Part II—Final Report, Tasks 3 through 8

FEASIBILITY STUDY OF AN INTEGRATED PROGRAM FOR AEROSPACE VEHICLE DESIGN (IPAD)

Volume VI: IPAD System Development and Operation

D6-60181-6
September 21, 1973

Prepared under Contract No. NAS1-11441 by
Boeing Commercial Airplane Company
P.O. Box 3707
Seattle, Washington 98124

for

Langley Research Center
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No

4. Title and Subtitle
FEASIBILITY STUDY OF AN INTEGRATED PROGRAM FOR AEROSPACE VEHICLE DESIGN (IPAD)
VOLUME VI - IPAD SYSTEM DEVELOPMENT AND OPERATION

5. Report Date
September 21, 1973

6. Performing Organization Code

7. Author(s)
D. D. Redhed, A. S. Kawaguchi
L. L. Tripp, R. E. Miller, Jr.

8. Performing Organization Report No.
D6-60181-6

9. Performing Organization Name and Address
Boeing Commercial Airplane Company
P. O. Box 3707
Seattle, Washington 98124

10. Work Unit No.

11. Contract or Grant No.
NAS1-11441

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, D. C. 20546

13. Type of Report and Period Covered
Contractor Report

14. Sponsoring Agency Code

15. Supplementary Notes
Project Manager, Dr. R. E. Fulton, Structures and Dynamics Division,
NASA Langley Research Center, Hampton, Virginia 23365

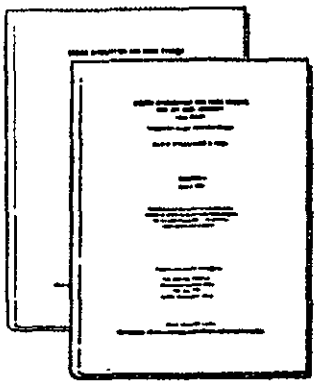
16 Abstract

Volume VI of the Boeing report on Tasks 3, 4, and 5 of the IPAD feasibility study presents the implementation plan. The strategy proposes a three phase development of the IPAD system and technical modules, and the transfer of this capability from the development environment to the aerospace-vehicle design environment. The system and technical module capabilities for each phase of development are described. The system and technical module programming languages are recommended. The initial host computer system hardware and operating system are also recommended. The cost of developing the IPAD technology is estimated. A schedule displaying the flowtime required for each development task is given. A PERT chart gives the developmental relationships of each of the tasks. An estimate of the operational cost of the IPAD system is given.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Integrated Systems Implementation Strategy Cost Schedule	

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	115	[redacted]

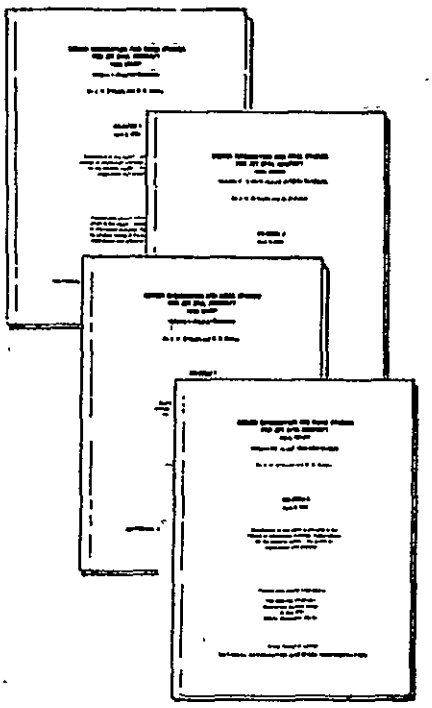
FEASIBILITY STUDY OF AN INTEGRATED PROGRAM FOR AEROSPACE VEHICLE DESIGN (IPAD)



Volume IA
Summary of IPAD Feasibility Study
D6-60181-1A

Volume IB
Concise Review of IPAD Feasibility Study
D6-60181-1B

Part I—Final Report, Tasks 1 and 2



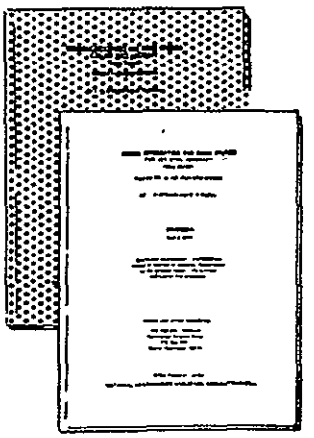
Volume II
The Design Process
D6-60181-2

Volume III
Support of the Design Process
D6-60181-3

Volume IV
IPAD System Design
D6-60181-4

Volume V
Catalog of IPAD Technical Program Elements
D6-60181-5

Part II—Final Report, Tasks 3 through 8



Volume VI
IPAD System Development and Operation
D6-60181-6

Volume VII
IPAD Benefits and Impact
D6-60181-7

SUMMARY

The implementation strategy proposes the development of a baseline system for delivery to NASA/Langley Research Center and a parallel effort to transfer IPAD technology from the development environment to the industrial aerospace-vehicle design environment. A three phase plan is presented, the first phase of which results in an initial operational system in four years.

The Phase 1 implementation of IPAD will provide a suitable balance between technical capability and system capability. Adequate technical modules will be available to support development and checkout of the IPAD concepts on some meaningful design case studies. The Phase 1 implementation of IPAD will provide some analysis capability for Project 1 subsonic class airplanes (see Volume II). This technical capability will provide the basis for vehicle analysis/design calibration since it will be possible to relate the subsonic class of airplane to both existing in-service airplanes and to studies which are continually in progress for this class of airplane. The Phase 1 implementation will also provide some analysis capability for Project 2 supersonic class airplanes (see Volume II). This technical capability will orient IPAD to future vehicle design problems. It is expected that technical capability for Phases 2 and 3 will reflect NASA and industry experience with Phase 1 capability.

The implementation of the IPAD system will basically follow the Phases 1, 2, and 3 enumerated in Table 2.1 of Volume IV. The initial operational system developed during Phase 1 will provide the principal IPAD design feature of continuity of task and time. Continuity will be supported through the development of the user/IPAD interface, and subtask and community libraries. During Phase 2, the system will be extended to include limited information control through the capability of defining project plans and reports. The full capability of the IPAD system design will be completed during Phase 3. This includes full support of project planning and reporting, specialized interactive capability, and full privacy and security provisions.

The successful transfer of IPAD technology from the NASA development environment to the industrial problem solving environment is mandatory. A technology transfer strategy which

considers both the commercial, proprietary interests of industry and the broad public interests of NASA is presented as follows:

- NASA will contract for the evaluation of the IPAD system software in realistic product design studies.
- Contractors will be required to perform these studies utilizing the IPAD system with the contractors own technical modules and deliver only reports covering technical and economic evaluation of their use of IPAD. . .

Cost and schedule estimates for two implementation plans are presented. The first plan, in which Phases 1, 2, and 3 are developed sequentially, requires nearly nine years and 13.2 million dollars for full IPAD development. The alternate plan, in which Phase 1, 2, and 3 development is overlapped, results in a fully developed IPAD in six years, but the cost is increased to 13.7 million dollars. For both plans, Phase 1 implementation is expected to be completed in four years and is estimated to cost 6 million dollars for IPAD system development and 3 million dollars for the development of the technical modules.

CONTENTS

CONTENTS (Continued)

ILLUSTRATIONS

TABLES

1.0 INTRODUCTION

The development of IPAD will produce two major products: a) the IPAD system capability and b) a technical vehicle design capability. The system design will accommodate technical capability limited only by the time and money invested in the Operational Modules (OMs). The technical capability may be oriented in any technical direction desired. In contrast, the system implementation will be general and will be finished in a finite time, except for continuing refinements that are added for convenience.

To ensure that IPAD as a tool is widely used, the IPAD technology must be transferred from its development environment to the using community at large. Consequently, the implementation plan also includes a specific strategy to transfer this IPAD technology to the industrial using community.

2.0 ANSWERS TO TASK 3 QUESTIONS 1, 2, AND 3

2.1 Can IPAD be developed by a single organization (company)?

For development purposes, the IPAD system and OMs should be considered separately. The system development should be done by a single organization since it must have a unified structure and logic throughout all the components. However, the OMs may be developed and contributed by a large (and even diverse) set of organizations. One of IPAD's goals is to make the use of OMs developed under such conditions more common in the future.

2.2 Is it appropriate or desirable to divide the development work among industry, government, university? Should anyone of the three develop it alone?

As noted in section 2.1, the system should be developed by a single organization and further to ensure an adequate emphasis on the requirements of the air vehicle design process, an aerospace firm should be the IPAD system developer.

From the OM point of view, no single company, university, or government agency has a monopoly on technical design and OMs. Thus, the OM development should take advantage of the expertise throughout the U.S.A. and should be contracted for on a broad basis.

2.3 What problems are associated with the inclusion of proprietary codes and ideas into the development?

Allowing proprietary codes and ideas to reside in IPAD poses two primary concerns for OM owners:

- Will I be adequately compensated for code and ideas which I am willing to release?
- Can I be assured of control of the code so that I am protected from
  - use by unauthorized user
  - copies being made of code which is not for sale.

The concern about adequate compensation applies only to "pre-existing" proprietary code which is desired for the IPAD technology. Some kind of cost reimbursement NASA procurement practice will be required if these codes are to be made part of the "public" IPAD technology.

The concern about control of the use of proprietary code will have to be met in the IPAD system design. Today every commercial computing service faces a similar problem in permitting competing customers to use their services simultaneously. Experience to date does not indicate that a totally secure system is possible. IBM is near the end of a $40 million security study (reference 1) which has produced some valuable recommendations and procedures, but no "final answers" to the security and control problem. In principle, the user with proprietary code can restrict the use of his code as he desires, (i.e., the IPAD design has security and control features which can protect the owner from all "normal" security violations). However, it must be finally noted that no system has been shown to be secure from the dedicated thief.

3.0 IMPLEMENTATION STRATEGY

The strategy for implementing IPAD technology is to develop a baseline system for delivery to NASA/Langley Research Center and to simultaneously transfer this technology from the development environment to industrial use (figure 3.1).

Because of the length of time involved in the total IPAD development, it is necessary to phase the implementation and to deliver, in a shorter time period, an initial operational system that demonstrates the concept of the IPAD technology. Thus the IPAD baseline system and technical capability will be implemented in three phases. Phase 1 is intended to be the minimum capability of an IPAD like system, while Phases 2 and 3 are aimed at enrichment of the basic capability. As soon as Phase 1 is complete, NASA usage and the transfer to industry will begin.

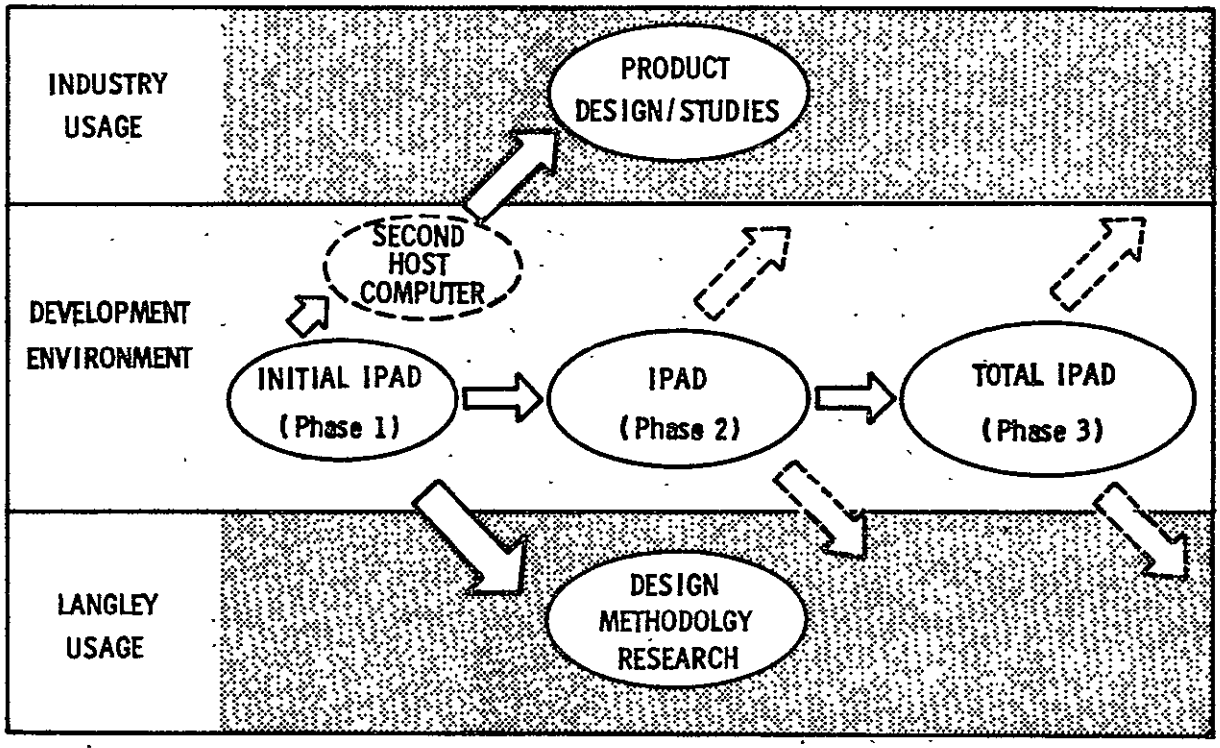


Figure 3.1 IPAD Implementation Strategy

The baseline system will be installed at Langley Research Center to:

(1) evaluate system features and to guide the remaining system development,

(2) evolve air vehicle design methodology consistent with the IPAD system concepts and,

(3) demonstrate IPAD system to potential users.

The transfer of IPAD technology to industrial use will be pursued during the development of the system. It is probable that changes to the baseline IPAD will be required to adapt the system to the industrial environment. These changes could involve conversion of baseline IPAD to a second host computer system or modifications of the system capabilities.

3.1 Implementation Phase Goals

The goals of each phase are:

Phase 1

- Demonstrate the utility of the IPAD concepts on total vehicle design studies.
- Demonstrate the task/time continuity capabilities of the IPAD system design.
- Develop a strategy for transferring Phase 1 IPAD technology from the development environment to an industrial environment.

Phase 2

- Extend the Phase 1 system capability to include limited information control through the capability of defining project plans and reports.
- Expand IPAD usage to industry.
- Demonstrate effective man/machine methodology on aerospace vehicle problems at Langley Research Center.
- Determine the impact of fourth generation computers on the development during Phase 3.

Phase 3

- Demonstrate the full capability of the IPAD system design.
  - Full support of project planning and reporting
  - Specialized interactive capability
  - Full privacy and security provisions.

3.2 Implementation Plans

The implementation goals are supported by plans which provide for phased technical and system capability, technology transfer through contracted studies by industry using IPAD, and usage support that ensures maintenance, training, and OM additions.

The Phase 1 implementation of IPAD will provide a suitable balance between technical capability and system capability. Adequate technical modules will be available to support the development and check out of the IPAD concepts on some meaningful design case studies.

Broad user acceptance of IPAD will be enhanced if some technical capability is developed for each design level during the initial implementation. The initial design and analysis capability for IPAD will be limited to modification of existing code and to the development of new code in a few design areas which are required to make the initial IPAD a useful design tool. This initial capability based primarily on existing code should establish an incentive to incorporate new code into IPAD.

Since a fully developed IPAD extends over nearly nine years if the development phases are sequential, an alternate plan for phase overlapping is presented. The sequential development has the following potential benefits:

- Maximum opportunity for feedback into Phases 2 and 3 from NASA and/or industry usage.
- A significantly lower yearly cash flow.

If Phases 2 and 3 are overlapped with Phase 1 and with each other, development time reduces to 6 years but:

- Feedback from actual usage will not impact the implementation.

- The yearly cash flow rises significantly.

No specific technical capability development is planned beyond Phase 1. It is expected that technical capability for Phases 2 and 3 will reflect NASA and industry experience with Phase 1 capability.

The implementation of the IPAD system will basically follow the Phases 1, 2, and 3 enumerated in Table 2.1 of Volume IV.

Appendix A presents considerations for the organization and management of a software project the size of IPAD.

3.2.1 Phase 1 Technical Capability

Phase 1 technical capability is described in detail in Volume II. The Phase 1 implementation of IPAD will provide Level II and III analysis capability for Project 1 subsonic class of airplanes. This technical capability will be adequate to support the development and checkout of the IPAD system. It will also provide a basis for calibration since it will be possible to relate the subsonic class of airplane to both existing in-service airplanes and to studies which are continually in progress for this class of airplane.

The Phase 1 implementation of IPAD will also provide Levels II and part of Level III analysis capability for Project 2 supersonic class airplanes. This technical capability will orient IPAD to future vehicle design problems. This will help demonstrate the value of the IPAD system for evaluation of complex design studies for which past experience is limited.

In addition, the interface with manufacturing at Level VI will be initiated during Phase 1 implementation.

3.2.1.1 Technical Module Selection

It is assumed that the primary modules from Level II and Level III subsonic transport networks can be obtained from such existing sources as the Boeing Computerized Preliminary Design System (CPDS) (references 2 and 3). Approximately 75% of the required code is assumed to exist in a form which can be incorporated into IPAD.

The primary modules for Level II and the geometry sizing part of Level III for the supersonic transport networks should be able to be derived from the subsonic modules. Approximately 15% more code is required. It is assumed that approximately 90% of this code exists. The existing code has been executed only as separate programs, therefore considerable work is required to develop the interface between these modules. Much work has been done in support of the Level V and VI interactive design of detailed parts.

The Phase 1 modules have been selected from those listed in Volume V. The list of modules is given in Appendix B.

3.2.1.2 Phase 1 Technical Module Language

It is proposed that the IPAD project adopt as its technical module language the American National Standards Institute (ANSI) standard FORTRAN in effect at the start of IPAD implementation and as implemented by the Control Data Corporation (CDC) for its CYBER machines. In the event the compiler for the standard FORTRAN is not available when needed, the alternate standard will be the FORTRAN language as implemented in the primary CDC FORTRAN compiler. It is anticipated this will be FTN (Reference 4).

The factors which affect the choice of a technical module language are:

- The life span of IPAD
- Past and present commitment to FORTRAN as a scientific applications language
- Resistance to change
- Continuing activities of the ANSI committee X3J3
- Technical improvements to programming languages
- Desire to improve the quality of software
- Benefit of using a standardized language

Since the approval of the last national standard FORTRAN some six years ago, the ANSI X3J3 committee has been considering changes to the standard. As of March 1973, 109 changes have been proposed. If approved in its present form, the proposed standard, known as FORTREV, will much more closely reflect the FORTRAN capabilities presently offered by major hardware

vendors. Presently, it is anticipated that the proposed standard will be presented for national adoption in the fourth quarter of 1973.

3.2.2 Phase 1 System

The initial operational system developed during Phase 1 will provide the principal IPAD design feature of continuity of task and time. Continuity will be supported through the development of the user/IPAD system interface, and subtask and community libraries. In addition, the initial system will include the capability of job construction and execution.

3.2.2.1 Use of Currently Existing Executive Routines

Since there are a number of "integrated systems" in existence today, the feasibility of beginning IPAD development with existing software as a base was examined. Prior to examining a few specific systems, some general observations were made. The basic characteristic of IPAD which distinguishes it from other systems is continuity over task and time. This requires specific software features to maintain continuity of the user's activities on the computer over time periods which are dictated by the nature of this work (see Volume IV section 3.1 for detail description). This characteristic shows up in several distinct system features, without which IPAD could not be clearly distinguished from several current systems. Thus a key question is how readily could the feature of continuity over task and time be added to another system, and how much of the original system would survive the change.

Related to this question is the contrast of innovation and evolution to gain desired system characteristics. If new characteristics represent a radical change, an evolutionary type development is not likely to produce the change in a reasonable time period. The potential savings realizable through use of existing software must be assessed, independent of the technical value of the software.

First, it is the code itself that is to be used, not its design features. The developers of IPAD are assumed to be under the burden of taking advantage of current software ideas whenever applicable. Once the IPAD design is complete, the judgement would have to be made about the suitability of the code. Possibly one would alter the design in an acceptable way in order to accommodate the code. The real question then is, how much code is really at stake? The majority of the "normal" executive functions are carried on in the IPAD design node F

(see Volume IV, Section 6.2), and some related functions are in a few other nodes. Node F is estimated at less than 2% of the code in the Phase 1 system. The risk of using such code is that it may not really fit the design in a proper fashion, resulting in some kind of adaption to make it fit. Thus the potential benefits are too small to justify much of any risk.

Second, there is the possibility of using another system in total as a starting point. Four representative systems are CPDS (references 2 and 3), ATLAS (reference 5), COMRAD (reference 6), and NASTRAN (reference 7). The IPAD characteristic of continuity over task and time is missing from all four systems. Each system has its own method of executive control, ideas from which IPAD will undoubtedly take direction in the appropriate design nodes. Only COMRAD is designed as an interactive system and thus is the only candidate for actual code usage. At this point the previous remarks concerning innovation versus evolution come to bear and the recommendation is that the IPAD design is innovative and should not be developed starting from a current system.

3.2.2.2 IPAD System Implementation Language

The proposed IPAD system implementation language is SYMPL (Reference 8). The language, SYMPL was originally developed by the Computer Sciences Corporation as part of the effort to implement the JOVIAL language on the UNIVAC 1108. The language was acquired by Control Data Corporation (CDC) as part of the JOVIAL language acquisition. The language has been or is being used by CDC to produce the SYMPL compiler, JOVIAL compiler, parts of the SCOPE 3.4 operating system, COBOL 5 compiler, PL/I compiler, ALGOL compiler and a data management system.

SYMPL offers the essential features of a system implementation language without any unnecessary overhead. The language has had extensive use on the recommended host CDC hardware. Appendix C contains an elaboration of the decision process used to select the implementation language. The AED-0 (reference 9) language is a serious contender with its major drawbacks being: lack of extensive use on the recommended host hardware, extra overhead of non-required features, and user documentation that is of quality less than desired.

3.2.2.3 Host System

The host hardware is assumed to be a CDC 6000 model. There are two choices of "standard" operating systems at this time; SCOPE 3.4 and KRONOS 2.1. Because of its greater orientation to

the IPAD environment of interactive processing, KRONOS 2.1 is selected. As discussed in Volume IV, the operating system will have to be modified to support IPAD, creating a special version of the operating system.

All currently planned modifications to the host computer operating system to support the IPAD system will be made during Phase 1. Two critical system capabilities absent on current CDC 6000 operating systems will be added. These are:

(1) multi-tasking within a single users terminal control and

(2) the ability to log off without losing continuity with activities in progress.

Multi-tasking is an operating system feature which allows a running program to command the execution of another program, usually in parallel with the originating program.

3.2.3 Phase 1 Usage Support

The intent of Phase 1 usage support is to provide the tools and procedures which will ensure system and OM maintenance, user training and OM additions.

3.2.3.1 Maintenance

IPAD maintenance during production usage must be capable of responding to user's requests for corrections and additions. The IPAD system features of continuity and integrity will be used to support all software which can impact the IPAD user's results. To do less would allow a serious erosion of the IPAD system design intent of "guaranteed" integrity of software. Thus the host operating system software (the modified version for IPAD) also needs to be maintained in the same manner as the OM and system software.

The procedures for long term maintenance of the system will be defined and implemented in Phase 1. These procedures include a means of reporting problems, suggesting solutions, suggesting improvements, and implementing changes.

Maintenance of OMs will be restricted to those considered part of "standard" IPAD and will be handled in a procedural sense similar to the system maintenance.

There will be periodic releases (updates) of IPAD with all known errors removed and additional capability added. The rational and procedures for such updates will be devised during Phase 1.

3.2.3.2 User Training

A program of training will be established for:

- Use of the IPAD system features.
- Design methodology in the IPAD environment.

Training will be provided, at NASA direction, during Phase 1.

3.2.3.3 Addition of OMs

Methods will be developed in Phase 1 for adding OMs. There are two basic categories of OMs at any installation of IPAD:

(1) Those OMs considered to be a part of the standard OM library.

(2) Those added by users to augment IPAD for their individual purposes.

The first category needs close control and well defined certification standards as well as criteria for judging future additions. The second category need be governed only by the local facility's ability to store and handle the OMs. The methods for adding OMs will encompass both requirements.

3.2.4 IPAD Technology Transfer

The transfer of the IPAD technology from the development environment to the industrial problem solving environment is mandatory for the success of the IPAD concepts. Normal commercial strategies for marketing and selling a high technology product do not apply directly in the IPAD situation because of the NASA role in the initial development. Nevertheless a technology transfer strategy, which considers both the commercial, proprietary interests of industry and the broad public, general industry interests of NASA can be structured in the following sequence:

- NASA contracts for the evaluation of the IPAD system software in realistic product design circumstances in

which contractor/s will be required to utilize the IPAD system with the contractor's technical modules on realistic case studies of product design.

- NASA selects successful contractor/s on the basis of cost, evaluation approach/depth, and technical quality/depth of case study design problems proposed by contractors.

- NASA, through their IPAD system contractor, will install the IPAD system software on the successful contractor's host computer.

- Contractors deliver only reports covering technical and economic evaluation of their use of IPAD.

It appears that the above strategy offers a flexible framework to achieve a balance of:

- The cost to NASA for installation on other host computer centers. However the resulting IPAD system software version could, after the contracting period, have multiple public use and availability.

- The cost to contractors to adapt a limited set of their technical modules to the IPAD system. Such cost to be borne by them but the modules also can remain proprietary.

- The benefits to NASA to achieve industry involvement in IPAD as a user, rather than the more sterile role of developer.

- The benefits to contractors to have an early opportunity, perhaps competitive advantage, to learn utilize, and evaluate a possible significant productivity advance in problem solving capability.

Initial exploratory work will be required to seek out interested organizations and to refine this technology transfer strategy.

The technology transfer activity may require the system to be operational on a host computer different from the NASA Langley host. If this is necessary, work may begin during phase 1 in parallel with the implementation at NASA Langley.

3.2.5 Phase 2

The product technical capability for Phase 2 is expected to primarily reflect the experiences of NASA and industry with Phase 1 capability. No specific technical module developments are planned.

The IPAD system capability for Phase 2 development is described in Volume IV. Detailed task planning for Phase 2 has not been done, but it is expected to be similar to Phase 1, with the following considerations of tasks completed in Phase 1:

- The total system design is complete.
- All basic operating system modifications are completed.
- Conversion, development and maintenance aids have been developed.
- Most standards and procedures have been developed.

The major Phase 2 development includes:

- Insertion of project management control capability
- Broader system utility support

3.2.6 Phase 3

The product technical capability for Phase 3 is expected to primarily reflect the experiences of NASA and industry with Phases 1 and 2. No specific technical module developments are planned.

The IPAD system capability for Phase 3 development is described in Volume IV. The Phase 2 IPAD will be extended to include:

- Completion of the control aspects of project plans
- Insertion of "extra" features to enhance usability
- Interactive graphics

The development of IPAD beyond Phase 3 could include broader system utility capability in the form of:

- Information retrieval and display
- General interactive support

4.0 IPAD IMPLEMENTATION

Implementation of the baseline IPAD is described in terms of development tasks, costs and schedules. Phase 1 costs and schedules were developed from detailed tasks (see Appendix D). Detailed tasks have not been developed for Phases 2 and 3. Two plans for implementation beyond Phase 1 are presented. The difference between the plans is only one of scheduling - the tasks are identical.

4.1 Phase 1 Baseline Development Tasks

The IPAD system capability for Phase 1 development is described in Volume IV. The tasks associated with the development of this capability are divided into four main groups and are listed below.

Technology Transfer

- Draft descriptions of work statements
- Survey potential bidders
- Formulate and issue Request for Proposals

Preparation of OMs

- Develop OM selection criteria
- Prepare OM language specifications
- Specify OM conversion aids
- Select and collect OMs
- Prepare the OM language
- Prepare the conversion aids
- Prepare the IPAD data bank
- Specify the OM certification tests
- Prepare the Phase 1 OMs
- Prepare OMs for system checkout

System Design

- Establish design procedures and standards

- Refine the Level 1 design
- Establish coding/certification standards and procedures
- Specify the host operating system modifications
- Specify the development and maintenance aids
- Produce the system language specifications
- Prepare the system language
- Produce the total system design

System Implementation

- Define the Phase 1 capability
- Produce all development and maintenance aids
- Perform the operating system modifications
- Establish training methods and train users
- Code and certify the system
- Certify the Phase 1 OMs in the system
- Plan and prepare computing facilities

4.2 Baseline Development Costs and Schedules - Plan 1

4.2.1 Resources and Flow Time

Individual tasks and their associated costs in man months and computer resources are given in tables 4.1, 4.2, 4.3, and 4.4. These costs should be interpreted using the following:

- Estimates for individual tasks are for technical work plus supervision and miscellaneous costs.
- Costs for computing time are estimated on current commercial rates from vendors of the CDC 6600.
- Dedicated machine costs are taken from a vendor's rate book for a 1 year lease.

Table 4.1 Summary Table for Technology Transfer

Activity Designator	Description	Start Month	Months flow time	Man Months		Computing Resources
				Prog.	Eng	
TT-1	Draft work statement descriptions	13	9	5	11	None
TT-2	Survey potential bidders	22	3	2	4	None
TT-3	Formulate and issue RFPs	25	12	2	7	None
	TOTALS		24 months duration	9	22	None

Table 4.2 Summary Table for Preparation of OM's

Activity Designator	Description	Start Month	Months flow time	Man-Months		Computing Resources
				Prog.	Eng.	
OM-1	Establish OM selection criteria	1	1	-	2	None
OM-2	Prepare OM language specifications	2	1	1	-	None
OM-3	Specify OM conversion aids	2	2	5	-	None
OM-4	Collect all Phase 1 OMs	2	3	-	7	None
OM-5	Prepare OM language	3	3	4	-	$600 computing time
OM-6	Prepare OM conversion aids	5	6	23	-	$4,000 computing time
OM-7	Prepare IPAD data bank	11	6	-	15	$1,000 computing time
OM 8	Specify OM certification tests	11	6	8	15	None
OM-9	Prepare Phase 1 OMs	11	26	247	259	$200,000 computing time
OM-10	Prepare system checkout OMs	11	2	2	5	$2,000 computing time
	TOTALS		36 months duration	290	303	$207,600 computing time

Table 4.3 Summary Table for System Design

Activity Designator	Description	Start Month	Months flow time	Man-Month		Computing Resources
				Prog.	Eng.	
SD-1	Establish design procedures and standards	1	1	3	-	None
SD-2	Refine Level 1 design	2	1	3	3	None
SD-3	Establish initial coding, certification standards	3	1	3	-	None
SD-4	Complete coding, certification standards	4	1	3	-	None
SD-5	Specify host operating system modifications	3	2	3	-	None
SD-6	Specify maintenance and development aids	3	2	6	-	None
SD-7	Produce system language specifications	1	3	4	-	None
SD-8	Prepare the system language	4	6	12	-	$2000 computing time
SD-9	Develop total system design	3	18	173	17	$2000 computing time
	TOTALS		20 months duration	209	20	$4000 computing time

Table 4.4 Summary Table for System Implementation

Activity Designator	Description	Start Month	Months flow time	Man-Months		Computing Resources
				Prog.	Eng.	
SI-1	Define Phase 1 system capability	21	1	4	4	None
SI-2	Produce development and maintenance aids	9	8	32	-	$6,000 computing time
SI-3	Perform basic operating system modifications	5	9	36	-	$3,000 computing time 1050 hours block time
SI-4	Complete all operating system modifications	14	3	12	-	$1000 computing time 156 hours block time
SI-5	Establish user training methods	22	2	4	4	$400 computing time
SI-6	Code and certify the system	22	16	196	-	Dedicated machine for $941,472
SI-7	Train users	38	2	1	1	$1,000 computing time
SI-8	System–OM Integration	38	13	92	92	Dedicated machine for $926,029
SI-9	Establish plan for the IPAD computing facilities	1	1	1	-	None
SI-10	Prepare computing facilities	2	38	5	-	None
	Phase 1 totals		Critical Path–50	383	101	$11,400 computing time 1206 hours block time $1,867,501 dedicated machine **
SI-11	Deliver IPAD Phase 2	51	30	160	50	$1,000,000 computing time
SI-12	Deliver IPAD Phase 3	81	24	120	40	$780,000 computing time

* A dedicated computer for SI-3 and SI-4 activities would cost $500,352 for a 9 months period and would replace the block and computing time estimates.

** If a dedicated computer is used for SI-2 and SI-4, the total dedicated machine cost would be $2,387,853.

The identifiers associated with each task have the following meaning:

TT-i - Technology Transfer

OM-i - Preparation of OMs

SD-i - System Design

SI-i - System Implementation

The detail estimates for each individual task are given in Appendix D.

Figure 4.1 shows the total system development costs including computer resources and figure 4.2 shows the OM preparation costs. The labor costs are based on the following rate table over the total IPAD development period.

Year	Average Labor Rate/Man Year
1975	$55,000
1976	$57,800
1977	$60,600
1978	$63,700
1979	$66,900
1980	$70,200
1981	$73,700
1982	$77,400
1983	$81,300

The dedicated machine costs reflect only hardware rental and maintenance. No figures are included for standard overhead items such as electricity, operators, paper, building, etc. An estimate for these items would be 15% of the hardware costs. The OM preparation costs do not include the cost to buy the existing OMs; only the cost of converting OMs to IPAD is included.

ORIGINAL PAGE IS
OF POOR QUALITY

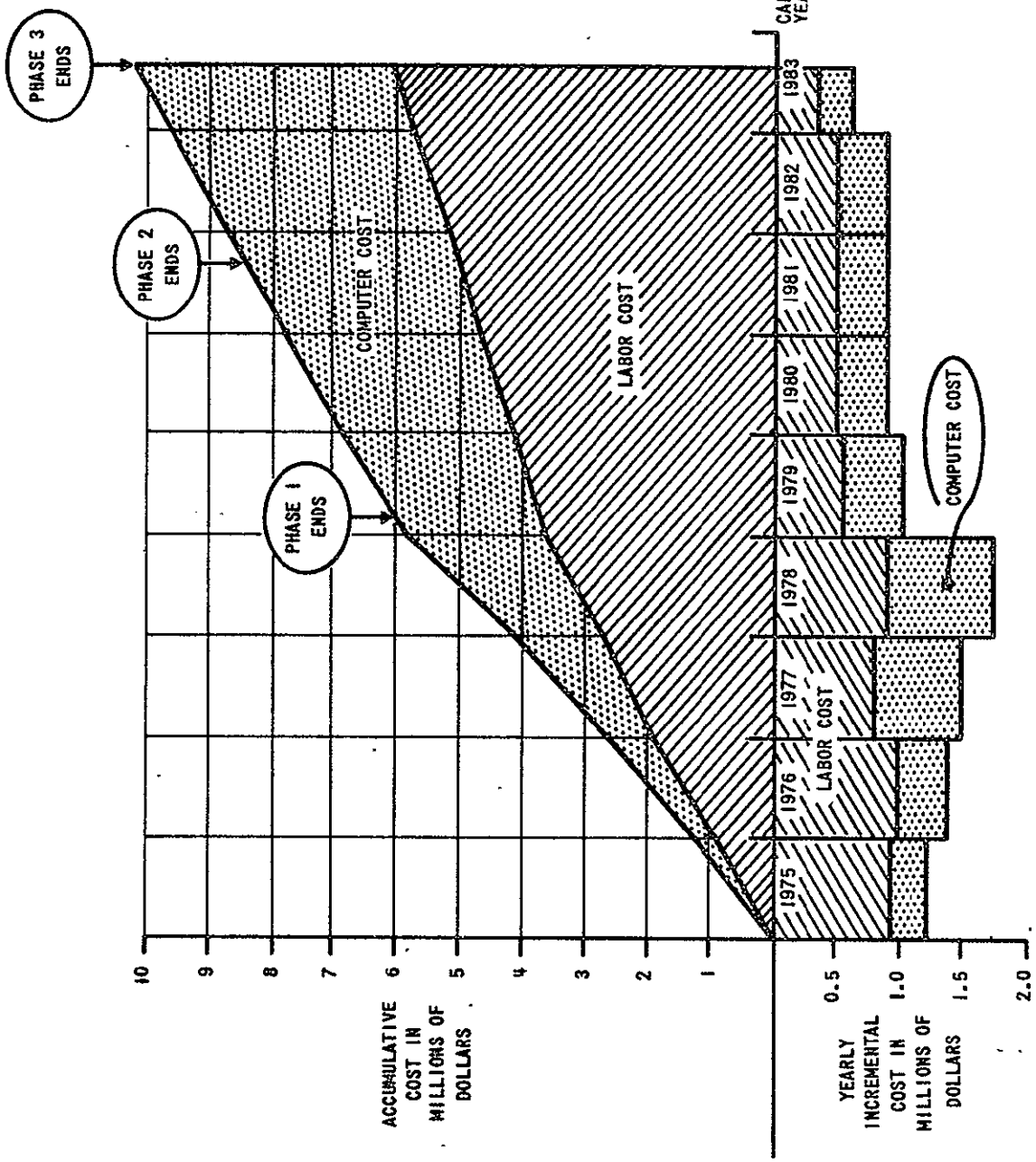


Figure 4.1 Summary of System Development Costs - Plan I

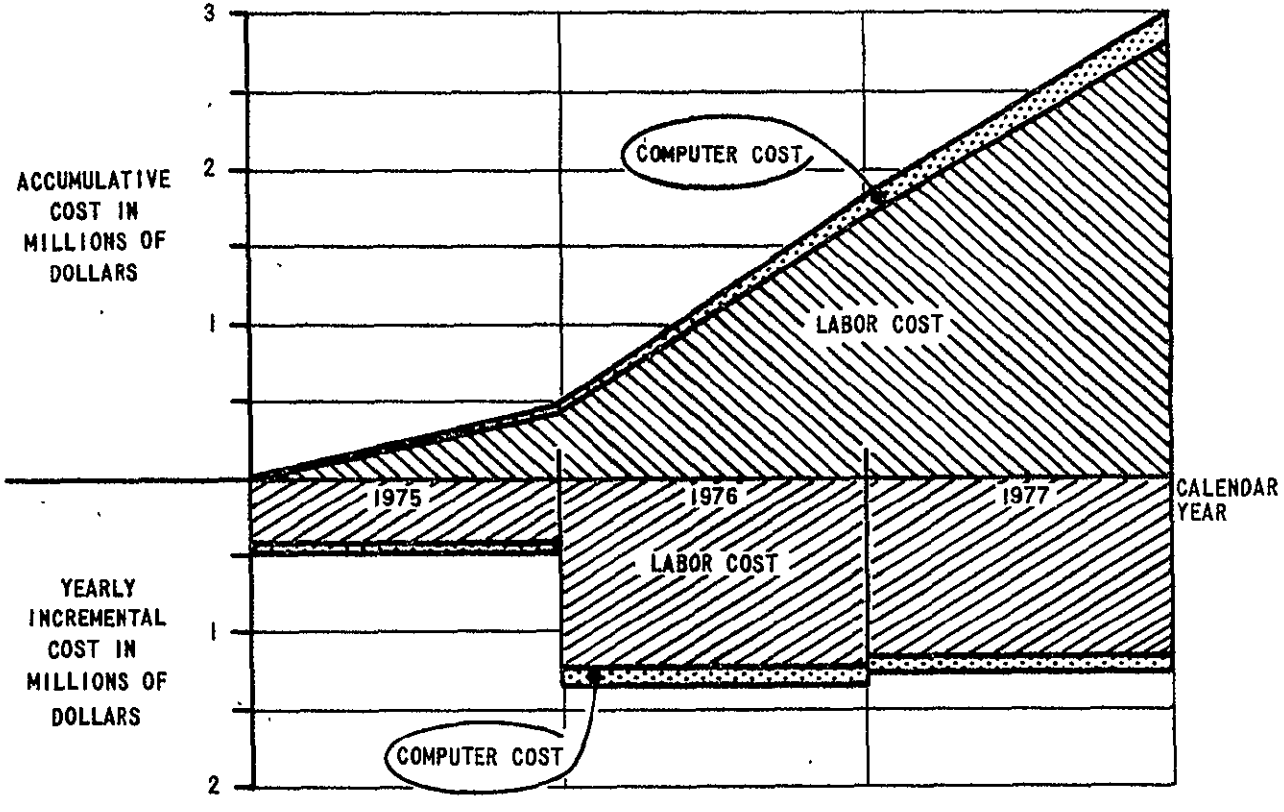


Figure 4.2 Summary of Phase I OM Preparation Costs

4.2.2 Schedules

Figure 4.3 shows the development PERT chart (each solid line represents an activity, each dotted line a schedule constraint, and each circle an event). The designators associated with each activity line correspond to the activities listed on tables 4.1, 4.2, 4.3, and 4.4. The activities with double lines are on the critical path.

Figure 4.4 is a development schedule based on the PERT chart. No attempt has been made to optimize manpower loading. This schedule shows the task relationships on a time basis to provide further insight to the critical path, which is identified by the double lines on the chart. The following activities represent the critical path for Phase 1.

- (SD-1) - Establish design procedures and standards
- (SD-2) - Refine Level II design specifications
- (SD-9) - Produce the total system design
- (SI-1) - Define Phase 1 capability
- (SI-6) - Code and certify system
- (SI-8) - System/OM certification

The major activities on the critical path are: (SD-9) - Produce total system design and (SI-6) - Code and certify system. Any attempt to shorten the Phase 1 development time will have to concentrate on these activities.

4.3 Baseline Development Costs and Schedules - Plan 2

The tasks for the overlap plan are identical to the non-overlap plan. As depicted in figure 4.5, Phase 2 overlaps Phases 1 and 3 and Phase 3 slightly overlaps Phase 1. Phase 2 begins when the system coding and checkout is finished. Phase 3 begins in the fifth year.

The schedules for the three individual phases are assumed to be unaffected by the overlap. However, there is a potential for some re-development in Phases 2 and 3 due to the overlap, and a 20% cost penalty was applied to the Phase 2 and 3 work.

The same manpower cost schedule is applicable, and the basic Phase 2 and 3 costs go down due to lower labor rates. The

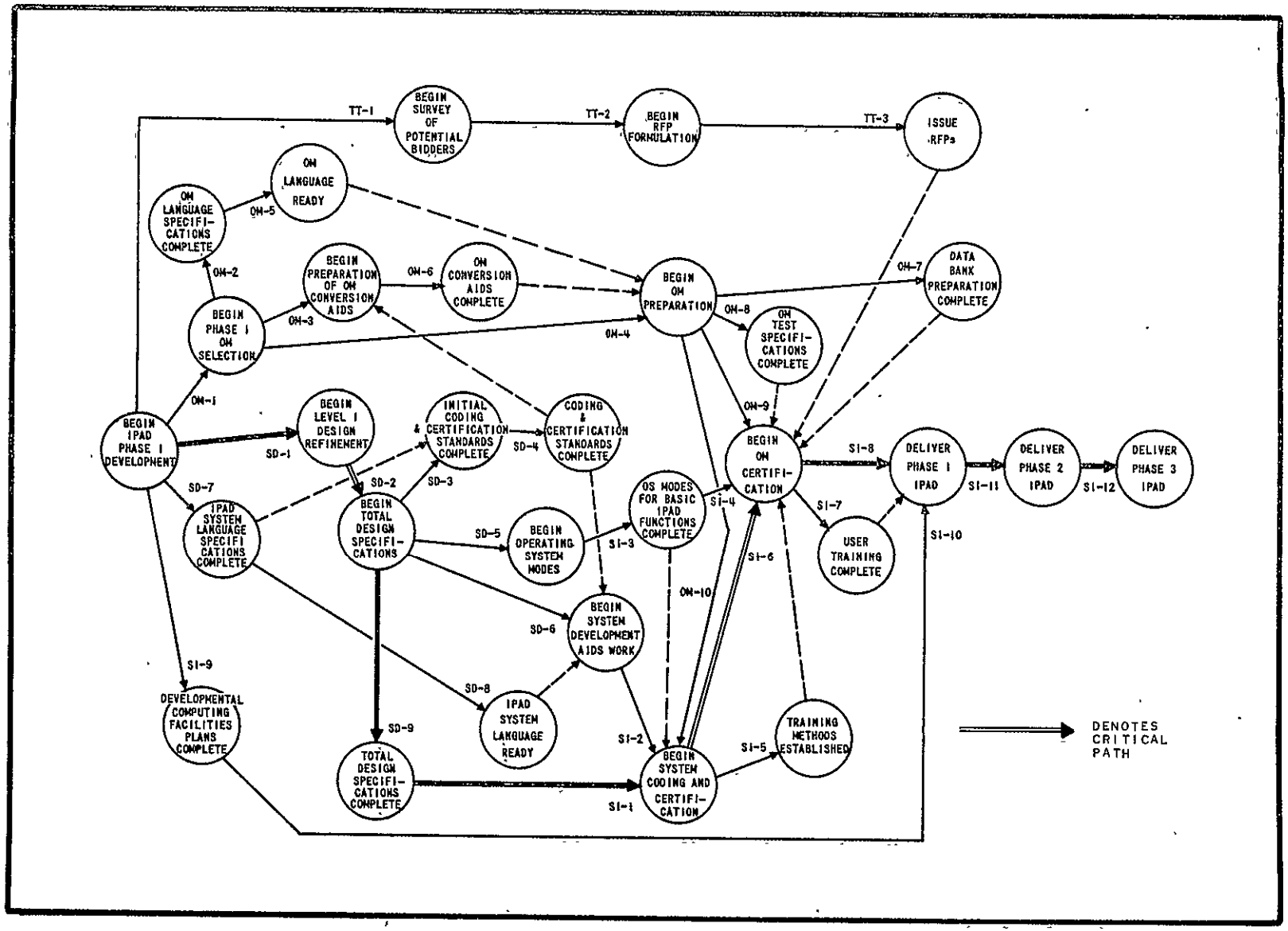


Figure 4.3 PERT Chart for Phase I IPAD Development

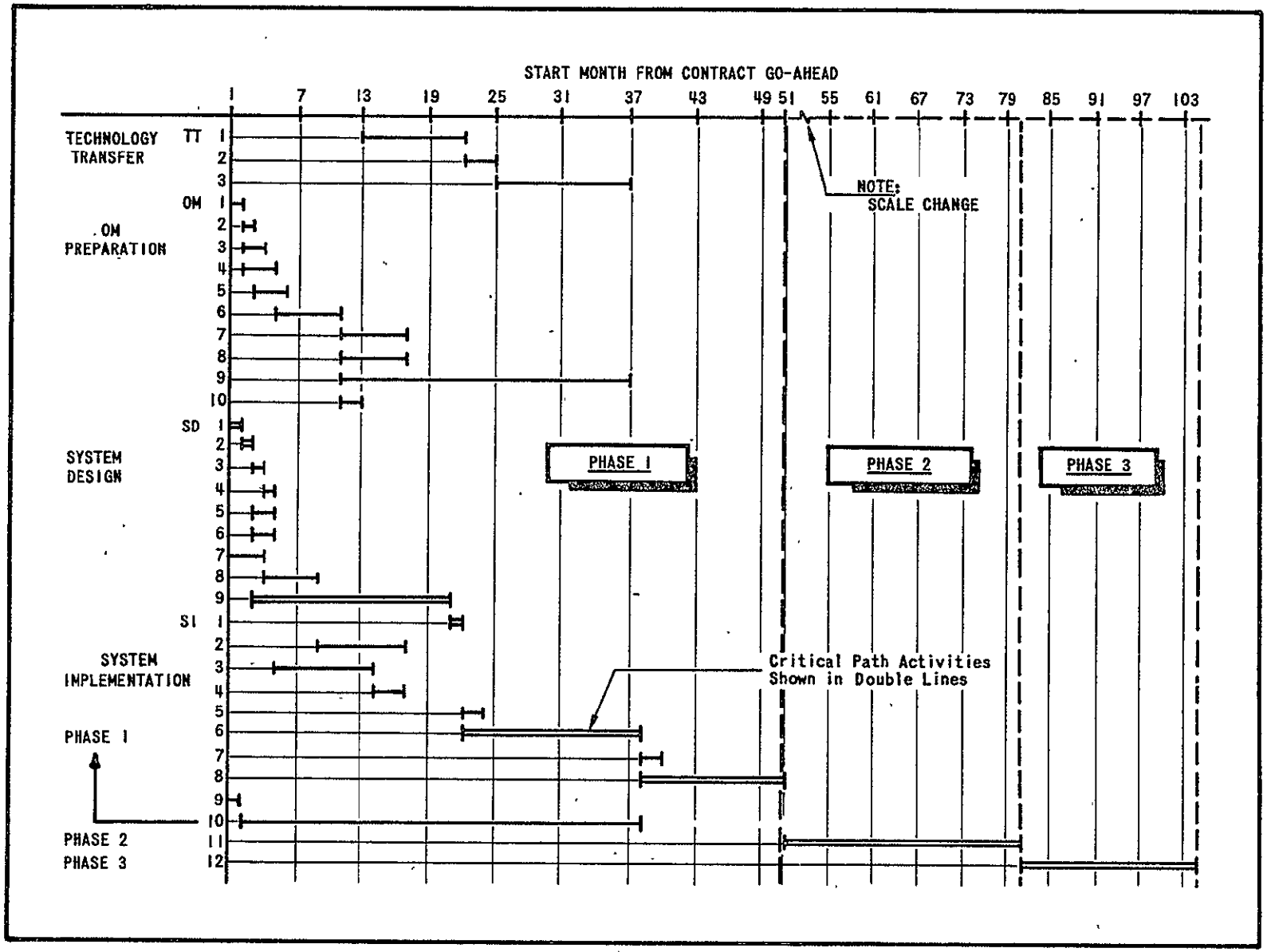


Figure 4.4 IPAD Development Schedule

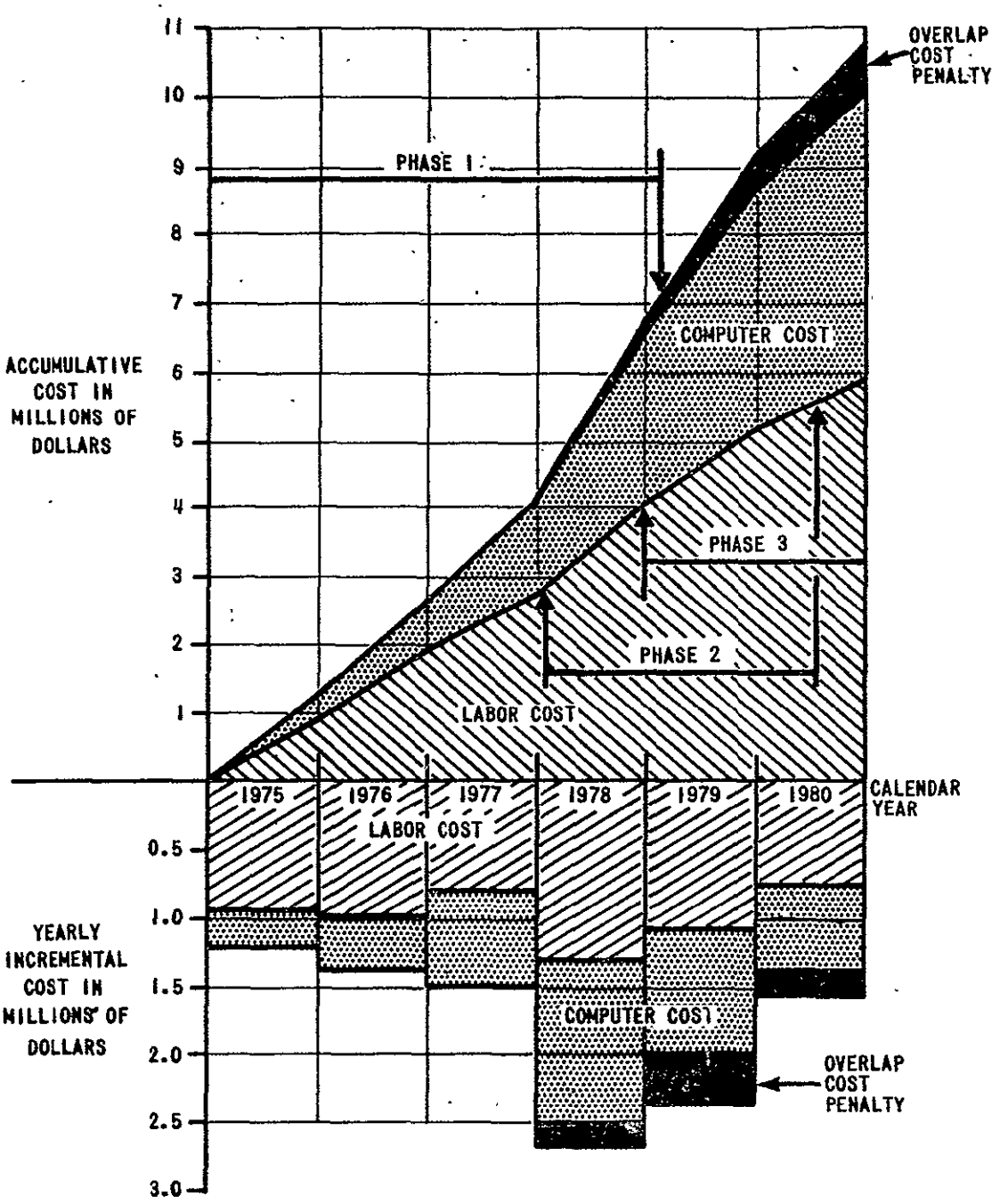


Figure 4.5 Summary of System Development Costs – Plan 2

added overlap penalty costs, however, give a net increase of about $500,000 for a total cost of $10,700,000.

4.4 Technology Transfer and NASA Usage Costs

4.4.1 Technology Transfer

The technology transfer plans call for the issuing of one or more RFP's. The costs shown in figures 4.1 and 4.5 include the effort to specify such RFP's, but not the cost of the resulting contracts. No information is provided on which to base the cost estimates.

In the event a decision is made to install the IPAD system on a different host computer than the baseline system, additional costs would arise. The tasks SI-6, -7, and -8 (see section 4.2) would have to be redone, and all the OM's would have to be converted. Table 4.5 shows the estimated costs for this additional work. The OM estimate assumes that the OM's have been prepared for IPAD and just need conversion to another host.

Table 4.5 Phase I IPAD Conversion Costs

	SI-6 CODE AND CERTIFY SYSTEM	OM CONVERSION	SI-8 SYSTEM/OM INTEGRATION	TOTAL
COMPUTING	$940,800	$30,000	$925,600	$1,896,400
LABOR	$983,000	$500,000	$983,700	$2,466,700
TOTAL	$1,923,800	$530,000	$1,909,300	$4,363,100

4.4.2 NASA (Langley) Usage

The strategy calls for research into design methodology to go on within NASA using IPAD. These activities are not specified nor have costs been estimated.

4.5 Operational Costs

4.5.1 Computing Costs

The IPAD operational costs are estimated on the basis of comparing what an analysis would cost in a non IPAD environment and in an IPAD environment using similar OM capability. The non IPAD environment was defined to be an analysis performed using individual programs in a batch computing mode. The three components of IPAD operational costs have been identified as:

(1) Integration - cost of IPAD executive and data manager.

(2) Time Sharing - cost of running in a time sharing mode as compared to a batch mode.

(3) Continuity - cost of facilities to provide task and time continuity in IPAD which is primarily the cost of a trillion bit memory device.

Component Cost Breakdown:

	Factor
• Integration ---------- This component includes a 1.05 cost factor for the executive and 1.25 for the data manager. This reflects the experience of CPDS and ATLAS systems at Boeing.	1.3
• Time Sharing ---------- It is assumed that a typical IPAD subtask will require 20% of its execution in a time sharing mode and 80% in a batch mode. It is further assumed that time sharing costs are a factor of 1.5 greater than batch. Thus the cost for time sharing in IPAD is: $(1.5 \times .2) + (1.0 \times .8) = 1.1$	1.1
• Continuity ---------- The fully burdened cost of the IPAD host machine is assumed to be \$150,000/month. The cost of a trillion bit IPAD data base storage device, fully dedicated to IPAD, is estimated at \$30,000/month. Assuming IPAD usage is one-half the host machine, the cost factor is then: $\frac{.5(150,000) + 30,000}{.5(150,000)} = 1.4$	1.4

The estimated total operational cost for an analysis performed with IPAD is the sum of these three extra costs. A factor of 1.8 times the cost of normal batch processing without IPAD.

4.5.2 Maintenance Costs

During the first year following the release of IPAD, system maintenance will require three people full time. Sustaining maintenance beyond this period will require an average of two man years per year. This does not include any maintenance of the OM code. In addition there will be clerical costs to support the release of modifications, update, manuals, etc.

4.5.3 User Support

User support for training, consultation, etc., is very dependent upon the number of users. In the early period of IPAD usage a staff of two engineers and one programmer would be needed to provide OM and system support.

5.0 REFERENCES

(1) "Interim Report from the IBM Data Security Study Sites", 1973 National Computer Conference and Exposition, June 4 - 8, 1973, New York. (Oral Report)

(2) LeRoy, P. L., "Computerized Preliminary Design System - Level I Development and Usage Reference Information", Document D6-42127TN, Boeing Commercial Airplane Company, May 29, 1973.

(3) Wallace, R. E., "A Computerized System for the Preliminary Design of Commercial Airplanes", AIAA Paper 72-793, August 1972.

(4) "FORTRAN Extended 3.0", Control Data Corporation Publication 60176600.

(5) "ATLAS An Integrated Structural Analysis and Design System Control and System Manual", Document D6-25400-2TN, Boeing Commercial Airplane Company, February 14, 1973.

(6) "Computer Based Integrated Design Systems", AFIPS Conference Proceedings Volume 42, Pages 319-363, 1973 National Computer Conference and Exposition, June 4 - 8, 1973, New York.

(7) "The NASTRAN Programmer's Manual", NASA SP-223 (01), September 1972.

(8) Control Data CYBER 70 Computer Systems SYMPL Reference Manual, 60328800, CDC Software Documentation, Sunnyvale, California, 94086

(9) Ross, D. T., "Introduction to Software Engineering with the AED-0 Language", Report ESL-R-405, M.I.T., October, 1969.

APPENDIX A

CONSIDERATIONS FOR THE MANAGEMENT OF IPAD SOFTWARE DEVELOPMENT

The purpose of this section is to list key items that should make up the IPAD software development management plan. The plan should contain the following ten items.

1) Overview

2) Phases

3) Task Organization and Structure

4) Tests

5) Configuration Controls

6) Documentation

7) Training

8) Reviews and Reports

9) Installation and Maintenance

10) Resources and Deliverables

Each of the above items are discussed below. References 1 and 2 provided the source for some of the key considerations presented.

A.1 OVERVIEW

Project objectives under the contract should be stated. Organization of the total plan should be clear. The assumptions and restrictions on which the plan is based should be listed. A gross schedule for the project should be present. A statement of the objectives of each of the remaining sections of the plan should be present.

A.2 PHASES

This plan element is a definition of the programming development effort in terms of a series of tasks that occur in sequence. Basic definitions should be established and the remaining sections of the plan should be tied to these definitions.

A.2.1 Definition Phase

- Primary Objectives
  - Problem analysis
  - Detailed project planning
- Secondary Objectives
  - Finding people
  - Understanding the customer
  - Forming tentative design ideas

A.2.2 Design Phase

- Primary Objectives
  - Baseline design for operational programs
  - Baseline design for support programs
- Secondary Objectives
  - Preparation for integration testing
  - Setting up configuration controls
  - Writing preliminary acceptance specifications
  - Constructing simulation models
  - Manning for subsequent phases
  - Preparation for programming training
  - Publication of programmer's handbook
  - Initial preparation for system test
  - Initial preparation for acceptance test
  - Setting up project libraries
  - Writing preliminary user documentation

A.2.3 Programming Phase

- Primary Objectives
  - Detailed design
  - Coding
  - Unit test
  - Integration test
  - Program documentation

- Secondary Objectives
  - Detailed preparation for system test
  - Detailed preparation for acceptance test
  - Writing final acceptance specification
  - Preparation for customer training

A.2.4 System Test Phase

- Primary Objectives
  - Execution and analysis of system tests
  - Testing against problem specification

- Secondary Objectives
  - Completion of acceptance test preparations
  - Customer training
  - Correction of descriptive documentation
  - Completion of user documentation
  - Manpower reassignment

A.2.5 Acceptance Phase

- Primary Objectives
  - Execution and analysis of acceptance tests
  - Signing of formal acceptance agreement

- Secondary Objectives
  - Completion of customer training
  - Cleanup of documentation

A.2.6 Installation and Maintenance Phase

- Primary Objectives
  - Assistance in installing system
  - Assistance in beginning operation

- Secondary Objectives
  - Testing on-site
  - Continuing maintenance and tuning
  - Continuing operation
  - Project evaluation

A.3 TASK ORGANIZATION AND STRUCTURE

The organization plan should define the organization of the project tasks and people and the assignment of responsibilities during various project phases. A sketch of the main flow of work within the organization, starting with problem analysis and design and proceeding through programming, integration, testing, documentation and delivery should be included. The detailed plan listed in section A.3.1 through A.3.5 represents the project during the programming phase.

A.3.1 Analysis and Design

- Writing Problem Specification
- Writing System Functional Specification
- Change Control of System Design after Design Phase
- Data Base Configuration Control
- Simulation Modeling
- Reviewing Detailed Design
- Writing User Documentation

A.3.2 Development

- Detailed Design
- Writing Unit Test Plan
- Coding
- Reading Other Programmer's Code
- Unit Testing
- Writing Program Documentation

A.3.3 Integration

- Receives New Code from Development
- Integrates New Code
- Performs Configuration Control
- Performs Integration Testing - Regression Tests
- Provides Scaffolding Development
- Enforces Standards
- Performs Technical Review
- Provides Progress Visibility

A.3.4 Test

- Writing System Validation Specifications
- Writing Acceptance Test Specifications
- Gathering and Generating Test Data

- Choosing and Obtaining Test Tools
- Setting Up Test Libraries
- Executing Tests
- Analyzing Test Results
- Documenting Test Results

A.3.5 Staff

- Computer Time Control
- Supplying Keypunch Services
- Planning and Installing Terminals
- Issuing Programmer's Handbook
- Training
- Special Technical Assignments
- Technical Liaison
- Document Control
- Report Control
- Contract Change Control
- Supplying Clerical Support
- Maintaining Project History

A.4 TESTS

Each performance requirement of the system software description should be verified. This verification should be made by test, demonstration or inspection.

Each performance requirement is to be verified using the software documentation as the first abstraction of the computer program.

Each system test should be completely reproducible by personnel not associated with the software development.

The testing process should be modular and certainty at one level should be established before proceeding to the next.

As an annex to each test plan, a test coverage matrix should be made, going from performance specification to specific subtest. The following defines the types of testing.

Unit Test
Testing done on the lowest level program module (a unit) before it is combined with other units to form higher level modules.

Integration Test
The process of combining tested units into progressively more complex groupings and testing these groupings until the entire program system has been put together and tested.

System Test
The retesting of the completed program system, in as nearly a live environment as possible, by personnel other than those who produced the programs.

Acceptance Test
The exercising of the program system under conditions agreed to by the customer in order to demonstrate to him that the system satisfies the customer's requirements.

Site Test
Testing of the program in its ultimate operational environment to assure readiness for operation. For each type of test, the objectives, responsibility, procedures and tools should be specified.

A.5 CONFIGURATION CONTROLS

The objective should define the procedures to be used for controlling change in the evolving and completed program systems. In order to provide visibility to both the contractor and NASA, certain baseline documents acceptable to both parties should be established and all events should be controlled relative to those baselines. Baseline documents serve as a reference point on questions raised. Changes whether revisions, additions or deletions should be negotiated relative to the baseline. A revised baseline document becomes the new baseline.

- Baselines
  - Problem specification
  - Design specification

- Proposing a Change
  - Who may propose a change
    - Project members
    - Customer
    - Other contractors
  - Change proposal document

- Investigating a Proposed Change
  - Who, how, when
  - The investigator's report
    - Summary of proposed change
    - Originator's name and organization
    - Classification of the change
    - Impact on costs, schedules, other programs
    - Recommendations

- Types of Changes
  - Type 1--the change affects a baseline or would cause a cost, schedule or other impact.
  - Type 2--the change affects no baseline and has negligible cost, schedule or other impact.

- Configuration Control Board
  - Membership
  - When it meets
  - How it operates

- Types of Recommendation
  - Acceptance
  - Rejection

- Customer Directed Changes

- Implementing a Change

- Estimating cost of change
- Approvals
  - Project management
  - Customer
- Documenting the change
- Testing the change

A.6 DOCUMENTATION

The documentation plan should define the procedures and resources required for the publication cycle and to outline a basic set of project documents.

- Publication Procedures
  - Preparation and approval
  - Typing
  - Proofing and editing
  - Reproduction
  - Distribution

- Project Documents
  - Problem specification
    - Scope--describes the requirements of the program system
    - Requirements--detailed description of program system
      - Performance
      - Data base requirement
      - Human factors

- Design Specification--defines a solution to the problem described in the problem specification. The design specification is the foundation for all program implementation. The design logic described here is detailed enough so that all required functions are satisfied, all interfaces are defined, system files are defined, and the logic connecting all program modules is defined. The design is done in sufficient detail that all logic problems are resolved and the complete program system "hangs together." The lowest level of program module is specified in terms of the functions it must perform and the interfaces it must have with other modules, but the actual internal design of these lowest level modules is left to the implementing programmers.

- Coding Specification

- Change Proposal

- Problem Specification Change Notice

- Design Specification Change Notice

- Test Specification

- Test Case

- Test Report
- Technical Note
- Administrative Note
- Programmer's Handbook
  - Introduction
    - Objective--the handbook is to be the source of basic technical information required by all programmers on the project.
    - Scope--handbook should be reasonably concise, useful and usable from a programmer's point of view.
    - Publication--near end of design phase.
  - Problem
    - Introduction--tutorial description of the customer, the environment, and the job to be done.
    - The problem specification--complete.
  - Testing--includes entire test plan
  - Support programs--describes programming tools available.
  - The design specification--includes overall design concept, design standards and conventions, coding standards and conventions, the baseline design.
  - Documentation
- Equipment
- Glossary
- Technical Status Report
- Project History
  - Scope
  - Significant events
  - Manpower history
  - Machine time history

A.7 TRAINING

The training plan defines the internal and external training required, the responsibility for each, and the resources required.

- Types of Training
  - Internal training
    - Understanding the overall project
    - Technical
      - Coding language
      - Use of test tools
      - Use of terminals
      - The data processing hardware
      - Interfacing with other subsystems
      - The problem
      - The baseline design
    - Non Technical
      - Management techniques
      - Configuration control procedures
      - Documentation control
      - Reporting requirements
      - Clerical procedures
  - External training
    - Installing the program system
    - Using the system
    - Modifying the system

- Resources
  - For each type of training identified, show:
    - training schedules
    - instructors required
    - training materials
    - facilities
    - number of trainees
    - special computer programs for training

C.8 REVIEWS AND REPORTS

This plan element addresses how project status should be communicated by oral project reviews and written reports.

- Reviews
  - Internal
    - Definition phase review--review the problem specification and determine readiness for the design phase.
    - Preliminary design review--when the baseline design has been completed, it should be reviewed to assure the validity of the design approach.
    - Critical design review--review the completed design specification to determine whether or not it satisfies the problem specification and is reasonable and programmable. Review project plan.
    - Programming phase review--review program integration results and determine readiness for the system test phase.
    - System test phase review--review system test results and determine readiness for the acceptance phase.
    - Post-mortem review--review and approve the project history document.

  External
  - Preliminary design review--review validity of design approach.
  - Critical design review--review in detail and concur on the design specification; review the contractor's project plan for entering the programming phase.
  - Acceptance review--review the results of the completed acceptance tests and determine any remaining problems that must be corrected before NASA Langley will formally accept the programs.

- Reports
  - By nonmanagers--biweekly, task oriented
  - Manager--biweekly, milestone oriented
  - Project manager--monthly and quarterly

A.9 INSTALLATION AND MAINTENANCE PLAN

This section should define the contractor's responsibility in installing, demonstrating and maintaining the accepted program system.

- Installation
  - Responsibility
  - Schedule
  - Multiple site considerations
- Maintenance
  - Change control procedures
  - Work location
  - Funding

A.10 RESOURCES AND DELIVERABLES

This plan element should gather in one place a summary of all resource estimates and a schedule for all deliverables.

- Manpower--total, by month and by category
- Computer Time--by month and by type
- Other Resources
  - Publication costs
  - Travel Costs
  - Relocation of employees and equipment
  - Equipment and supplies
  - Special purchases or rentals
- Delivery Schedules--dates for all deliverables
- Milestone Charts
- Budget

Observations on NASTRAN development management as reported by reference 3.

1) Design teams were not allowed time to produce functional specifications,
2) Users' views were obtained on user input,
3) Users' views should have been obtained on program diagnostics,
4) Planning for training was vital,
5) PERT was used in progress analysis,
6) Availability of computer time should have been scheduled,
7) The project's development and certification progress should have been displayed more clearly,
8) A tutorial manual should have been written,
9) A system support manual should have addressed system maintenance.

REFERENCES

1. Metzger, Philip W., "Managing a Programming Project", Prentice-Hall, 1973.

2. Smith, Don Jr., "An Organization for Successful Project Management", pp. 129-140, AFIPS SJCC, 1972.

3. Butler, T. G., "Considerations for the Design of a General Purpose Structural Analysis Program", ASME, Synthesis of Vibrating Systems Winter Annual Meeting, Washington, D. C., November 30, 1971, pp. 102-124.

APPENDIX B

PHASE 1 TECHNICAL PROGRAM ELEMENTS

The Technical Program Elements (TPE's) selected for Phase 1 IPAD implementation are presented in appendix B. The TPE's are listed by catalog number and title consistent with Volume V. The status of the TPE's is denoted by:

1 - operational

2 - in development

3 - not programmed

The OM group number identifies the TPE as providing technical capability for an airplane type and design level as follows:

OM Group	Airplane Type	Design Level	No. of TPEs
1	Subsonic	II	9
2	Subsonic	III	46
3	Supersonic	II	14
4	Supersonic	III (Geometry sizing only)	27
5	Subsonic and Supersonic	V & VI	3

There are sixty-five unique TPE's. Many of the TPE's provide technical capability for more than one airplane type and design level.

Catalog No.	Title	Status	OM Group
ARO-5	Analysis & Design of Wing-Body Combinations	1	4
ARO-6	Calculation of Slender Body Effects for A/C Matrix Formulation	1	4
ARO-7	Subsonic Cruise Drag Module for Transport Configurations	1	1,2,3,4
ARO-8	Low Speed Lift and Drag Module for Transport Configurations	1	1,2,3,4
ARO-9	Wave Drag and Supersonic Area Rule	1	3,4
ARO-10	Calculation of Supersonic Drag Due to Lift and Wing Nacelle Interference Drag	1	4
ARO-11	Supersonic Drag and Pressure Distribution on Bodies of Revolution	1	4
ARO-12	Supersonic Skin Friction Prediction	1	3,4
ARO-13	Influence of Non-smooth Geometries on Sonic Boom	1	4
ARO-14	Propagation Characteristics of Sonic Booms in Non-homogeneous Atmosphere	1	4
ARO-17	Calculation of Lift and Induced Drag	1	3
ARO-21	Parmetric Estimate of Supersonic Drag of Complete Configurations	3	3
DCA-1	Airplane Geometry Control	1	2,4
DCA-2	Airplane Geometry Parameters	3	1,3
DCA-3	Computerized Space Arrangement Mockup	3	2,4

Catalog No.	Title	Status	OM Group
DCA-4	Level III Configuration Sizing Driver	3	2,4
DGL-1	Airplane Exterior Geometry Loft (SHAPE)	1	2,4
DGL-9	Perspective Projections of 3D Data (PERSPE)	1	2
DSA-1	Wing Structural Arrangement Definition	3	2
DSA-2	Body Structural Arrangement Definition	3	2
DSA-3	Empennage Structural Arrangment Definition	3	2
DSA-4	Landing Gear Structural Arrangement Definition	3	2
DSA-5	Interactive Design - Structural	3	5
DSA-6	Frame Design Program	3	5
DSA-7	Floor Beam Design Program	3	5
MIS-1	Configuration Management	3	2
PNZ-1	Noise Prediction	1	2,4
PRF-1	Airplane Performance Calculations for Market Analysis	3	1,3
PRF-2	Calculation of Flight Performance	1	2,4
PRF-3	Takeoff and Climbout Performance	2	2,4
PRF-4	Landing Performance	2	2,4
PRF-5	Performance Summary	3	2
PRO-1	Nacelle Design	1	2
PRO-2	Nacelle Design	3	4

Catalog No.	Title	Status	OM Group
PRO-3	Engine Performance (cycle matching)	1	1,2,3,4
PRO-4	Engine Performance (cycle matching) GSA	1	2,4
PRO-5	Engine Performance (table lookup)	1	1,2,3,4
PRO-6	Engine Installation	3	1,2,3,4
S&C-1	Preliminary Airplane Balance and Tail Sizing Program	1&2	2
S&C-2	Maneuver Margin Increment Due to SAS	3	2
S&C-20	SST Preliminary Airplane Balance, Tail Sizing, Gear Location and Lateral Control Check	3	4
SFL-10	Flutter Matrix Formulation and Solution	1	2
SLO-1	Rigid Wing Aerodynamics	1	2
SLO-2	Aeroelastic Wing Loads Distribution	1	2
SLO-3	Body and Empennage Loads Distribution	1	2
STM-1	System Requirements Analysis	3	2,4
STR-1	Preliminary Wing Gross Stress Analysis and Sizing	1	2
STR-2	Preliminary Body and Empennage Stress Analysis and Sizing	1	2
STR-5	Fatigue Analysis and Design	2	2
WTS-1	Class 0 Weight Estimation	3	1,3
WTS-2	Level I Weight and Balance System (type A weights)	1	1,2,3,4

Catalog No.	Title	Status	OM Group
WTS-3	Wing Geometry and Dead Weight Generating and Distribution (GEMPAK)	1	2
WTS-4	Body/Empennage Design System	1	2
WTS-5	Wing Primary Structure (type B weights)	1	2
WTS-6	Body/Empennage Primary Structure (type B weights)	1	2
WTS-7	Wing-Secondary Structure (type B weights)	1	2
WTS-8	Body/Empennage Structure (type B & C weights)	1	2
WTS-9	Landing Gear (type D weights)	1	2
WTS-10	Propulsion and Fixed Equipment	1	2
WTS-11	Fuel Distribution	1	2
WTS-12	Mass Properties	1	2
WTS-13	Weight Statement	1	2
WTS-14	Airplane Mass Distribution	1	2
WTS-15	Weights Update Control	3	2
WTS-24	Parametric/Statistic Weight Estimating (type A weights)	2	3,4

APPENDIX C

SYSTEM IMPLEMENTATION LANGUAGE SELECTION

C.1 GENERAL REQUIREMENTS

The IPAD system implementation language should be general enough to satisfy the requirements of the various IPAD subsystems. The considerations delineated below begin with those which influence the execution of the IPAD system, and end with those which influence the writing and designing of the IPAD subsystems.

C.1.1 Target Code Efficiency

The target code of a systems language should make full use of all the capabilities of the machine. A serious attempt should be made to have every statement produce optimal code for the particular machine in the environment in which it is used. The User's Guide for the language should list the estimated storage requirements and execution overhead of all language constructs.

C.1.2 Run-Time Environment

A run-time library should be provided to meet the common run-time needs of the IPAD subsystems. The user should be informed of the approximate cost, in execution time and core, of each routine.

C.1.3 Error Checking

A comprehensive set of compile-time checks should be provided. Optional implicit run-time error checking should be available such as array bound checking, etc.

C.1.4 Debugging Facilities

The debugging facilities should be flexible enough to provide selective dumps of blocks of code and/or core in a static or dynamic mode.

C.1.5 Syntax Considerations

After the provision of efficiency and cost information, an important consideration for a system language is that it be as readable as possible. The instructions themselves should read as easily as if they were part of the documentation; comments within the program should not be the sole method for explaining the algorithm or logical flow, but rather they should enhance the meaning of a set of instructions.

C.1.6 Adaptability

To meet changing demands during the life of the project, the system language compiler ought to support definitions of new types of operations and operands. A good, general purpose macroprocessor can aid considerably in making the language adaptable to the user.

C.1.7 Program Modularity

The system implementation language must provide facilities for linking the object code modules it produces to each other, and to those produced by FORTRAN, assembly language, or any other language that may be used.

C.1.8 Machine Dependence

The degree of machine dependence of the systems language should be limited to those features of the host machine that are required to meet the system performance objectives.

C.1.9 Language Support and Availability

The system implementation language should be supported and made available by an organization which under contract or as part of product support will provide the necessary support to the language and its compiler. This support will permit the language's effective use over the life of the IPAD development.

C.1.10 Long Term Potential

The system implementation language should reflect the current state of the art in language development and have design flexibility which would allow new features to be added conveniently so that the language will be attractive for use over the life of IPAD.

C.2 SPECIFIC REQUIREMENTS

The previous section delineated the global requirements of the IPAD system implementation language. It considered the overall criteria of such a language both at a source and at the target levels. This section lists the specific language facilities required for implementation of an IPAD system which will function nearly as efficiently as if hand-coded in assembler language.

C.2.1 Data Facilities

Fixed point variables are needed to serve as counters, to contain quantities such as the current length of a varying length character string, or to be used as pointers to reference other data items. All basic scalar arithmetic operations must be available for manipulating these quantities. There is need for floating point variables and arithmetic operations as well. Both bit and character data types are required: the first might contain flags and masks; the second, parsed data items and command sequences.

C.2.2 Storage Allocation

In order to perform data manipulation efficiently, the system implementation language should provide various methods of storage allocation on two basic levels: program (static) and statement (explicit dynamic). Multiply dimensioned arrays with cell size specification facility and storage mode specification facility should be available in the program (static) storage schemes.

The other mode of storage allocation (explicit dynamic) is required for the statement level and is directly controlled by the user. That is, the user issues a command to allocate a variable and at some time later, he issues an explicit command to free it again. During the period of accessibility, the variable is "based on" (referenced by) a pointer (whose value is an address) that has previously been allocated in the program (on the program or statement level). An implementation of this feature would be through based array and a pointer variable.

C.2.3 Data Manipulation

String manipulation facilities must be provided as part of the language syntax to manage bit and character data. Typical operations are pattern matching, accessing a substring,

concatenating two (or more) strings, inserting, deleting, substituting, rearranging, etc.

C.2.4 Program Control and Segmentation

Flow of control must be maintained within individual modules, as well as between separate components of the system. A clear program logic, which is probably the most crucial aspect of systems design, can be maintained only through the use of iterative loops and conditional and unconditional jumps. Constructs that should be available for this purpose are IF ... THEN ... ELSE, WHILE, FOR, and CASE statements. Facilities for overlays and run-time linking should be available.

C.2.5 I/O Facilities

Stream I/O should be available for unit record devices. Record I/O should be available for direct access as well as sequential files. Means should be provided for the buffering of data to and from files.

C.2.6 Debugging Facilities

Simple I/O should be provided for the user so that he may debug with a minimum of excess work. Facilities for printing out the contents of small areas of memory or for dumping values of given variables should be included for this purpose. Such facilities should be provided on demand (that is, the user indicates the point at which he wants the contents of the variable printed), or universally (that is, the user states that whenever a certain identifier is used, the value is to be dumped).

To be compatible with today's interactive environment, debugging aids should be provided for debugging on-line as well as off-line.

C.3 LANGUAGE EVALUATION

In considering which language to use for IPAD system implementation, there are two inextricably intertwined issues - management and technical. To concentrate on one to the exclusion of the other is short-sighted, particularly because they are not truly independent. The language selection decision must be based on meeting both the technical and management requisites. There are a spectrum of systems languages. On the specialized end of the spectrum, there are languages (such as PL/360, ref. 1) which are designed for a specific machine. At the generalized extreme there are languages such as AED-0 (ref. 2) which are designed for maximum portability. In the middle and toward the generalized direction are languages such as CDC's SYMPL (ref. 3), BLISS (ref. 4) and others (ref. 5).

C.3.1 Evaluation Criteria

The evaluation criteria used here is divided into two parts: qualification criteria and decision criteria. The qualification criteria is the minimum set of capabilities that the system implementation language should have. These were elaborated in section C.2 The decision criteria is a set of qualities that the choice of the IPAD system implementation should be based on, as discussed in section C.1.

C.3.1.1 Qualification Criteria

Item	Points
Data Facilities	20
Storage Allocation	14
Data Manipulation	16
Program Control and Segmentation	25
I/O Facilities	10
Debugging Facilities	15
	100

The point assignment for each of the above items was done on the basis of how important each item will be in the IPAD implementation. Key factors affecting assignment are: processing of command sequences, support to structured programming, support of IPAD system directories and

dictionaries, and IPAD data manager requirements. A total of 100 points was divided among the six items.

Each language was rated in each of the above categories by estimating how much of the qualifying item the language contains as defined in sections C.2.1 through C.2.6. For example, if the language has been estimated to have 80% of the data facilities as defined in section C.2.1, it would be awarded 16 points.

C.3.1.2 Decision Criteria

Item	Weight
Target Code Efficiency	.9
Run-Time Environment	1.0
Error Checking	1.0
Debugging Facilities	1.1
Syntax Considerations	1.2
Adaptability	.8
Program Modularity	1.0
Machine Dependence and Portability	1.1
Language Support and Availability	1.0
Long Term Potential	.9

The weights were assigned on the basis of those items affecting implementation being highest and those items affecting efficiency being lowest.

Each item above was rated on a scale of 1-10. The scores were assigned on a relative basis, among languages reviewed, with the following interpretation:

1-2 Inferior Capability

3-4 Below Average Capability

5-6 Average Capability

7-8 Above Average Capability

9-10 Superior Capability

The final score for each item is obtained by multiplying by the weight. The decision criteria score is obtained by summing the above item weighted scores. The final decision criteria score is obtained by multiplying the decision criteria score by the sum of qualification criteria percentages. The maximum possible score is 100.

C.3.2 Evaluation

The above criteria and procedures are applied to the following languages: AED-0, BCPL, BLISS, CDC FORTRAN, PASCAL, PL/I and SYMPL, (refs. 2, 6, 4, 7, 8, 9, 3). Below is the summary of results given in order of decreasing final score. The sections C.3.2.1 through C.3.2.7 give the detailed scores and intermediate totals.

Language	Qualification Total	Decision Total	Final Score
SYMPL	88	83.9	74
AED-0	89	68.6	61
PL/I (IBMF)	92	57.4	53
BCPL/6600	77	60.5	47
PASCAL	78	57.9	45
BLISS	74	58.2	43
CDC FORTRAN EXTENDED	58	63.0	37

C.3.2.1 Language - SYMPL

Qualification Criteria

Item	Score
Data Facilities	20
Storage Allocation	14
Data Manipulation	16
Program Control and Segmentation	23
I/O Facilities	5
Debugging Facilities	10
Qualification Total	88

Decision Criteria

Item	Raw Score	Weight	Score
Target Code Efficiency	9	0.9	8.1
Run-Time Environment	7	1.0	7.0
Error Checking	7	1.0	7.0
Debugging Facilities	8	1.1	8.8
Syntax Considerations	10	1.2	12.0
Adaptability	9	0.8	7.2
Program Modularity	9	1.0	9.0
Machine Dependence and Portability	7	1.0	7.7
Language Support and Availability	9	1.0	9.0
Long Term Potential	9	0.9	8.1

Decision Total 83.9

Final Score .88 x 83.9 = 74

C.3.2.2 Language - AED-0

Qualification Criteria

Item	Score
Data Facilities	18
Storage Allocation	14
Data Manipulation	16
Program Control and Segmentation	23
I/O Facilities	6
Debugging Facilities	12
Qualification Total	89

Decision Criteria

Item	Raw Score	Weight	Score
Target Code Efficiency	5	0.9	4.5
Run-Time Environment	7	1.0	7.0
Error Checking	6	1.0	6.0
Debugging Facilities	7	1.1	7.7
Syntax Considerations	6	1.2	7.2
Adaptability	6	0.8	4.8
Program Modularity	8	1.0	8.0
Machine Dependence and Portability	10	1.1	11.0
Language Support and Availability	7	1.0	7.0
Long Term Potential	6	0.9	5.4

Decision Total 68.6

Final Score = .89 x 68.6 = 61

C.3.2.3 Language - PL/I (IBM F)

Qualification Criteria

Item	Score
Data Facilities	20
Storage Allocation	14
Data Manipulation	16
Program Control and Segmentation	21
I/O Facilities	8
Debugging Facilities	13
Qualification Total	92

Decision Criteria

Item	Raw Score	Weight	Score
Target Code Efficiency	4	0.9	3.6
Run-Time Environment	6	1.0	6.0
Error Checking	6	1.0	6.0
Debugging Facilities	8	1.1	8.8
Syntax Considerations	5	1.2	6.0
Adaptability	4	0.8	3.2
Program Modularity	8	1.0	8.0
Machine Dependence and Portability	5	1.1	5.5
Language Support and Availability	4	1.0	4.0
Long Term Potential	7	0.9	6.3

Decision Total 57.4

Final Score = .92 x 57.4 = 53

C.3.2.4 Language - BCPL/6600

Qualification Criteria

Item	Score
Data Facilities	20
Storage Allocation	10
Data Manipulation	14
Program Control and Segmentation	22
I/O Facilities	6
Debugging Facilities	5
Qualification Total	77

Decision Criteria

Item	Raw Score	Weight	Score
Target Code Efficiency	6	0.9	5.4
Run-Time Environment	4	1.0	4.0
Error Checking	4	1.0	4.0
Debugging Facilities	5	1.1	5.5
Syntax Considerations	9	1.2	10.8
Adaptability	8	0.8	6.4
Program Modularity	8	1.0	8.0
Machine Dependence and Portability	9	1.1	9.9
Language Support and Availability	2	1.0	2.0
Long Term Potential	5	0.9	4.5

Decision Total 60.5

Final Score = .77 x 60.5 = 47

C.3.2.5 Language - PASCAL

Qualification Criteria

Item	Score
Data Facilities	18
Storage Allocation	10
Data Manipulation	12
Program Control and Segmentation	25
I/O Facilities	6
Debugging Facilities	7
Qualification Total	78

Decision Criteria

Item	Raw Score	Weight	Score
Target Code Efficiency	5	0.9	4.5
Run-Time Environment	4	1.0	4.0
Error Checking	4	1.0	4.0
Debugging Facilities	5	1.1	5.5
Syntax Considerations	10	1.2	12.0
Adaptability	7	0.8	5.6
Program Modularity	5	1.0	5.0
Machine Dependence and Portability	9	1.1	9.9
Language Support and Availability	2	1.0	2.0
Long Term Potential	6	0.9	5.4

Decision Total 57.9

Final Score = .78 x 57.9 = 45

C.3.2.6 Language - BLISS

Qualification Criteria

Item	Score
Data Facilities	16
Storage Allocation	8
Data Manipulation	12
Program Control and Segmentation	25
I/O Facilities	6
Debugging Facilities	7
Qualification Total	74

Decision Criteria

Item	Raw Score	Weight	Score
Target Code Efficiency	5	0.9	4.5
Run-Time Environment	3	1.0	3.0
Error Checking	4	1.0	4.0
Debugging Facilities	5	1.1	5.5
Syntax Considerations	9	1.2	10.6
Adaptability	8	0.8	6.4
Program Modularity	8	1.0	8.0
Machine Dependence and Portability	8	1.1	8.8
Language Support and Availability	2	1.0	2.0
Long Term Potential	6	0.9	5.4

Decision Total 58.2

Final Score = .74 x 58.2 = 43

C.3.2.7 Language - CDC FORTRAN Extended

Qualification Criteria

Item	Score
Data Facilities	10
Storage Allocation	7
Data Manipulation	8
Program Control and Segmentation	15
I/O Facilities	10
Debugging Facilities	8
Qualification Total	58

Decision Criteria

Item	Raw Score	Weight	Score
Target Code Efficiency	8	0.9	7.2
Run-Time Environment	7	1.0	7.0
Error Checking	7	1.0	7.0
Debugging Facilities	8	1.1	8.8
Syntax Considerations	4	1.2	4.8
Adaptability	1	0.8	0.8
Program Modularity	7	1.0	7.7
Machine Dependence and Portability	4	1.1	4.4
Language Support and Availability	9	1.0	9.0
Long Term Potential	7	0.9	6.3

Decision Total 63.0

Final Score = .58 x 63.0 = 37

C.4 DISCUSSION OF EVALUATION

The science of evaluating computer programming languages is presently in an elementary stage (ref. 10). This fact along with a desire to provide some basis for a language choice, the foregoing decision process was devised. It is recognized that the process is subjective. The first three highest ranking languages will be discussed as to critical points in the evaluation process.

SYMPL

1) Extensive use on the host system and hardware anticipated for initial implementation (CDC 6600),

2) No CASE construct in language syntax,

3) Some of the I/O facilities would have to be done by a run-time library,

4) Compiler for other host machines would have to be implemented.

AED-O

1) Small host system user base,

2) No CASE construct in language syntax,

3) Lack of global debugging facilities,

4) Limited array facilities,

5) Awkward language syntax,

6) Awkward method for based array,

7) Limited I/O facilities,

8) Too many syntax facilities buried in run-time libraries,

9) Lack of visibility on cost of language constructs and facilities,

10) User documentation is less than adequate.

PL/I

The evaluation of this language was complicated by the fact that it is not presently readily available on the host system. The evaluation was based in part on implementation on other vendor's equipment. Most PL/I implementations appear to have

been characterized by excellent design specifications but their actual performance has been disappointing. The present non-availablity of PL/I on the host system eliminates it from serious consideration.

REFERENCES

1. Wirth, N., "PL360, A Programming Language for the 360 Computers", JACM, Vol. 15, No. 1 (Jan. 1968).

2. Ross, D. T., "Introduction to Software Engineering with the AED-0 Language", Report ESL-R-405, M.I.T.

3. Control Data CYBER 70 Computer Systems SYMPL Reference Manual, 60328800, CDC, Software Documentation. Sunnyvale, California 94086.

4. Wulf, W. A. et al, Bliss Reference Manual, Carnegie-Mellon University, Department of Computer Science, Pittsburg, Pa. 15213 (Jan. 1970).

5. Sammet, J. E., "Brief Survey of Languages Used in Systems Implementation", SIGPLAN NOTICES Vol. 6, #9 (Oct. 1971).

6. Richards, M. "BCPL", Proc. SJCC, Vol. 34 (May 1969).

7. Control Data CYBER 70 Computer Systems FORTRAN Extended Reference Manual, 60176600, CDC, Software Documentation. Sunnyvale, California 94086.

8. Wirth, N., "The Programming Language PASCAL", Acta Information, Spring-Vorlag, Vol. 1 (1971).

9. IBM System/360 Operating System PL/I(F) Language Reference Manual, IBM, Data Processing Division, C28-8201, White Plains, N.Y. 10601.

10. Sammet, J. E., "Problems In, and a Pragmatic Approach to Programming Language Measurement", FJCC Vol. 39 (Nov. 1971).

APPENDIX D

DETAILS OF IPAD DEVELOPMENT COSTS AND SCHEDULE ESTIMATES

D.1 SUMMARY TABLES

Individual tasks and their asscciated costs in man months and computer resources are given in Tables D.1, D.2, D.3, D.4. These costs should be interpreted using the following:

- The total estimates for individual tasks include technical work plus supervision and miscellaneous extra costs.
- Supervision resources are estimated at 10% of technical.
- The miscellaneous category covers such items as
  - secretaries
  - documentation
  - training
  - reporting
- The portion of the total cost estimate due to supervision and miscellaneous costs is given in parenthesis following each estimate.
- Costs for computing time are estimated on current commercial rates from vendors of the CDC 6600.
- Dedicated machine costs are taken from a vendor's rate book for a 1 year lease.

Following these tables each individual task is described in more detail. The identifiers associated with each task have the following meaning:

TT-i - Technology Transfer

OM-i - Preparation of OMs

SD-i - System Design

SI-i - System Implementation

Table D.1 Summary Table for Technology Transfer

Activity Designator	Description	Start Month	Months flow time	Man-Months		Computing Resources
				Prog.	Eng.	
TT-1	Draft work statement descriptions	13	9	5	11	None
TT-2	Survey potential bidders	22	3	2	4	None
TT-3	Formulate and issue RFPs	25	12	2	7	None
	TOTALS		24 months duration	9	22	None

Table D.2 Summary Table for Preparation of OM's

Activity Designator	Description	Start Month	Months flow time	Man-Months		Computing Resources
				Prog.	Eng.	
OM-1	Establish OM selection criteria	1	1	-	2	None
OM-2	Prepare OM language specifications	2	1	1	-	None
OM-3	Specify OM conversion aids	2	2	5	-	None
OM-4	Collect all Phase 1 OMs	2	3	-	7	None
OM-5	Prepare OM language	3	3	4	-	$600 computing time
OM-6	Prepare OM conversion aids	5	6	23	-	$4,000 computing time
OM-7	Prepare IPAD data bank	11	6	-	15	$1,000 computing time
OM-8	Specify OM certification tests	11	6	8	15	None
OM-9	Prepare Phase 1 OMs	11	26	247	259	$200,000 computing time
OM-10	Prepare system checkout OMs	11	2	2	5	$2,000 computing time
	TOTALS		38 months duration	290	303	$207,600 computing time

Table D.3 Summary Table for System Design

Activity Designator	Description	Start Month	Months flow time	Man-Month		Computing Resources
				Prog.	Eng.	
SD-1	Establish design procedures and standards	1	1	3	-	None
SD-2	Refine Level 1 design	2	1	3	3	None
SD-3	Establish initial coding, certification standards	3	1	3	-	None
SD-4	Complete coding, certification standards	4	1	3	-	None
SD 5	Specify host operating system modifications	3	2	3	-	None
SD-6	Specify maintenance and development aids	3	2	5	-	None
SD-7	Produce system language specifications	1	3	4	-	None
SD-8	Prepare the system language	4	5	12	-	$2000 computing time
SD-9	Develop total system design	3	18	173	17	$2000 computing time
	TOTALS		20 months duration	209	20	$4000 computing time

Table D.4 Summary Table for System Implementation

Activity Designator	Description	Start Month	Months flow time	Man-Months		Computing Resources
				Prog.	Eng.	
SI-1	Define Phase 1 system capability	21	1	4	4	None
SI-2	Produce development and maintenance aids	9	8	32	-	$6,000 computing time
SI-3	Perform basic operating system modifications	5	9	36	-	$3,000 computing time 1050 hours block time
SI-4	Complete all operating system modifications	14	3	12	-	$1000 computing time 156 hours block time
SI-5	Establish user training methods	22	2	4	4	$400 computing time
SI-6	Code and certify the system	22	16	196	-	Dedicated machine for $941,472
SI-7	Train users	36	2	1	1	$1,000 computing time
SI-8	System–OM Integration	38	13	92	92	Dedicated machine for $926,029
SI-9	Establish plan for the IPAD computing facilities	1	1	1	-	None
SI-10	Prepare computing facilities	2	36	5	-	None
	Phase 1 totals		Critical Path–50	383	101	$11,400 computing time 1206 hours block time $1,867,501 dedicated machine **
SI-11	Deliver IPAD Phase 2	51	30	160	50	$1,000,000 computing time
SI-12	Deliver IPAD Phase 3	81	24	120	40	$780,000 computing time

* A dedicated computer for SI-3 and SI-4 activities would cost $500,352 for a 9 months period and would replace the block and computing time estimates.

** If a dedicated computer is used for SI-2 and SI-4, the total dedicated machine cost would be $2,367,853.

ACTIVITY IDENTIFICATION TT-1

Activity Description

Draft a work statement for future Request for Proposals (RFP) for the transfer of the IPAD technology from a development environment to industry use.

Results Expected

Preliminary work statements for technology transfer RFP's which indicate the type/depth of case study design problems, and the technical/economic factors to be evaluated by the contractors as a consequence of the use of IPAD.

Cost and Flowtime Estimates

Labor (man-months)
5 programming (1)*
11 engineering (2)*

Flowtime (months)
9

Computing Resources
None

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

ACTIVITY IDENTIFICATION TT-2

Activity Description

Survey potential bidders to test the validity of the Phase 1 technology transfer plans.

Results Expected

Evaluation of the potential bidders survey and personal bidder contacts so that the intended work statements can be finalized.

Cost and Flowtime Estimates

Labor (man-months)
2 programming (1)*
4 engineering (1)*

Flowtime (months)
3

Computing Resources
None

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

ACTIVITY IDENTIFICATION TT-3

Activity Description

Formulate and issue Request for Proposals for the technology transfer.

Results Expected

RFP's issued to industry. Timing of the contractor's efforts and expected results to be coordinated with Phase 2 plans.

Cost and Flowtime Estimates

Labor (man-months)
2 programming (1)*
7 engineering (1)*

Flowtime (months)
12

Computing Resources
None

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

ACTIVITY IDENTIFICATION OM-1

Activity Description

Establish constraints and standards for the selection of OMs for Phase 1 capability.

Results Expected

Constraints and standards for engineering and computing requirements; for example:

Engineering	Computing
disciplines	language
engineering theory	run time
modeling techniques	documentation
mathematical methods	machine dependent features

Cost and Flowtime Estimates

Labor (man-months)
2 engineering

Flowtime (months)
1

Computing Resources
None

ACTIVITY IDENTIFICATION OM-2

Activity Description

Prepare the OM language specifications - this will depend upon the activity of OM-1 and will consist of collecting existing information.

Results Expected

Formal language specification for the OM language or languages to be used during Phase 1 development.

Cost and Flowtime Estimates

Labor (man-months)
1 programming

Flowtime (months)
1

Computing Resources
None

ACTIVITY INFORMATION OM-3

Activity Description

Specify the OM conversion aids - that software and/or reference material to be used during preparation of the OMs.

Results Expected

Specifications of all software to be developed; identification of all reference material available and useful.

Cost and Flowtime Estimates

Labor (man-months)
5 programming (1)*

Flowtime (months)
2

Computing Resources
None

Additional Scheduling Information or Cost Information

This activity will overlap the activity of OM selection, (OM-4) but some information from that activity will be necessary.

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

ACTIVITY IDENTIFICATION OM-4

Activity Description

Selection and collection of all Phase 1 OMs.

Results Expected

Source code for all OMs intended to be part of Phase 1 IPAD. In addition, any available documentation and checkout aids should be obtained.

Cost and Flowtime Estimates

Labor (man-months)
7 engineering (1)*

Flowtime (months)
3

Computing Resources
None

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

ACTIVITY IDENTIFICATION OM-5

Activity Description

Prepare the OM language.

Results Expected

The OM language(s) should be operational and performing to specifications on the host system.

Cost and Flowtime Estimates

Labor (man-months)
4 programming (1)*

Flowtime (months)
3

Computing Resources
$600 computing time

Additional Scheduling Information or Cost Information

Coordination required between this activity and preparation of conversion aids activity (OM-6).

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

ACTIVITY IDENTIFICATION OM-6

Activity Description

Preparation of conversion aids - this will require gathering of information about the types of conversions which will be done and may involve writing some translators or code scanners.

Results Expected

Conversion aids meeting the specifications developed in OM-3. Any software developed will be operational on the host system at this point.

Cost and Flowtime Estimates

Labor (man-months)
23 programming (5)*

Flowtime (months)
6

Computing Resources
$4000 computing time

Additional Scheduling Information or Cost Information

This activity cannot start until SD-4 is completed which provides system coding standards.

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

ACTIVITY IDENTIFICATION OM-8

Activity Description

Specify the OM certification tests.

Results Expected

Specifications of the tests to be performed and the results to be obtained in order to declare the OM as certified.

Cost and Flowtime Estimates

Labor (man-months)
8 programming (2)*
15 engineering (3)*

Flowtime (months)
6

Computing Resources
None

Additional Scheduling Information or Cost Information
This activity must be coordinated with the OM preparation activity (OM-9).

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

ACTIVITY INFORMATION OM-7

Activity Description

Prepare the data bank for IPAD Phase 1 - gather all data required and put it into machine format. Requirements will be developed following OM selection.

Results Expected

All data required for the Phase 1 data bank, in the host system representation, ready for IPAD.

Cost and Flowtime Estimates

Labor (man-months)
15 engineering (3)*

Flowtime (months)
6

Computing Resources
$1000 computing time

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

ACTIVITY IDENTIFICATION OM-9

Activity Description

Prepare the selected Phase 1 OMs.

Results Expected

All OMs operational in the host system and performing well enough to make it feasible to begin certification. All information necessary to incorporate the OMs into IPAD should be available.

Cost and Flowtime Estimates

Labor (man-months)
247 programming (50)*
259 engineering (54)*

Flowtime (months)
26 months (5)*

Computing Resources
$200,000 computing time

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

Additional Scheduling Information or Cost Information

The information for incorporating OM's into IPAD is dependent upon the specifications for code and OM incorporation coming from the system design activity (SD-9).

The following information is used in estimating the labor for preparation of Phase 1 OMs.

- The Technical Program Elements (TPE) listed in Appendix B are divided into two categories: conversion only (status 1) and development (status 2 and 3).
- Source card estimates from Volume V are used to estimate TPE size.
- Development labor is estimated at 800 source statements per programmer man month.
- Conversion labor is assumed to be 1/10 of development labor.
- During development and conversion 3/4 of an engineering man month is estimated to be required to support a programmer man month.
- Engineering specifications for development are assumed to take 50% of the engineering support during development.
- Preparation for interfacing in IPAD is assumed to take 10% of the cost to develop.
- There are 108,000 cards to be developed and 294,000 cards to be converted.

The labor figures are:

301	mm for program preparation
51	mm for engineering specifications
50	mm for IPAD preparation
402	mm Total

The computing costs are estimated at $200,000 ($500 per man month).

ACTIVITY IDENTIFICATION OM-10

Activity Description

Prepare a set of OMs to aid system checkout. This activity is intended to solve problems associated with OMs to use for system checkout.

Results Expected

Operational OMs in the host system and the necessary information to insert them into IPAD. This will be a subset of the Phase 1 OMs, amounting to 5-20% of the total OM set.

Cost and Flowtime Estimates

Labor (man-months)
2 programming
5 engineering (1)*

Flowtime (months)
2

Computing Resources
$2000 computing time

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

ACTIVITY IDENTIFICATION SD-1

Activity Description

Establish the design procedures and standards.

Results Expected

Detailed specifications of how the system design will be approached and how it will be documented. This will produce a formal description of the design tree, state diagrams, etc.

Cost and Flowtime Estimates

Labor (man-months)
3 programming (1)*

Flowtime (months)
1

Computing Resources
None

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

ACTIVITY IDENTIFICATION SD-2

Activity Description

Refine the level 1 design specifications given in Volume IV, section 6.2.

Results Expected

A complete description of the system at level 1, consistent with the standards set up in activity SD-1.

Cost and Flowtime Estimates

Labor (man-months)
3 programming (1)*
3 engineering (1)*

Flowtime (months)
1

Computing Resources
None

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

ACTIVITY IDENTIFICATION SD-3

Activity Description

Establish the initial coding and certifying standards and procedures.

Results Expected

Standards and procedures for coding and certifying practices. This will not include any detail about the system language since the specifications will not have been available.

Cost and Flowtime Estimates

Labor (man-months)
3 programming (1)*

Flowtime (months)
1

Computing Resources
None

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc).

ACTIVITY IDENTIFICATION SD-4

Activity Description

Complete the coding and certifying standards and procedures.

Results Expected

All standards and procedures governing any coding and certifying work in the IPAD system.

Cost and Flowtime Estimates

Labor (man-months)
3 programming (1)*

Flowtime (months)
1

Computing Resources
None

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

ACTIVITY IDENTIFICATION SD-5

Specify the host operating system modifications necessary to support Phase 1 IPAD.

Results Expected

Detailed external specifications of the operating system modifications required by the IPAD system design.

Cost and Flowtime Estimates

Labor (man-months)
3 programming (1)*

Flowtime (months)
2

Computing Resources
None

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.)

ACTIVITY IDENTIFICATION SD-6

Activity Description

Specify the development and maintenance aids. This activity should include a thorough search for aids used in the development of sizeable software.

Results Expected

Specifications of all aids to be used during development and for maintenance of the system. These aids may or may not exist, so develcpment may not be necessary.

Cost and Flowtime Estimates

Labor (man-months)
5 programming (1)*

Flowtime (months)
2

Computing Resources
None

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

ACTIVITY IDENTIFICATION SD-7

Activity Description

Produce the IPAD system language specifications.

Results Expected

Collect complete specifications for SYMPL and any available utility packages and organize it for IPAD use.

Cost and Flowtime Estimate

Labor (man-months)
4 programming (1)*

Flowtime (months)
3

Computing Resources
None

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

ACTIVITY IDENTIFICATION SD-8

Activity Description

Prepare the system language for development use.

Results Expected

The language and its specified capabilities should be operational on the host system. Training manuals should be available, along with plans for training.

Cost and Flowtime Estimates

Labor (man-months)
12 programming (2)*

Flowtime (months)
5

Computing Resources
$2000 computing time

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

ACTIVITY IDENTIFICATION SD-9

Activity Description

Develop the total system design to support the level 1 specifications. The design will be taken down to the level of code in the IPAD system language for all nodes. Concurrent with the design effort will be the development of a set of data management specifications to support a search for data management software.

Results Expected

The design tree, state descriptions, and state diagrams for all nodes, plus any other information specified in the design standards activity SD-1. Data management software will be found, in part or whole, or the search will specify why the required software does not exist.

Cost and Flowtime Estimates

Labor (man-months)
173 programming (60)* (10 people max.)
17 engineering (3)*

Flowtime (months)
18 months (4)*

Computing Resources
$2000 computing time

Additional Scheduling Information or Cost Information

See the following two pages. Figure D.1 presents a PERT chart of the design refinement activities. The tables on the page following the figure give the detail labor estimates, in man-weeks, for refining the design of the system utilities and executive. Engineering support is estimated at 1 man full time during the activity.

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

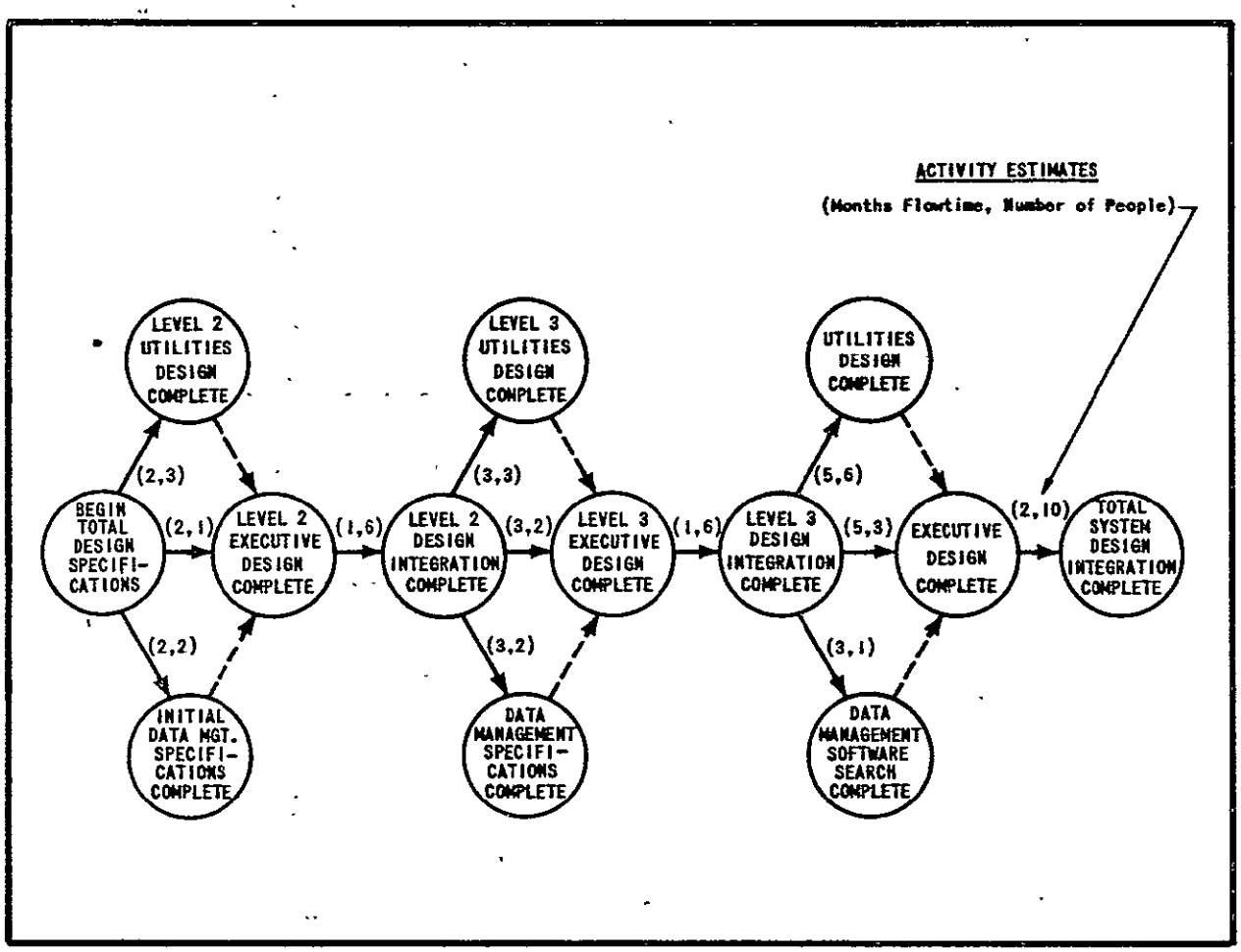


Figure D.1 PERT Chart for SD 9 Activities

DETAILED LABOR ESTIMATES FOR REFINEMENT OF THE UTILITIES AND EXECUTIVE DESIGN ONLY

Estimate for Refinement of Utilities Design

Utility Node* Designator	Level 2	Level 3	Remainder
G	3	6	15
H	3	5	12
I	2	6	15
K	1	5	10
M	2	4	20
N	2	5	12
O	1	2	6
P	3	6	20
Q	2	4	10
W	3	5	12
Total (man-weeks)	22	48	132
Total (man-months)	5	10.9	30

Estimate for Refinement of Executive Design

Executive Node* Designator	Level 2	Level 3	Remainder
E	2	4	10
F	1	3	7
T	2	4	10
U	2	4	15
V	2	4	15
STS Interruption	2	4	12
Total (man-weeks)	11	23	69
Total (man-months)	2.5	5.2	15.7

*See Volume IV, figure 6.2.

Total Estimate for Refinement of the Utilities and Executive Design

	Level 2	Level 3	Remainder
Utilities	22	48	132
Executive	11	23	69
Total (man-weeks)	33	71	201
Total (man-months)	7.5	16.1	45.7

ACTIVITY IDENTIFICATION SI-1

Activity Description

Define the Phase 1 capability based on the Phase 1 capabilities defined in Volume IV, section 2.0, and current design specifications.

Results Expected

Identification of all nodes in the design which will be coded for Phase 1.

Cost Estimates and Flowtime Estimates

Labor (man-months)
4 programming (1)*
4 engineering (1)*

Flowtime (months)
1

Computing Resources
None

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

ACTIVITY IDENTIFICATION SI-2

Activity Description

Produce all development and maintenance aids. This will include collecting software and developing it, or possibly modifying existing software.

Results Expected

Functional software for support of development and maintenance ready to be used as the host system.

Cost Estimates and Flowtime Estimates

Labor (man-months)
32 programming (8)*

Flowtime (months)
8 months

Computing Resources
$6000 computing time

Additional Scheduling Information or Cost Information

Activity SD-4 must be completed first.

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

ACTIVITY IDENTIFICATION SI-3

Activity Description

Perform all operating system modifications in support of the basic executive functions.

Results Expected

Functioning host operating system with all modifications supporting the basic executive - operating system relationship.

Cost and Flowtime Estimates

Labor (man-months)
36 programming (9)*

Flowtime (months)
9

Computing Resources
$3000 computing time + 8 hour blocks of dedicated time/day for 6 months = 1050 hours.

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

ACTIVITY IDENTIFICATION SI-3 (continued)

Additional Scheduling Information or Cost Information

EQUIPMENT FOR SI-3 AND -4 IF A DEDICATED MACHINE IS USED

		Monthly Rent Incl. Maint.	
1	73-13 CPU (49,162 words CM)	$25628	
1	6612 Console	0	
2	7054 Controllers	4000	
2	844-2 Disc Drives	1300	
2	6681 Data Channel Converters	0	
1	3528-3 Controller Tape Channel	1574	
2	657-4 Tape Drives	1934	
1	6676 Terminal Controller (600 baud)	2089	
3	Terminals (CRT with some hard copy output)	300	(approx.)
1	3547 Card Reader Controller	250	
1	405 Card Reader	441	
1	3446 Card Punch Controller	501	
1	415 Card Punch	325	
2	3555 Printer Controllers	1308	
2	512 Printers	2046	
		$41696	

(Also possible charge for KRONOS $4500 + 1200/mo.)

Total Rental Time = 12 months (SI-3 + SI-4)
Total Cost = $500,352

ACTIVITY IDENTIFICATION SI-4

Activity Description

Complete all host operating system modifications.

Results Expected

The host operating system ready to run system checkout.

Cost and Flowtime Estimates

Labor (man-months)
12 programming (3)*

Flowtime (months)
3

Computing Resources
$1000 computing time + 3-4 hour blocks of dedicated time per week = 156 hrs. of block time.

Additional Scheduling Information or Cost Information

For detail of dedicated machine costs, see same section of ACTIVITY IDENTIFICATION SI-3.

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

ACTIVITY IDENTIFICATION SI-5

Activity Description

Establish training methods for the users.

Results Expected

All information and resources ready to train the users.

Cost and Flowtime Estimates

Labor (man-months)
4 programmer (1)*
4 engineering (1)*

Flowtime (months)
2

Computing Resources
$400 computing time

Additional Scheduling Information or Cost Information

User documentation from the system design must be available.

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

ACTIVITY IDENTIFICATION SI-6

Activity Description

Coding and certification of the system.

Results Expected

Stable IPAD system capable of performing all the Phase 1 functions specified in level 1 design in the environment of the OMs selected for checkout.

Cost and Flowtime Estimates

196 programming (56)*

Flowtime (months)

16 months (2)*

Computing Resources

Dedicated machine 1 1/2 shifts per day for the working period.

Additional Scheduling Information or Cost Information

The size of the system is in the 150,000 to 200,000 source statement range and some experience indicates 50 checked out statements per man day output is possible.

Additional equipment needed for activity SI-6 compared to that for activity SI-3 if a dedicated machine is used.

		Monthly Rent Inc. Maint.
2	844-2 Disc Drives	$ 1300
2	6681 Data Channel Converters	688
1	3553 Disc Drive Controller	922
1	821 Disc Drive	6018
1	659 Tape Drive	978
7	Terminals (CRT with some hard copy output)	700
1	10264-2 Memory upgrade to 65,536	6540
		$17146

Total Rent is now $58842
Total rent time = 16 months
Total cost = $941,472

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, trailing, reporting, etc.).

ACTIVITY IDENTIFICATION SI-6 (continued)

Node* by Node Estimates of Source Statements

	Full Design	% for Phase 1	Phase 1 Est
E	5000	80	4000
F	2000	100	2000
T	3000	80	2400
U	10000	60	6000
V	5000	50	2500
Subtask step interruption	3000	100	3000
Data Management	50000	80	40000
Control	10000	20	2000
G	10000	60	6000
H	5000	50	2500
I	15000	80	12000
K	5000	50	2500
M	10000	60	6000
N	5000	60	3000
O	3000	20	600
P	20000	50	10000
Q	5000	40	2000
W	10000	100	10000
	176000		116500

*Nodes refer to System Design, Volume IV figure 6.2.

ACTIVITY IDENTIFICATION SI-7

Activity Description

Train users in preparation for usage after delivery.

Results Expected

Potential users acquainted with the system and ready to do actual problems.

Cost and Flowtime Estimates

Labor (man-months)
1 programming
1 engineering

Flowtime (months)
2

Computing Resources
$1000 computing time

ACTIVITY IDENTIFICATION SI-8

Activity Description

System OM integration. Certification of Phase 1 OMs to the standards specified in activity OM-8. This includes insertion of all OMs and the associated data banks.

Results Expected

All Phase 1 OMs operational under IPAD and satisfying all certification standards.

Cost and Flowtime Estimates

Labor (man-months)
92 programming (20)*
92 engineering (20)*

Flowtime (months)
13 months (1)

Computing Resources
Dedicated machine 2 shifts per day for the duration of the work (4220 hr. block).

Additional Scheduling Information or Cost Information

Additional equipment needs for activity SI-8 compared to activity SI-6 if a dedicated machine is used.

		Monthly Rent Inc. Main.
1	7054 Disc Drive Controller	$ 2000
2	844-2 Disc Drives	1300
1	10264-3 Memory increment to 96,304	8891
2	Terminals (CRT with some hard copy output)	200
		$12391

Total Monthly Rent $71,233
Total rental time = 13 months
Total cost = $926,029

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

ACTIVITY IDENTIFICATION SI-9

Activity Description

Establish plans for the IPAD developmental computing facilities.

Results Expected

A plan and schedule for all computing resources necessary to support the complete Phase 1 developmental effort. This will include equipment and operating system specifications necessary. Where lead time is required, ordering dates will be given.

Cost and Flowtime Estimates

Labor (man-months)
1 programming

Flowtime (months)
1

Computing Resources
None

ACTIVITY IDENTIFICATION SI-10

Activity Description

Prepare computing facilities for activities SI-3, SI-6, SI-8, SI-9.

Results Expected

The computing facilities specified for the specific activities prepared when the activity begins.

Cost and Flowtime Estimates

Labor (man-months)
5 programming (1)*

Flowtime (months)
36

Computing Resources
None

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

ACTIVITY IDENTIFICATION SI-11

Activity Description

Develop and deliver Phase 2 IPAD.

Results Expected

Additional system capability. See Volume IV section 2.0, Table 2.1.

Cost and Flowtime Estimates

Labor (man-months)
160 programming (40)*
50 engineering (10)*

Flowtime (months)
30

Computing Resources
$1,000,000 computing time

Additional Scheduling Information or Cost Information

The computing cost estimate was based on an average of $4900 per man month during the Phase 1 SI-6 and SI-8 activities.

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).

ACTIVITY IDENTIFICATION SI-12

Activity Description

Develop and deliver Phase 3 IPAD.

Results Expected

System design implementation complete.

Cost and Flowtime Estimates

Labor (man-months)
120 programming (30)*
40 engineering (10)*

Flowtime (months)
24

Computing Resources
$780,000 computing time

Additional Scheduling Information or Cost Information

The computing cost estimate was based on an average of $4900 per man month during the Phase 1 SI-6 and SI-8 activities.

*This is the portion of the total estimate due to supervision and miscellaneous costs (secretaries, documentation, training, reporting, etc.).