (see reference document 1);
- b. the organization responsible for the Part I Specification including, in that document, appropriate references to design practices (see reference document 2);
- c. the contractor producing, during the programming process, a practice manual according to the needs of the task and the requirements given in the Statement of Work and the Part I Specification.

In some cases it may be appropriate to specify explicitly in the SOW certain practices (e.g., flow chart conventions) to be adhered to by the contractor. In all cases, those practices not delineated in the SOW which should be applied to a programming task should be specified in the Part I Specification and/or in the contractor-produced practice manual. Usually, practices included

in the Part I Specification are those pertaining to overall design of the CPCEI, plus any other practices which the procuring agency, since the time the SOW was written, had determined to be appropriate. The contractor-produced practices manual, on the other hand, usually covers practices pertaining to subprogram design, coding, checkout, documentation, and adaptation.

### 3.0 CHECKLIST OF AREAS SUBJECT TO PROGRAMMING PRACTICES

This section contains a checklist of the design and implementation aspects of computer programming for which specific practices may be appropriate. This list is representative of the standards that could be established in a programming task. The areas covered in an actual set of practices depend on the specific application and may be more or less comprehensive than the list given below.

The main categories of this checklist are overall computer program design, subprogram design, coding, data structuring, computer program documentation, computer program application and maintenance, product inventory, special programming techniques, and other programming practices.

#### 3.1 Overall Computer Program Design

Practices related to the design of the computer program, such as:

##### 3.1.1 Program Organization

Practices and guidelines which affect the structure of the computer program, including focal points of program control, criteria for subprogram construction, and criteria for data communication between subprograms with common data requirements.

##### 3.1.2 Functional Allocation to Subprograms

Procedures for functionally identifying subprograms; criteria for determining the criticality of subprograms, and the functional hierarchy of subprograms.

##### 3.1.3 Modes of Operation

A list of permissible computer and computer program operating conditions, including identification of modes and functional purposes, rules for initializing and operating in the mode, options available within the mode, and options for possible mode changes.

### 3.1.4 Overall Data Organization

The requirements for the sequence and order of data, including identification of the structures for conversion, storage, retrieval, and processing; rules for when and where to use data.

### 3.1.5 Testability

Practices that specify the requirements and methods to ensure the ability to test the computer program during design, implementation, acceptance, and operation.

## 3.2 Subprogram Design

The criteria for subprogram design, such as:

### 3.2.1 Modularity of Design

Practices affecting the construction methods, permissible sizes of subprograms, and methods of specifying data linkages.

### 3.2.2 Input-Output Isolation

Input and output functions are system sensitive, i.e., they may be altered by a change in system configuration. These functions should be isolated from the computational and data processing functions of the program as separate subprograms. This practice will facilitate modifications arising from adapting the program to a different system or to a new system configuration.

### 3.2.3 Transfer of Control

A list of the permissible methods of designing and coding control transfers between subprograms, the rules for determining the use of transfer of control, and identification of the consequences of both dynamic (e.g., executive control) and static (e.g., subroutine calling sequence) transfers.

### 3.2.4 Communication Conventions

A set of standards covering the data communications between subprograms, as for example how push-down-pop-up, hopper, pointer-queue, flags, and other communications techniques are to be implemented and under which circumstances each of these types of techniques are to be used in subprogram design.

### 3.3 Coding

The permissible methods of subprogram coding, such as:

#### 3.3.1 Language

The permissible programming languages to be used and the rules for determining which language will be used for which types of subprograms.

#### 3.3.2 Special Purpose MACROS

A list identifying the standard and special purpose MACROS available, the rules for using such MACROS, and the consequences of using each MACRO available.

#### 3.3.3 Subprogram Element Names

The conventions for titling data items, data tables, and transfer points as they are developed in the programming process.

#### 3.3.4 Coding Techniques

A list of all preferred special coding techniques not described elsewhere.

#### 3.3.5 Special Procedures and Limitations

Practices identifying special programming and hardware features available (e.g., interrupts), and specifying the methods and rules governing their use.

### 3.4 Data Structuring

Rules, procedures, and standards pertaining to the design and coding of all data aspects of the computer program. These rules may cover such areas as the following:

#### 3.4.1 Data Categorization

An identification of the types of data inserted into the computer program during coding, during compiling, during loading (insertion of the program into computer memory), and during program operation. This identification includes rules used to determine the data category and method of insertion of each given set of data.



### 3.4.2 Storage Allocation

An identification of the various sections of computer memory and the data, subprograms, and "scratch memory" areas allocated to each section. If storage allocation is dynamic, this identification includes the practices that ensure proper storage manipulation, allocation map updating, and subroutine linkage.

### 3.4.3 Data Identification

A list of all rules of data identification (tables and items) to ensure convenient access to data by both programming personnel and the computer program. This may include specifying, for each set of data, records of identification of the data, the source and destination of the data, the subprograms that use and/or modify the data, the number of datum in the data set, the units used, and the rules for establishing and recording symbolic tags used by the data set.

### 3.4.4 Table and Block Structures

Practices defining the permissible table and block structures (e.g., compact, parallel) and the rules used to determine the structure type applicable to any data set.

### 3.4.5 Table Look-Up Techniques

A set of rules that specify the permissible table look-up techniques (i.e., binary search, direct and indirect indexing), the methods of implementing each technique, and the planned use of each technique on specific data types.

### 3.4.6 Data Conversion, Packing, and Separation

Practices specifying data conversion techniques (between different codes), and the rules for packing (multiple data per word) and separating (leaving unused bits to provide for growth) data.

### 3.4.7 Data Codes and Formats

A list of the permissible data codes and types of data formats, including rules for determining when and how to use each.

### 3.4.8 Protected Storage Requirements

Practices specifying the methods of achieving memory protection, and the location, access means (including coding techniques), and contents of computer memory areas containing critical, protected data.

## 3.5 Computer Program Documentation

Practices concerning the several types of computer program technical documentation, such as those covering the following:

### 3.5.1 Flow Chart Conventions

A set of standards used to symbolically express the computer program logic and techniques in flow chart form. This includes rules used to determine, for each subprogram, the type of flow chart applicable (logical or functional), the level of detail required, and the format for the physical appearance of the flow chart (including commentary, identification of flow chart elements, etc.).

### 3.5.2 Program Listings

A set of procedures for inserting commentary in the program listings. Commentary is information intended for programming personnel that is ignored by the computer. Such commentary explains key logic portions of the code, all transfer of control points, and such identification items as the name of the element, name of the programmer, a brief history of the revisions to the element, etc.

### 3.5.3 CPC Specifications

A set of rules for documenting individual computer program components. Such documentation, together with the flow charts and listings, is expected to completely define the CPC (including the logic flow, interfaces with other CPC's timing idiosyncracies, algorithm details, operational limitations, etc.). An example of a format of such program documentation is given in reference document 2 (specifically Section 3.2 of the CPCEI Specification Part II).

### 3.6 Computer Program Application and Maintenance

Practices to ensure the applicability and maintainability of the finished computer program, such as the following areas (where applicable):

#### 3.6.1 Generic Programming

Generic programming is the concept of developing a basic CPCEI for use in several different computer-based systems. In such an approach, all computer-based dependencies (peripheral equipment, operational load levels, etc.) are represented as data to be provided at compilation or installation of the CPCEI. Such an approach minimizes reprogramming time for each installation. If such an approach is used, all programming procedures, conventions, techniques, and formats required to implement generic programming must be specified.

#### 3.6.2 Adaptation to Specific Installations

A set of procedures required to adapt the completed CPCEI to specific installations. A separate set of procedures may exist for each installation, and they may involve procedures from simple instructions for assembly to complete reprogramming of certain subprograms.

#### 3.6.3 Updating and Deleting Procedures

A list of procedures to be followed in maintaining subprogram coding and listings of the code during CPCEI checkout and/or during revisions to reflect changing user's requirements. These procedures may include the requirements for maintaining superseded versions as backup, identification of permanent versions, and rules for determining when a new version is accepted as the master version.

#### 3.6.4 Status Accounting and Reporting

Practices that specify the requirements and methods for maintaining status records (a record of when, by whom, and why a subprogram was altered) for each physical representation of the product.

### 3.7 Product Inventory

A set of procedures used to guard the computer program against calamities (e.g., fire) and to maintain an accurate up-to-date record of status, covering such aspects of CPCEI physical handling as the following:

### 3.7.1 Subprogram Identification

Procedures for identifying each subprogram, including such items as title, person generating the subprogram, classification of the subprogram, date generated, etc.

### 3.7.2 Duplication Requirements

Rules for determining the number and location of physical representations of the programming product to ensure adequate access and protection against loss.

## 3.8 Special Programming Techniques

Rules and procedures pertaining to special techniques as appropriate to ensure the program adequately satisfies objectives. This is particularly true of real-time programs. While special programming techniques rely only on the specific application, examples of practices for real-time systems are:

### 3.8.1 Real-Time Break

If the computer-based system is designed to operate in real-time with dynamic sequencing control of the programmed functions on a time-priority basis, the set of all programming rules, techniques, and formats to be used to implement the dynamic task sequencing (including communications between the executive sequence control subprogram and the several task-performing subprograms) must be specified. This communication may involve each subprogram informing the executive control subprogram of requests for control to do work. It may also involve the techniques whereby a subprogram shall, under various conditions, relinquish control back to the executive sequence control subprogram before a task is completed, with special requests for later return of control for completion of the task.

### 3.8.2 Interrupt Control

If the computer is equipped with the capability of one or more hardware-triggered program interrupts, the complete set of programming rules and techniques for design and coding of the interrupt subprograms, and all communications and other aspects of interfaces between such interrupt subprograms and the other subprograms of the CPCEI, must be specified.

### 3.8.3 Rollback

If the "rollback" technique of CPCEI operation is to be incorporated into the design of the individual CPC's the set of all programming rules and techniques for doing so must be specified. "Rollback" refers to the concept whereby when computer or CPCEI operational malfunctions are detected, program operation is automatically suspended and control is returned to an earlier point in CPCEI operation. This often results in repeating those portions of CPCEI operation during which the malfunction occurred. Rollback programming techniques pertain to identifying those points in CPCEI operation to which control may be safely returned (so-called "rollback" or "check" points), how data is to be properly safeguarded and updated at each check point, and how the control suspension and "rolling back" is to be implemented. These techniques are usually employed when hardware-triggered interrupt capabilities (see above discussion) are used for detection of operational malfunctions and for rollback control.

### 3.8.4 Subprogram Hierarchy Control

If a CPCEI design permits subprogram "nesting" during operation, the set of programming rules to properly effect control up and down all permissible nesting hierarchies must be specified. These rules may involve the use of "push-down-pop-up" lists of return addresses, index register contents storage, and deferred programmed diagnostic indications (e.g., the disposition of error diagnostics, such as divide underflow, within the subprogram rather than by an executive monitor system such as IBSYS).

### 3.8.5 Hardware Error Detection/Diagnosis By Subprograms

The set of all programming rules and techniques pertaining to the automatic detection and processing, by the subprograms, of hardware (i.e., computer and peripheral equipment) malfunctions during operation. These techniques may involve the establishing and maintaining of hardware malfunction data tables, the sequencing and control of special hardware malfunction diagnosis and reporting subprograms, and interfaces between hardware-triggered interrupt subprograms and the other subprograms of the CPCEI (see discussion above).

### 3.8.6 Autonomous Multi-Processor Control

If a CPCEI is to operate in a multi-processor computer in which the several central processors operate partially or completely autonomously with respect to each other, all special programming techniques necessary for the operation of several portions of the CPCEI simultaneously must be identified. These special techniques might include memory area queueing logic, first-come-first-served or master-slave rules for task and subprogram seizure by the several processors, special subprogram hierarchy considerations (see above discussion), and special rules for the design of task list and status data tables.

### 3.9 Other Programming Practices

Any other programming practices, rules, procedures, or techniques not specified elsewhere, which are appropriate for a particular programming effort. These might, for example, cover such areas as the following:

#### 3.9.1 Programming Activity Coordination

Practices specifying the use of automated aids (e.g., computer controlled information systems) and administrative procedures for coordination of the efforts of several programmers working as a team to develop a CPCEI.

#### 3.9.2 Use of Available Library Subprograms

Practices governing the acquisition, adaptation, and use of externally developed computer programs, such as those available from user libraries, computer manufacturers, and government sources.

#### 3.9.3 Physical and Mathematical Conventions

Practices specifying mathematical models, calculation methods, coordinate systems, rounding and truncation techniques, engineering units, and accuracy requirements.

#### 3.9.4 Terminology and Symbolology

Practices which identify and define (or reference documents that identify and define) all permissible technical terms, abbreviations, symbols, characters, etc.

REFERENCE DOCUMENTS

1. Appendix I of this report, "Guidelines for the Preparation of Technical Statements of Work for Computer Programming Procurement Contracts."
2. Appendix III of this report, "Preparation of Contract End Item Detail Specification (Computer Program)."

BELLCOMM, INC.

APPENDIX VII

GLOSSARY



GLOSSARY

- Acceptance: The act by which the procuring agency assents to ownership of computer programs, systems, or components thereof; or approves specific services rendered as partial or complete performance of a contract.
- Acquisition of Computer Program:  
The activities involved in the production of a computer program system after the preparation of performance specifications; the design, development, testing, and acceptance of the programs.
- Activity: The effort directed towards attaining a goal. An activity can be: the preparation of plans, inputs, and/or specifications; an analysis and design within the computer program process; the planning, organizing, defining, identifying, and/or establishment of criteria, requirements, procedures and policies.
- Approval: The formal establishment of acceptability of a product by the procuring agency.
- Assembly: The act of putting together the constituent parts of an entity; also see Computer Program Assembly.
- Attachment: A separate supporting document referred to by, and supplementary to, a guidelines document or manual; a document supplementing a technical work statement providing general supporting information and/or guidelines.
- Baseline: (1) (Definition in Exhibit XIX to AFSCM 375-1, June 1, 1964): An approved and defined point of departure for control of future changes in system or equipment performance and design. A baseline is documented by a specification and other documents and is technically defined by formal approval of the specification, or a part thereof, by the procuring agency.

(2) (Expansion of official definition for use in this report): A set of concepts, information, or specifications, documented and subject to formal configuration control procedures, which prescribes, to a progressively greater level of detail, an officially acceptable configuration of a computer program/system which is either existing, being implemented, or to be developed.

Baseline Configuration:

The documented and approved design concept, or arrangements of components, as established at a given point in the procurement cycle for systems or equipment (the physical system, proposed or implemented).

Baseline Documentation:

All documentation which at some time has been approved and is subject to formal configuration management (change control) procedures; the documents required to provide a complete, accurate description of the system and to monitor the computer program development; the official system specifications subjected to configuration management procedures in order to control system integrity.

Baseline Establishment:

An event during the programming process when, as a result of a review wherein the procuring agency approves or "concurs" with certain documentation, that documentation becomes an official portion of the baseline documentation of the system and hence subject to formal configuration management procedures.

Checkout:

The application of diagnostic or other testing procedures to a computer program element or equipment in order to establish the element's integrity.

Code: To translate and write information commands, etc., in a form understandable to the intended receiver; e.g., to write machine instructions or symbolic notation from flow diagrams and/or program design specifications; also, the set of instructions so produced.

Component: See Computer Program Component.

Computer Based System:

A system which contains a digital computer as a controlling element.

Computer Program: The ordered set of instructions and data required to control the operation of a digital computer.

Computer Program Assembly:

The process of integrating subprograms into a main program; also, the process performed by a special computer program (an assembler) operating on symbolic code to produce, on a one-to-one basis, instructions that are understandable to a computer.

Computer Program Compilation:

The process, usually performed by a special computer program (called a compiler), of producing a machine language routine from a routine written in functional rather than computer-oriented source language.

Computer Program Component (CPC):

A segment of a CPCEI, containing a functionally related set of instructions and data, whose development is specifically identified for monitoring by the procuring agency; the separately produced subprograms or functionally related groups of subprograms which compose a CPCEI. This group of subprograms is usually separately compilable and performs a specified function or set of related functions.

Computer Program Contract End Item (CPCEI):

A computer programming end product whose development is designated by the procuring agency to be subject to configuration management.

Concurrence: The mutual agreement between two parties; for example, the agreement between the procuring agency and its contractor.

Configuration: The order or positional description of elements within a system, subsystem, etc.

Configuration Accounting:

The act of reporting and documenting changes made to systems/equipment/computer programs subsequent to the establishment of a base-line configuration in order to maintain an accurately defined configuration status.

Configuration Control:

Systematic evaluation, coordination, and approval or disapproval of proposed changes to any baseline.

Configuration Identification:

The technical documentation defining the approved configuration of systems/equipment/computer programs under design, development, test and manufacture.

Configuration Management:

The formal set of procedural concepts by which a uniform system of configuration identification, control, and accounting is established and maintained for systems/equipment/computer programs and components thereof.

Contractor: The organization which is under contractual obligation to perform study activities and/or produce documents, products, and/or fully integrated systems. As used throughout this report, "contractor" denotes the organization responsible for developing the computer program.

Contract End Item: Same as End Item.

Conversion: Same as Data Conversion.

Data Base: All static and dynamic information supplied to the operational computer program.

Data Conversion: The processing and formatting of the raw data base information into a form acceptable for system utilization.

Debugging: The process of checkout of individual sub-programs.

Deliverable Item: An essential product of the programming process produced by the contractor according to well established standards; an item used in support of the end item whose delivery is called for in a work statement.

Design Evaluation: The activity which confirms the system design to satisfy the system requirements and to perform the operational functions required by the user; studies concerning the adequacy of design of a system for user's needs.

Design Requirements:

Those requirements necessary to formulate an approach to the desired solution to a problem on the computer.

Documentation: The systematic organization and presentation of recorded, specialized knowledge in order to maintain a complete record of it and of reasons for changes in it; the information that is generated to record data required for control of design, production, procurement, maintenance, and supply of materials of an equipment, computer program, or system.

Dynamic Data: Data which is subject to change during the operation of a computer program.

Element: A distinct part of a subset of a computer program system that performs a function or set of functions; a tool or set of tools.

used in support of design, implementation, or testing of a system; a subsection of a work statement chosen to provide a convenient breakdown for the specification of contractor tasks and obligations.

End Item: The deliverable item whose procurement is the primary objective of a development process; an arbitrary designation for the portions of a system/equipment/computer program identification as a result of a formal functional analysis; a functional entity physically related to and selected for the purpose of system development, procurement, and logistics.

End Product: Same as End Item.

Environment: The aggregate of all conditions and influences outside the system that affect the operation of equipments, computer programs, or systems.

Equipment Options: Specifications of permissible variations in a system configuration and how such variations affect the computer program design and implementation.

Event: A point in a process denoted by the completion of one or more activities; e.g., the delivery of documents, or customer-contractor concurrence on specifications.

Executive Subprograms: The control element of a computer program system that regulates the sequence of program functions.

Exhibit: A separate supporting document attached to a contract work statement containing specific clauses or statements of practices which apply to the contract; a set of explicit supporting statements and/or clauses.

Facilities: A physical installation which houses and/or provides for the operation of a system (buildings, shelters, utilities, shops, etc.).

- Flow chart: A drawing, using standard symbology, which represents the sequence of operations and the flow of data in a computer program, or more generally, in any system of elements.
- Flow Diagram: Same as Flow Chart.
- Format: The specified arrangement of information; e.g., words, digits, characters, records, fields, pages, etc.
- Generative Routine:  
A computer routine that generates data for inclusion in the data base, for use directly by other programs, or for use as intermediate results in the computer programming process.
- Guidelines: A set of standards and/or instructions which serve as a guide to a technical representative in preparing something; e.g., task description and/or specification documents, computer programs, etc.
- Hardware: The physical equipment or devices forming a computer, peripheral equipment, and related components.
- Information Flow: The progression of data through a computer-based system; the sequence of operations made on data from the point of introduction of that data into the system until the last output affected by that data has been generated.
- Input-Output: Information or data transferred between external media and the internal storage of a computer, or among the computer program subsystems' components.
- Input-Output Equipment:  
The hardware used to transfer data into and out of a computer-based system.
- Integration: The process of assuring that the major hardware, computer program and human elements of a system are assembled, tested, and operated in such a manner as to be compatible with each other and to satisfy the system objectives.

Interface: The relationship between components, end items, or functional areas required to achieve the integrated system design objective; a common boundary between automatic data processing systems or parts of a system.

Interface Considerations:

Those elements or aspects of the total effort that are considered in developing, defining, and/or establishing the common boundaries between the various systems or parts of the systems.

Life Environment: The environment for the integration of the various subsystems into the whole system. The live environment approximates as closely as possible the operational environment. It is involved in testing the operational system and in training the personnel, preceding release of the system for operational usage.

Macro: A source language statement which may result in a variable number of machine instructions.

Maintenance: The function of assuring continuous integrity of a system or component; the process of updating computer programs for changing system requirements, succeeding missions, or design oversights discovered during system operation.

Milestone: A significant, distinct, objectively identifiable point, which can be used as a means of evaluating progress in terms of an estimated time schedule.

Mode: A descriptive term applied to a particular methodology or sequence of performing one of several possible tasks.

Module: A combination of components contained in one package or common to one element.

Off-line: Descriptive of a system and of the peripheral equipment or devices in a system in which the operation of such equipment is not under control of the central processing unit and in which the information is processed separately from the main flow of operations.



On-line: Descriptive of a system and of the peripheral equipment or devices in a system in which the operation of such equipment is under control of the central processing unit and in which information reflecting current environment is introduced directly into the main flow of operations.

Operating Conditions:

All variables that affect the execution of a program, subsystem, or system.

Operational: The state of a system where all design, development, environment integration, and checkout has been completed and the system is available for use. A computer-based system is operational when it is free of known programming errors, all associated products are available, the system has been accepted by the procuring agency as having passed a set of acceptance tests and is now in, or ready to be in, use by the procuring agency.

Operational Computer Programs:

The computer programs of a particular system, exclusive of support and utility computer programs, that operate in real time and/or in direct support of the system functions and in direct relationship to the system environment; the computer programs that directly perform those functions necessary to satisfy system objectives.

Operational Environment:

The environment for the utilization and application purposes in which the system was designed to operate.

Operators: The personnel who control a system's operations; the personnel of a system's operational environment and with whom the computer programs have direct interfaces.

Operator Position:

A point or place in the system where manual interaction is defined. This may be a manned position at a computer console, display media, communication media console, output station or input station.

- Option Factors: Considerations that determine the need for, or existence of, alternate approaches to a phase or activity in the development of computer programs.
- Package: A group of computer subprograms or CPC's, some of interim versions, which can operate as an entity in a computer to perform some subset of the system's functions.
- Performance: The execution of an action; that which is accomplished via some effort. In computer program production, this refers to the operations of a computer program subsystem or element thereof to effect a desired result; the result of actions taken by a contractor.
- Phase: A stage or interval in a development cycle or process.
- Practices: Accepted agreements on the procedures and standards which shall apply in a given project or activity.
- Procedure: A formal instruction, carrying management approval, which governs and prescribes the means by which personnel are to operate to accomplish an objective or produce an intermediate or end item (within the programming process).
- Procurement Process:  
All activities, associated with the lettering of a contract, during the time period from the identification of the need for computer programming end items to the award of a contract for procurement of those end items.
- Procuring Agency: The organization or group within NASA, or a system prime contractor which acts as the NASA agent, which contracts for the procurement of system components, subsystems, or other systems; the NASA buyer agent. In this report, unless specifically indicated otherwise, the procuring agency is considered to be the same as the using agency.

Programming Process:

All activities required for the production of a working documented computer program.

Qualification:

The test activity which serves as a basis for formal recognition that an implemented CPCEI is ready for shipment to a user's facilities (if appropriate), integration into a computer-based system, acceptance by NASA, and/or use.

Routine:

A set of coded instructions arranged in proper sequence to direct the computer to perform a desired operation or sequence of operations; a subdivision of a program consisting of two or more instructions that are functionally related. (See Computer Program.)

Simulated Environment:

An environment for testing and integrating computer programs into the operational subsystem, using computer programs and/or equipment which represent the operational environment.

Simulation:

The representation of physical systems and phenomena by computers, models or equipment; e.g., an imitative type of data processing in which a computer is used as a model of some entity.

Source Language:

The form of symbology used by a programmer in specifying the sequence composing a computer program.

Specification:

A detailed description of the characteristics of a computer program end product and of the criteria which must be used to determine whether the end product is in conformity with the stated description; also, the document, containing the detailed descriptions, that is used as the basis for configuration management of computer programming efforts.

- Standard: An accepted criteria for terminology, performance, practice, design, size, etc., to be adhered to in computer program development.
- Statement of Work: That portion of a contract or a request for a proposal which specifies a system to be developed or studies to be made (the task to be done by the contractor), the procuring agency's management system for technical monitoring of the task, and the technical practices and management control procedures to be followed by the contractor.
- Storage: A general term for any media which can retain information and from which information can be obtained at a later time.
- Subfunction: A subset of actions for which a person, equipment, computer program, or system is specially fitted or used.
- Subprogram: A part of a larger computer program which can be converted (from source language) into machine language independently; a separately compilable group of program instructions. This term is used throughout this report as synonymous with "Computer Program Component," except in Appendix VI where it is noted to be defined as a segment of the lowest level computer program component.
- Subroutine: A set of instructions, possibly a portion of a routine, necessary to carry out a well-defined mathematical or logical operation; a commonly used, specific purpose routine.
- Subsystem: A major functional subassembly or grouping of items, equipment, computer programs, or personnel which is essential to operational completeness of a system.
- Support Computer Programs: All computer programs required as aids in the design, integration, and/or testing of other computer program systems and/or in personnel training activities associated with a computer-based system; e.g., system simulation programs, simulation tests data generating programs, and system installation adaptation tools.

- System: Any self-sufficient combination of parts, assemblies, computer programs, and personnel grouped together to perform an operational function or functions.
- System Limits: The specifications of the maximum acceptable amounts by which the actual characteristics of a system or the components thereof may differ from the desired characteristics.
- System Tolerances: The specifications of the total permissible variation in the characteristics of a system or the components thereof.
- System Validation: See Validation.
- System Verification:  
See Verification.
- Technical Representative:  
The person (administrator, manager, or engineer) representing the procuring agency (NASA Center or agent) who is directly responsible for all technical aspects of the procurement of a computer program or computer-based system.
- Technical Work Statement:  
Same as Statement of Work.
- Tradeoff: A compromise, in the design of an equipment, computer program, system, or component thereof, between the cost elements of its procurement and operation and its performance characteristics, or between various design optimization factors.
- Training: Instructing personnel in the use of equipment, system operations, etc; conducting demonstrations and performing related activities.
- Transfer Function: An expression used for the relationship between the outgoing and the incoming signals of a process or control element, or the relationship between physical conditions at two different points in time or space in a given system.

User: The personnel or organizations utilizing the computer system (computer program and/or hardware).

Utility Computer Programs:

Computer programs used to generate the operational and support computer programs. These include compilers, assemblers, loaders, tape manipulation programs, program debugging tools, and program updating tools.

Validation: The activity which confirms the system design to satisfy the system requirements and perform the operational functions needed by the user; studies concerning the adequacy of design of a system for user's needs.

Verification: The test activity which checks a computer program element for its accuracy in meeting the stated requirements; studies concerning how completely an implemented system satisfies the system design.

Video Display: Presentation of display information on a cathode ray tube.

BELLCOMM, INC.

TABLE OF ACRONYMS

ADP	- Automatic Data Processing
APO	- Apollo Program Office
ASA	- American Standards Association
CCB	- Change Control Board*
CCBD	- Change Control Board Directive *
CDR	- Critical Design Review
CEI	- Contract End Item
CPC	- Computer Program Component
CMO	- Configuration Management Office
CPCEI	- Computer Program Contract End Item
DCPI	- Deliverable Computer Program Item
DPR	- Definition Phase Review
DRB	- Design Requirements Baseline
DRD	- Documentation Requirement Description
DRL	- Document Requirements List
ECP	- Engineering Change Proposal
FACI	- First Article Configuration Inspection
I/O	- Input/Output
MCC	- Mission Control Center
MSC	- Manned Spacecraft Center
MSF	- Manned Space Flight
NASA	- National Aeronautics & Space Administration

---

\*Current literature uses "CCB" (and CCBD) as an acronym for both "Change Control Board" and "Configuration Control Board"; both titles represent the same functional entity.

PDR - Preliminary Design Review  
PERT - Program Evaluation and Review Technique  
SCL - Specification Change Log  
SCN - Specification Change Notice  
SOW - Statement Of Work  
RFP - Request For Proposal  
TSR - Test Specification Review



# BELLCOMM, INC.

## DISTRIBUTION LIST

### NASA Headquarters

W. E. Beebe - MAP  
J. Costantino - MSR  
P. E. Cotton - MS  
L. E. Day - MAT  
J. H. Disher - MLD  
H. Hayman - MAT  
J. K. Holcomb - MAO  
D. M. Jones - MD-P  
T. A. Keegan - MA  
N. F. Lamb - MSP  
D. A. Linn - MSP  
W. E. Miller - MOG  
R. V. Murad - MAT  
J. H. Turnock - MA  
K. R. Webster - TN  
W. J. Willoughby - MAR  
S. Winn - MAP  
NASA Hq. Library - ATSS 10

### NASA Manned Spacecraft Center

H. L. Bent - ED  
W. C. Bradford - EB5  
E. H. Brock - ED  
A. Cohen - PS6  
R. C. Duncan - EG  
L. Dunseith - FM2  
R. K. Everett - ED  
S. Faber - CF33  
M. A. Faget - EA  
J. A. Frere - FS13  
C. C. Kraft - FA  
R. W. Lanzkron - PF  
H. E. Leech - FS3  
J. P. Loftus - PS4  
J. P. Mayer - FM  
H. G. Miller - FC  
J. A. Miller - FC  
D. T. Myles - FS2  
K. L. Nelson - PH  
J. H. Robinson - BR9  
J. C. Stokes - FM2  
T. Woods - ED

### NASA Marshall Space Flight Center

R. C. Balentine - R-QUAL-PS  
G. P. Barr - R-ASTR-ES  
C. L. Bradshaw - R-COMP  
C. L. Collins - R-ASTR-NDS  
P. J. Elliott - R-QUAL-PSC  
B. J. Funderburk - R-QUAL-PSC  
H. Hoelzer - R-COMP  
L. G. Richard - R-ASTR-S  
C. G. Riley - R-QUAL-PC  
T. H. Smith - E-S  
C. N. Swearingen - R-ASTR-ND  
G. L. Turner - R-ASTR-ND

### NASA Goddard Space Flight Center

O. M. Covington - 500  
J. J. Donegan - 554  
J. T. Mengel - 500  
T. Roberts - 512  
L. R. Stelter - 570  
H. F. Thompson - 512

### NASA Kennedy Space Center

R. H. Bruns - INS-1  
J. F. Burke - EDV-3  
B. A. Corbett - INS-112  
B. G. Griffin - INS-11  
R. K. Jenke - LVO-24  
J. R. Medlock - LVO-241  
R. D. Smith - PPR-12  
C. A. Whiteside - LVO-24

### Bureau of Standards

N. Ream - NBS  
S. Alexander - NBS

## BELLCOMM, INC.

### Distribution List Contd.

#### Bell Telephone Laboratories

J. A. Harr -HO  
T. N. Lowry - CB  
A. H. Scheinman - NY  
C. S. Sherrerd - WH  
R. B. Smith - HO  
H. E. Vaughn - HO

#### Bellcomm

J. O. Cappellari  
K. R. Carpenter  
F. C. Childs  
R. E. Gradle  
E. L. Gruman  
D. R. Hagner  
W. G. Heffron  
W. C. Hittinger  
B. T. Howard  
B. Kaskey  
C. M. Klingman  
R. J. Litschgi  
W. J. Martin  
J. Z. Menard  
V. S. Mummert  
I. D. Nehama  
J. M. Nervik  
J. J. Rocchio  
I. M. Ross  
P. S. Schaenman  
A. L. Schreiber  
R. V. Sperry  
W. Strack  
C. M. Thomas  
T. H. Thompson  
R. L. Wagner  
Department 1023  
Central File  
Library

9 Dec 66 31258

2 cys

