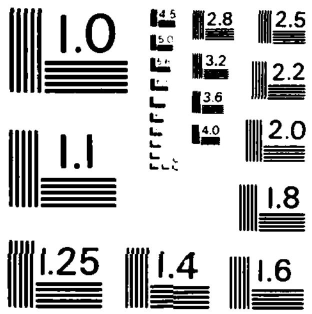


MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ESD-TR-84-151 MTR-9005

AD-A145 413

DESCRIPTION OF THE SOFTWARE ACQUISITION PROCESS (SWAP) MODEL

By
O. SHAPIRO

AUGUST 1984

Prepared for
DEPUTY FOR ACQUISITION LOGISTICS & TECHNICAL OPERATIONS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
Hanscom Air Force Base, Massachusetts

Project No. 6810
Prepared by
THE MITRE CORPORATION
Bedford, Massachusetts
Contract No. F19628-82-C-0001

REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.

ROY A. FLOWERS, 2LT, USAF
Project Officer, Management
Control Technology

WILLIAM J. LETENDRE, GS-14
Chief, Computer Engineering
Applications Division

FOR THE COMMANDER

WALTER W. TURGISS, GM-15
Director, Engineering and Test
Deputy for Acquisition Logistics
and Technical Operations

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) MTR-9005 ESD-TR-84-151	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION The MITRE Corporation 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code) Burlington Road Bedford, MA 01730	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Deputy for Acquisition Logistics (cont.) 8b. OFFICE SYMBOL (If applicable) ALEE	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F19628-82-C-0001
8c. ADDRESS (City, State and ZIP Code) Electronic Systems Division, AFSC Hanscom AFB, MA 01731	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO.: — PROJECT NO.: 6810 TASK NO.: — WORK UNIT NO.: —
11. TITLE (Include Security Classification) Description of the Software Acquisition Process (SWAP) Model	

12. PERSONAL AUTHOR(S)
O. Shapiro

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
Final Report	FROM ______ TO ______	1984 August	192

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	Computer Sofware; Software Acquisition; Embedded Software; Software Cost Estimation; Military Software

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The Software Acquisition Process (SWAP) Model is a software cost and schedule estimating tool that operates by simulating the acquisition process. Its current implementation is tailored to reflect the Air Force 800 series regulations that apply to the Full Scale Development phase of the system acquisition process. This document details the acquisition process depicted by the Model, the concepts underlying its simulation, its current status, and instructions for operating the Model.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☒ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Diana F. Arimento	22b. TELEPHONE NUMBER (Include Area Code) (617) 271-7454 22c. OFFICE SYMBOL

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

8a. and Technical Operations

ACKNOWLEDGEMENTS

This report has been prepared by MITRE Corporation under Project No. 6810, Contract F19628-82-C-0001. The contract is sponsored by the Electronic Systems Division, Air Force Systems Command, Hanscom Air Force Base, Massachusetts.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Concluded)

LIST OF ILLUSTRATIONS

SECTION 1

INTRODUCTION

Planning, coordination, and effective management of the acquisition of large scalc military systems depends on an ability to anticipate the cost and schedule for all components of the system being procured. However, forecasting methods currently used for software components have not demonstrated the accuracy or consistency needed to meet the requirements of the military procurement community.

The Software Acquisition Process (SWAP) Model was developed in response to this need. It is based on the idea that a simulation of the acquisition/development process provides a better basis for cost estimation than the commonly used algorithmic methods. In addition because software lateness can cause system completion delays that are often more costly than the software development itself, the SWAP Model is as concerned with schedule as it is with costs.

The SWAP Model development began in 1979 and continued at a low level of support until 1 January 1983. During this period, the Model's concepts and implementation have matured, but the Model is not yet ready for general use as a software cost/schedule estimation tool. This report was prepared to consolidate and preserve the current state of the Model, and is intended to support usages such as:

- Continuation or resumption of this development.
- Creation or improvement of other cost models.
- Creation of a training course on software acquisition management.
- Documentation of the capabilities and concepts of the current version of the Model, termed Release 1, in enough detail to allow potential users to evaluate its ability to fit into their own acquisition environments.

For these purposes the report provides a detailed description of the acquisition process to which the Model has been tailored, a characterization of the concepts employed to simulate the process generically, and a set of instructions that can be used to operate SWAP Release 1.

This report is organized into nine sections and four appendixes as follows:

Section 2 provides a general overview of the SWAP Model concepts and the premises and assumptions on which the Model is built.

Sections 3 through 5 describe various aspects of the underlying concepts related to simulation, output reporting, and project size scaling.

Section 6 describes the user interface, both the current version and that planned for future versions.

Section 7 describes the ideas behind an abridged version of the Model that can provide a reduced (but considerable) capability in an earlier time frame.

Section 8 describes the current status of the Model and discusses some of its growth capabilities.

Section 9 briefly states our recommendations for the project.

Appendix A provides a detailed description of the acquisition process. It includes a detailed diagram (and associated commentary) of the whole full scale development process along with a mid-level diagram that reflects the abridged version of the process proposed as the Short-SWAP Model.

Appendix B describes the tables that underpin the SWAP simulator and provide parameter values for the boxes that constitute the current base project. It is keyed to the detail level diagram.

Appendix C provides sample output reports that can be produced by the current model as well as graphical versions of these reports that are planned for future models.

Appendix D is a set of operating instructions for the Release 1 version of the Model.

SECTION 2

SWAP DESCRIPTION AND PREMISES

This section presents a general overview of the SWAP Model concepts and the underlying ideas and assumptions behind these concepts. The relationships between the current SWAP Model, SWAP Release 1, and future versions of the SWAP Model are also discussed.

2.1 OVERVIEW

This overview briefly describes the activities of the acquisition process and the method of representing this process in the SWAP Model.

2.1.1 Software Acquisition Process

Viewed as a whole, the software development process converts a set of computer program requirements (e.g., the specification) into a set of computer program products, that include the computer programs, data, and documentation. The process uses resources such as manpower and developmental facilities that account for the cost. These resources are usually provided by a contractor. When the activities of the contracting agency are included, e.g., defining the requirements, awarding the contract, monitoring the development, and accepting the products, the entire operation is defined as the acquisition process (see appendix A).

On any project involving embedded software, which is the current purview of the SWAP Model, a number of major programs (officially referred to as computer program configuration items or CPCIs) are generally to be acquired. In the SWAP Model these may be simulated individually, or in small groups. In the description that follows, it is assumed that the acquisition of a CPCI such as a major operating program, is being simulated.

In the SWAP Model, the acquisition process is represented by a network of boxes; each box reflects a major activity or decision that is to occur during the acquisition. The configuration of boxes, which is termed the project's activity network, is represented by a diagram and also a network linkage table.

Each activity in the diagram is reflected in the network by a rectangular box in which the activity is compactly described. Other tables, which are referred to as the network's activity/decision data base, contain data that indicate the nominal levels of manning and the duration of each activity (rectangular box) and exit (yes/no) probability data for each decision (diamond shaped box). A set of example tables, which are keyed to the detail level diagram (LOSIM) are shown in appendix B.

2.1.2 Method of Operation

SWAP operates by simulating (enacting) the acquisition process that is represented in the project's activity network. It uses the box interconnection data contained in the network linkage table (appendix B, table B-1) to follow the network box by box. The Model resolves (selects an exit for) each decision box, and assigns manning and duration for each activity box until it reaches the end of the network. It keeps track of time and resources used as it progresses and uses that information to create its output forecasts.

Each forecast is driven by the data tables that are used to quantify the simulation. The method used to establish the values in these tables is a formalized extrapolation that converts a set of existing values for a known (base) project into a new set for a user's (target) project. This technique is shown in figure 2-1. All squares referenced in this section relate to this figure.

The base project comprises two components: an activity network (square 1B) and an activity/decision data base (square 1A). Square 2 illustrates the adjusting of the base project's network and data base to reflect the differences between the base and target projects. The first of these adjustments involves altering the network configuration to account for any expected differences from that shown for the base project, i.e., by adding or deleting boxes from the base project's network or altering their interconnection logic (square 2C).

To convert the base project's activity/decision data base into one that represents the target project, the Model first reads in descriptors of the target project (square 2A). The Model then derives a set of scaling factors by comparing the descriptors of the target and base projects. These scaling factors are used to convert the base project's data base into one that reflects the target project (square 2B).

A simulator (square 3) subsequently enacts the entire acquisition process. It follows the network box by box by resolving each decision box while keeping track of manpower use and time. During any one pass through the network, the path followed and the values used for box mannings and durations are subjected to random-

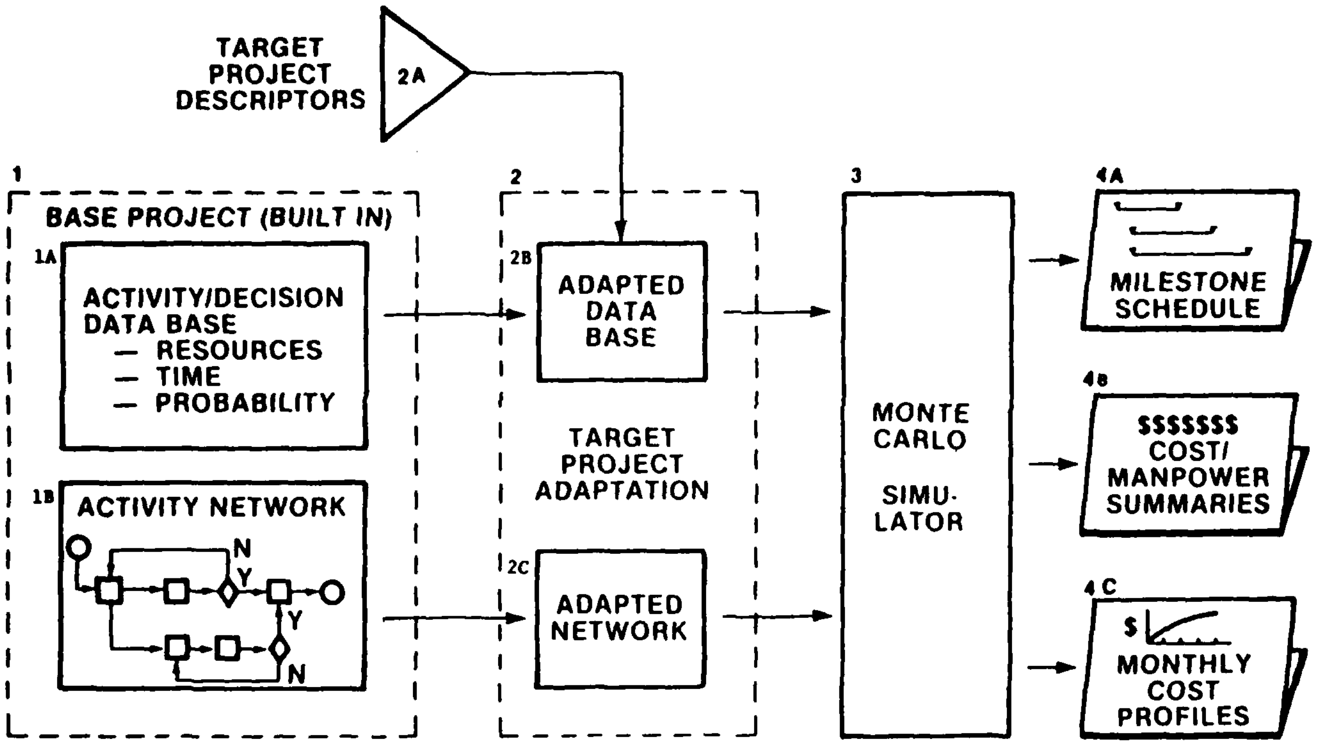


Figure 2-1. SWAP Concepts and Mechanism Overview

like variations. For this reason, many passes are made through the network to allow statistical methods to treat the randomness of the process. The results of the simulation are provided in a number of output reports. These reports include milestone schedules, cost/manpower summaries, and monthly cost summaries (squares 4A, 4B, and 4C).

2.2 BASIC PREMISES

During preparation of the SWAP Model, it was found necessary to delineate the Model and to limit the scope of the initial effort to fit within a limited budget and schedule. The set of basic premises discussed below was established, therefore, as guidance for the initial phases of this work. Some of these apply to the acquisition process itself, others to simplifications introduced for application to early versions of the Model.

2.2.1 Conformance to Military Standards

The acquisition process modeled is intended to conform to all military standards and regulations that are normally applied to software acquired during Electronic Systems Division (ESD) procurements. These include MIL-STD-483, Configuration Management Practices for Equipment, Munitions, and Computer Programs;[1] MIL-STD-1521A, Technical Reviews and Audits for Systems, Equipment, and Computer Programs;[2] AFR 800-2, Acquisition Program Management;[3] and AFR 800-14, Vol. II, Acquisition and Support Procedures for Computer Resources in Systems.[4] If deviations from these practices are found to be necessary, these will be explicitly described (and explained) at each point in the process where each occurs.

2.2.2 System, Segment, and CPCI Relationships

The relationships among activities associated principally with a system, its segments, and its CPCIs will be considerably simplified in the early implementations. In particular, system segments can be used in different ways on different contracts and are therefore not fully amenable to generic implementation. For this reason, the Model addresses the CPCI (level 3) and one level higher. While this higher level is referred to as "system" (level 1) it could as readily represent "system segment" (level 2). The choice depends on the nature of the system and the specific contract(s) being simulated.

In addition, while the Model is designed to accommodate a number of CPCIs, it will treat these initially in a somewhat simplified manner. As thus modeled, all CPCIs will initiate and terminate together (e.g., in the system test), and proceed independently in between. In actual practice, the various CPCIs often have dependency relationships that can be of critical importance to the success of a project. Later versions of the Model will be designed to accommodate these relationships.

2.2.3 Validation Phase Activities

The process model of the full-scale development (FSD) phase presumes that a full validation phase has already been completed. However, since many projects omit this phase but incorporate some of its activities in the FSD phase, provision should be made for such activities (e.g., preparation of development specifications) in the FSD phase model. Extension of the Model to the validation phase is planned for later implementation. The process flow developed for that phase will be designed so that selected activities can be readily moved into the FSD phase.

2.2.4 Support Facilities

The Model presumes that the test and programming support functions are each provided by separate facilities. On some projects, such facilities may be shared (in whole or in part) to support both functions. The Model can reflect any combined use of these facilities.

While the current Model, SWAP Release 1, provides for accumulating the costs of creating, operating, and maintaining support facilities and for the impact resulting from their late availability, it does not include the effect of contention between facility users or the results of unscheduled down time. These capabilities can be added in later versions.

2.2.5 Staged Implementation Provisions

Procurement regulations allow design reviews to be conducted on a single or on an incremental basis. The Model is designed to represent the incremental approach. While this decision adds to the complexity of the Model, it was made because the single design review approach would not support the trend toward staged development, particularly for larger systems. The Model will also accommodate the single design review approach, simply by setting the number of design increments to one.

2.2.6 Incidental Activities

While the Model includes all significant mainstream acquisition activities, it does not include a number of incidental tasks that are essential to a project but would add needless complexity to the Model. Instead, the cost and loading impact of such activities are aggregated into larger mainstream activities. Similarly, certain events and activities judged infrequent or inconsequential to the Model (though not to the acquisition process) are not included. If experience or collected data indicate that some of these incidental activities should be added to the Model, this will be done in a later version.

2.2.7 Resource Utilization

Each process activity uses project resources such as:

a. Contractor manpower in various job categories;

b. Government manpower in various job categories;

c. Development support facilities;

d. Test support facilities;

e. Miscellaneous other resources.

In the current implementation only manpower resources, which are the principal cost components, are assigned to specific process activities.

2.2.8 Excluded Acquisition Cases

The Model assumes that the project being modeled will follow an orderly progression through the network and will reach a successful (acceptable) conclusion. On some projects, however, the original orderly plan may not be followed and/or the products developed may not be acceptable to the government. Generally, these situations develop when the contractor's progress is slower than originally planned and he "short cuts" the process in an effort to catch up. Once a project departs from proper acquisition practices, the Model's forecasts no longer apply. For this reason, only orderly acquisition process data, which are associated with projects that have reached a successful conclusion, are planned for inclusion within the Model's data base.

2.3 APPLICATIONS

The SWAP Model can be usefully applied throughout the definition and development phases of the project, as follows:

- Early Project Planning Phase - The Model's estimating capability is useful for establishing system requirements via cost/benefit analyses and for evaluating alternative system configurations, as well as alternative contractual approaches.

- Contract Award Phase - Contractor proposals can be evaluated by comparing each contractor's proposed methods, allocated staffing, schedule, and development plans for reasonable consistency with the Model's forecast, based on conditions that would apply if that contractor were selected. After the contract is awarded, the Model can be used to forecast a fine-grain schedule, showing the sequence and timing of events, against which actual progress can be compared.

- Contract Monitoring Phase - As the development proceeds, actual results become available that can be compared with those forecast. When the actual data begin to deviate from those estimated, the actual data can be used by the Model as a basis for a revised estimate. This usage mode can be applied throughout the whole development period. In addition, the Model's forecasts can be used to estimate the cost and schedule impacts for any major engineering change proposals (ECPs) being evaluated on the project.

SECTION 3

SIMULATION CONCEPTS

In this section, the techniques used to represent the acquisition process and to conduct the simulation are explained in some detail. This conceptual information is intended primarily for persons who plan to continue the Model's development or to apply it for other applications or in other environments. Some of the concepts discussed are complex and, therefore, not appropriate to the casual reader.

3.1 ACQUISITION PROCESS REPRESENTATION

The acquisition process, as presented in appendix A, is defined at three levels termed HISIM, LOSIM, and MIDSIM. Each of these levels is described below.

3.1.1 High-Level (HISIM)

The high-level representation provides a global view of the FSD phase of the overall process. Because of the vagueness with which the various component interdependencies can be shown at this level, it is not suitable for simulation. It does provide, however, a reasonably clear division of the whole process into its main component activities. It is used, therefore, to describe the level at which project size scaling takes place (see paragraphs 5.2 and 5.3).

3.1.2 Low-Level (LOSIM)

The low-level representation shows the level at which interactions between the government and contractor are shown explicitly. It also shows the major go/no-go decisions and includes the associated iteration for no-go situations. This is the level at which most development work has been done and the intended level for most SWAP simulations. Extensive amplification notes (see appendix A, table A-4) are provided to describe all the activities and decisions shown at the LOSIM level.

3.1.3 Mid Level (MIDSIM)

The mid-level representation was developed to reflect the process at an intermediate level for two purposes:

a. To facilitate the calibration of the Model;

b. To support an earlier usable version of the Model (Short SWAP, see section 7).

3.2 BOX DATA CONTENT

The process flow diagram, which shows the sequence of activities and decisions involved in the acquisition/development of embedded software, provides a qualitative description of the acquisition process. Each box in the diagram must also have its quantitative characteristics specified. The values for these characteristics, which are defined below, are given in a set of tables (one table per box type) that are associated with the diagram. Tables keyed to the LOSIM diagram are shown in appendix B.

a. Activity Boxes are given a nominal duration (in days) and manning level for each of five contractor and three government personnel types. Manning levels may be specified in fractions (i.e., to the nearest tenth of a man) to allow for personnel time sharing. The data include factors for altering (independently) the duration and manning level for up to three subsequent iterations. It is also possible to assign a waiting time (e.g., document shipping time) before the activity may begin.

b. Decision Boxes, which represent the results of an evaluation activity, are given a value for the probability of a YES exit. Probability values for up to three iterative entries (for a total of four values) are given. Counter decision boxes that are used to model staged development (par. 3.4.2) require no explicit quantitative information.

c. Special Event Boxes bear no quantitative values.

d. Personnel Boxes are given parameter values that establish and adjust the levels of contractor personnel types assigned to the project.

3.3 PERSONNEL

The personnel associated with an acquisition are categorized into six contractor types and three government types and assigned to boxes according to their roles.

3.3.1 Personnel Types

a. Contractor Personnel. Personnel quantities in five job categories can be individually assigned to each activity box:

   (1) Systems Engineer or Analyst

   (2) Designer

   (3) Programmer

   (4) Test Engineer

   (5) Support (e.g., equipment operator, librarian, technical writer)

   A sixth category, management, is not specifically assigned to each activity. Because of the difficulty in estimating management resource expenditure on a per activity basis, management is treated as a continuous activity with a controllable utilization profile.

b. Government Personnel. Three job classifications were selected for personnel assignment to specific activities; these reflect the three principal commands involved in system acquisition:

   (1) Developing Command e.g., ESD

   (2) Using Command, e.g., Tactical Air Command (TAC)

   (3) Supporting Command, e.g., Air Force Logistics Command (AFLC)

3.3.2 Levels Assigned to Project (P.Boxes)

Assignment of personnel to boxes is treated differently for the contractor than for the government. This is due to the different roles played by the two parties. In the case of the government, SWAP operates as if enough personnel are available to be assigned to any box that is ready to start. Whenever government personnel are needed to start a box, they are assigned immediately. During the simulation, the quantities of government personnel assigned to all boxes are tracked. After the simulation the model can provide a profile of government manpower usage that can be useful for project planning. This arrangement reflects project reality in that most of the government work during full scale development involves interaction with the contractor, as follows:

a. The government's work can begin only after the contractor has completed some segment of his work, and

b. The government's work must be completed within a contractually imposed time period; it is not determined by the size of its task or by the availability of adequate staffing.

For example, when reviewing a test procedure, the government review and response must be completed within the designated time limit (e.g., 30 or 45 days) or the document is automatically accepted. Thus the quantity of government personnel available will determine the thoroughness of the review rather than its duration.

In dealing with contractor staffed activities, the size of the pool of contractor personnel available has a direct effect on the level of personnel assigned to all of the activity boxes and thus on the duration of the project. The simulator uses personnel boxes (P.Boxes) to control the number of each of the six types of contractor personnel available to the project as a whole.

The personnel pool contains the quantity of contractor personnel, by type, available for individual assignment to project activities. When an activity is able to start (i.e., all prerequisite entry conditions have been met) it must be assigned sufficient quantities of each required personnel type (par. 3.3.3) before it can actually begin. Any personnel that are assigned to one activity box cannot be assigned to another until the prior box completes and releases its personnel.

The first box in the process flow diagram is a P.Box. It establishes initial contractor personnel levels. Additional P.Boxes in the diagram are used to modify (up or down) the level of contractor personnel available for project activities.

3.3.3 Levels Assigned to Boxes

The supply of and demand for contractor personnel varies dynamically during the simulation. P.Boxes change the quantities of personnel (by type) in the pool, while activity boxes (A.Boxes) take personnel from the pool and later return them. The Model reacts to this supply/demand variation in two ways. It can adjust the actual quantitative manning mix assigned to a box (that is ready to start) to reflect the relative availability of personnel. Or if personnel availability levels are below a threshold, it can cause a box to wait in a queue until sufficient personnel become available. The techniques used are briefly described below.

Most activity boxes are designated as predominantly design, program, or test activities; the other activity boxes are termed "general" and are discussed later. The supply/demand availability level for the predominant activity type of a box is used to determine its actual manning (versus nominal manning given in the table) at the time that box is ready to start. The other personnel (termed auxiliary) are scaled proportionately with the dominant type. If there are shortages of any of the auxiliary personnel types, then fewer can be assigned (within limits). Whenever the quantities of personnel assigned to a box are changed from the nominal values given in the table, the box duration is inversely modified to reflect a weighted average of the quantity of personnel actually assigned.

For general activity boxes, the nominal manning levels are assigned, if they are all available. If the available level for any personnel type is less than nominal (within limits), fewer of that type will be assigned and the activity duration increased to reflect a weighted average of all personnel assigned. If the manning availability for any personnel type falls below the threshold, the box waits in a queue until sufficient personnel become available (e.g., by a P.Box or the completion of an A.Box).

3.3.4 Priorities

Boxes that are in queue waiting to start are assigned personnel (and started) in the following priority order:

1. Boxes that are entered for iterative processing.

2. Boxes that have waited (for personnel) for more than D (=20 initially) days.

3. Boxes entered with lower group numbers.

4. All other boxes.

Boxes within each of the above priority classes are started in first-in, first-out (FIFO) order.

3.4 STAGED DEVELOPMENT

Design reviews are often conducted on an incremental basis. The Model is designed to accommodate an incremental or staged development approach.

3.4.1 Staged Development Concept

a. Each CPCI is defined in terms of functional and other requirements to be met at the completion of the current procurement contract. While certain follow-on requirements may also be explicitly or implicitly defined, these are treated as beyond the scope of that contract.

b. The contractor may divide the development of the fully defined capability into several developmental stages called developmental integration groups (DIGs). This division would be defined in a phased implementation plan that would normally be described in the computer program development plan (CPDP).

c. As shown in figure 3-1, the contractor would begin first with an overall global design. He would then proceed with the design of the functional capability planned for the first DIG (DIG-I). The work on this DIG would then pass successively through the high level then the detail level of the design process (including preliminary design review (PDR) and critical design review (CDR)), and through coding, debugging, integration and checkout (I&C), and the contractor's internal computer program test and evaluation (CPT&E). The DIG-I functional capability might also be subject to preliminary qualification testing (PQT), but <u>not</u> to formal qualification testing (FQT).

d. The design and implementation of the functions associated with each of the other DIGs would proceed in order behind DIG-I. Work on the second DIG (DIG-II) would begin after

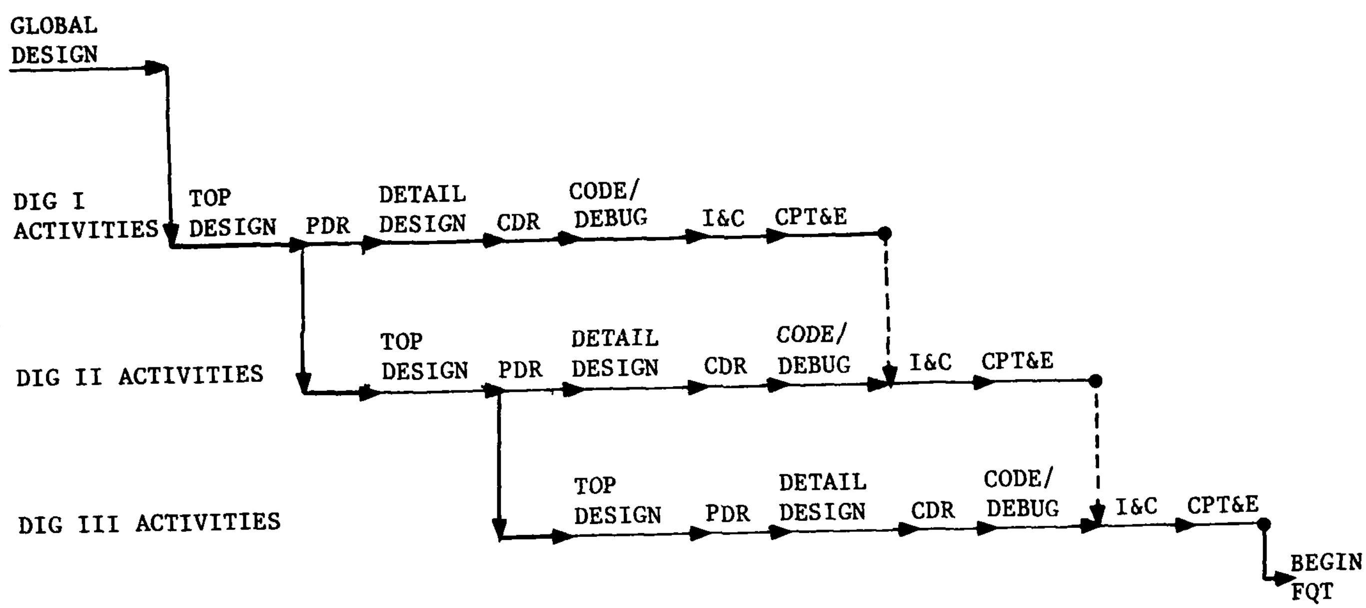


Figure 3-1. Staged Group Development Example

completion of high-level design on DIG-I; DIG-III would similarly start after DIG-II, etc. The incremental PDRs and CDRs and other development activities for each DIG would proceed in the same order.

e. During each phase of development, the capabilities associated with each successive DIG would add to and build onto the aggregated preceding DIGs. In other words, a single CPCI would be built in successive stages; it would not be built as separate DIGs to be joined together at the end.

f. When the last DIG passes through each development phase, the total implementation to that phase would be complete. Therefore, each last DIG design review would be extended to survey the totality of the design, in addition to that of the last functional increment.

3.4.2 Staged Development Implementation

The staged development concept has been implemented generically in the Model, as described below:

a. Counter boxes are decision boxes that are identified by a "C" in the left corner. They are used to cause the activity progression through the network to repeat for each staged sequence of activity/decisions as many times as there are stages. The YES exit in the counter boxes is taken only when the box has been entered for its last stage; i.e. all other stages have already passed through the box sequence.

b. The user can specifiy how many phases (DIGs) are planned and also the percentage of the total effort that is to be accomplished in each DIG. This technique allows the staging to be treated generically; if computer program component (CPC) groupings for each stage had to be specified, the process would need to be separately treated for each project.

c. The simulation program adjusts the nominal durations of all phased activity boxes to reflect the percentage of effort assigned to each stage. For example, if a box duration is 40 days and stage one includes 30% of the whole task, then the duration for that stage will be 12 days. The nominal manning level for a box is not affected by staging.

d. Whenever a box sequence completes a stage, it enters the counter box. If it was not the final stage, the NO exit will enable the first box in the sequence to be ready to start, but on the next stage. The flow logic causes the stage number to increment each time a staged activity sequence is reentered.

3.5 BOX BURSTS

Some activity boxes represent multiple parallel operations. For example, box 10A in figure 3-2, actually represents many similar and concurrent activities being individually and separately conducted. While each of the separate activities requires a low manning level, the box reflects a much higher (aggregated) manning level. Unless the simulator is designed to respond properly to this situation, its behavior will not reflect reality.

For example, since the simulator permits no activity box to begin until adequate levels of personnel are available for it, any box that requires high manning would have to wait in queue until enough personnel accumulate to staff it. During this time, the accumulating personnel would be treated as idle, or might be assigned to other smaller tasks. These conditions are artificial because such processes can begin (in reality) as soon as a minimum of manpower becomes available. Resolving this problem by individually modeling all such activities (instead of aggregating them) could swamp the Model with excess detail and could remove the generic quality from the process representation. The box burst technique described below was devised to obtain the characteristics of reality while also retaining the generic approach.

A burst is conducted over a string of successive boxes. Each string begins with a "start box" and ends with one or more "end" boxes, while all intervening boxes are termed "continue" boxes. The processing associated with each burst type is as follows:

1. When Start Boxes are initiated, they subdivide into N equal sub-boxes (with N set to 5). The nominal activity duration of all activity sub-boxes is the same as for the whole box but each duration is separately randomized. The manpower requirement for each sub-box is 1/N of the whole box manpower. Thus, when a start box is initiated, the number of sub-boxes that start depends on the manning level available. Some may start immediately, others later as additional personnel become available.

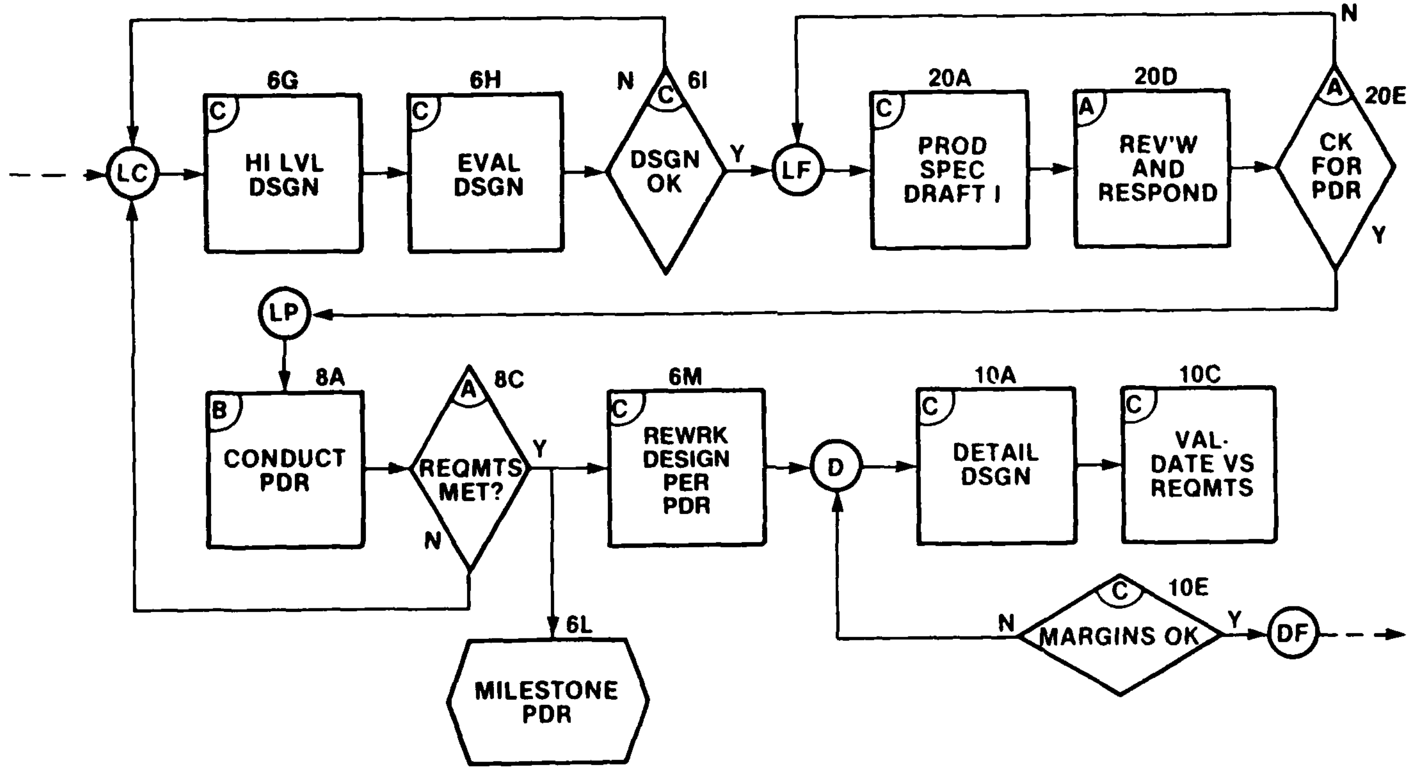


Figure 3-2. FSD Activity Network Representation Simulation Level Example

2. Continue Boxes are treated like start boxes, except that these start and end on a sub-box basis.

3. Each End Box is treated like a non-burst box, except that it cannot start until the last of its sub-boxes flows into it.

3.6 RECURRING ACTIVITIES

Certain activities, such as program management review (PMR), recur periodically until some point near the end of the project. Other activities, such as the operation and maintenance of the development support facilities, are on-going until they are shut down when no longer needed. A remote action box is used to conclude these recurring or on-going activities.

An on-going activity is represented by an activity box with itself as a successor. Once initiated, it remains active for an assigned duration (e.g., 10 days) after which it flows to reactivate itself. Recurring activities have a similar representation with one difference: a waiting time is assigned to provide the periodicity of occurrence. For activities represented by a series of boxes, the last box in the series has the first box as its successor.

Upon activation, the remote action box causes its target box to become ineligible for starting. This causes the recurring or on-going activity box to stop rather than flow back to itself.

3.7 SUBNETWORKS

The model allows the overall network to be functionally decomposed into subnetworks for the purpose of obtaining cost and schedule estimates for any portions of the network that are so designated by the user (see section 4.3). The user defines a subnetwork by assigning it a name (e.g., "Documentation", "All Test", or "Formal Test"), and identifying its constituent boxes. Up to 15 subnetworks may be defined. Output reports can be requested for any subnetwork, or any combined group of subnetworks. For example, "Test Documentation", "Informal Test", and "Formal Test" could each constitute a separate subnetwork. The user could get reports on each of these individually or could combine them all into a single "All Test" report.

SECTION 4

OUTPUT REPORTING CONCEPTS

The simulator can produce the following kinds of output reports:

- Milestone Schedules
- Cost/Manpower Summaries
- Monthly Cost/Manpower Profiles

This information can be made available in two formats: tabular and pseudographic. The tabular format presents the information in more detail and with greater precision than does the pseudographic. The latter, however, provides ample information and adequate precision for many purposes, and provides a clearer grasp of the results. Examples of typical output reports are provided in appendix C.

4.1 REPRESENTATION OF VARIABILITY IN OUTPUT REPORTS

The range of variation is presented to the user via three values for each data item reported. These values are mid-range, optimistic, and pessimistic. The values are based on run data that are segregated into three groups as follows:

- Mid-Range - Averages the middle 50% of passes
- Optimistic - Averages the better 20% of passes
- Pessimistic - Averages the poorer 20% of passes

The range of variation is shown on the pseudographic reports by the letters M, O, and P; see figure 4-1, Summary Forecast. This figure shows, for example, that the loaded cost could vary between an optimistic value of $3M and a pessimistic value of $6.6M, with a most likely value of $4.6M.

The three groups are formed by placing the data from each complete pass through the network into one of the three groups on the basis of the time required to reach the physical configuration audit (PCA), earlier being better. Note that the best and worst 5% of the cases are not used. These were deliberately omitted so that the outlying case data do not influence the results.

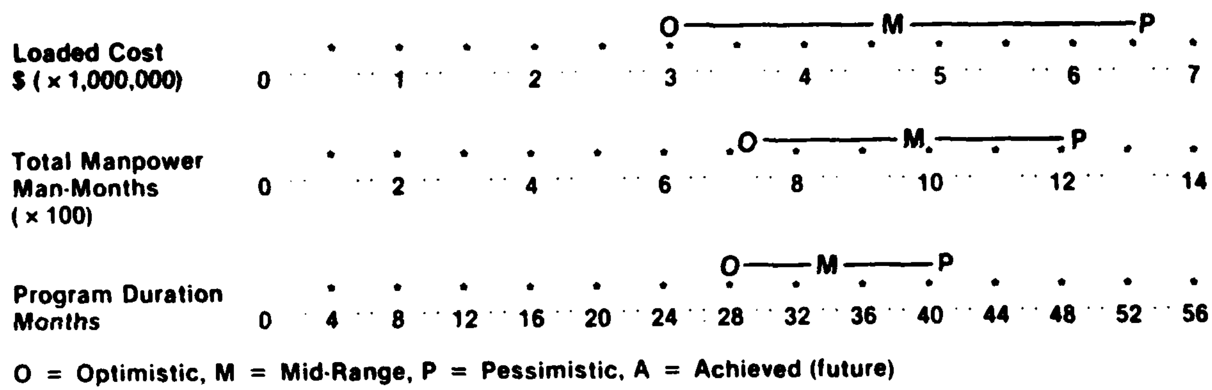


Figure 4-1. Summary Forecast

The three group division thresholds are formed by first finding the mean time and its standard deviation (sigma) to reach the PCA milestone. The thresholds are then established on the basis of the percentage of sigma above and below the mean, assuming a normal distribution. The percentage values used are available for alteration by the user.

4.2 PSEUDOGRAPHIC REPRESENTATION

The pseudographic reports are printed in a format that can be produced by a typical line printer. This format was selected so that the user will not be required to have a particular graphic device. While the granularity of the data presentation is coarse, it is adequate for most purposes. If finer granularity is found necessary, other graphic formats will be considered.

The report generation program uses the printer to print its own background grid and other reading aids; this insures that each report is scaled correctly regardless of line and letter spacings. In addition, the scale markings (for cost and manning) are planned to be automatically adjusted (e.g., by factors of 2, 5, 10) depending on the magnitude of the data. Schedule data, however, are planned to be shown on a fixed monthly scale, with a time extension page being added when necessary.

All of the reports have appropriate titles and markings to make them understandable to anyone acquainted with the software acquisition process. Representative examples of the graphic reports are shown in appendix C, figures C-1 to C-4.

4.3 SUBNETWORK REPORTS

The Model has the capability of printing any report using only the data from an individual subnetwork (par. 3.7), or a group of subnetworks. Each such report will include the number and name of each subnetwork included. An example of a subnetwork report is shown in appendix C, figure C-9.

SECTION 5

PROJECT SIZE SCALING CONCEPTS

As previously described, the acquisition process for software procured as part of a large military system can be represented by an activity/decision network that is made quantitative by the assignment of data values to each of the boxes. The Model allows a user to establish these data values for his (target) project by utilizing either of two methods:

- The user can estimate the manning and duration values for each activity and enter these directly into the data base. He would also estimate and enter the manning profile for the project and probability data for each decision.

- The user can describe his project in terms of defined project attributes that are used by the Model to establish values for all the boxes in the data base.

The second process, which is much easier to accomplish, is expected to be the one commonly used. This method makes use of an existing set of box data that were obtained for some prior (similar) project; this earlier project is termed the "base" project, and its data the "base data set." The process works by quantitatively comparing the attribute values of the user's (target) project with the corresponding values for the base project. This formal comparison process yields a set of scaling or conversion factors that are applied to convert the base data set into one that reflects the target project. This scaling process is further described below.

5.1 BASE PROJECT CONCEPTS

a. Each base project is intended to reflect the actual results experienced on an earlier project. It also can be obtained by other means such as: modifying actual results from an earlier project by eliminating the effect of uncharacteristic kinks in the process; or combining results from several projects; etc. As the Model is put into wider usage, the number of base data sets will increase; see paragraph d. below. The diversity of base data sets will make it possible for a user to select one that most closely matches his own project's functionality, size, developmental methods, and acquisition regulations. As the similarity of the target and base projects increases, the scaling factor values approach unity and the cost/schedule estimates become more accurate.

b. On the current Model, only one base project is available and it contains synthetic data (i.e., it was created from prior experience but is reflective of earlier developmental practices). Its initial purpose was to provide data needed to debug and test the Model's simulation programs and to check out the logic of the acquisition process representation. This initial base data set is included in the Release 1 version (see appendix B).

c. The initial base data set is planned to be replaced by the best available data that can be obtained from an actual prior project (e.g., Combat Grande or PAVE PAWS). Such base data would be suitable for use on one or more early pilot projects.

d. As the Model is used on various target projects, its forecasts will be compared with the actual results subsequently obtained. The observed differences will be used to improve the Model. At the same time, new base project data sets can be created by using the target project results. Because the early uses of the Model are likely to be on ESD projects, the early base data sets will be ESD oriented. As the Model is used by other agencies, each will develop its own sets of base projects to reflect their own acquisition practices and functionality.

5.2 ACTIVITY EFFORT SCALING

The conduct of the software development involves a number of different types of activity. For the purpose of project scaling, these have been grouped into a set of seven "parent" activity categories, as follows:

1. System Analysis and Program Design (Design)

2. Program Coding through Integration (Program)

3. Program Test (Test)

4. Composite (or mixed) Developmental Activities (General)

5. Formal Design and Program Documentation (Design Documentation)

6. Formal Test Documentation (Test Documentation)

7. Formal User Documentation (General Documentation)

For any given project, the amount of effort that goes into each of these parent activities is derived by the Model from the attributes of that project. The magnitudes of these individual effort categories are used in turn to scale the effort expended on the boxes that comprise the acquisition network. Each box is influenced by just one of the seven activity categories, so that the total effort (staff-days) assigned to a box will be scaled by an amount that is influenced by the size of the box's parent activity. Thus the number of staff-days expended in each design activity box for a target project will be scaled twice as high as on the base project, if the target projects design activity factor is equal to two.

5.3 EFFORT APPORTIONMENT BETWEEN MANNING AND DURATION

When a box's effort level is scaled upwards, this can be accomplished by increasing its manning, duration, or both. In the Model, this apportionment is determined by another box attribute, called "growth pattern." Three patterns are defined:

1. F = Fragmented activities. This mode of growth is used for tasks that are minimally impacted by other on-going (concurrent) activities. Although many persons may be performing such a task, they can work independently or in small groups. Fragmented activities are adjusted for differences in effort level by changing the assigned manning level, with only minimal change in the task's nominal duration.

2. I = Integrated activities. This growth mode is given to tasks that are accomplished by teams of people working together closely and interactively. The need for close coordination in these cases causes these tasks to be adjusted in size by changes in both manning and duration.

3. K = Constant activities. This growth mode is assigned to tasks that do not lend themselves to adjustment by scaling,

or to tasks that the user individually sets for his own project. These activities retain their given manning and duration regardless of the scaling factor value.

5.4 BOXES SUBJECTED TO SCALING

The box types used to represent the acquisition process are described in appendix A (figure A-4). Scaling applies to three of these box types: activity, decision, and personnel as follows:

a. Activity box manning levels and durations are adjusted by the scale factor associated with each activity type and growth pattern.

b. Decision box probability is adjusted to reflect the personnel quantity scaling associated with the activities that precede the decision. The YES exit probability diminishes as the number of personnel involved increases. The quantitative relationships and scaling methods are discussed in paragraphs 5.6 and 5.7.

c. Personnel box manning increments also reflect the activity type and growth pattern.

5.5 PROJECT ATTRIBUTES USED FOR SCALING

The activity categories described in paragraph 5.2 are derived from project attributes that are entered by the user. These attributes are organized into the following four groups:

1. Products - These attributes encompass capabilities and characteristics associated with the products to be developed and delivered. All of these can be defined at the CPCI level; some can be further subdivided to the CPC level. Product attributes include program size and some measure of its difficulty and newness; these also include the documentation and test requirements.

2. Methods and Tools - These characterize the methods and tools to be used by the contractor in designing, programming, and testing the computer programs.

3. Staff - These characterize the productivity to be expected from each of the different types of developmental personnel (i.e., designers, programmers, and testers) that will be assigned to the project by the contractor.

4. Contractual Attributes - These are subdivided into the following three categories:

- Contract
- Contracting Agency (i.e., the government)
- Contractor

Contractual factors tend to apply to the acquisition as a whole, while the other factors tend to apply selectively to one or more parent activities.

The specific attributes described below are a tentative set. After experience gained in using the Model, some may be dropped because they have too little effect, or are too difficult to define or obtain, etc. Others may be added, e.g., computer configuration, architecture, or word size, if they have significant impact.

The Model requires the user to enter numerical data (e.g., program size) or for him to select among given alternatives (e.g., program language). Each choice is ultimately converted to numeric values that reflect its impact on each of the basic activities that it affects. The user is also given the option of entering other (unspecified) alternatives and their associated numerical impacts data.

5.5.1 Specific Product Attributes

For each of the attributes listed below, the user selects one of the alternatives presented. All of these attributes can be entered at the CPCI level; a designated few can be entered, alternatively, at the CPC level.

5.5.1.1 CPCI Level Attributes

a. Program/Design Documentation Requirement (select one):

   (1) Full product spec per standard data item description (DID) (e.g., DI-E-3120B), usually with further explicit direction

   (2) Full product spec content - contractor format

   (3) High-level design description plus annotated listing

(4) Contractor's own content/format

(5) Annotated listing only

(6) None required

b. Test Documentation Requirement

(1) Approval Formality (select one):

(a) Formal per DID-PQT and FQT

(b) Formal per DID-FQT only

(c) Formal FQT but with government defined supplementary tests permitted

(d) Formal acceptance at system level only

(2) Detail Level for Procedures

(a) Fully explicit. Each input is explicitly defined in terms that relate directly to the entry device. All expected outputs and acceptability ranges are similarly defined.

(b) Inputs are described in functional terms; actual outputs are evaluated for correctness.

(3) Planned Usage

(a) Documents are used only for formal test.

(b) Documents are to be delivered for on-going baseline testing after government acceptance of the CPCI.

c. User Documentation Requirement

(1) Per specified document format, government approved

(2) Contractor format, government approved

(3) Contractor format, no approval

d. Software Metrics Requirement

A list of metric types (e.g., maintainability, reliability, quality, portability, reusability, integrity, etc.) will be presented to the user. He will indicate the required level for each metric imposed on the target project.

e. Special Test Requirements

(1) Load/Capacity Test

(a) Fully specified for CPCI

(b) Contractor defines

(c) None

(2) Flight Testing Required

(a) Yes

(b) No

(3) Site Testing

(a) At military base

(b) At multiple sites?

(c) None

f. Direct Program Attributes

The parameters listed in paragraph 5.5.1.2 may be entered alternatively at the CPCI level.

5.5.1.2 <u>CPC Data</u>

Any or all of the following information should be entered at the CPC level, if estimates at that level are available. Any information not broken down to the CPC level, should be entered at the CPCI level. If data are entered for any CPC, the size of that CPC must also be entered. If these data are entered at both the CPC and CPCI level, the former will be used by the Model for all cases where it is available. The CPCI level data are then used to fill in any data gaps at the CPC level. If neither data are present, built-in default values at the CPCI level are used.

a. CPC Size

In machine oriented language (MOL) executable instructions (X100). These data are mandatory. (If preferred, the size can be entered in source instructions; in this case, the Model will convert it to MOL, based on the language expansion factor.)

b. CPC Complexity Factor

(1) High

(2) Medium

(3) Low

The user can "trim" his selection by adding a plus or minus. A plus will cause the cost estimating relationship (CER) value to increase by one-third of the given incremental range; a minus will correspondingly decrease the CER value.

c. Programming Language (If more than one is selected, give percentage of each.)

(1) Basic Assembler

(2) Enhanced Assembler (e.g., Macros, Library, Data Definitions, etc.)

(3) FORTRAN

(4) PL-1

(5) JOVIAL (J-73)

(6) CMS-2

(7) PASCAL

(8) Ada

(9) Other

d. Newness (Give percentage for each; 100% = completely new.)

(1) Of Program Design

(2) Of Computer Program

(3) Of Test

e. Criticality Factors (Enter only if critical.)

(1) Ratio of storage needed vs. available (if greater than 0.5)

(2) Ratio of processing time needed vs. available (if greater than 0.5)

f. CPC Functional Reliability Requirement (If CPCI level, give percentage of each.)

(1) High

(2) Medium

(3) Low

This may be trimmed per paragraph b. above.

5.5.2 <u>Developmental Methods Data</u>

a. Design Representation Methods

(1) High-Level Design (Give percentage of each.)

(a) Manual Flow Charts

(b) Chapin Charts

(c) Decision Tables

(d) HIPO Diagrams (Hierarchial input/output (I/O))

(e) PDL (Program Design Language)

(f) FSD (Functional Sequence Diagram)

(g) OSD (Operational Sequence Diagram)

(h) Other

(2) Detail Design (Give percentage of each.)
(Same alternatives as in (1) High Level Design)

b. Programming Methods

(1) Developmental Facility Quality

A facility can provide support for functions such as program library maintenance, linking, loading, debugging, exercising, configuration management, etc. The quality of a facility is derived from considerations such as:

(a) Are all functions supported?

(b) How well are they supported?

(c) How seasoned are the support programs?

With answers to these questions, the user will categorize the facility as: excellent, good, or adequate. This choice may be trimmed per paragraph 5.5.1.2b.

(2) Machine Access Method (Select one.)

(a) Punch card open shop (3 accesses/day)

(b) Punch card closed shop (3 hr. turnaround)

(c) Time sharing option (TSO) terminals, batch (response time)

(d) UNIX terminals, batch (response time)

(e) Interactive, interpretive terminal

(f) Other (enter estimate)

c. Test Methods

(1) Availability of Facility

(a) Physical Access (Select one.)

1) Same building (short walk)
2) Another building (long walk)
3) Must drive to facility

(b) Capacity (Select one.)

This is a measure of the utilization of the total test facility during peak test period.

1) Can get use on day requested
2) Can get use within 2-hours of request
3) Must schedule several days ahead (i.e., test priorities needed)

(c) Reliability (Select one.)

1) 10% unscheduled downtime
2) 5% unscheduled downtime
3) 20% unscheduled downtime

(2) Utility of Facility

A test support facility's utility is based on the flexibility and ease with which it allows a user to:

(a) Enter the conditions that create a test environment and implement a "canned" test scenario

(b) Conduct the test

(c) Isolate and interpret the relevant test results

Using the above, the user will categorize the facility as excellent, good, or adequate, and may assign a plus or minus per paragraph 5.5.1.2b.

d. Project Development Staging (DIGs) (Enter quantity and percentage for each.)

If the user does not know, the following default values are applied:

Size (MOL Inst)	Quant.	% Each DIG
L.T. 20K	1	100
20K-50K	2	60/40
50K-100K	3	40/30/30
100K-200K	4	30/25/25/20
G.T. 200K	5	25/20/20/20/15

5.5.3 Technical Staff Productivity Data (To Be Developed (TBD))

a. Designers

b. Programmers

c. Testers

5.5.4 Contractual Data

These data values are organized into three sets as described and decomposed below. It should be understood that while each data element does impact the development, the quantitative effects of these will reflect subjective judgment. A default value is provided for unknown situations (e.g., contractor not yet selected). Otherwise, the users will be given descriptive guidance and examples to aid in establishing subjective or consensus values.

5.5.4.1 Contract Factors

a. Contract Type

(1) Cost plus fixed fee (CPFF)

(2) Cost sharing (indicate formula)

(3) Fixed price

(4) Contract extension

(5) Other

b. Requirements Definition Quality

This quality is influenced by the completeness, clarity, and verifiability of the functional and quality assurance requirements as expressed in the governing specification. These will be evaluated in terms of excellent, good, or adequate, with plus and minus trimming per paragraph 5.5.1.2b.

c. Schedule Urgency

Based on the size of the project, a moderate start to finish completion time will be presented to the user. He can accept or alter the schedule by indicating high, medium, or low urgency, with plus or minus trimming. These choices will affect the average levels of manpower assigned to each task by the Model.

d. Cost Realism

(1) Sole source negotiation

(2) Normal competition

(3) "Buy in"

(4) Other

5.5.4.2 Buyer (Procurement Agency) Factors

a. System Program Office (SPO) Constituency

(1) Single command

(2) Multiple commands

(3) Multiservice participation

b. Monitoring Policy

(1) Relationship

(a) Primarily formal paper interchange

(b) Informal formative approach

(c) Some defined work sharing.

(2) Distance

(a) Both parties together at contractor site

(b) Both parties together at using site

(c) Frequent technical/administrative interchange

(d) Infrequent interchange (e.g., low travel budget)

c. Willingness to Modify Requirements

(1) System capabilities

(2) Documentation quality

d. Staff Experience (TBD)

5.5.4.3 Maker (Contractor) Factors

a. Management Organization

(1) Single contractor for custom hardware and software

(2) Cocontractors for custom hardware and software

(3) Multicontractors for software

(4) Other

b. Technical Organization

(1) Chief programmer teams

(2) Designer/programmer teams

(3) Designer teams/programmer teams

(4) Other

c. Developmental Practices

(1) Design and Program Verification

(a) Independent interface reviews

(b) Design/program walkthroughs

(c) Other

(2) House Standard Practices

The house practices contribution is influenced by their completeness, quality, and enforcement. These are characterized by the user as excellent, good, or adequate; with plus and minus trimming (par. 5.5.1.2b).

(3) Design Approach

(a) Top down

(b) Doer's choice

(c) Other

d. Managerial/Systems Experience

e. Manning Stability/Turnover Rate

f. Manning Availability Level

The cumulative staffing needs of all the contractor's current projects can meet, exceed, or be less than the current personnel supply. This demand/supply ratio (which can vary during the project) can influence the manning level assigned to the project. The user can indicate whether the supply is high, equal, or low vs. that needed; he can trim his selection (par. 5.5.1.2b).

5.6 ATTRIBUTE/ACTIVITY MATRIX COMPOSITION

Each of the input attributes affects the acquisition process and thus the amount of effort needed to accomplish the general developmental activities. For purposes of simplifying the implementation, these input attributes are separated into two categories:

1. Selective attributes can affect each of the general activities differently (or not at all).

2. Global attributes can affect the project as a whole and are therefore applied equally to all the activities.

5.6.1 Selectively Applied Attributes

These attributes are organized into a matrix similar to that shown in figure 5-1. For each attribute listed in the left most column, a numeric value is assigned that reflects its relative influence on each of the basic activities that head the other columns. If there is no influence on a given activity, no value is assigned; the value in these cases is treated as unity by the program. It should be noted that the values assigned have relative but not absolute significance. This is because they are applied to

PROJECT ATTRIBUTES (USER INPUT)	BASIC ACTIVITIES						
	DEVELOPMENTAL				DOCUMENTATION		
	DSGN	PRGM	TEST	MIXED	PRGM	TEST	USER
PRODUCT (P)	(PD_1)	(PP_1)	(PT_1)	(PM_1)	(PPD_1)	(PTD_1)	(PUD_1)
________	(PD_2)	(PP_2)	(PT_2)	(PM_2)	(PPD_2)	(PTD_2)	(PUD_2)
________	(PD_3)	(PP_3)	(PT_3)	(PM_3)	(PPD_3)	(PTD_3)	(PUD_3)
METHOD (M)	(MD_1)	(MP_1)	(MT_1)	(MM_1)	(MPD_1)	(MTD_1)	(MUD_1)
________	(MD_2)	(MP_2)	(MT_2)	(MM_2)	(MPD_2)	(MTD_2)	(MUD_2)
STAFF (S)	(SD_1)	(SP_1)	(ST_1)	(SM_1)	(SPD_1)	(STD_1)	(SUD_1)
________	(SD_2)	(SP_2)	(ST_2)	(SM_2)	(SPD_2)	(STD_2)	(SUD_2)
CONTRACT (C)	(CD_1)	(CP_1)	(CT_1)	(CM_1)	(CPD_1)	(CTD_1)	(CUD_1)
________	(CD_2)	(CP_2)	(CT_2)	(CM_2)	(CPD_2)	(CTD_2)	(CUD_2)
________	(CD_3)	(CP_3)	(CT_3)	(CM_3)	(CPD_3)	(CTD_3)	(CUD_3)
CUMULATIVE EFFECTS	D	P	T	M	PD	TD	UD

Figure 5-1. Project Attribute/Activity Matrix

both the base and target project cases so that their absolute values "wash out" when the ratio of the two cases is taken. In general, the most likely case is assigned an absolute value of one.

Once the values have been assigned for all attributes, these are then aggregated (vertically) to determine their cumulative effects on each of the basic activities. For example, when the design factor values for all attributes that affect the design effort are multiplied together (aggregated), the product is a measure of the total design effort. In the same way, each of the other activity effort factors are multiplicatively combined.

5.6.2 Globally Applied Attributes

These attributes, which affect the overall productivity of the developmental personnel, are aggregated and then applied equally to all seven of the basic activities. Some of these are applied directly; others exert an indirect influence, as explained below.

5.6.2.1 Direct Attributes

All direct attributes are multiplied together to obtain their combined effect on project productivity. This product can then be applied (multiplied) to each of the selective activity factors (par. 5.6.1).

5.6.2.2 Indirect Attributes

These project attributes deal with the level of staffing assigned to the project, and with their effects on staff productivity and the development schedule.

5.7 PROJECT SCALING

After the target project attributes (par. 5.5) are converted into seven activity effort factors (par. 5.6) they must then be converted into seven activity effort scaling factors (par. 5.2). These effort scaling factors are calculated by dividing the set of activity effort factors for the users target project by a similar set derived for the base project selected by the user. Once these effort scaling factors are obtained, they are used to convert the base project data set into one that applies to the target project. As part of this process, the effort scaling factors must be apportioned between staffing levels and activity durations (par. 5.3). Because of this involvement in staffing levels, the indirect attributes (par. 5.6.2.2) also exert their influence, as follows:

5.7.1 Productivity vs. Project Size

The scaling ratio of a basic activity has an exponentially inverse effect on the productivity (P.SIZE) associated with doing that activity. Every doubling of the scaling ratio results in the activity's productivity decreasing by 10%. If the expansion ratio halves, productivity will increase by 10%. The following table illustrates this result:

Expansion Ratio	0.25	0.50	1.00	2.00	4.00
Productivity (P.SIZE)	121%	110%	100%	90%	81%

5.7.2 Decision Probability vs. Size

SWAP also uses expansion ratios to adjust the decision probabilities of the networks. As the expansion ratio doubles, the YES exit (i.e., successful) likelihood decreases by 6% of the original base project's probability. Conversely, as the expansion ratio halves, the NO exit (i.e., iteration) likelihood decreases by 6%. The following table shows this effect:

Expansion Ratio	0.125	0.25	0.5	1.0	2.0	4.0	8.0
Probability of YES Exit (%)	34	30	25	20	19	18	16
	51	47	44	40	38	35	33
	67	65	62	60	56	53	49
	84	82	81	80	75	70	66

5.7.3 Productivity vs. Manning Level

The productivity expected for any activity is also affected by the manning level attributes. Two inputs contribute to the manning level:

1. Schedule urgency (par. 5.5.4.1c)

2. Contractor manpower availability (par. 5.5.4.3f)

These inputs are combined into a manning multiplier matrix to produce a manning level index as shown below:

		Staff Availability		
		High	Medium	Low
	High	1.3	1.2	1.0
Schedule	Medium	1.1	1.0	0.9
Urgency	Low	1.0	0.8	0.7

Thus, if a project has a high schedule urgency and a low staff availability, the manning level index will be 1.0. Each activity's productivity (P.STAFF) is affected by this index as shown below:

Manning Index	0.70	0.80	0.90	1.00	1.10	1.20	1.30
Productivity (P.STAFF)	110%	107%	104%	100%	95%	90%	85%

If the user has used plus or minus moderators these are applied to adjust the manning index and productivity values by interpolation.

5.7.4 Activity Scaling

The scaling factor values that apply to each activity box must adjust both the nominal staffing level and duration for that box. The technique used is to first determine the staffing level based on the activity factor value (par. 5.2) and the growth category (par. 5.3). Then the two productivity influences are obtained: P.SIZE (par. 5.7.1), and P.STAFF (par. 5.7.3). Finally, the activity duration scaling factor (DUR) is computed. It reflects the influences of the staffing scale (STAFF) the staff productivity (PROD) and the scaling of the total work to be done (WORK).

a. Productivity (PROD) is obtained as the product of its two components:

$$(PROD) = (P.SIZE)(P.STAFF)$$

b. The total work scale (WORK) for the box is the same as the activity effort factor. It is related to staffing, duration, and productivity factors as follows:

$$(WORK) = (STAFF)(DUR)(PROD)$$

c. Thus: (DUR) = (WORK)/(STAFF)(PROD)

5.7.5 Accounting For Randomness

Thus far, all project size scaling has been described as a deterministic process. There is, however, much randomness to the process. Its causes and the methods of dealing with them are as follows:

a. The project attributes listed in par. 5.5 are not the only ones that affect the project. All other attributes, which exert unknown effects, are treated by the application of pure randomness to all scaling.

b. The CER weighting ascribed to all the defined project attributes are only approximations that will gradually be refined as knowledge improves; they will never become exact values. For this reason, the Model adds a random component to these weights each time it scales a target project.

c. Many attributes are determined by subjective evaluations; these attribute weights are inherently subject to judgmental variation. The Model therefore randomizes these weights over a wider range than those used in item b. above.

d. Many attributes are unknown to the user at the time an estimate is performed. For example, the contractor's methods and tools, etc. would not be known before the contractor is selected. The Model assigns "most likely" values for these attributes, but increases the amount of randomness to reflect this greater uncertainty.

e. There is an element of pure chance associated with any human creative effort. System development is a heuristic process; its paths are guided by intuitive insights that may or may not lead to an optimal formulation or even sometimes to one that can work. The Model treats this by introducing randomness into every activity.

Before each pass through the network, the randomness elements are introduced into the scaling process that converts the base project nominal data set into one that applies to the target project. On each pass through the network, therefore, the nominal data associated with each box will be somewhat different. In addition, because of the randomness associated with each decision box, the path through the network will be different for each pass. Because of these differences, the dynamic relationship between the availability of and demand for personnel will differ on each pass. The Model dynamically adjusts activity box staff levels (and durations) from their nominal values to reflect the supply/demand situation.

5.8 MANUAL ENTRY OF BOX DATA

The user may prefer to provide his own values for all the boxes in the network. This would be done if no base project exists that is suitable for scaling, or if the user wants to create a new base project. The Model supports this direct method of data entry. This direct method is the only one available on Release 1; the systemized method of scaling the base data set, as described in paragraphs 5.5 through 5.7, is not available for that version.

SECTION 6

USER INTERFACE

The user interface (USI) provides the means by which an operator can access and control the SWAP Model to enter his project descriptors, run the simulation, and obtain his results. In this section, the currently used USI is described in sufficient detail to allow a potential SWAP user to operate the Release 1 version. In addition, the planned approach for a future "friendly" USI is briefly characterized.

6.1 CURRENT RELEASE 1 VERSION USI

The Release 1 User Interface (USI-1) is oriented primarily for use by developmental programming personnel. It uses TSO for data entry and job control language (JCL) for some of the operational control. Both of these are available in the MITRE Bedford Computing Center. While a more friendly interface is planned, USI-1 is not difficult to use or to learn. A set of operating instructions is provided in appendix D.

6.2 PLANNED FUTURE USI

The future USI planned will use a menu technique that permits the operator to control the whole simulation process by following a hierarchical sequence of displays at his terminal. A USI definition document is being prepared to define the technique, describe by example the content and format of each display, indicate the operator/system interactions needed to perform a simulation, and identify the rules for uniquely identifying each simulation run. The USI document will be initially used as a means for consolidating the views of cognizant personnel, before the USI is implemented. It will then be used as the basis for implementing the enabling computer program, which is planned to be written in the PL-1 programming language. Lastly, the document will be expanded to become a SWAP users manual. Each of the functions of the USI is briefly described below.

6.2.1 Simulation Run Identification Label

SWAP can be used to model a given project many times over a long period of time. In order to distinguish the reports obtained from the many different runs, and to avoid label duplication, a standard naming sequence is specified. Each simulation run is identified by a four-field label, as follows:

a. Project Name. The commonly used project name (e.g., JTIDS, E3-A, COMBAT GRANDE, or PAVE PAWS) will provide the first field.

b. Computer Program (CPCI) Name. A number of CPCIs are usually provided on each project. A separate simulation will normally be run for each major C.CI, or on a group of smaller ones. The functional name for the CPCI (e.g., Operating Program, TADIL B Program, Test Support Program, Simulation Program, etc.) constitutes the second field.

c. Simulation Name. Any given CPCI may be simulated several times. For example one might wish to separately simulate the proposals provided by a number of competing contractors (e.g., IBM, SDC, GE, GTE/Sylvania, Hughes) and compare these with one or more independent cost estimates (e.g., ICE #2). The name of the simulation forms the third field.

d. Option Index. Any prior simulation may need to be repeated, but with some alternative conditions (e.g., with different documentation, less test, a different programming language). Each alternative will be given an index number which automatically increments for each such run. The user will separately define each option to name its purpose and describe its specific changes.

6.2.2 Simulation Run Selection

To begin a run, the user must identify the specific situation he plans to run. This situation may be an entirely new project or it can be as little as just a change in option. The USI will provide a sequence of displays that enables the operator to select a previously entered condition or enter a new one. These displays will identify (in turn) all previously run projects, all previously run programs for the selected project, all previously run simulations for that program, and all previously run options for that simulation.

6.2.3 Task Selection

Once the simulation run conditions are established, the user can identify the task he wishes to perform. All tasks are identified on a main menu to allow the user to make his selection. He may choose such tasks as:

a. Altering the process network

b. Entering attributes of his project that allow the Model to establish the effort level associated with each box in the network

c. Conducting the simulation

d. Requesting output reports

Each of these tasks will be associated with a sequence of displays that guide and enable the operator's actions.

SECTION 7

PROPOSED ABBREVIATED MODEL (SHORT SWAP)

Many potential SWAP users have expressed strong interest in obtaining a usable capability in an earlier time frame. For this reason, an abbreviated version of the Model, dubbed Short SWAP, has been formulated.

7.1 OBJECTIVES

Three objectives were established for the Short SWAP capability:

1. To provide an earlier usable capability by scaling down the Model, not by scaling up the developmental effort.

2. To preserve as much of SWAP's unique capabilities as is possible.

3. To support the Model's future growth to full potential. Changes should not constitute a developmental detour away from the full planned capability.

All of these objectives appear to be realizable by the Short SWAP concepts described in this section.

7.2 SHORT SWAP CONCEPTS

Short SWAP is planned to be the same as regular SWAP except for the following changes:

7.2.1 Acquisition Process Modeling Level

a. Each box in the regular SWAP Model represents an activity or decision that is taken by either the contractor or the government. It also includes the "both" case, when the work directly involves the two parties. Whenever decisions cause rework of an earlier activity this iterative behavior is treated explicitly. At this level the process requires about 150 activity/decision boxes.

b. On Short SWAP, each box reflects a broader task, so that both parties commonly participate in most higher-level boxes. In addition, decision boxes are not used so that iterative behavior is not explicitly shown. The Model does separate the government and contractor contributions, however, by treating separately the contractor's personnel types and those of the government. It also tries to take into account the iterative behavior of the acquisition/development process by the assignment of time durations and manning levels that reflect the total effort levels normally expected. At this level, the total full scale development process can be expressed in about 40 activity boxes.

7.2.2 Developmental Personnel Usage Treatment

a. In regular SWAP the levels of personnel assigned to the project are explicitly controlled, and these levels dynamically regulate the rate at which the acquisition activity proceeds. SWAP does this by adjusting personnel levels (and durations) assigned to individual boxes on the basis of personnel availability at the time of assignment. It also causes boxes to wait when inadequate quantities of personnel are available.

b. In Short SWAP, each box is assigned a fixed personnel level that is assumed to be available during the simulation. The box duration is similarly fixed. Personnel quantities are not explicitly controlled, and no box ever waits for personnel. This model does reflect the supply of personnel available, but in a fixed way. The general level of personnel availability, which is determined by user inputs (par. 5.7.3), is used to establish the staffing levels and box durations before the simulator passes through the network. Short SWAP does respond to expected staffing levels, therefore, but not in a dynamic way.

7.2.3 Process Variability and Project Attributes

a. Regular SWAP recognizes that each pass through the network will be different because of randomness in resolving decision branches, dynamic differences in box staffing and differences in box durations caused by project uncertainties (par 5.7.5). The variability in results is reported in terms of optimistic, pessimistic, and most likely forecasts.

b. The initial version of Short SWAP will not deal with project uncertainties. As a result, only one pass through the network will be needed to produce a mean value or most likely forecast.

c. A later version (i.e., Short SWAP +) will include a wider set of project attributes and, more importantly, will contain the effects of unknown project attributes, iteration variability, cost driver uncertainties, etc., to reflect projection of uncertainties into its forecasts.

d. The full SWAP Model plans to directly include the effects of almost 50 project attributes in its forecasts. The initial Short SWAP will include a smaller set.

7.3 APPLICATIONS

The Short SWAP version should be emminently suitable for longer range (e.g., project planning) applications. The large range of data (attribute) uncertainty during the early time period would prevent the advantages of a full SWAP implementation from being realized.

In addition, the higher-level reflection of the process, as provided by Short SWAP, will be useful for Model calibration. The higher-level representation corresponds more closely with the data that can be gleaned from prior projects. In particular, the deterministic outputs produced by the initial Short SWAP version will be more readily compared with the available data.

7.4 GROWTH

While the initial Short SWAP version can fulfill the objective of an earlier usable model, it will also support the Model's longer range objectives. This version can grow and improve in ways that support both models. The Model can add:

- Effects of variability into its forecasts
- Effects of a full set of project attributes
- Improved accuracy through easier calibration
- Friendly user interface that can be the same as for full SWAP

In these ways, Short SWAP can produce an improving product on its own, while at the same time supporting the longer-term full SWAP capability.

SECTION 8

CURRENT STATUS AND NEEDS

In this section, the current status of the Model is described, followed by an evaluation of two alternatives for achieving an initial operating capability. Some longer range capabilities and applications for the Model are also described.

8.1 CURRENT CAPABILITY

The FSD phase of the acquisition process is thoroughly defined, (appendix A). It is described at three levels: a very high level (HISIM), an intermediate level (MIDSIM), and a detail level (LOSIM). A comprehensive set of notes is provided to explain and clarify the process depicted at the LOSIM level.

Implicit in the above is the existence of a notation that defines all the elements of the process and the logic of their interconnection and interaction. This notation allows the current diagrams to be altered or replaced to represent other acquisition techniques, or other products, etc.

The computer programs that conduct the simulation have been written and are in an operable state. Some of the programs need additional work, however. In particular, routines that deal with the staffing level assigned to a project and with the dynamic assignment of these personnel to specific tasks on the project need to be exercised and perfected until the simulation can reflect actual experience adequately, insofar as cost and schedule impacts are concerned. In addition, routines that scale the base program data base to create a data base for a target project need to be altered to implement the matrix scaling technique described in section 5.

8.1.1 User Interface

a. A user input and operational control interface is fully implemented and usable. It is programmer oriented, however, and is dependent on the TSO capability of the MITRE IBM mainframe for its operation. No provision has yet been made for remote access.

b. A large set of output reports can be obtained to meet operational and developmental needs. All of these reports are in a tabular format; see appendix C. None of the graphic formats shown in the appendix have been fully implemented.

8.1.2 Data Base

a. Data for a base project has been synthesized as shown in appendix B. These data are adequate for exercising and checking the simulator computer programs, or for conducting some trade studies or sensitivity analyses. The data base is not considered adequate to serve as a basis for project cost/schedule estimation, however.

b. The CERs needed for insertion into the scaling matrix (par. 5.6) have not yet been fully formulated.

8.2 ALTERNATIVE DEVELOPMENTAL APPROACHES

Any further development of the SWAP Model can be conducted according to either of two options: (1) complete the regular SWAP, or (2) proceed with the Short SWAP version described in section 7. Each of these alternatives is described in this subsection with respect to the effort needed to achieve an initial operational capability (IOC), as well as one step beyond this.

Certain capabilities are needed regardless of the option selected. These include the following:

a. Development, refinement, and diversification of the data base must be considered to be the major on-going activity.

b. Any model version should be applied to one or more pilot projects before it is put into wider use.

c. The user system interface improvement (par. 6.2), should apply equally well to either version.

d. A remote access capability should also apply equally to either.

The two options are described, compared, and evaluated below.

8.2.1 <u>Regular SWAP Option</u>

a. In order to get the regular SWAP option into a form that can be used initially by ESD for software cost estimation, the following tasks would need to be done:

(1) The staffing algorithm (par. 3.3) must be brought into a usable state.

(2) At least one base project data set (par. 3.2) must be formulated.

(3) One set of CERs must be formulated for a reasonable subset of the project attributes listed in par. 5.5.

(4) The matrix scaling algorithm (par. 5.7) must be implemented.

b. The next step after the initial capability should include:

(1) At least one additional base project data set.

(2) Further CER refinements and additional project attributes.

(3) An improved user interface, including graphical output reports per appendix C.

(4) Some form of usable remote access if there is other than ESD sponsorship.

8.2.2 <u>Short SWAP Option</u>

a. In order for this option to become usable for software cost estimating at ESD, the following tasks need to be accomplished:

(1) A set of tables equivalent to those shown in appendix B (tables B-1, B-2, B-4, and B-5) needs to be created.

(2) A simulator program change needs to be instituted that would allow a box to flow to each of its successors at different times. This is because the MIDSIM boxes, each of which contains an implied network of LOSIM boxes, can exit to other MIDSIM boxes at different points within the implied network.

(3) Tasks listed as items (2) through (4) in paragraph 8.2.1a would also need to be accomplished for Short SWAP.

b. The next step for Short SWAP includes the same items listed for regular SWAP (par. 8.2.1b).

8.2.3 Options Compared

a. While most tasks are the same for both options, the few differences are significant. Task a(1) for regular SWAP can require a considerable effort to recursively define, implement, checkout, and evaluate this aspect of the Model's behavior. In contrast, tasks a(1) and a(2) for Short SWAP are relatively simple and routine.

b. The creation of base projects is needed for both SWAP models but that activity is much easier for Short SWAP (see par. 7.3).

c. An initial set of CERs can be established and installed into the matrix driven scaling program for both options, either with or without built-in variability.

In the case of regular SWAP, however, the variability due to decision path selection probabilities is already built in. By itself, this single contributor gives a variability range that is misleadingly narrow (see par. 5.7.5).

d. As a result of the above considerations, it appear that a Short SWAP version can be fielded in an earlier time period, and could provide a good initial capability.

8.3 LONG TERM CAPABILITIES

Some examples of long term capabilities are briefly explored in this subsection.

8.3.1 Data Base Improvements

It is probably axiomatic to state that no model can be better than the data on which it is based. As a corollary, it follows that a model can improve as the quality and extent of the available data base grow. The SWAP Model is better able to make full use of such data than are other models because it directly allows each CER to be narrowly focused on only those activities that it affects. By this means, it may be able to avoid much of the extreme scatter that characterizes the performance of existing models versus the data bases on which they have been calibrated.

8.3.2 Added Capabilities

Certain capabilities will not be available on early versions of the Model. Examples of some that can be added later are briefly described below:

a. Critical Path and Slack. These program evaluation and review technique (PERT) type parameters can be obtained by further processing of SWAP results.

b. ECP Modeling. The engineering change proposal is the mechanism for introducing, controlling, and managing changes in requirements after a contract has been awarded. The Model can be altered to explicitly model the processing of ECPs, including their effects on the on-going effort.

c. Constrained Estimation. Some users need to constrain an estimate to obtain specific types of solutions. For example, a user may want to impose a fixed schedule or cost, or may seek an earliest possible solution or one that will obtain the lowest cost. The Model can be designed to find these special case solutions.

d. Program Sizing. The SWAP Model, and practically all others, uses program size as a critical cost parameter. In many cases, however, the cost estimate must be done before the size has been established with any degree of certainty. The SWAP Model can be configured to internalize the size estimation process, based on the user's description of the functionality of the system and the technology of the implementation.

e. Progressive Refinement of an Estimate. As a general rule, the earlier the time period for an estimate, the less is known about the system. The range of variability in the forecast, therefore, should become narrower as the system definition matures. The SWAP Model is designed to reflect this situation. Once the contract has been awarded and as the development proceeds, the maturation continues; but now actual performance data become available. The Model can be modified to use the actual data as a basis for reworking the forecast so as to refine the estimate further.

8.3.3 Diversified Applications

The current SWAP Model has been tailored to reflect a limited range of applications. Specifically, these are confined to embedded software CPCIs acquired in accordance with ESD practices and only for the full scale development phase of the process.

The same basic model can be tailored to apply to other products, other acquisition regulations, and other phases. For example, the Model can be configured to apply to other software, or to the development of custom hardware, or (at a higher level) to a system as a whole. The process diagrams can be altered to reflect other acquisition practices such as those followed by other projects or other services. The same simulator can also model both the pre-FSD phases (concept/validation) and the post-FSD phases (system turnover, operational testing, and program maintenance).

SECTION 9

RECOMMENDATIONS

MITRE recommends that the development of the SWAP Model be continued, with the focus on bringing the Short SWAP version into operational use, including the conduct of at least one pilot application. The Short SWAP version should be improved also, at least to the point where it can be used to provide probabilistic estimation.

LIST OF REFERENCES

1. MIL-STD-483, Configuration Management Practices for Equipment, Munitions, and Computer Programs, 31 December 1970.

2. MIL-STD-1521A, Technical Reviews and Audits for Systems, Equipment, and Computer Programs, 2 January 1975.

3. AFR 800-2, Acquisition Program Management, 14 November 1977.

4. AFR 800-14, Vol. II, Acquisition and Support Procedures for Computer Resources in Systems, 26 September 1975.

APPENDIX A

SOFTWARE ACQUISITION PROCESS DESCRIPTION

APPENDIX A

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

A.1 INTRODUCTION

This appendix provides a full description of the acquisition process depicted in the Software Acquisition Process (SWAP) Model. The process has been diagrammed at three levels: high, medium, and low; a detailed commentary keyed to the low level diagram is provided.

A.2 SCOPE

The process depicted has been tailored to reflect an environment that is typical for the Air Force Electronic Systems Division (ESD). As such it reflects the following:

- An acquisition managed under the Air Force 800 series regulations.
- Activities in the full scale engineering phase of the acquisition process.
- Acquisition of software that is embedded in a major military system
- Textual description of the process intended to apply to a major computer program configuration item (CPCI) such as an on-line mission-critical program.

The process representation method shown preserves the generic properties of the acquisition process. It can be easily tailored to reflect other regulations or products.

A.3 USE

This appendix is intended to clarify the acquisition process for SWAP users so that they can tailor this representation to reflect the situation expected on their own projects. The information also provides work descriptions for the various boxes to aid in establishing staffing and duration values for each box in the network. By these means, the user can create new base projects or run direct (unscaled) simulations. While the process description is intended to be generic, it has been described in explicit terms that may not exactly conform to expectations. As long as the differences do not significantly affect manning or schedule, they can be safely ignored.

The low–level diagram and associated notes provide a coherent event-by-event description of a typical military system acquisition. Such a description can have multiple useful applications. It can provide a basis for a training course on software acquisition that can help prepare government and MITRE personnel for assignment to a System Program Office (SPO). To this end, the notes offer some advice on selecting preferred techniques and paths, where choices are available. For this usage, the notes could be expanded to explore further the consequences of such choices. They should also be annotated by references to appropriate paragraphs in the applicable military regulations and standards.

The diagrams and notes can also provide a valuable aid for those who are planning a full scale development contract. These allow the user to anticipate the activities, events and decisions that are normally encountered, so that none will be overlooked; also, he can plan and arrange to have necessary time and resources available. If the diagram and notes are used with the simulation model, a sense of time and effort can be determined for each activity, along with a probable milestone schedule. This information, which allows the actual project progress to be compared with that forecast by the Model, permits evidences of slippage - either in quality or progress - to be detected earlier, so that appropriate responses can be quickly decided and instituted.

A.4 ORGANIZATION AND FORMAT

A.4.1 Box Identification Labels

The box numbering method used for all three level diagrams is intended to reflect the successive levels of decomposition by which the diagrams were prepared. A single numeric field is used to identify each box on the HISIM (high level) diagram, figure A-1. The leading digit in this field identifies the function in a broad way. For example, the tens deal with requirements, the twenties with design, the thirties with programming, the forties with test plans and procedures, etc.

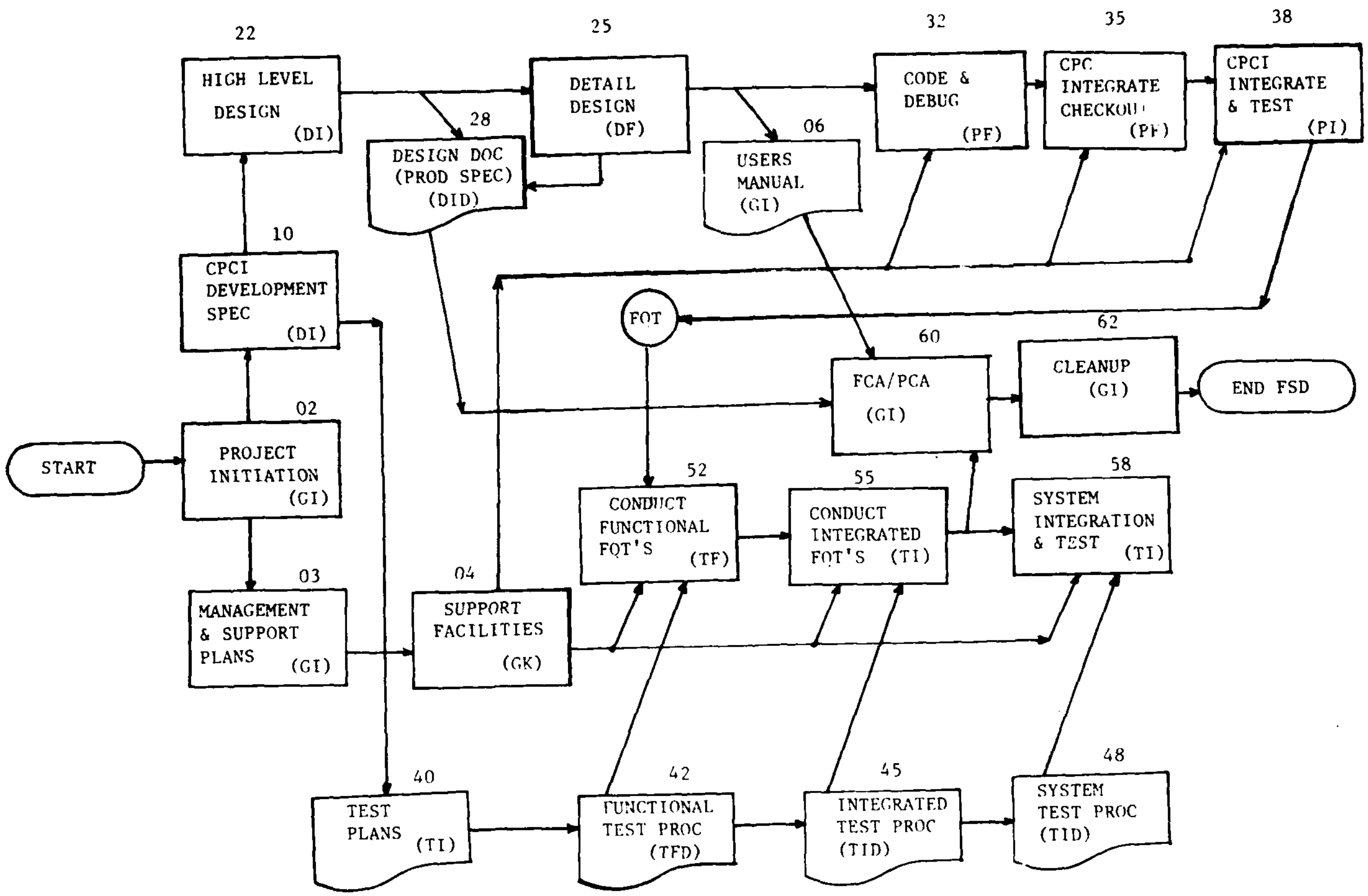


Figure A-1 Software Acquisition Process Flow Diagram High Level (HISIM) Representation.

When the high level boxes are decomposed for the MIDSIM level diagram (figure A-2), the numeric field for each MIDSIM box is taken from its parent HISIM box. A single letter field is then added to provide a unique identification for each box. A few of the letters are restricted to specific box types. For example, the letter "S" is used for stage counters, "M" for milestones, "R" for remote actuators, "P" for personnel boxes (not shown), etc.

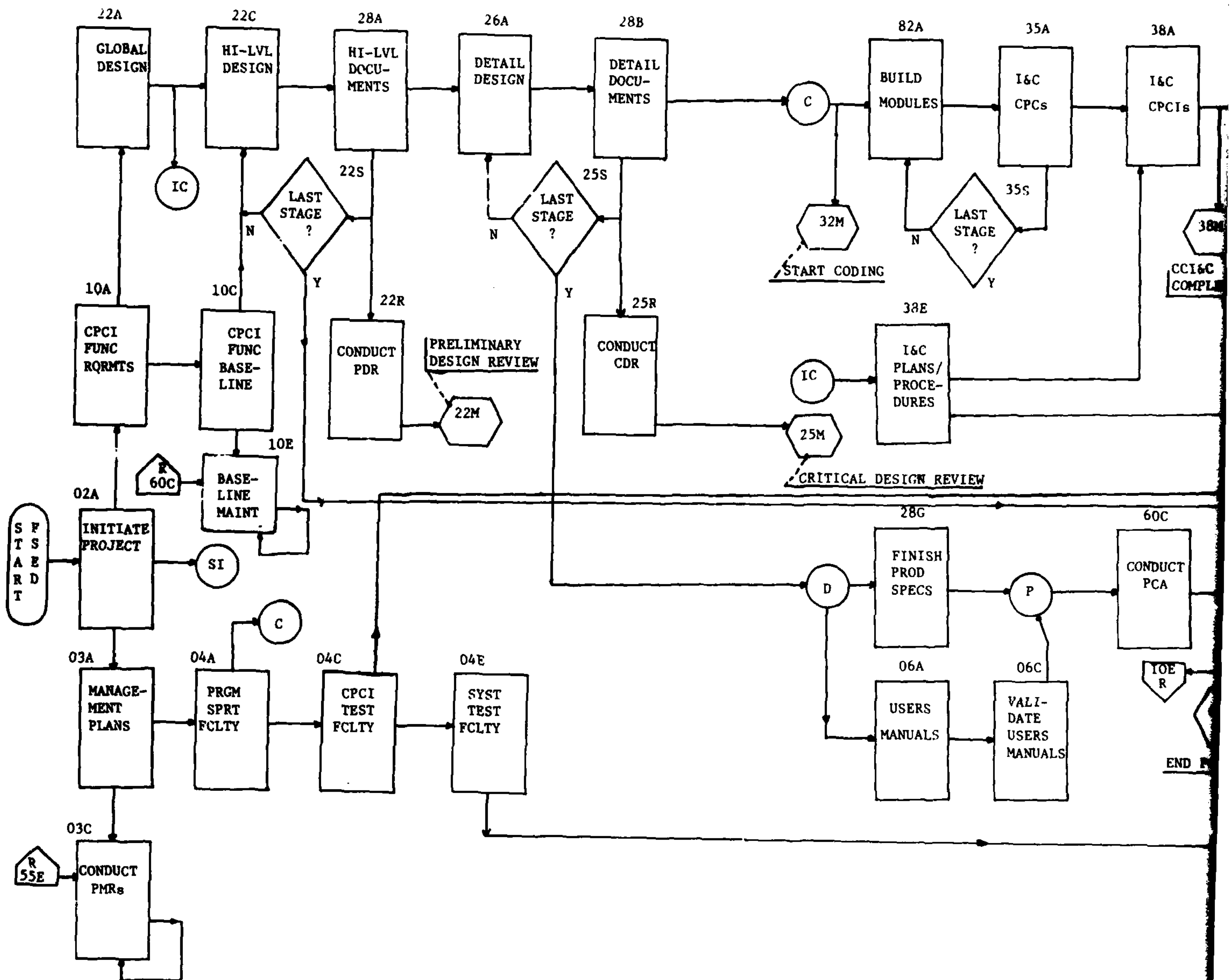
When the MIDSIM boxes are decomposed to obtain the LOSIM level representation, each low level box will carry its parent MIDSIM label plus an added numeric field to create a unique identification for each box. The current LOSIM diagram (figure A-3), which is derived from a previously used two-level decomposition, does not reflect the labeling technique described herein; it will be updated at a future time.

A.4.2 Flow Diagram Interpretation Aids

Along with the diagrams and notes that describe the acquisition process, additional information is provided that can materially aid in the use and interpretation of the diagrams. Figure A-4 describes and explains the abbreviated notation used on the LOSIM level diagram (figure A-3) and explains three basic elements: function boxes, auxiliary elements, and lines of flow. Table A-1 greatly reduces the effort in following the flow through the eleven pages that constitute the LOSIM level diagram (figure A-3). It also shows the page number on which each box appears. Table A-2 allows the user to find all amplification notes that reference any LOSIM level box. Table A-3 expands all abbreviations used in the diagrams.

A.4.3 Amplification Notes Format

A table of contents for the amplification notes is provided in table A-4. The notes themselves (table A-5) are arranged so that each note deals with a functional sequence indicated by its title. Each note covers boxes listed at the left margin; a hyphen following a numeric box designator indicates that all boxes using that number are included. The contributions of individual boxes are given by the insertion of their box numbers at appropriate points in the text.



22A
GLOBAL DESIGN
22C
HI-LVL DESIGN
28A
HI-LVL DOCU-MENTS
26A
DETAIL DESIGN
28B
DETAIL DOCU-MENTS
C
82A
BUILD MODULES
35A
I&C CPCs
38A
I&C CPCIs
IC
22S
LAST STAGE ?
N
Y
25S
LAST STAGE ?
N
Y
32M
START CODING
35S
LAST STAGE ?
N
Y
CCI&C
10A
CPCI FUNC RQRMTS
10C
CPCI FUNC BASE-LINE
22R
CONDUCT PDR
PRELIMINARY DESIGN REVIEW
22M
25R
CONDUCT CDR
25M
CRITICAL DESIGN REVIEW
38E
I&C PLANS/ PROCE-DURES
IC
10E
R 60C
BASE-LINE MAINT
02A
START
FSED
INITIATE PROJECT
SI
28G
FINISH PROD SPECS
D
P
60C
CONDUCT PCA
03A
MANAGE-MENT PLANS
04A
PRGM SPRT FCLTY
C
04C
CPCI TEST FCLTY
04E
SYST TEST FCLTY
06A
USERS MANUALS
06C
VALI-DATE USERS MANUALS
TOE R
END
03C
R 55E
CONDUCT PMRs

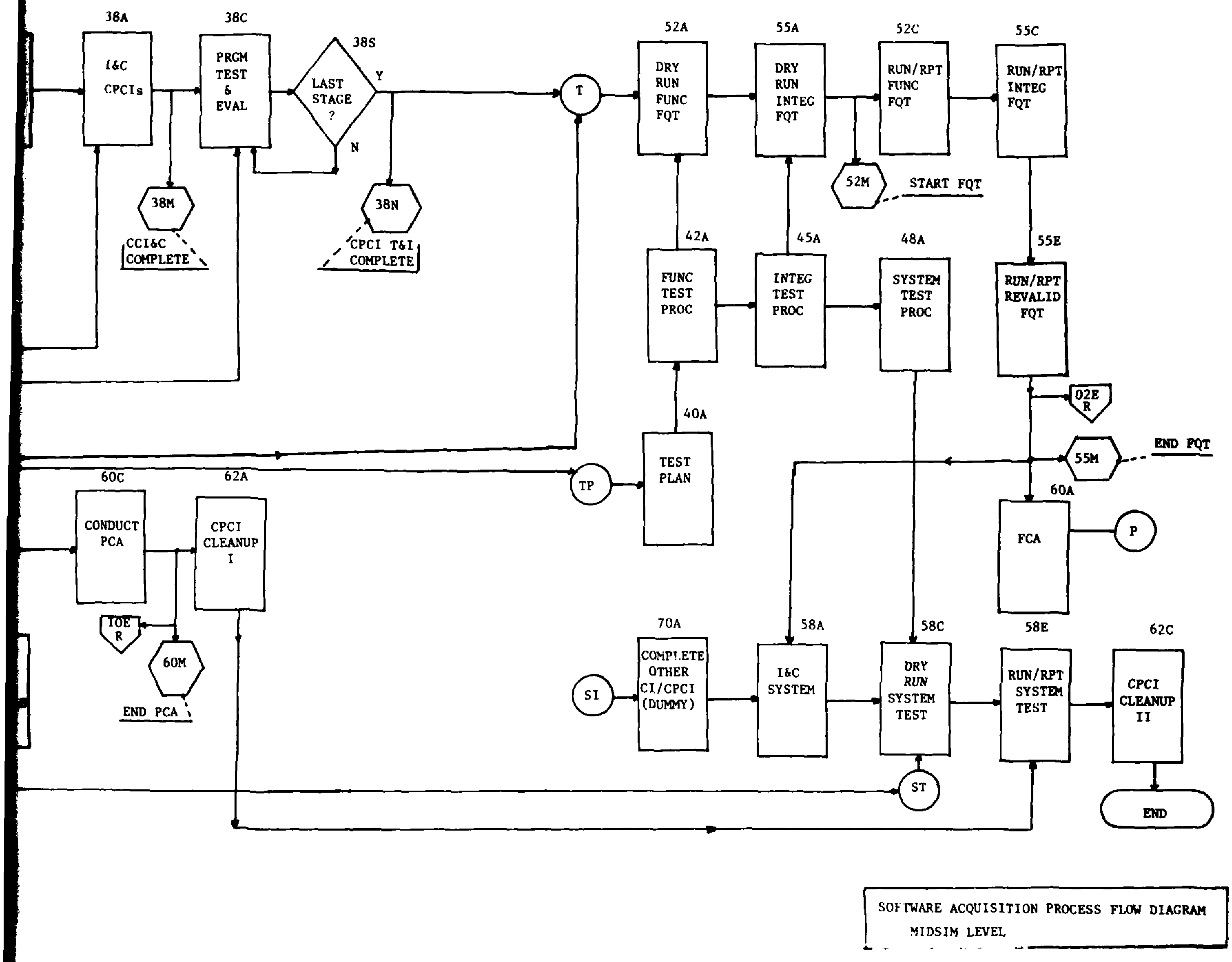


Figure A-2. Software Acquisition Process Flow Diagram MIDSIM Level

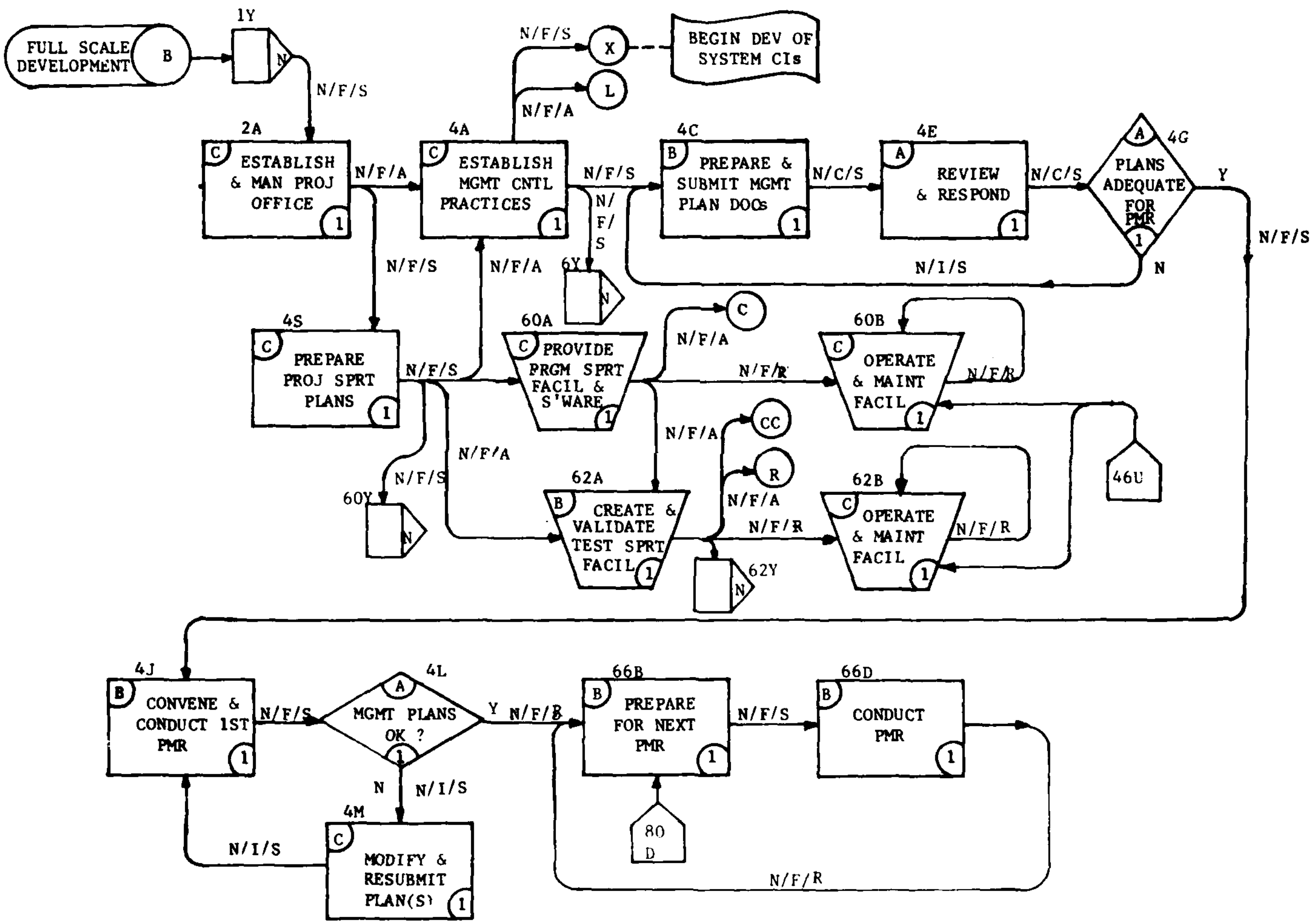


Figure A-3 (Rev.4). Software Acquisition Process Flow Diagram-LOSIM Level
Sheet 1 - Organization, Management, Support

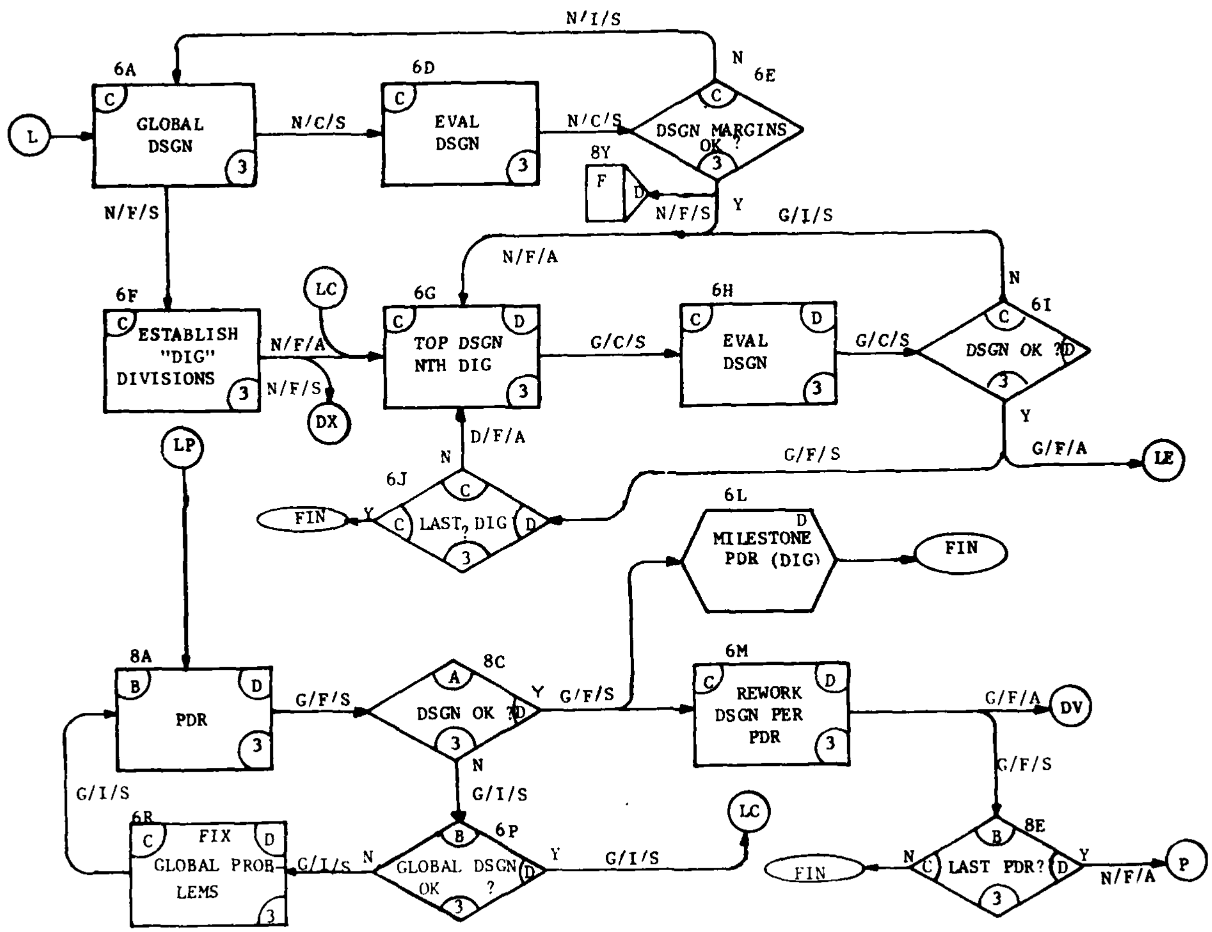


Figure A-3 (Rev.4). Software Acquisition Process Flow Diagram-LOSIM Level
Sheet 2 - Top Level Diagram Through PDR

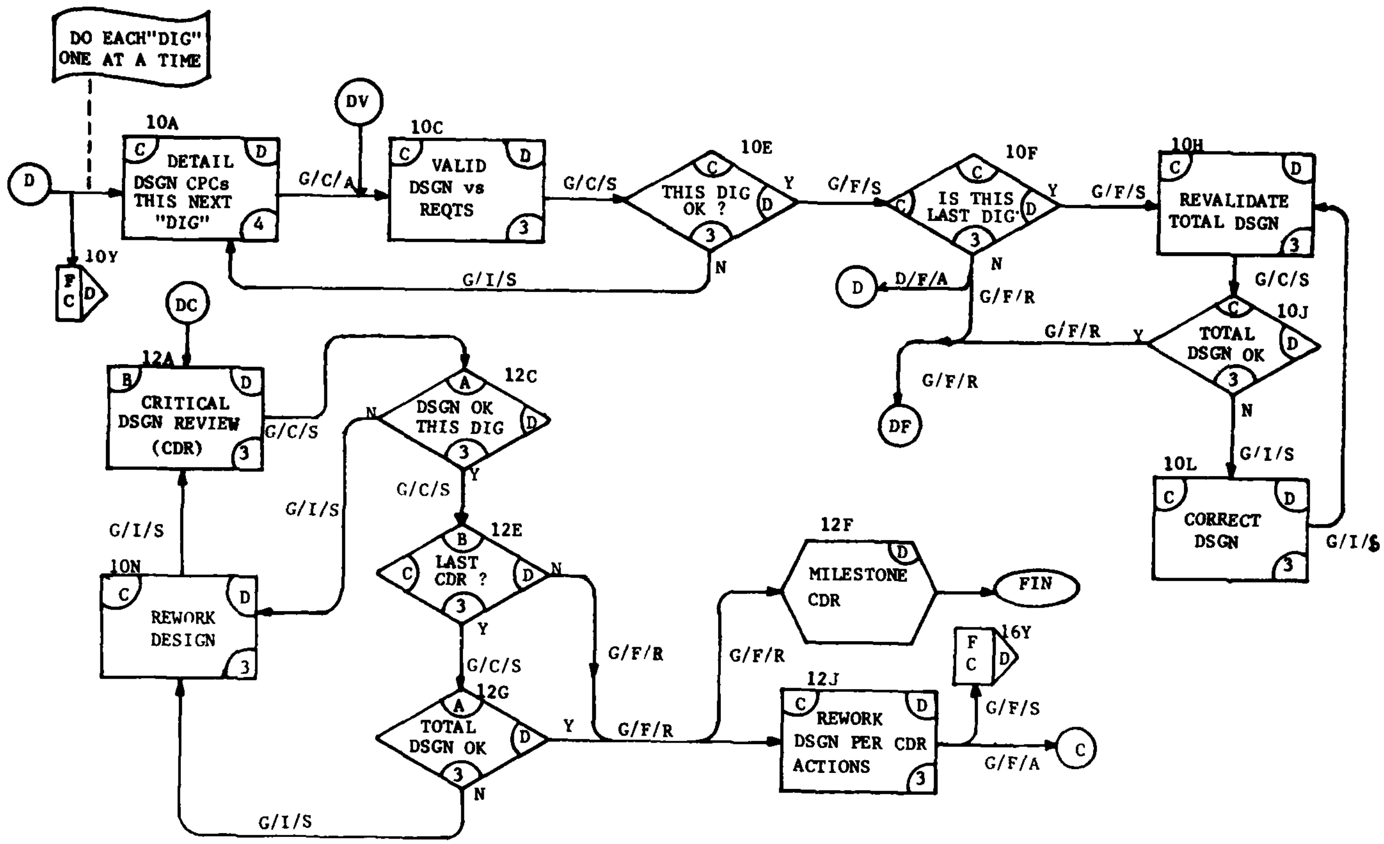


Figure A-3 (Rev.4). Software Acquisition Process Flow Diagram-LOSIM Level
Sheet 3 - Detail Design Through CDR

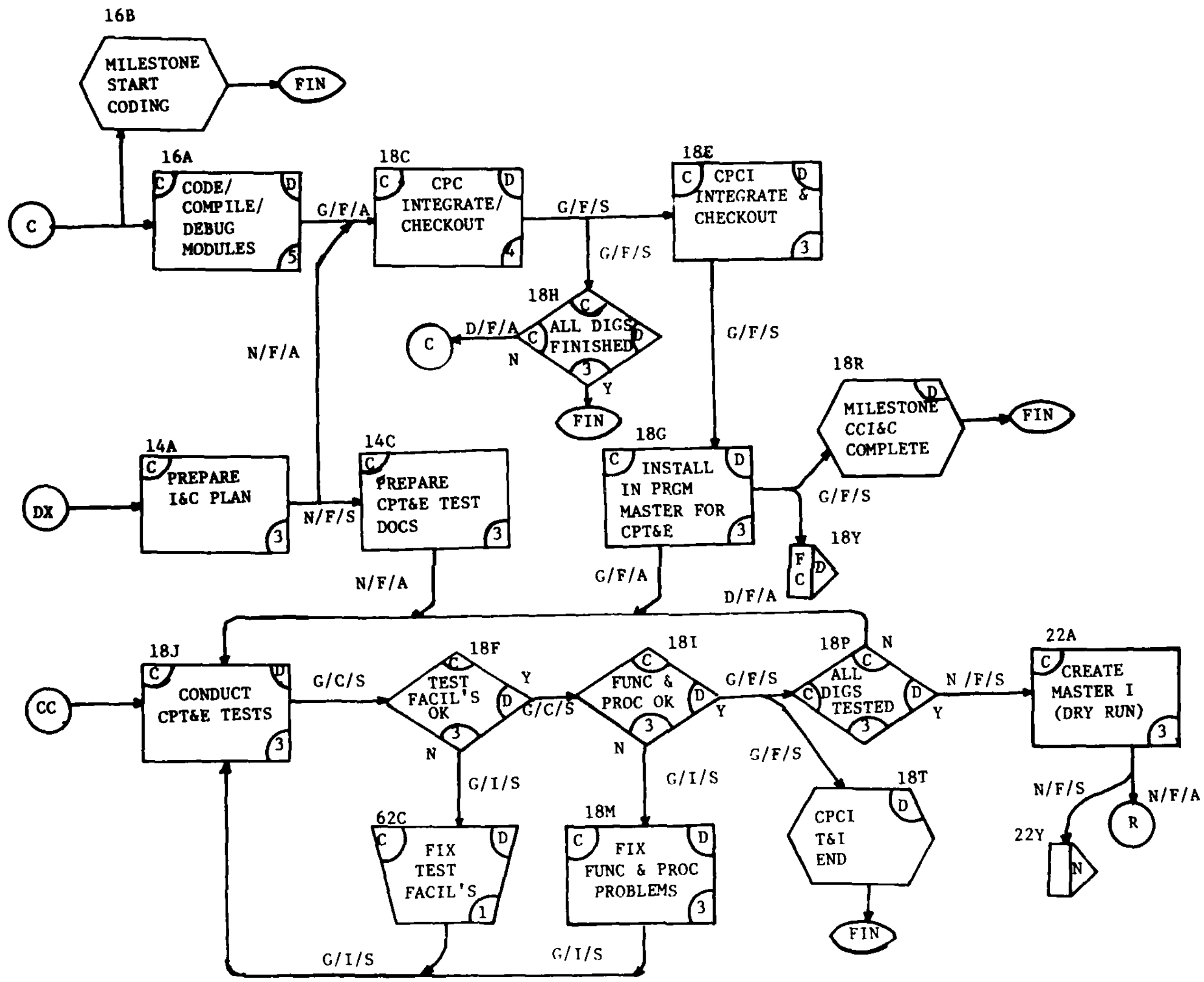


Figure A-3 (Rev.4). Software Acquisition Process Flow Diagram-LOSIM Level Sheet 4 - Code/Compile/Debug/Integrate/Checkout

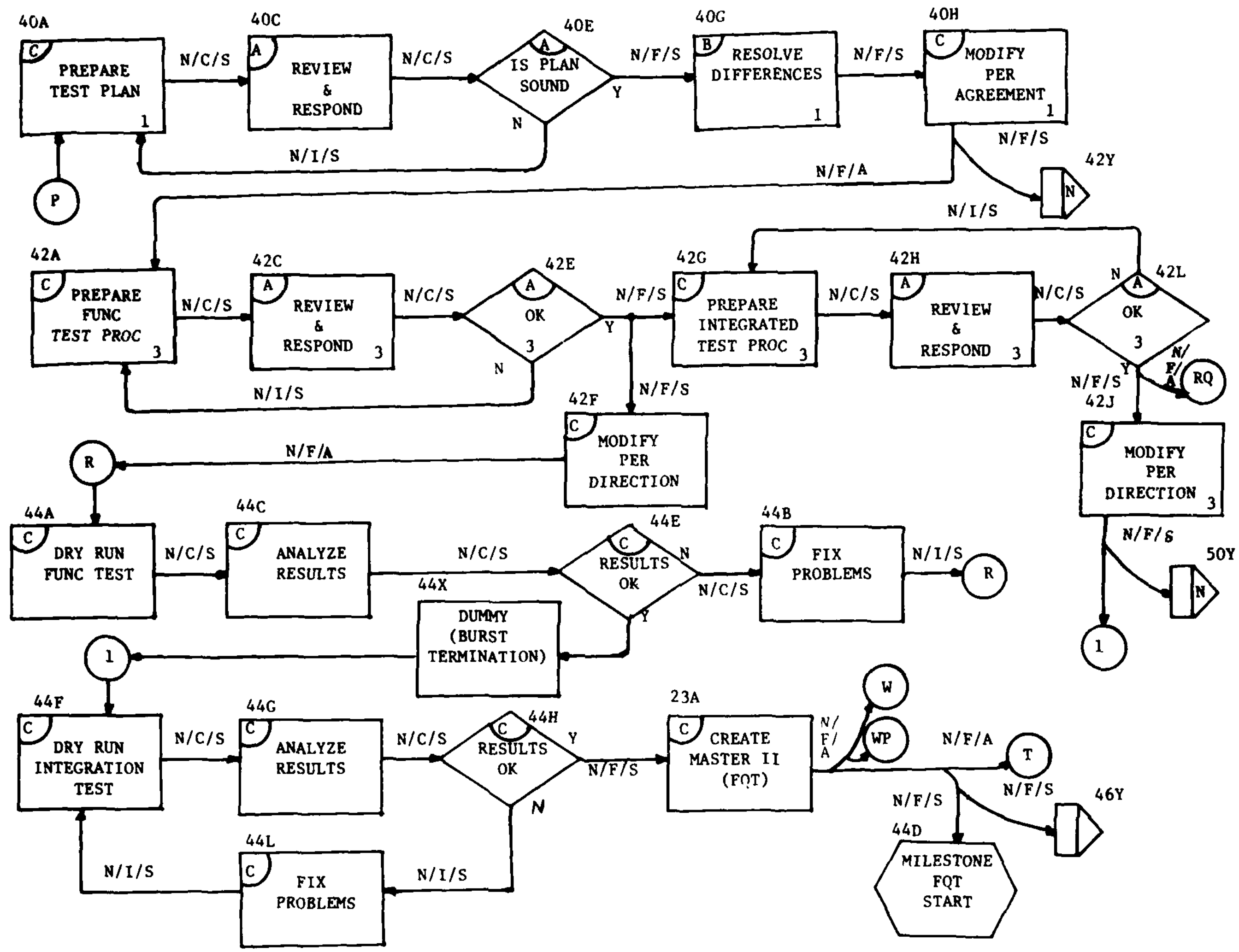


Figure A-3 (Rev.4). Software Acquisition Process Flow Diagram-LOSIM Level
Sheet 5 - Test Plan/Procedures/Dry Runs

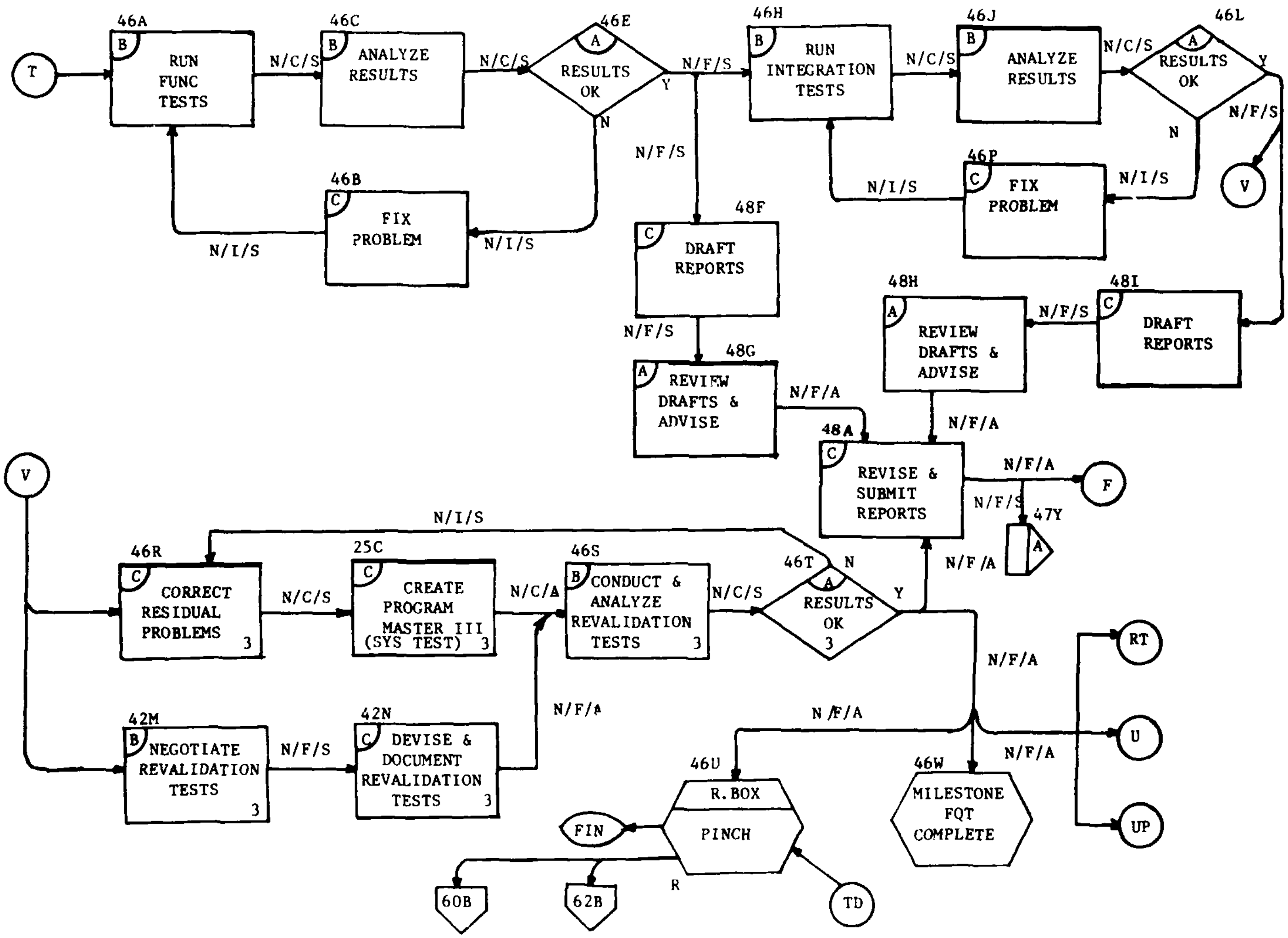


Figure A-3 (Rev.4). Software Acquisition Process Flow Diagram-LOSIM Level
Sheet 6 - Conduct Qualification Tests

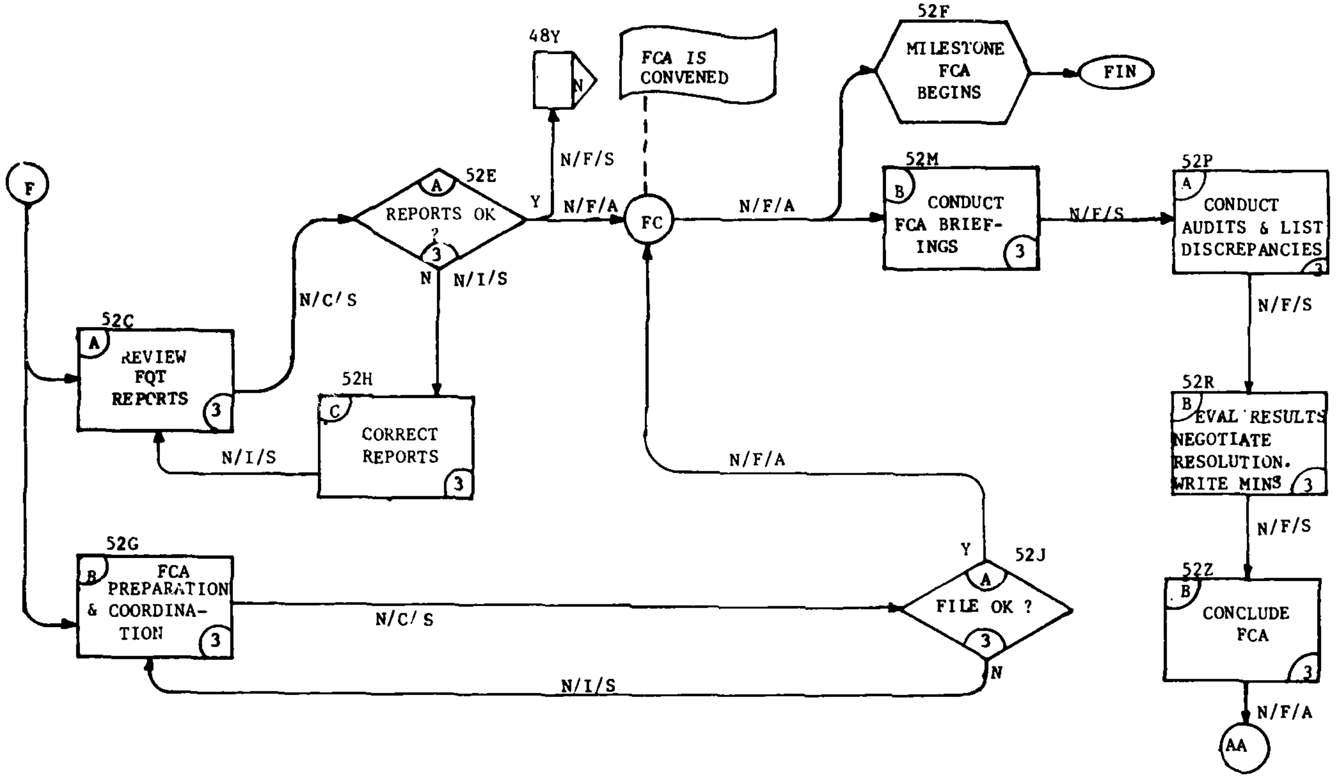


Figure A-3 (Rev.4.) Software Acquisition Process Flow Diagram-LOSIM Level
Sheet 7 Funtional Configuration Audit - FCA

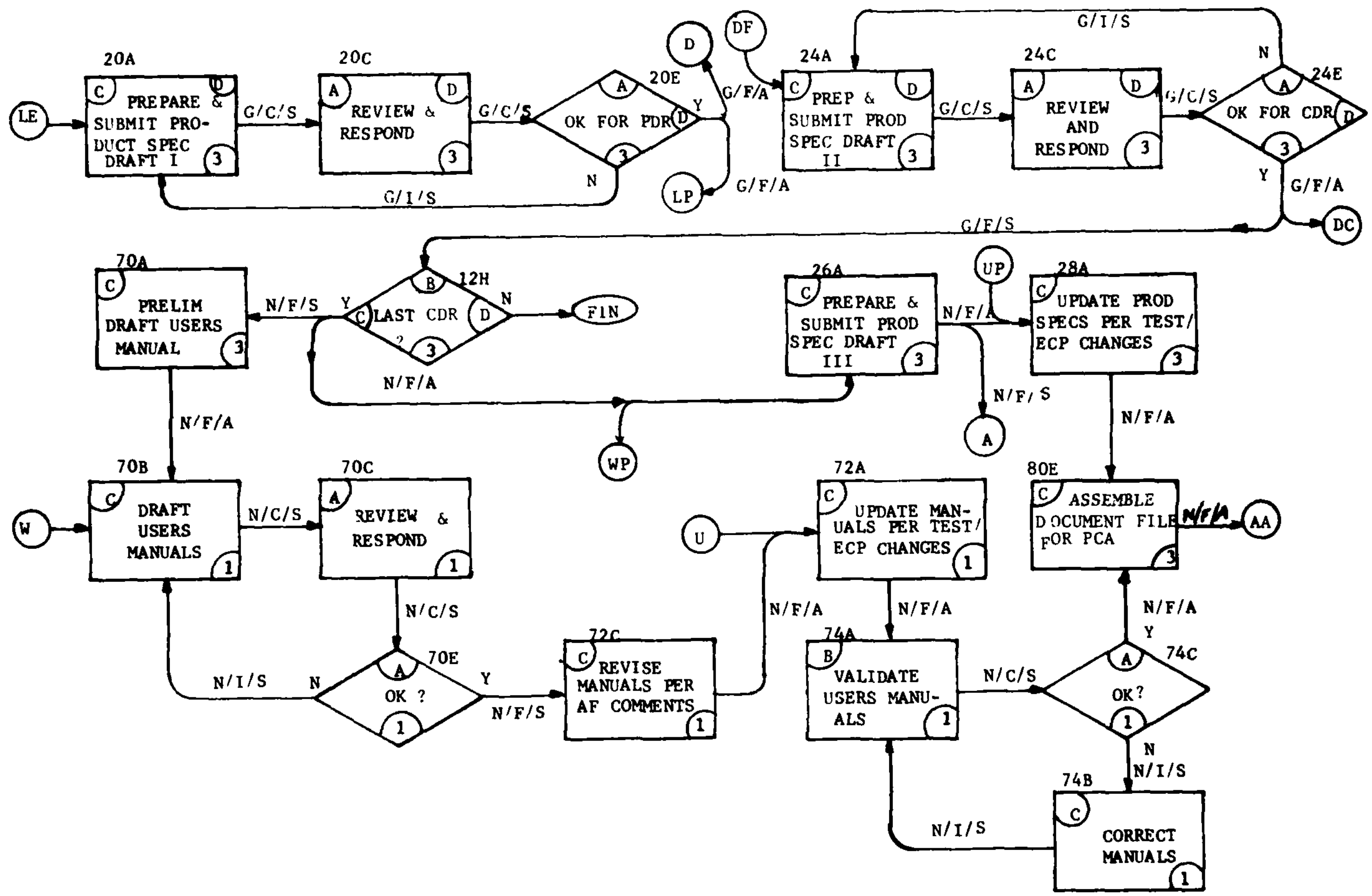


Figure A-3 (Rev.4). Software Acquisition Process Flow Diagram-LOSIM Level
Sheet 8 - Product Specs/Users Manuals

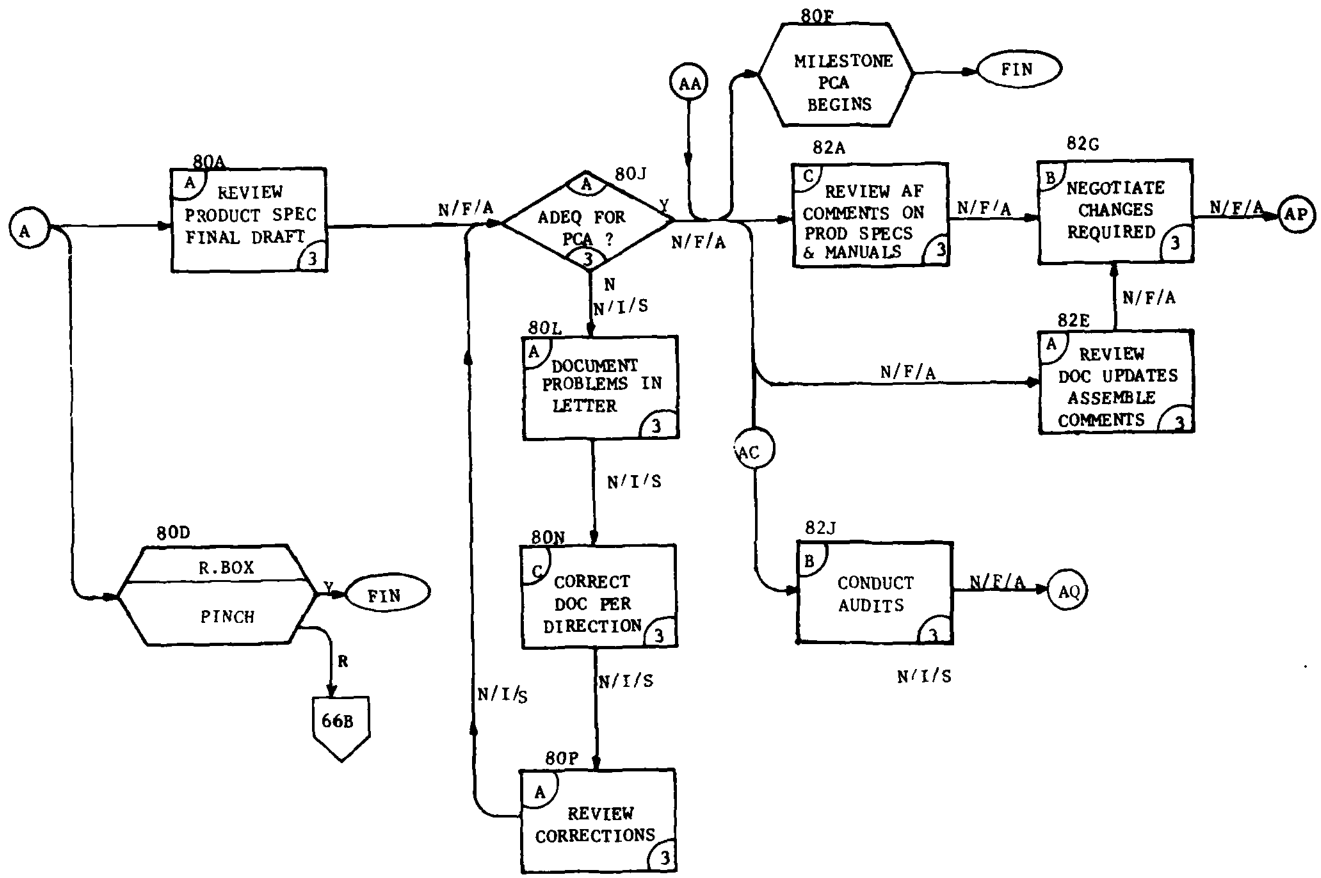


Figure A-3 (Rev.4). Software Acquisition Process Flow Diagram-LOSIM Level
Sheet 9 - Physical Configuration Audit - PCA

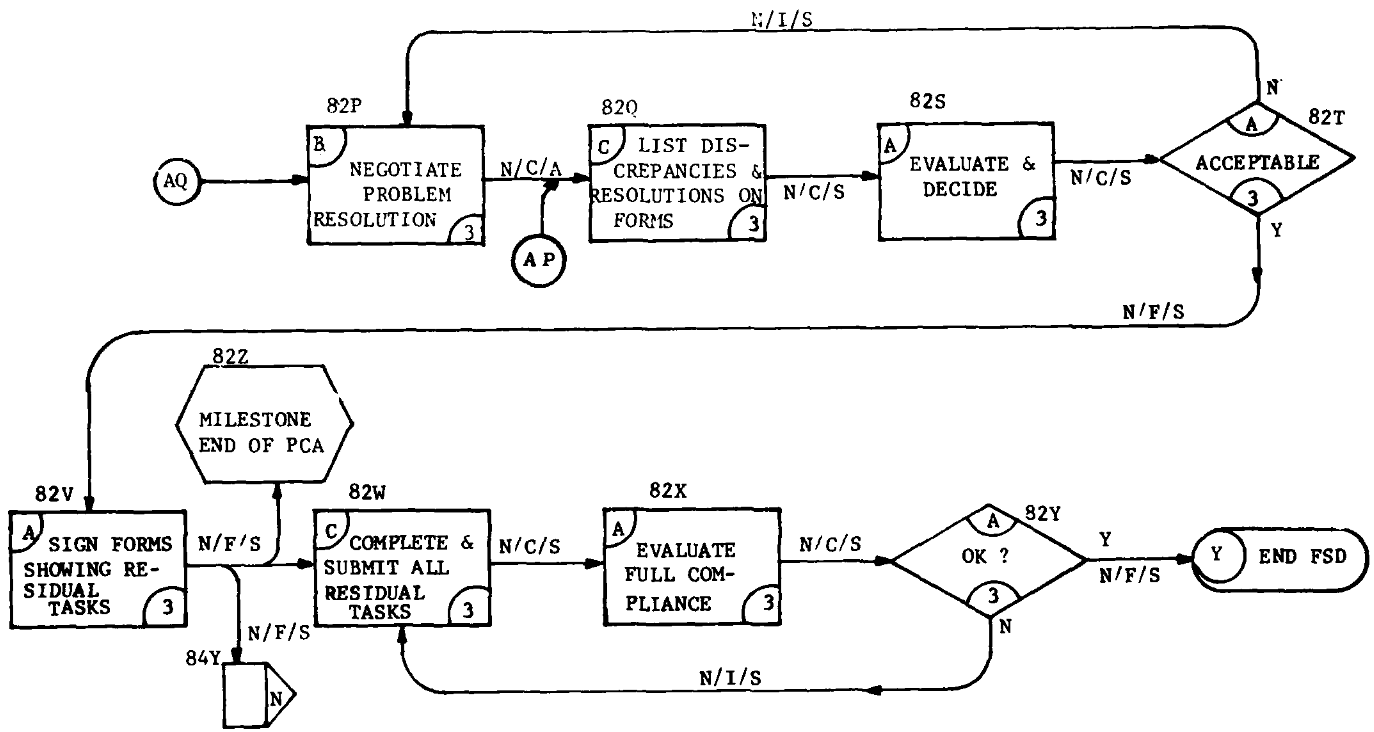


Figure A-3 (Rev.4). Software Acquisition Process Flow Diagram-LOSIM Level
Sheet 10 - Physical Configuration Audit - PCA Concluded

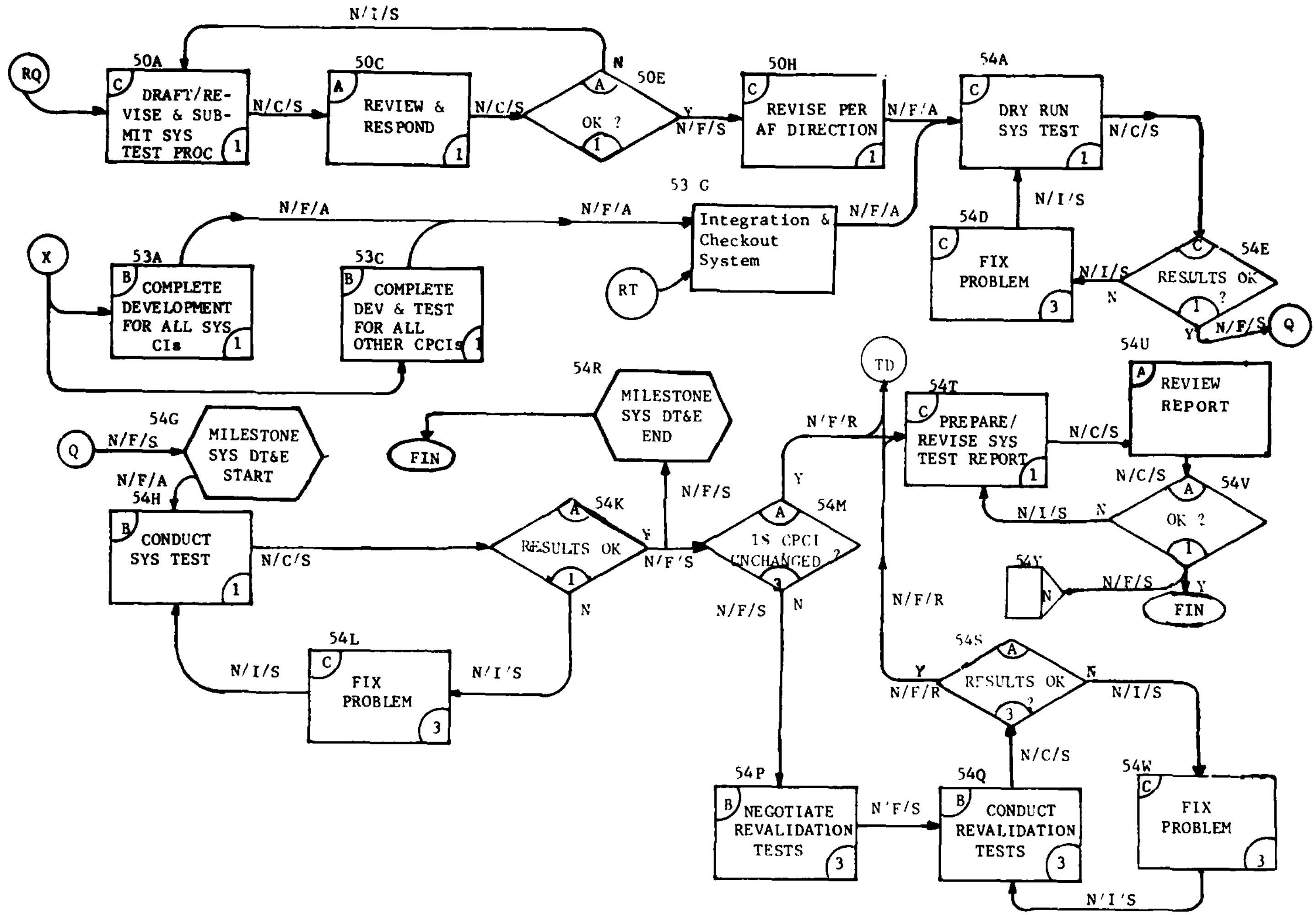
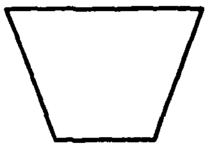


Figure A-3 (Rev.4). Software Acquisition Process Flow Diagram-LOSIM Level Sheet 11 - Systems Test

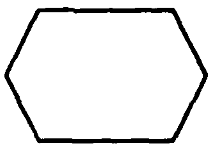
A-4.1 FUNCTION BOXES

A-4.1.1 Shapes

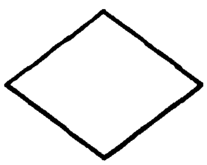
Mainstream Activity Box. Used only to represent mainstream activities (i.e., activities of principal importance).



Support Activity Box. Used to represent support activities. Both mainstream and support activities consume time and resources.



Special Event Box. Used for two functions: (1) The milestone box, marks a point and supplies a name for use in creating the milestone schedule. (2) A remote action box alters the action of another box at a remote location.



Decision Box. Depicts any procedure which selects between two mutually exclusive exits. By convention, these include no time or resource expenditures, which are included instead in preceding activities.

Personnel Box. Used to alter the manpower levels assigned to the project.

A-4.1.2 <u>Labels</u>

Each function box has a label, printed just above the box. Each label is a one- or two-digit number suffixed by a letter.

Figure A-4. Flow Diagram Notation

A-4.1.3 Features

Each box may contain several field designators, identified by corner positions within the box, as shown by letters X, D, Z and C.

X - indicates doer; i.e., the organization responsible for the function: A = government (e.g., Air Force), C = contractor, B = both.

D - indicates development integration group (DIG); blank indicates that the function is not divided into groups.

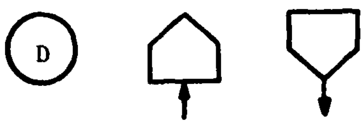
Z - indicates the level at which the work is conducted: 1 = system, 2 = segment, 3 = CPCI, 4 = CPC (computer program component), 5 = lower level module.

C - present on any decision box used as a counter.

A-4.2 AUXILIARY ELEMENTS

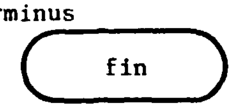
A-4.2.1 Shapes

Connectors



Used to indicate a specific point in the process flow. May be used to show connection between physically separated elements on flow diagrams. (A given label must apply uniquely to only one input point in the process flow.) The two shapes other than the circle are used to point to a box that is to be remotely actuated.

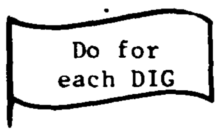
Terminus



Used to mark a start or end point of a process. When labeled "fin" it marks the end of the specific flow path.

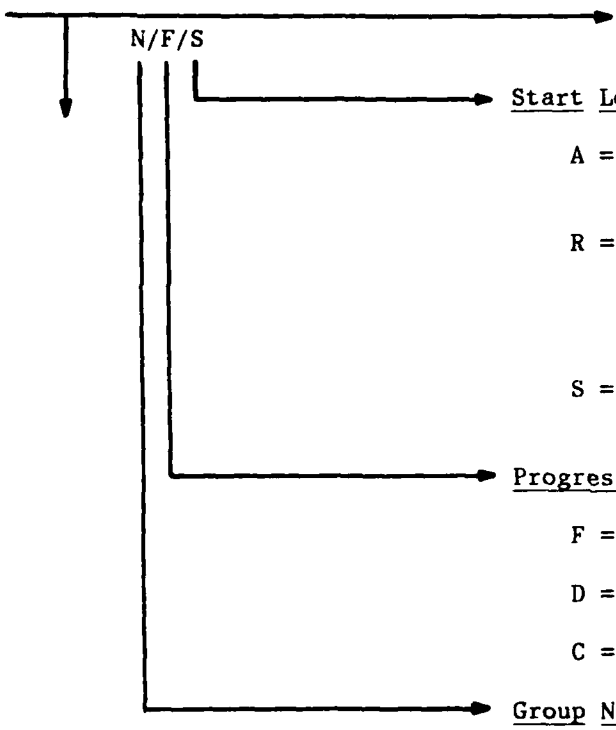
Figure A-4. Flow Diagram Notation (Continued)

Flag — Used to annotate flow diagrams.



A-4.3 LINES OF FLOW

The lines of flow have arrows to indicate direction, plus three alphabetic designators, as follows:



<u>Start Logic</u>

A = logical "AND' relationships (The input is necessary to start the box.)

R = logical "OR" relationship (Only one of these is necessary to start a box; inputs of other types may also be necessary, however.)

S = start immediately (This input by itself will start a box.)

<u>Progression Mode (PM)</u>

F = normal forward progression

D = iterative progression

C = continue progression mode (F or I) of

<u>Group Number Controller</u>

N = no group involvement

D = increment DIG number

G = retain predecessor's group number

Figure A-4. Flow Diagram Notation (Concluded)

Table A-1

Index to Figure A-3 Connectors and Boxes

Connector	A	AA	AC	AP	AQ	B	C	CC	D	DC	DF	DV	DX	E	E1	E2	F	FC
Input Sheet	9	9	9	10	10	1	4	4	3	3	8	3	4	12	12	12	7	7
Output Sheet	8	7,8	9	9	9	1	1,3,4	1	3,8	8	3	2	2		12	12	6	7

Connector	L	LC	LE	LP	P	Q	R	RQ	RT	T	TD	U	UP	V	W	WP	X	Y
Input Sheet	2	2	8	2	5	11	5	11	11	6	6	8	8	6	8	8	11	
Output Sheet	1	2	2	8	2	11	1,4,5	5	6	5	11	6	6	6	5	5	1	10

Box Number Index

Box No.	1Y	2	4	6	8	10	12	12H	14	16	18	20	22	23	24	25	26	28	40	42	42 M&N	44
Sheet	1	1	1	2	2	3	3	8	4	4	4	8	4	5	8	6	8	8	5	5	6	5

Box No.	46	46Y	47Y	48	48Y	50	50Y	52	53	54	60	62	62C	66	70	72	74	80	80E
Sheet	6	5	6	6	7	11	5	7	11	11	1	1	4	1	8	8	8	9	8

Box No.	82 A-J	82 P-Z	84Y	96
Sheet	9	10	10	12

SOFTWARE ACQUISITION PROCESS FLOW DIAGRAM - -
LOSIM LEVEL

Table A-2

Box To Amplification Notes Cross Reference List

Box Name	Notes #	Box Name	Notes #	Box Name	Notes #
2A	1	12A	8	25C	16d(4)
		C	8		
4A	2c	E	8	26A	7g, 13f
C	2f	F	8		
E	2e			28A	71, 16d(4)
G	2e	12G	8		
		H	18b	40A	6F, LLA,
4J	2g	J	8	C	11b
L	2g			E	11b
M	2g	14A	9d	G	11c
S	3B	C	10b(2)	H	11c, 20d
6A	4a, 6b	16A	9a	42A	12b(3)
D	4b	B	9a	C	12a(4)
E	4b			E	12a(3), 13a
F	4c	18C	9b	F	11a(3), 13a
		E	9d		
6G	5c, 6d	F	10d(1)	42G	12b(4)
H	5c	G	9g, 10b(1)	H	12b(4)
I	5c			J	12b(4)
J	5e	18H	9e	L	12b(4), 20d
		I	10d(2)		
6L	6a	J	10b(2)	42M	16d(4)
M	6f, 7b(4)	M	10d(2)	N	16d(4)
P	6e				
R	6d	18P	10e	44A	13c(1), 15c
		R	9g	B	13c(3), 15c
8A	5f, 6	T	10e	C	13c(2), 15c
C	6d			D	13f
E	6f	20A	5g, 7a		
		C	5d	44E	13c(3)
10A	5h, 7a	E	5h, 6, 7d	F	13e
C	6f, 7b			G	13e
E	7c	23A	13f,14a		
F	7e			44H	13f
		24A	7f	L	13e
10H	7e	C	7f	X	13d
J	7e	E	7f		
L	7e				
N	8				

Table A-2 (Concluded)

Box To Amplification Notes Cross Reference List

Box Name	Notes #	Box Name	Notes #	Box Name	Notes #
46A	14a(1), 16d(2)	54A	20e	80A	7h
B	16d(2)	D	20e	D	2h
C	14a(1), 16d(2)	E	20b	E	7i, 18d
		G	20b	F	7h(3)
46E	16d(2)				
H	16d(2)	54H	20f	80J	7h(3)
J	16d(2)	K	20f	L	7h(1)
		L	20f	N	7h(2)
46L	16d(2)	M	20f	P	7h(3)
P	16d(2)				
R	16d(4)	54P	20f	82A	19b, 20c
		Q	20f	E	19b, 20c
46S	16d(4)	R	20f	G	19b, 20c
T	16d(4)	S	20g	J	19c, 20c
U	3d, 16d(5), 20g				
W	6d(5), 7g	54T	20g	82P	19d, 20c
		U	20b	Q	19d, 20c
48A	14b, 16d(5)	V	20b	S	19e, 20c
F	14b	W	20f	T	L9E, 20C
G	14b, 16d(5)				
H	14b, 16d(5)	60A	3c, 9a	82V	19e, 20c
I	14b	60B	3d, 20g	W	19f, 20c
				X	19g, 20c
50A	20d	62A	3f, 10b(3)	Z	19e, 20c
C	20d	B	3d, 20g		
E	20d	C	10d(1)		
H	20e				
		66B	2h		
52C	16d(5)	D	2h, 12a(3)		
E	16d(5)				
F	17b	70A	7g, 18b		
G	17a	B	13f, 18c		
H	16d(5)	C	18c		
		E	18c		
52J	17a				
M	17b	72A	16d(4), 18d		
P	17c	C	18c		
R	17d				
Z	17e	74A	18d		
		B	18d		
53A	20a	C	18d		
C	20a				
G	20e				

Table A-3

Abbreviations

AF	Air Force
ADEQ	adequate
CCB	Configuration Control Board
CCI&C	code, compile, integrate & check
CDR	critical design review
CDRL	contract data requirements list
CI	configuration item
CPC	computer program component
CPCI	computer program configuration item
CPDP	computer program development plan
CPT&E	computer program test & evaluation
CRISP	computer resources integrated support plan
CRIT	critical
CTL	control
DEMO	demonstrate
DESCR	description
DEV	develop
DID	data item description
DIG	developmental integration group
DISCREP	discrepancies
DIST	distribute

Table A-3 (Continued)

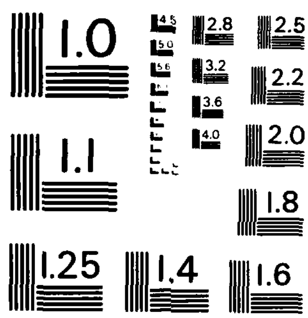
DOC	document
DSGN	design
ECP	engineering change proposal
EVAL	evaluate
FACIL	facility
FCA	functional configuration audit
FIN	end of this process flow diagram path
FQT	formal qualification testing
FUNC	functional
HIERARCH	hierarchial
HWARE	hardware
I&C	integration and checkout
IMPL	implementation
INTEG	integration
LOSIM	low simulation
LVL	level
MAINT	maintain
MGMT	management
MGR	manager
MISC	miscellaneous

Table A-3 (Continued)

ORG	organization
PCA	physical configuration audit
PCKG	packaging
PDR	preliminary design review
PRGM	program
PMR	program management review
PQT	preliminary qualification tests
PREP	prepare
PRF	problem reporting form
PROB	problem
PROJ	project
PSD	product specification document
PT&E	program test & evaluation
QA	quality assurance
REQT	requirement
REVAL	reevaluation
REVW	review
SCHED	schedule

AD-A145 413 DESCRIPTION OF THE SOFTWARE ACQUISITION PROCESS (SWAP) MODEL(U) MITRE CORP BEDFORD MA O SHAPIRO AUG 84 MTR-9005 ESD-TR-84-151 2/3

UNCLASSIFIED F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Table A-3 (Concluded)

SEMP	system engineering management plan
S'WARE	software
SPEC	specification
SPRT	support
STD	standard
SYS	system
SWAP	Software Acquisition Process (Model)
SZ	size
TECH	technical
TEMP	test and evaluation master plan
T&I	test and integration
VDD	version description document

Table A-4

Amplification Notes Table of Contents

Table A-5

Process Flow Diagram Amplification Notes

Note 1 - Project Initiation

Sht.	Box	Notes
1	2A	The assignment of key contractor personnel at the initiation of a project is generally a slow process. Each person selected for a new project usually has an existing assignment which must be transitioned to a successor; the successor may also need to transition his job to another, etc. Advance planning by the contractor helps in the personnel assignment process, but the uncertainties associated with the timing of the award on this contract (as well as with the contractors other pending bids) make startup a traumatic event that usually gets under way slowly.

Note 2 - Management Control Practices, Plans, and Reviews

Sht.	Box	Notes
1	4A	a. Management control practices (Box 4A) must be put into effect almost immediately on any major project that is to succeed. But management control preparation for each project will be unique because of considerations such as:
1	4C	
1	4E	(1) The degree to which the contractor's developmental organization (on this project) has, uses, and enforces a pre-existing set of practices.
1	4G	
1	4J	(2) The managerial and reporting requirements imposed by the contracting service (or agency), and the degree to which these are compatible with the contractor's normal practices.
1	4L	
1	4M	
1	66B	
1	66D	
9	80D	

(3) The attitudes and prior experiences of the personnel comprising the management team assembled for this project.

(4) The particular needs for this project, e.g., the tightness of schedule, the availability of money, the availability of qualified personnel, the technical risk, etc. imposed on the contractor; all of these against a background of these same factors on the totality of all projects currently in process (or expected shortly) at the contractor's facility.

b. Because the above conditions are highly variable from project to project, the duration and manning level for Box 4A will probably need to be individually adjusted for each application that takes place after the project requirements are well established and specific contractors are being considered, or have been selected. The activity itself, however, is generic and must be included in every simulation.

c. Once the planned practices are formulated (Box 4A), the main line development of the CPCI being simulated can begin (via connector L), as can also the development of other system CPCIs and CIs (via connector X).

d. The control plans must be formally documented within standard management documentation as shown in Box 4C. These management plans, which are shown to be a joint contractor/government activity, include:

- a system engineering management plan (SEMP)
- a test and evaluation management plan (TEMP)
- a computer resources integrated support plan (CRISP)

e. The above plans are usually developed during the validation phase activities, but they need to be updated by the contractor per note 2a above. The final proposed drafts (Box 4C) must be reviewed by the government (Box 4E) and found to be adequate (Box 4G) before they can be finalized; see Note 2g. The effort level assigned to the creation of these plans includes only that which is directly applicable to this CPCI.

f. The computer program development plan (CPDP) is generally addressed in the contractor's proposal. The activity included in Box 4C covers the rewrite and extension necessary before the CPDP can be put into contractual effect.

g. The management plans are frequently resolved at the first full-scale overall program management review (PMR), as shown in Boxes 4J, 4L and 4M. If they become urgent issues, they can be treated at a separate meeting. If not controversial they can be treated by mail and phone. The process diagram and box parameters can be adjusted to cover any expected case.

h. PMRs (Box 66D) are generally conducted on a periodic basis (e.g., monthly or bimonthly) throughout the entire contractual period. The duration of Box 66B is set to reflect the amount of time between PMRs. PMRs are shown here because the preparation and conduct activities consume considerable manpower on an intermittent basis and thereby can impact the development process. Note that a special event box (Box 80D) will cause the PMR activity for this CPCI to stop at the start of the PCA. It is common to include technical interchange "splinter sessions" on a variety of system components and activities; e.g., see Note 12a(3). The manning assigned to Box 66d is intended to reflect that portion of the PMR associated with the CPCI being modeled.

Note 3 - Developmental Support Facilities

Sht.	Box		Notes
1	4S	a.	The project support plan, which is formulated in Box 4S, must provide for:
6	46U		
1	60A		

1 60B (1) Developmental, management, and test support.

1 62A

1 62B (2) All support hardware and software; the plan must address:

- whether the hardware or software already exists, needs modification, or will be new;
- whether it will be purchased, or developed in-house;
- whether it is contractually specified and deliverable or not;
- whether it will be used in-plant or on-site, or both;
- how it will be checked out and installed and validated;
- its availability in time to satisfy needs.

b. The support facility needs defined in Box 4S must be designed, implemented, and installed. Separate boxes are shown for the program development (and management) support (Box 60A) and for test support (Box 62A). These facilities are treated separately because they are needed at different times, even though they may employ the same or different physical facilities. While support facility work is a generic activity on all projects, the time and effort consumed can vary widely, depending on the degree to which the contractor's existing facilities meet the needs of this project.

c. If any of these facilities is contractually specified and deliverable (as a separate configuration item) its development should be separately simulated (using SWAP) to determine its likely time of availability, so that appropriate duration values can be assigned to Boxes 60A and 62A; in this case no manning need be assigned to these boxes because the costs will be included in the separate simulation.

d. The operation and maintenance of the support facilities (Boxes 60B and 62B) are carried on by support personnel from the period where each becomes operational until it is no longer needed. Both boxes are turned off for this CPCI by Box 46U, when all the acceptance test activity on the CPCI is successfully completed.

e. If several different support facilities are planned (e.g., for in-plant vs. on-site, or for unit function tests vs. multifunctional tests, or simulation testing vs. live testing, etc.), additional boxes may be added to the diagrams.

f. Any non-trivial special (i.e., not deliverable) equipment or software that is to be used to support qualification testing, must be evaluated to ensure that it is valid for its intended use. As examples, a facility may be needed to emulate a nonavailable interfacing component (hardware or software) or to produce radar returns representing a flying aircraft, etc. This type of validation effort is included within Box 62A.

Note 4 - Global Design

Sht.	Box	Notes
2 2 2 2 1	6A 6D 6E 6F 6Y	The diagrams of the FSD process start with the assumption that the developmental specification (often referred to as the "B-5" specification, per MIL-STD-490, Specification Practices) has been prepared and accepted as the functional baseline for the CPCI. If B-5 Specification is not yet prepared or approved, appropriate boxes for this critical activity should be added to the diagrams. Once the baseline has been developed to an appropriate level, the product development of the CPCI can begin with the overall global design activity (Box 6A). This needs a team of senior system analysts and designers, as shown via Box 6Y. Their work begins with the synthesis of an overall design solution for this CPCI. This work includes the following kinds of activities:

a. System Synthesis (Box 6A)

(1) Establishing algorithmic solutions for all major functional requirements.

(2) Establishing a data base and data flow configuration.

(3) Establishing an overall task controller or executive.

(4) Dividing the whole CPCI into a set of product components (CPCS).

(5) Analyzing precision requirements for each function. These include decisions on whether fixed or floating point, or double precision, etc., arithmetic, is to be used.

(6) Establishing the total storage needs for each storage medium that is to be used, and a storage budget for each product component.

(7) Analyzing response time and buffering requirements and selecting methods of meeting these.

(8) Analyzing individual and collective timing requirements, and a timing budget for each product component (CPC).

b. System Evaluation (Boxes 6D & 6E)

After an overall solution is synthesized it must be evaluated to determine that all functional requirements will be met with adequate margins, that the storage requirements and availability are in balance, that adequate processing power is available, etc. When all needs appear to be safely met, which is unlikely until several passes through this iterative sequence, a storage and processing time usage budget will be established (per Notes 4a(6) and (8)). It will quantitatively assign resources to each major function (and its data) and retain some for contingencies and also set aside amounts needed to meet specified growth requirements.

c. After the global design is established, the set of product components that comprise the total specified capability may be clustered into developmental integration groups (DIGs). These product clusters, which allow product development to proceed in overlapping sequential stages, are generally used only on medium or large CPCIs. The task (Box 6F) requires that the DIG divisions cut along lines that minimize breaks in functional dependencies. The order of DIG development, which must begin with the implementation of the stages that provides essential service functions, is also established in Box 6F.

d. The division of a development into DIGs and the assignments of individual CPCs to these DIGs are unique for each project. The SWAP Model treats these assignments at a higher (generic) level. The user can indicate the quantity of DIGs and the relative size (in percent of total effort) of each. If the user does not provide a DIG-by-DIG size breakdown, SWAP defaults to a breakdown that is program size dependent. A 100K object instruction program, for example, will default to three DIGs, which are sized at 40%, 30%, and 30% of the total instructions.

Note 5 - High Level Design and Documentation

Sht.	Box	Notes
2	6G	a. In Box 6G, a high-level design is synthesized
2	6H	for all defined functions allocated to
2	6I	a particular DIG. This work includes
2	6J	the following types of tasks, which can proceed
2	8A	separately for each of the included functions:
8	20A	
8	20C	(1) An algorithm (or design concept) is
8	20E	established for each function contained
		within the DIG.

(2) The functional and interface requirements of each product component (CPC) are established, and each CPC is subdivided into a set of subfunctional components.

(3) The functional and interface requirements of each subfunction are defined.

(4) The timing and storage utilization needs for each subfunction (and its associated data) are estimated.

b. The synthesized design is evaluated in Box 6H to determine the adequacy of the design, e.g.:

(1) All applicable requirements in the CPCI specification that apply to the CPCs included in the DIG appear to be attainable.

(2) The total timing and storage utilization remains compatible with the budgets established in the global design; the budgets are re-allocated, if necessary (i.e., some components are assigned less, others more, etc.).

c. Generally, the design synthesis (Box 6G) and its evaluation (Box 6H) proceed iteratively (via Box 6I) until all functional requirements for the DIG appear to be safely accommodated. At this point, the computer resource utilization budgets are extended downward to a more detailed level for those functions included in the DIG.

d. As the DIG functions are subdivided and their subcomponents are defined, these must be documented (Box 20A) for review and coordination with the contracting agency (Box 20C). The documentation format, which is usually the same as that planned for the CPCI product specification (also known as the C-5 Specification, per MIL-STD-490, Specification Practices), will also be reviewed for compliance with the data item description (DID) requirements. If the documentation is not adequate to support the PDR, Box 20E will take the NO exit, back to Box 20A. If it is adequate, the PDR (Box 8A) and detail design (Box 10A) can begin.

e. As each designer completes his documentation tasks, he becomes available to begin design on the next DIG via Box 6J.

f. As described in Note 5a. above, the high-level design activity consists of a number of concurrent activities, one for each function contained in the DIG. This type of concurrency is represented in the Model by the "burst" concept (see par. 3.5), which allows these activities to be reflected generically. Thus, each burst sub-box represents a different function (or group of functions) whose design proceeds separately through the network until the preliminary design review (PDR). The PDR (Box 8A), which ends the burst string, cannot proceed until all applicable functional designs (i.e. all burst sub-boxes) have completed the course.

g. Note that proper contractor management would itself review the design documentation to determine its completeness and adequacy before submitting it in the pre-PDR package. This iterative step is not shown separately. Instead, it is implicitly included within Box 20A, in order to reduce complexity in the diagram. If the contractor does not properly review this draft document, the government decision (Box 20E) is much more likely to require rework on the documentation.

h. Once the pre-PDR design documentation is found to be acceptable to the government (Box 20E) the PDR can begin. At that same time, the contractor can begin work on the detail design (Box 10A). Any such work before an acceptable PDR proceeds at increased risk to the contractor. Such advance work is often necessary, however, because some of the designers have become available for new assignments before the PDR is completed; it is not economic to keep them idle, and not practical to assign them to some other project. For these reasons, the Model allows Box 10A to begin as soon as the staff become available.

i. After the design is well underway, the contractor must begin to develop his concepts on how the delivered program is to be tested for conformance with the specified requirements. These test concepts will also be discussed at one or more of the PDRs. As indicated in Note 6f, the test formulations should be conceptually acceptable by the last PDR.

Note 6 - Preliminary Design Review (PDR)

Sht.	Box
2	6L
2	6M
2	6P
2	6R
2	8A
2	8C
2	8E

Notes

After the government has reviewed the high level of design documentation and found it adequate (Box 20E), the first (incremental) PDR is conducted (Box 8A). The contractor describes and elaborates on his design algorithms and shows that they can meet all the functional requirements that apply to the DIG functions. He also shows his time and storage budget revisions to show that the global design is still valid. From this review, several alternative findings can be obtained:

a. The design may be found to be generally "on-track" so that Box 8C takes the YES exit and milestone 6L is achieved. Even though the design is found to be satisfactory, it is not uncommon for the review effort to expose a number of detail problems that are then corrected via Box 6M. (See case f. below.)

b. The design may not meet all requirements, but the government may waive (or alter) the requirements to allow the design to proceed. This situation follows the same Box 8C exit as in a. above; waiver and ECP processing will be added to the SWAP Model at a later time.

c. The design may meet all specified requirements, but during the review, it becomes apparent that the system will not meet the user's needs. In this case, a cost ECP may be required. This situation is treated as in b. above.

d. The design can be found so inadequate that major rework is necessary. In this case, Box 8C takes the NO exit. If the problem is with the global design (Box 6P), Box 6R is entered. Notice that the path could have returned to Box 6A instead of to 6R. This latter path, which produces a long loop, was not used because such loops can cause network flow problems (closed loops or multiple exits) that cause user difficulties. For this reason SWAP uses special "fix" boxes rather than long loops.

e. If the PDR outcome was inadequate (as in d. above), but the global design (Box 6P) is not at fault, then the identified functional design is reworked by return to Box 6G. In both this case and case d., another PDR must be conducted after the contractor has corrected the problem.

f. After Box 6M completes, the action can continue at Box 10C via connector DV. When PDRs for all DIGs have been successfully completed, counter Box 8E will cause Box 40A (via connector P) to become eligible to start work on the test plan; see Note 11. The test plan work can start earlier, but usually the test formulation does not clarify until the preliminary program design is established; see Note 5i.

Note 7 - Detailed Design and Documentation

Sht.	Box
3	10A
3	10C
3	10E
3	10F
3	10H
3	10J
3	10L
8	12H
8	24A
8	24C
8	24E
8	26A
8	28A
6	46W
9	80A
8	80E
9	80J
9	80L
9	80N
9	80P

Notes

a. The detailed design process (Box 10A) begins with a high-level design that was documented per Box 20A (see Note 5d). Its objective is to define a structured set of modular components that will embody the higher-level design. The logic of each module must be expressed in a manner that allows it to be readily transformed into computer program language statements that exactly implement its defined design and interfaces. Also, any special (e.g., timing, storage, or precision) requirements must be stated in the design documentation.

b. The validation of the detailed design (Box 10C) requires that its logic be reviewed by persons other than those who implemented the detailed design. The review team considers whether the detail logic:

(1) Correctly reflects the higher-level source logic.

(2) Is unambiguous and provides for all cases that can arise.

(3) Does not impose any narrowing restrictions on the higher logic; or if it does, these are compatible with the specified functional requirements.

(4) Correctly reflects all design modifications that were obtained at the PDR (per Box 6M). This prevents the review from being conducted until after a successful PDR.

(5) Conforms to all special requirements, as described in Note 7a above.

c. If the review team finds problems (Box 10E), the design must be reworked and revalidated. Such iteration is a normal expectation, if the review is carefully conducted.

d. Whenever the detailed design has been completed for a DIG (Box 10F), detail work can begin on the next DIG, provided the high-level design documentation (per Box 20E) has been successfully completed for that next DIG.

e. When the detailed design for the last of the DIGs has been reviewed (Box 10F), the whole detailed design should be reviewed (Box 10H) to see if the individual DIGs are collectively consistent and all functions have been accounted for. In particular, the timing and storage budgets should be revised and compared with the global plan. If problems are found (Box 10J), they are corrected (Box 10L).

f. At this point, detailed documentation of the design (per Boxes 24A, 24C, and 24E), which is needed to support the critical design review (CDR), will follow the same procedures described in Note 5. All revisions to the high-level design occasioned by the detail design process, or because of requirements changes (ECPs, etc.). are to be reflected in the pre-CDR document submission; i.e. the documentation is to be up to date and consistent at all levels.

g. When the pre-CDR documentation is complete for all DIGs, work can begin on the CPCI Users Manual (Box 70A); see Note 18. The full product specification draft can also be completed by the contractor (Box 26A) and submitted for government review. However, the completion of the full specification is not usually done until after the program has been implemented and tested and has become reasonably stable. This is reflected in the Model by connector WP, which does not enable further work on the specification until the start of FQT (Box 46W).

h. The product specification is given a quick review (Box 80A) to determine that the document is complete and suitable for beginning the PCA. If it is not (Box 80J), a corrective sequence is entered, as follows:

   (1) The major document deficiencies are explained via Box 80L.

   (2) These are corrected by the contractor (Box 80N) and resubmitted.

   (3) The revised document is reviewed (Box 80P) and its adequacy to support the PCA evaluated (Box 80J). The document must be ready before the PCA can begin (Box 80F).

i. The government continues to review (usually on a sampling basis) the product specification to determine that it is complete, consistent and reasonably up-to-date. It accumulates and filters the comments from the members of the review team. During this same period, the contractor must update the specification (Box 28A) to reflect all changes incorporated during the FQT plus those made necessary by ECPs. This revised version is then placed into the PCA support file (Box 80E).

Note 8 - Critical Design Review

Sht.	Box	Notes
3	12A 12C 12E 12F 12G	The conduct of each CDR comprises the same kinds of activities and decisions described for a PDR (per Note 6) except that the work now embraces the design down to a more detailed level. This is therefore a larger task so that more time is normally required.

12J 10N	The government does not necessarily review the total design expression (Box 12C). Usually, it examines changes that have been introduced into the design since the last review, and the reasons for them. It also <u>samples</u> the low-level design to see if it is complete and consistent with the higher level. In addition, any areas perceived to be difficult or risky can be followed at the detail level. Again, the resource requirements budget changes are reviewed and evaluated to determine whether contingency provisions are being depleted.

Note 9 - Computer Programming, Integration, and Checkout

Sht.	Box	Notes
4	14A 16A 16B 18C 18E 18G 18H 18R	a. Once the apparent adequacy of the detail designs for a DIG have been confirmed in the CDR, the main line programming activity can begin (Box 16A), provided the development facility (Box 60A) has also become available. While this is the preferred activity order, it is not uncommon for the contractor to begin coding sooner. While this earlier start is risky, its occurrence is often made necessary when programming personnel become available for this work before the CDR is satisfactorily concluded. The diagram shows the more conservative approach, but it can be drawn either way. At this point milestone 16B is reached. While each module must be coded, compiled, and debugged (in that order), these three programming activities usually proceed concurrently because of the iterative nature of this work. In Box 16A, all programmers are working independently in programming and checking out single modules or small groups of closely tied modules. b. In Box 18C, all the modules that comprise a CPC, which is usually a major functional component, are being combined and exercised until their ability to function together correctly is established. This low Level integration, which involves many independent small groups of

programmers, is complete when the coherently functioning CPC programs are entered into the CPCI program library.

c. Module integration must be planned ahead (Box 14A) in order to ensure that module operational dependencies will be satisfied as integration proceeds; this avoids or minimizes the need for special "test driver" code.

d. When several of the CPCs for a particular DIG have been placed into the CPCI library, the integration of these CPCs (Box 18E) can begin. Because this work involves a larger group of programmers, usually at least one from each CPC being integrated, this is treated as an "integrated" (as opposed to "fragmented") activity (see par. 5.3). The CPCs, which have been placed in the library, are joined together per the I&C plan (Box 14A). These components are exercised together and appropriately corrected until stable, coherent, and functionally correct operation is obtained.

   The above step is a highly iterative set of activities that involve problem detection, isolation, correction, and checkout on a gradually increasing segment of the program. Because this activity is so inherently and pervasively iterative, no need was seen for showing the process as a series of action/ decision loops.

e. As step b. completes and step d. gets underway, those programmers who are not directly involved in the CPCI integration can begin programming the modules that constitute the next DIG, via Box 18H.

g. After step d. is complete, a program master (Box 18G) is created. This master will initially contain all the components of the first DIG. As subsequent DIGs are completed it will contain the latest DIG along with <u>all</u> previously completed DIGS. In other words, it always contains the totality of all the completed DIGs. As each DIG is installed into the master, milestone 18R is reached.

Note 10 - Computer Program Test and Evaluation (CPT&E) or Preliminary Qualification Tests (PQTs)

Sht.	Box		Notes
4	14C 18F 18I 18J 18M 18P 18T 22A 62C	a.	The program master prepared in Box 18G must be systematically tested to determine that all the functions assigned to the completed DIGs operate correctly. The test procedures prepared in Box 14C can be informal (CPT&E) or official (PQT). In the former case, as shown on the diagram, they require no government approval or participation. In general, unless special circumstances indicate otherwise, the informal technique is preferred. As described below, these tests tend to be intricate and lengthy. If they must be formally documented and approved by the government, they will introduce considerable delay and cost into the process. Because these tests are conducted on components of an unfinished system, however, they provide <u>no</u> assurance that the results observed will hold for the final delivered product. The primary purpose of these tests for the government is the visibility they extend. They provide an indication of the current state of the program and, equally important, a feel for the amount of care and attention being shown by the contractor in finding and correcting the program problems, before the formal tests begin. These tests generally seek to determine whether the objectives of the program design have been realized by the programs. They therefore attempt to exercise all the code by establishing conditions that cause all logical paths to be followed, all loops to be exercised, etc. Unlike the integration and checkout work,

however, these tests are conducted primarily by test oriented personnel, usually the same persons who will later prepare and conduct the formal acceptance tests.

b. In order to begin the tests for a DIG (Box 18J) three conditions are required:

   (1) The program master is available per Box 18G.

   (2) The test procedure is available per Box 14C.

   (3) The test facility is available per Box 62A.

c. As each test is conducted, the expected results will be obtained or the test can fail for a number of reasons:

   (1) The test facility can behave differently than expected, per Box 18F.

   (2) The test procedure used may be incorrect in terms of setting up initial conditions, operating directions, expected results, etc.

   (3) The tested computer program may not operate correctly.

d. Any test run can fail from any combination of the above conditions. The logic to express all possible outcomes and the appropriate responses is complex (8 cases). In the interest of simplicity only two cases are treated, but the assigned probabilities of failure and the associated corrective efforts are selected to reflect the summation of all cases.

   (1) Test facility failure (Box 18F) and its correction (Box 62C) is taken as one (much less common) case because it involves different personnel types.

   (2) Problems with both the procedures and the programs are usually expected on the first running of a test. Boxes 18I and 18M show

this result. After several tries, however, the procedure corrections are usually complete before the program errors are fully corrected. The assignment of probability to Box 18I reflects the "inclusive or" case, while the manning/duration assigned to Box 18M reflects the likely mix of cases. Note, however, that program correction during this and all subsequent test phases is conducted per Note 15.

e. When each DIG passes its tests, milestone 18T is reached, and work can begin on the next DIG via Box 18P, but only after the program master for that DIG is ready (Box 18G).

f. After all DIGs have passed their tests, the program is complete and must be readied for acceptance testing. The first step is the creation of a new program master per Box 22A.

Note 11 - Test Plan

Sht.	Box	Notes
5	40A 40C 40E 40G 40H	The test plan is a formal document that defines the organizational breakdown of the planned formal (and sometimes informal) test effort and describes the objectives, methods, and each identified test or test component. The plan most commonly is formulated at the CPCI level, in which case it would be subordinate to a separate system or segment test plan. Depending on the system size and other considerations, the applicable plan might address instead, all software or a group of CPCIs. These latter cases are advantageous when there are strong functional interrelationships or commonality among the CPCIs. In these diagrams, since just one test plan is shown (Box 40A), it is assumed to be at the system or segment level. Whatever the plan level only the effort associated with the particular CPCI being simulated is included in the Box 40 data.

a. The contractor prepares the test plan per Box 40A. The CPCI portion of the plan should be based on the quality assurance (QA) sections of the system and CPCI specifications and should also reflect the content of any test concepts paper included with the statement of work. The work can begin at any time, but usually begins at about the time the last PDR is conducted; see Note 6f.

b. The government carefully reviews the document (Box 40C) to be sure that it is complete, clear, and sound. If it is too vague or unduly restrictive in scope, it should be sent back for major rework per Box 40E. The government must respond within the contractual time limits imposed via the contract data requirements list (CDRL), or the plan is automatically accepted.

c. When the submitted plan is basically sound but needs some specific changes, these can be negotiated (Box 40G) and then incorporated via Box 40H.

Note 12 - CPCI Formal Qualification Test (FQT) Procedures

Sht.	Box	Notes
5	42A 42C 42E 42F 42G 42h 42J 42L 66D	FQT procedures, as defined in the approved test plan, are prepared by the contractor and reviewed and accepted by the government. Two sets of procedures are indicated in the diagrams, as follows: a. Functional Tests (1) The functional test procedures, which are shown in Boxes 42A, 42C, and 42E, are normally performed first. These generally are used to show compliance with the requirements for each function defined in the CPCI specification. Each unit test generally concentrates on one or several closely related functions so that all unit requirements are exercised via a number of test cases, and shown to operate correctly.

(2) These test procedures are usually prepared by the contractor (Box 42A) and submitted for evaluation by the government (Box 42C) via a sequence of test group submittals. This keeps the level of effort uniform for both parties. This mode of submittal is treated generally in the model by the "burst" technique, see par. 3.5.

(3) When the government review finds the submittal to be "near enough to the mark" that it can accept the procedures intact, or (more commonly) with explicitly described changes, an accept (yes) decision is taken (Box 42E). The government requested changes, if any, are then made per Box 42F. Phone coordination is necessary to make this formal document exchange work. Sometimes, technical interchange meetings are also required; these are included in the Model as "splinter sessions" conducted during the program management reviews (PMRs) (Box 66D); see Note 2h. If any contractor submittal is too deficient for the above treatment, it is rejected for stated cause (via Boxes 42C and 42E), and the flow returns to Box 42A for rework.

(4) The time for government review is usually contractually established via the CDRL (see Note 11b); the time assigned to Box 42C must never exceed this value.

b. Multifunctional or Integration Tests

(1) The integration tests are prepared to demonstrate that all the functions specified for the CPCI can work together correctly and without adverse side effects. The emphasis in these tests is to exercise all the interfaces in a manner that closely follows the expected operational usage of the CPCI, including the effects of operator and machine error.

(2) These tests, when specified, will also include one or more load and capacity tests

to show that the CPCI can operate correctly under its fully defined maximum loads. These tests may also probe degraded mode operation.

(3) When specified, a carefully designed multi-functional test that can exercise all functions and their normal interactions in a concise capsule manner may be required. This test, which is to be used and maintained for the post-acceptance maintenance of the CPCI, is intended to provide a quick and economical means of determining that any program changes introduced subsequent to CPCI acceptance do not introduce undesirable side effects. This test is also used for CPCI qualification.

(4) The preparation (Box 42G), review (Box 42H), acceptance (Box 42L), and modification (Box 42J) methods are the same as described for the functional tests per Note 12a above.

Note 13 - FQT Dry Run

Sht.	Box	Notes
5	23A	a. Once the FQT procedures are approved (Boxes 42E, F) and the program master has been created (Box 22A), a "dry run" test must be conducted. Its purpose is threefold:
5	44A	
5	44B	
5	44C	
5	44D	
5	44E	
5	44F	
5	44G	
5	44H	
5	44L	
5	44X	

(1) It must verify that the FQT procedures are correct (e.g., provide all required inputs, in the order needed, using correct notation, etc.).

(2) It must verify that the program being tested operates correctly, i.e. that it accepts all inputs defined in the FQT procedures and responds with outputs that lie with defined limits, and that no adverse unforeseen results are produced.

(3) It serves to train the test operators and government observers.

b. While the tests are primarily conducted by and for the contractor, advance notice is given to the government so that its representatives can attend and observe the tests. This provides an opportunity for training the government test observers.

c. The individual functional tests are run first; the sequence for each test group or component is iterative, as follows:

(1) Run the test per Box 44A and determine whether all immediately confirmable behavior is correct. All noted problems are documented; see Note 15.

(2) Analyze all performance data that could not be evaluated at the time of test conduct (Box 44C). This activity includes any data reduction effort, the inspection and evaluation of the data for compliance with expected results, searching for evidence of adverse side effects, and documenting any suspected problems.

(3) If problems were noted in steps (1) and (2) above, they are treated (Box 44B) as described in Note 15, and the tests rerun via Box 44A.

d. The many different functional tests are generally run individually or in small groups; some runs being successful others not. This behavior is modeled by the burst concept, see par. 3.5. Box 44X, a dummy box is used to end the burst chain. It allows the multifunctional (integration) tests to begin only after the problems noted during the unit-functional test have been substantially corrected. A dummy box is required here because the current SWAP program does not allow a single box to be designated as both a burst "starter" and "ender."

e. The integration test sequence (Boxes 44F, G, H, L) follows the same logical procession described in c. above; only the nature and complexity of the tests are different.

f. After all the problems noted during the dry run of the FQT procedures have been fully corrected and verified (Box 44H), a new program master is created per Box 23A. This allows the "FQT start" milestone (Box 44D) to be reached and also allows documentation updating to begin via connectors W and WP to Boxes 26A and 70B.

Note 14 - FQT Conduct, Analysis, and Reports

Sht.	Box
6	46A
6	46B
6	46C
6	46E
6	46H
6	46J
6	46L
6	46P
6	48F
6	48G
6	48H
6	48I

Notes

a. Once the dry run tests have been completed, all known problems corrected (or temporarily set aside by per Note 16c) and a correct master program created (Box 23A), the FQT conduct (Box 46A) can begin. The tests are conducted as described for the dry run (Note 13c, e) except for the following:

(1) A government observer must witness the running of each formal test (Box 46A), unless this right is waived.

(2) The government observer may review or sample the analysis of the results (Box 46C).

(3) Under these conditions the test conduct proceeds slowly and meticulously so that each result is carefully checked by a full crew of operators, test analysts and observers. Problems that were missed during the dry runs are often found, as are other problems that may have been created during the earlier program correction activities.

(4) Each test begins with a pre-test briefing in which the contractor's test conductor describes the plan for the test, the operating positions, and personnel techniques for controlling the test pace so that time can be had for noting and confirming the existence of problems or

anomalies. In addition, any changes introduced in the previously approved formal test documentation are described, along with the reasons for these changes. The test does not begin until the cognizant government witness is satisfied that the test conditions are adequate for conducting the test.

(5) After each test, a post-test debriefing is conducted during which all test problems and/or anomalies noted by the operators and observers are described and discussed, and then entered into a test log. Later, after the recorded verification data has been examined by the contractor (and selectively by the government) any additional problems noted will be added to the problem list.

b. As each group of tests (burst sub-unit) runs successfully, the draft report can be documented (Boxes 48 F, I). The government reviews these drafts for content and format (Boxes 48G, H). The government's comments will be negotiated and then incorporated in the final test reports (Box 48A) after the revalidation tests are concluded per Note 16.

c. When FQT begins, the whole computer program and the approved test procedures are normally placed under configuration management, so that no changes can be introduced without the knowledge and consent of the government; see Note 16.

Note 15 - Program Maintenance During Test

Sht.	Box	Notes
		The test conduct and analysis, problem identification and correction, and test rerun sequence, as shown on the process flow diagram, is a simplification of actual procedures that are too detailed to reflect on

the diagram. While the representation of the sequence has been simplified, the assigned timing and effort parameter values reflect the following activity/decision pattern.

a. During the inclusive test period, usually starting with the CPT&E tests (Box 18J), three major ongoing and interrelated activities implement the process of progressively advancing the status of the software until its performance is acceptable:

   (1) Program testing (which is carried on by test personnel) methodically exercises the programs to determine their functional correctness and to identify problems.

   (2) Program correction (which is accomplished primarily by design and programming personnel) investigates and isolates problems, formulates and implements appropriate program changes, confirms that each change works properly, and installs the changed program into the CPCI library.

   (3) Program management (which is done primarily by system engineers and managers) evaluates problems to determine their urgency and difficulty. It decides the order or priority in which problems are addressed and the "clustering" of the changes into successive program masters or versions.

b. The typical sequence proceeds as follows:

   (1) It begins with test/analysis activities during which problems are initially identified and characterized on a problem reporting form (PRF).

   (2) The change/correction management group reviews each PRF to determine whether the problem is new or another instance of a previously reported problem.

(a) If the problem is not new, the PRF can be merged with the earlier report if it provides additional data.

(b) If it is new, it must first be evaluated:

- Is it a problem; does it violate requirements? If it is not, the PRF originator is informed.

- What is its impact: if it is immediate and serious (i.e., the test effort is strongly impeded), it is given a high priority for immediate attention.

- All other problems are given lesser priorities and entered onto a problem list. They are also assigned to specific CPCs, when applicable.

(c) The prioritized problem list guides the program repair activity and can also be used to evaluate program status.

(3) Design/programming teams, which are usually organized along program functional lines (e.g., per CPC), investigate each assigned problem, identify its causal mechanism, formulate and implement an appropriate correction, and then debug and check it out. The corrected program is placed in the program library (with annotation) and the change management group informed.

(4) As the program changes accumulate, they are selectively grouped for incorporation into an ongoing sequence of new program versions. Also, appropriate change documentation is released as informal version description documentation.

(5) Each new version is then tested by the program test group to determine that the initially reported problems have been cleared up, and that new problems have not been created. New or revised PRFs can be created by this activity.

(6) The change management group maintains the problem list by deleting those problems for which corrections have been confirmed by retest.

c. In the process flow diagram representation, work done per Note 15 a(1) above is included within the run test (e.g., Box 44A) and analyze results (e.g., Box 44C) activities. The fix problem work (e.g., Box 44B) accounts for the activities described in Notes 15 a(2) and (3).

d. Problems that arise during the official FQT (Boxes 46-) are also subjected to the change control procedures described in Note 16. This additional work is reflected in the effort levels assigned to these boxes.

Note 16 - Program Change Control During FQT

Sht.	Box	Notes
6	25C	a. The formal qualification of a CPCI is
6	46-	accomplished by means of a long series of formal
6	42M	tests that cumulatively establish that all
6	42N	specified requirements have been successfully
6	48A	satisfied. The validity of this process depends
6	48G	on:
6	48h	
8	72A	(1) Acceptable results being obtained on all
7	52C	tests.
7	52E	
7	52H	(2) No changes being made on the computer
		program (CPCI) during the whole formal test
		series (see b. below).

b. The actual conduct of FQT seldom (if ever) satisfies the above criteria. As described in Note 14, problems do occur during the tests (Boxes 46 E, L) and corrections (Boxes 46 B, P) are inserted; the CPCI master does not remain static during the test period. Any program changes inserted after some required capabilities have been confirmed by tests, however, could (inadvertently) adversely impact one or more of the confirmed capabilities. The validity of the earlier formal tests would be consequently undermined.

c. At the end of the formal test conduct period, therefore, the program usually contains a number of mid-test corrections (sometimes in "patch" form) along with some uncorrected (bypassed) problems.

d. In order to deal with the mid-test change problem, the following change discipline is usually imposed on the formal test process:

(1) Once FQT begins, no changes may be made in the program master, without the notification and agreement of the government. Any permitted changes must be documented:

- code revisions (annotated source form)
- problems that are corrected (PRF number)
- intended functional impact of the change

(2) After each new mid-test program version is installed (Boxes 46B, P) it is used to confirm by retest (Boxes 46A, C, H, J) that the installed modifications work properly (Boxes 46E, L).

(3) After all the defined formal tests have been completely run, the contractor must correct (or obtain waivers or other dispositions

for) the bypassed problems (Box 46R) and create a new progam master (Box 25C) that includes all needed corrections, usually in deliverable (not patched) form.

(4) At the same time, the contractor creates a functional profile of the problems encountered and a structural profile of the program components (and modules) altered. These profiles are reviewed by the government and a set of "revalidation tests" are negotiated (Box 42M). These usually include selective rerun of parts of the original tests, but may also include new tests or changes in the original tests. The total content of the revalidation tests, including test procedures for any new or modified test, are documented by the contractor (Box 42N). When the retest documentation is acceptable to the government, the retesting is conducted and analyzed (Box 46S). If the results are not fully satisfactory (Box 46T), the correction string (Boxes 46R, 25C, and 46S) must be repeated. When the results are correct, the product specification (Box 28A) (via the UP connector) and users manual (Box 72A) (via the U connector) must be updated to reflect the test findings. Also, the revised program master (Box 25C) can now be used for system test via the connector RT.

(5) When test results are acceptable, the FQT complete milestone (Box 46W) is reached and the support facilities are turned off (Box 46U). The previously reviewed test reports (Boxes 48G and 48H) are revised by the contractor (Box 48A) to reflect the revalidation test documentation and results. Also, the revised reports will respond to any appropriate government comments (Boxes 48G, H). The revised reports are then

submitted for governmental review (Box 52C) via connector F. If the reports are found to be inadequate (Box 52e), the reports must be improved (Box 52H) and resubmitted. Otherwise FCA can begin.

Note 17 - Functional Configuration Audit (FCA)

Sht.	Box	Notes
7	52-	a. Once all formal qualification tests and inspections have been conducted satisfactorily, preparations for the FCA (Box 52G) can begin. An FCA agenda is produced and coordinated while the contractor prepares a QA related document file for use by the FCA team. The content of the file is decided during the agenda discussion, and it is inspected for adequacy (Box 52J) just prior to the FCA conduct.
		b. As the FCA convenes, milestone 52F is reached. The contractor generally conducts a briefing (Box 52M) that identifies the documentary evidence that shows compliance with all QA requirements. Any known discrepancies or departures from requirements are put on record, and dispositions recommended.
		c. The government conducts audit reviews on the QA compliance documentation (Box 52P) to establish the following:
		(1) All specified requirements have been tested and found to be in compliance.
		(2) All interfaces have been checked per the above.
		(3) All problems observed during the tests have been corrected and rechecked.
		(4) All ECP changes have been incorporated and checked.
		(5) All PDR/CDR action items have been resolved.

d. Any deviations from requirements obtained in steps b. and c. above are negotiated (Box 52R) to obtain an agreed resolution: e.g., waiver, ECP, pre-DD-250 correction, post-DD-250 correction, etc.

e. The contractor prepares a plan for implementing the resolution negotiated in step d. and obtains government concurrence. The contractor also documents the FCA results and conclusions for input to the PCA. These actions conclude the FCA (Box 52Z).

Note 18 - Users Manual(s) and Other CDRL Items

Sht.	Box	Notes
8	70- 72- 74-	a. The users manual(s) as provided per these boxes includes any system operators manuals, including positional handbooks, that are contractually specified (e.g., via the CDRL and DIDs) and which are closely associated with the functions supported by the target CPCI. The same box sequence may also be used to represent other CDRL items, if any, that are closely associated with the CPCI.
		b. While work on the initial draft of the users manuals (Box 70A) can begin earlier, it does not generally begin until after the detailed design has been completed (Box 12H). The preliminary draft, which does <u>not</u> need to be the completed document, is needed to suppport the test effort, i.e., the writing of the FQT procedures and the conduct of the FQT dry run. The connection to these activities is <u>not</u> shown on the diagram, however, because the activities can proceed (though with less dispatch) using the informal user interface design descriptions that are prepared to support the design work.
		c. After the manual(s) have been used on the FQT dry runs, manual changes found to be necessary are incorporated into the now completed document(s)

(Box 70B). The manuals are then reviewed by the government (Box 70C). If they are inadequate (Box 70E) they must be redone via Box 70B. If they are generally acceptable, any conditions attached to the government acceptance are accomplished in Box 72C.

d. After the FQT is completed (via connector U) the manuals are updated based on FQT usage experience (Box 72A). The manuals are now validated (Box 74A) by an independent group who use the manuals in an operational manner. Any problems (Box 74C) are corrected (Box 74B) and the corrections validated (Box 74A). The validated manuals now become part of PCA support file (Box 80E).

Note 19 - Physical Configuration Audit (PCA) and Final Cleanup

Sht.	Box		Notes
9	80F		
9	82-	a.	The PCA provides the occasion for the final review and inspection of all deliverable products to determine that they are available, complete, and in satisfactory condition. In addition all product acceptance paper work is usually accomplished.
10	82		

The PCA is diagramatically shown to be conducted independently of the system test; see Note 20C. On many projects the software is not accepted until after the system test has confirmed its adequacy in a more realistic and natural environment. If this latter case is followed, the PCA start (Box 80F) would not begin until after Box 54V reaches the YES exit.

b. The product specification document (PSD) is a major concern at the PCA. As the PCA (Box 80F) begins, the government supplies its comments on the PSD for the contractor to review and evaluate (Box 82A). At the same time, the government inspects and evaluates (Box 82E) the latest revisions (from Box 28A) to the PSD, which reflect all FQT and ECP revisions; see Note 7i. The two parties then jointly review all residual comments (Box 82G) to reach concurrence on all PSD changes required.

c. During the same period, all deliverable products are inspected (Box 82J) (usually on a selective sampling basis) e.g.:

   (1) Does the PSD reflect the final tested version of the program?

   (2) Does the final tested program exactly reflect the delivered source program master; i.e., no patches or overlays are superimposed?

   (3) Are the version description document, the users manuals, and other CDRL submittals also consistent with the tested version?

   (4) Are all copies of delivered computer programs (source, object, and loadable) available in the required quantities, properly packaged, and correctly labeled, etc.?

   (5) Have all FCA cleanup items been correctly completed?

d. All discrepancies and questions raised by the audits are then jointly discussed (Box 82P) and their dispositions resolved. These agreed upon results are documented by the contractor (Box 82Q) into an action plan that lists: all problems that require action, and a plan for their disposition.

e. The government reviews the action list/plan for compliance with the negotiated agreement (Box 82S); if it does not comply (Box 82T), the above sequence is repeated. Once the list/plan is correct, the acceptance forms, which list all residual tasks that must be completed before final acceptance

(Box 82V), are signed. Also, all waivers and deviations are executed. The PCA is now over, per milestone (Box 82Z).

f. While step e. above completes the formal CPCI acquisition, there remain some residual tasks to be accomplished (Box 82W), e.g.:

(1) To fix, validate, and document problems that were not corrected before the PCA.

(2) To update, publish and deliver all applicable documents in their final form:

- Product Specification
- Users Manual(s)
- Version Description Document
- Special Test Procedures

This cleanup activity generally proceeds along with the work being done on the next phase of the software acquisition process.

g. The government must continue to inspect the products and witness and approve the tests (Box 82X) until all are found to be acceptable (Box 82Y). While this completes the CPCI acceptance, its use in the overall system (or segment) integration and test activity (see Note 20) can interact with the final acceptance.

Note 20 - System (or Segment) Test

Sht.	Box	Notes
5	40H	a. After the system (or segment) configuration items, including both hardware critical items (CIs) (Box 53A) and software (CPCIs) (Box 53C), have each individually passed FQT, they are all brought and joined together to be integrated and checked out (Box 53G). This activity can take place either in-plant or at a field site, depending on the circumstances. Boxes 53A and C are assigned no manning because their own development would be separately modeled. They are shown here to prevent the completion of system integration (Box 53C) until all system components have become available.
5	42L	
6	46U	
11	50-	
11	53-	
11	54-	
1	60B	
1	62B	

b. The system test is conducted to establish that all the components of the system, when properly interfaced and operated, meet all requirements defined in the system specification. The test also shows that system operation is free of adverse or destructive side effects. When possible, some or all of the system test should be conducted at a field site to show that proper function is obtained in a "live" environment. Because the system test is shown here as part of a CPCI acquisition, the manning and duration figures given for Boxes 50-, 53G, and 54- are intended to reflect just those portions that are directly associated with the target CPCI.

c. The diagram shows that the PCA (Boxes 82-) for the target CPCI can occur before, during, or after the system test; see Note 19A. When practical, it is better to conduct the PCA after the system test, so that all the formal delivered documentation and program masters, etc. can reflect all the changes introduced into the target software during the test activities; see Note 20F. In this case also, the "ownership" of the target CPCI during the system test remains with the contractor, as does the responsibility for correcting problems that arise during the test.

d. The preparation of the system test procedure can begin any time after the test plan has been approved (Box 40H), but usually the draft is begun at about the time the CPCI (and CI) FQT procedures have been acceptably completed (Box 42L). As the procedures are completed (Box 50A), usually in increments, they are incrementally submitted for review and approval by the government (Box 50C). If major inadequacies or omissions are noted (Box 50E), they are sent back for rework (Box 52A). Otherwise, any other problems that were noted during the review are documented by the government (Box 50C) and discussed with the contractor who corrects them per Box 50H. No government review of these corrections is shown, but if the corrections are not in conformance with the government's direction, contractor rework would be indicated.

e. After all system components have been integrated (Box 53G) and the test procedures are ready (Box 50H), a dry run (Box 54A) is conducted for three purposes:

   (1) To ensure that all system components work together properly.

   (2) To determine that the test procedures are workable and correct.

   (3) To provide a training experience for all test participants, both contractor and government.

   Any problems noted and any changes made (Box 54D) all take place in the program correction environment described in Note 15 and are subjected to change control per Note 16.

f. After the dry run has proceeded correctly, the formal system test, which repeats the conduct of the test procedures (and the subsequent analysis) but very slowly and carefully, is performed (Box 54H). Again any problems noted are corrected (Box 54K), subject to Note 15. Once the results are satisfactory (Box 54K), milestone (Box 54R) is reached. Any changes introduced into the target CPCI (Box 54M), subsequent to FQT acceptance, must be revalidated per procedures agreeable to the government (Box 54P). If the revalidation test and analysis (Box 54Q), reveals any problems (Box 54S), these are corrected (Box 54W).

g. Once the results are acceptable (Box 54s) the system test report can be prepared (Box 54T). At the same time, the remote activator (Box 46U) is reached (via connector TD). This will turn off the charges for the CPCI development and test support (Boxes 60B and 62B). The test report is then reviewed by the government (Box 54Q). If it is inadequate, rework is accomplished via Box 54T, otherwise the system test for the target CPCI is complete.

APPENDIX B

MODEL DEFINITION DATA

APPENDIX B

TABLE OF CONTENTS

TABLE OF CONTENTS (Concluded)

LIST OF TABLES

B.1 INTRODUCTION

This appendix incorporates and describes the tables that define the software acquisition process (SWAP) logic and parameter values to the simulator. These tables are input via a terminal to computer files and may easily be altered, as explained in appendix D. The simulator reads these files, reformats the tables, and interprets the revisions to develop simulation results. Within broadly defined limits the tables may be modified to represent more or less detail, differences in process logic, or revised parameter values. For example, the MIDSIM diagram (appendix A, figure A-2) can be reflected into these tables. Without needing revision itself, the simulator will interpret the modified tables and develop corresponding simulation results.

Table B-1, SWAP Model Network Linkage, is a tabular representation of the entire LOSIM process flow diagram (appendix A, figure A-3). Table B-2, SWAP Model Activity Box Parameter Data, contains the manning and duration parameter value estimates for the activities depicted in the LOSIM diagram. Table B-3, SWAP Model Decision Box Parameter Data, contains estimates of the decision outcome probabilities for all LOSIM decision boxes except counters. Table B-4, SWAP Model Counter and Special Event Box Parameter Data, contains the LOSIM flow diagram counter decision box limits and special event box parameters so far defined. Table B-5, SWAP Model Personnel Box Parameter Data, contains parameters that establish and adjust the levels of personnel assigned to the project. Table B-6, SWAP Model Subnetwork Titles, gives the names of the various subnetworks for labeling of output reports.

The columns of these tables, and the values that the data in each column may legitimately contain, are explained below.

B.2 TABLE B-1

Table B-1 represents the SWAP Model network. It must contain an entry for each box in figure A-3.

B.2.1 Box Data

a. Box Name: This is the box's label (see figure A-4).

b. Box Type:

A = mainstream activity box.

B = branching box (i.e., a normal decision box).

SOFTWARE ACQUISITION PROCESS MODEL

CS

TABLE B-1 NETWORK LINKAGE 12/10/82

GAP VERSION

MODEL 1 DATA BOX BURST

BOX		GENERAL DATA					SUCCESSORS				
NAME	TYPE	TRANS CLASS	DOER	GROUP	BURST	SUB NET	BOX	EXIT	GROUP	PROG. MODE	START LOGIC
01Y	P	GI	C	N	N	0	02A	Y	N	F	S
02A	A	GI	C	N	N	10	04A	Y	N	F	A
							04S	Y	N	F	S
04A	A	GID	C	N	N	10	04C	Y	N	F	S
							06A	Y	N	F	A
							06Y	Y	N	F	S
							53A	Y	N	F	S
							53C	Y	N	F	S
04C	A	GID	B	N	N	10	04E	Y	N	C	S
04E	A	GID	A	N	N	10	04G	Y	N	C	S
04G	B	GID	A	N	N	10	04J	Y	N	F	S
							04C	N	N	I	S
04J	A	GID	B	N	N	10	04L	Y	N	F	S
04L	B	GID	A	N	N	10	66B	Y	N	F	R
							04M	N	N	I	S
04M	A	GID	C	N	N	10	04J	Y	N	I	S
04S	A	GID	C	N	N	10	04A	Y	N	F	A
							60A	Y	N	F	S
							62A	Y	N	F	A
							60Y	Y	N	F	S
06A	A	DI	C	N	N	2	06D	Y	N	C	S
							06F	Y	N	F	S
06D	A	DI	C	N	N	2	06E	Y	N	C	S
06E	B	DI	C	N	N	2	06G	Y	N	F	A
							06A	N	N	I	S
							08Y	Y	N	F	S
06F	A	DI	C	N	N	2	06G	Y	N	F	A
							14A	Y	N	F	S
06G	A	DI	C	D	S	2	06H	Y	G	C	S
06H	A	DI	C	D	C	2	06I	Y	G	C	S
06I	B	DI	C	D	C	2	20A	Y	G	F	A
							06G	N	G	I	S
							06J	Y	G	F	S
06J	C	DI	C	D	E	2	06G	N	D	F	A
06L	M	DI	N	D	N	9					
06M	A	DI	C	D	N	2	08E	Y	G	F	S
							10C	Y	G	F	A
06P	B	DI	B	D	N	9	06R	N	G	I	S
							06G	Y	G	I	S
06R	A	DI	C	D	N	2	08A	Y	G	I	S
06Y	P	DI	C	N	N	0					
08A	A	DI	B	D	E	9	08C	Y	G	F	S
08C	B	DI	A	D	N	9	06L	Y	G	F	S
							06P	N	G	I	S
							06M	Y	G	F	S

Table B-1 (Continued)

08E	C	DI	B	D	N	9	40A	Y	N	F	A
08Y	P	DI	C	D	N	O					
10A	A	DF	C	D	C	2	10C	Y	G	C	A
10C	A	DF	C	D	C	2	10E	Y	G	C	S
10E	B	DF	C	D	C	2	10F	Y	G	F	S
							10A	N	G	I	S
10F	C	GK	C	D	C	2	10H	Y	G	F	S
							24A	N	G	F	R
							10A	N	D	F	A
10H	A	DI	C	D	C	2	10J	Y	G	C	S
10J	B	DI	C	D	C	2	24A	Y	G	F	R
							10L	N	G	I	S
10L	A	DF	C	D	C	2	10H	Y	G	I	S
10N	A	DF	C	D	N	2	12A	Y	G	I	S
10Y	P	DF	C	D	C	O					
12A	A	DI	B	D	E	9	12C	Y	G	C	S
12C	B	DI	A	D	N	9	12E	Y	G	C	S
							10N	N	G	I	S
12E	C	GK	B	D	N	9	12G	Y	G	C	S
							12F	N	G	F	R
							12J	N	[illegible]	F	R
12F	M	GK	N	D	N	9					
12G	B	DI	A	D	N	9	12J	Y	G	F	R
							10N	N	G	I	S
							12F	Y	G	F	R
12H	C	GK	B	D	E	O	26A	Y	N	F	A
							70A	Y	N	F	S
12J	A	DF	C	D	N	2	16A	Y	G	F	A
							16B	Y	G	F	A
							16Y	Y	G	F	S
14A	A	PI	C	N	N	3	14C	Y	N	F	S
							18C	Y	N	F	A
14C	A	PF	C	N	N	3	18J	Y	N	F	A
16A	A	PF	C	D	S	3	18C	Y	G	F	A
16B	M	GK	C	D	N	3					
16Y	P	PF	C	D	N	O					
18C	A	PF	C	D	C	3	18E	Y	G	F	S
							18H	Y	G	F	S
18E	A	PI	C	D	C	3	18G	Y	G	F	S
18F	B	PI	C	D	N	3	62C	N	G	I	S
							18I	Y	G	C	S
18G	A	PI	C	D	E	3	18J	Y	G	F	A
							18R	Y	G	F	S
							18Y	Y	G	F	S
18H	C	GK	C	D	E	3	16A	N	D	F	A
							16B	N	D	F	A
18I	B	PI	C	D	N	3	18M	N	G	I	S
							18P	Y	G	F	S
							18T	Y	G	F	S
18J	A	PI	C	D	N	3	18F	Y	G	C	S
18M	A	PI	C	D	N	3	18J	Y	G	I	S
18P	C	PI	C	D	N	3	22A	Y	N	F	S
							18J	N	D	F	A
18R	M	GK	N	D	N	3					
18T	M	GK	C	D	N	3					
18Y	P	PI	C	D	N	O					

Table B-1 (Continued)

20A	A	DI	C	D	C	7	20C	Y	G	C	S
20C	A	DI	A	D	C	7	20E	Y	G	C	S
20E	B	DI	A	D	C	7	20A	N	G	I	S
							08A	Y	G	F	A
							10A	Y	G	F	A
							10Y	Y	G	F	S
22A	A	PI	C	N	N	O	44A	Y	N	F	A
							22Y	Y	N	F	S
22Y	P	PI	C	N	N	O					
23A	A	PI	C	N	N	O	46A	Y	N	F	A
							44D	Y	N	F	S
							46Y	Y	N	F	S
							26A	Y	N	F	A
							70B	Y	N	F	A
24A	A	DFD	C	D	C	7	24C	Y	G	C	S
24C	A	DFD	A	D	C	7	24E	Y	G	C	S
24E	B	DFD	A	D	C	7	12H	Y	G	F	S
							24A	N	G	I	S
							12A	Y	G	F	A
25C	A	PI	C	N	N	O	46S	Y	N	C	A
26A	A	DID	C	N	N	7	28A	Y	N	F	A
							80A	Y	N	F	S
							80D	Y	N	F	S
28A	A	DID	C	N	N	7	80E	Y	N	F	A
40A	A	TID	C	N	N	4	40C	Y	N	C	S
40C	A	TID	A	N	N	4	40E	Y	N	C	S
40E	B	TID	A	N	N	4	40G	Y	N	F	S
							40A	N	N	I	S
40G	A	TID	B	N	N	4	40H	Y	N	F	S
40H	A	TID	C	N	N	4	42A	Y	N	F	A
							42Y	Y	N	F	S
42A	A	TFD	C	N	S	4	42C	Y	N	C	S
42C	A	TFD	A	N	E	4	42E	Y	N	C	S
42E	B	TFD	A	N	N	4	42G	Y	N	F	S
							42F	Y	N	F	S
							42A	N	N	I	S
42F	A	TFD	C	N	N	4	44A	Y	N	F	A
42G	A	TID	C	N	S	4	42H	Y	N	C	S
42H	A	TID	A	N	C	4	42L	Y	N	C	S
42J	A	TID	C	N	E	4	44F	Y	N	F	A
							50Y	Y	N	F	S
42L	B	GK	A	N	C	4	50A	Y	N	F	A
							42J	Y	N	F	S
							42G	N	N	I	S
42M	A	TID	B	N	E	4	42N	Y	N	F	S
42N	A	TID	C	N	N	4	46S	Y	N	F	A
42Y	P	TFD	C	N	N	O					
44A	A	TF	C	N	S	5	44C	Y	N	C	S
44B	A	TF	C	N	C	5	44A	Y	N	I	S
44C	A	TF	C	N	C	5	44E	Y	N	C	S
44D	M	TF	N	N	N	5					

Table B-1 (Continued)

44E	B	TF	C	N	C	5	44X	Y	N	F	S
							44B	N	N	C	S
44F	A	TI	C	N	S	5	44G	Y	N	C	S
44G	A	TI	C	N	C	5	44H	Y	N	C	S
44H	B	TI	C	N	E	5	44L	N	N	I	S
							23A	Y	N	F	S
44L	A	TI	C	N	N	5	44F	Y	N	I	S
44X	A	TI	C	N	E	5	44F	Y	N	F	A
46A	A	TF	B	N	S	5	46C	Y	N	C	S
46B	A	TF	C	N	C	5	46A	Y	N	I	S
46C	A	TF	B	N	C	5	46E	Y	N	C	S
46E	B	TF	A	N	C	5	46B	N	N	I	S
							46H	Y	N	F	S
							48F	Y	N	F	S
46H	A	TI	B	N	C	5	46J	Y	N	C	S
46J	A	TI	B	N	C	5	46L	Y	N	C	S
46L	B	TI	A	N	C	5	46P	N	N	I	S
							46R	Y	N	F	S
							48I	Y	N	F	S
							42M	Y	N	F	S
46P	A	TI	C	N	C	5	46H	Y	N	I	S
46R	A	TI	C	N	E	5	25C	Y	N	C	S
46S	A	TI	B	N	N	5	46T	Y	N	C	S
46T	B	TI	A	N	N	5	48A	Y	N	F	A
							46R	N	N	I	S
							46W	Y	N	F	S
							28A	Y	N	F	A
							72A	Y	N	F	A
							53G	Y	N	F	A
							46U	Y	N	F	A
46U	R	GK	N	N	N	O	60B	R			
							62B	R			
46W	M	GK	N	N	N	5					
46Y	P	TF	C	N	N	O					
47Y	P	TFD	C	N	N	O					
48A	A	TID	C	N	E	5	52C	Y	N	F	A
							52G	Y	N	F	A
							47Y	Y	N	F	S
48F	A	TFD	C	N	C	5	48G	Y	N	F	S
48G	A	TFD	A	N	C	5	48A	Y	N	F	A
48H	A	TFD	A	N	C	5	48A	Y	N	F	A
48I	A	TFD	C	N	C	5	48H	Y	N	F	S
48Y	P	TID	C	N	N	O					
50A	A	TID	C	N	C	6	50C	Y	N	C	S
50C	A	TID	A	N	C	6	50E	Y	N	C	S
50E	B	TID	A	N	C	6	50A	N	N	I	S
							50H	Y	N	F	S
50H	A	TID	C	N	C	6	54A	Y	N	F	A
50Y	P	TID	C	N	N	O					
52C	A	TID	A	N	N	5	52E	Y	N	C	S
52E	B	TID	A	N	N	9	52F	Y	N	F	A
							52H	N	N	I	S
							52M	Y	N	F	A
							48Y	Y	N	F	S
52F	M	TID	N	N	N	9					
52G	A	TID	B	N	N	9	52J	Y	N	C	S
52H	A	TID	C	N	N	9	52C	Y	N	I	S

Table B-1 (Continued)

52J	B	TID	A	N	N	9	52F	Y	N	F	A
							52G	N	N	I	S
							52M	Y	N	F	A
52M	A	TID	B	N	N	9	52P	Y	N	F	S
52P	A	TID	A	N	N	9	52R	Y	N	F	S
52R	A	TID	B	N	N	9	52Z	Y	N	F	S
52Z	A	TID	B	N	N	9	80F	Y	N	F	A
							82A	Y	N	F	A
							82E	Y	N	F	A
							82J	Y	N	F	A
53A	A	GK	B	N	N	O	53G	Y	N	F	A
53C	A	GK	B	N	N	O	53G	Y	N	F	A
53G	A	TI	C	N	S	6	54A	Y	N	F	A
54A	A	TI	C	N	C	6	54E	Y	N	C	S
54D	A	TI	C	N	C	6	54A	Y	N	I	S
54E	B	TI	C	N	C	6	54D	N	N	I	S
							54G	Y	N	F	S
54G	M	GK	B	N	E	6	54H	Y	N	F	A
54H	A	TI	B	N	S	6	54K	Y	N	C	S
54K	B	TI	A	N	C	6	54L	N	N	I	S
							54M	Y	N	F	S
							54R	Y	N	F	S
54L	A	TI	C	N	C	6	54H	Y	N	I	S
54M	B	TI	A	N	E	6	54P	N	N	F	S
							54T	Y	N	F	R
							46U	Y	N	F	R
54P	A	TID	B	N	N	6	54Q	Y	N	F	S
54Q	A	TI	B	N	N	6	54S	Y	N	C	S
54R	M	GK	B	N	E	6					
54S	B	TI	A	N	N	6	54W	N	N	I	S
							54T	Y	N	F	R
							46U	Y	N	F	R
54T	A	TID	C	N	N	6	54U	Y	N	C	S
54U	A	TID	A	N	N	6	54V	Y	N	C	S
54V	B	TID	A	N	N	6	54T	N	N	I	S
							54Y	Y	N	F	S
54W	A	TI	C	N	N	6	54Q	Y	N	I	S
54Y	P	TI	C	N	N	O					
60A	H	GK	C	N	N	10	60B	Y	N	F	R
							16A	Y	N	F	A
							62A	Y	N	F	A
60B	H	GK	C	N	R	10	60B	Y	N	F	R
60Y	P	GK	C	N	N	O					
62A	H	GK	B	N	N	O	62B	Y	N	F	R
							18J	Y	N	F	A
							44A	Y	N	F	A
							62Y	Y	N	F	S
62B	H	GK	C	N	R	O	62B	Y	N	F	R
62C	H	GK	C	D	N	O	18J	Y	G	I	S
62Y	P	GK	C	N	N	O					
66B	A	GK	B	N	R	10	66D	Y	N	F	S
66D	A	GK	B	N	R	10	66B	Y	N	F	R
70A	A	GID	C	N	N	8	70B	Y	N	F	A
70B	A	GFD	C	N	N	8	70C	Y	N	C	S

Table B-1 (Concluded)

70C	A	GID	A	N	N	8	70E	Y	N	C	S
70E	B	GID	A	N	N	8	72C	Y	N	F	S
							70B	N	N	I	S
72A	A	GID	C	N	N	8	74A	Y	N	F	A
72C	A	GID	C	N	N	8	72A	Y	N	F	A
74A	A	GID	B	N	N	8	74C	Y	N	C	S
74B	A	GID	C	N	N	8	74A	Y	N	I	S
74C	B	GID	A	N	N	8	80E	Y	N	F	A
							74B	N	N	I	S
80A	A	DFD	A	N	N	7	80J	Y	N	F	A
80D	R	GK	N	N	N	O	66B	R			
80E	A	GID	C	N	N	O	80F	Y	N	F	A
							82A	Y	N	F	A
							82E	Y	N	F	A
							82J	Y	N	F	A
80F	M	GK	N	N	N	9					
80J	B	DFD	A	N	N	7	80F	Y	N	F	A
							80L	N	N	I	S
							82A	Y	N	F	A
							82F	Y	N	F	A
							82J	Y	N	F	A
80L	A	DFD	A	N	N	7	80N	Y	N	I	S
80N	A	DFD	C	N	N	7	80P	Y	N	I	S
80P	A	DFD	A	N	N	7	80J	Y	N	I	S
82A	A	DID	C	N	N	9	82G	Y	N	F	A
82E	A	DID	A	N	N	9	82G	Y	N	F	A
82G	A	DID	B	N	N	9	82Q	Y	N	F	A
82J	A	DID	B	N	N	9	82P	Y	N	F	A
82P	A	DID	B	N	N	9	82Q	Y	N	C	A
82Q	A	DID	C	N	N	9	82S	Y	N	C	S
82S	A	DID	A	N	N	9	82T	Y	N	C	S
82T	B	DID	A	N	N	9	82V	Y	N	F	S
							82P	N	N	I	S
82V	A	DID	A	N	N	9	82Z	Y	N	F	S
							84Y	Y	N	F	S
							82W	Y	N	F	S
82W	A	DID	C	N	N	9	82X	Y	N	C	S
82X	A	DID	A	N	N	9	82Y	Y	N	C	S
82Y	B	DID	A	N	N	9	82W	N	N	I	S
82Z	M	GK	A	N	N	9					
84Y	P	GFD	C	N	N	O					

SOFTWARE ACQUISITION PROCESS MODEL

TABLE B-2: ACTIVITY BOX PARAMETER DATA

10:20 CS
12/10/82

MODEL 1 DATA

BOX				MANPOWER: CONTRACTOR					MANPOWER: AIR FORCE						DURATIONS				
NAME	TYPE	GROUP	SST	SYST	DSGN	PRGM	TEST	SPRT	DVLP	USE	SPRT	IT	FCTR		DAYS	IT	FCTR		WAIT
02A	A	N	P					1							10				
04A	A	N	P	2	2		2	1							10				
04C	A	N	P	2	2			2	4	1	1	5	5	5	12	3	1	1	
04E	A	N	F						6	1	1	5	5	5	10	3	1	1	5
04J	A	N	F	1.5	1		1	1	10	2	1	5	3	3	4	3	2	2	10
04M	A	N	F	1	1		1	1				5	5	5	3	6	3	3	
04S	A	N	P	1	1		2	1							15				
06A	A	N	P	2	6							10	5	3	18	3	2	2	
06D	A	N	F	3	6							5	3	1	5	5	2	2	
06F	A	N	P	2	2										8	-1			
06G	A	D	P	2	10	4						3	2	1	40	4	4	4	
06H	A	D	F	3	6							5	5	5	9	4	2	2	
06M	A	D	P	1	4			1				5	2	2	6	3	3	3	
06R	A	F	F	2	5							8	6	4	5	6	5	4	
08A	A	F	F	2	3	.6	.6	2	8	2	1	8	4	2	4	5	3	1	3
10A	A	D	P	3	20	5						6	4	2	56	3	2	1	
10C	A	D	F	3	4	1						5	3	2	9	6	4	2	
10H	A.	F	F	5	6	2						6	4	2	5	6	4	2	
10L	A	F	P	1	8							6	4	2	10	6	4	2	
10N	A	D	F	1	8							9	7	5	20	6	4	2	
12A	A	F	F	3	4	2	1.5	2	15	4	2	8	4	2	5	3	1	1	5
12J	A	D	P	2	6	1	1	2							15	-1			
14A	A	N	P	1.5	1.5		2	1				5	5	5	20	4	2	2	
14C	A	N	P	1.5	.3		3	1				5	3	1	30	5	3	1	
16A	A	D	P		5	30		1							60				
18C	A	D	P		5	20		1							28				
18E	A	D	F		5	12		1							36				
18G	A	F	F				1	1							3	5	5	5	
18J	A	D	P				4	1				10	6	4	15	6	4	2	
18M	A	F	F	1	2	4	2	1				9	7	5	5	10	10	10	
20A	A	D	F	1	6			2				5	2	1	18	5	3	1	
20C	A	D	F						8	2	2	6	3	2	18	5	3	1	5
22A	A	N	F			1	1	1							4				
23A	A	N	F			1	1	1							4	1	1	1	
24A	A	D	F	1	12			4				5	4	2	30	5	3	1	
24C	A	D	F						8	2	2	6	3	2	27	5	3	1	

Table B-2 (Continued)

25C	A	N	F			1	1	1						4	5	3	[illegible]
26A	A	N	P	2	3			4			4	3	2	40	5	3	1
28A	A	N	F	1	2.4			2			4	2	1	15	5	3	1
40A	A	N	F	2			3	2			6	3	2	25	6	4	2
40C	A	N	F						4	1	6	4	2	25	6	4	3
40G	A	N	P	2			3	1	4	1				6			
40H	A	N	P	2			4	1						10			
42A	A	N	F	2			6	2			6	4	2	40	6	4	2
42C	A	N	F						6	1	5	3	1	20	6	4	4
42F	A	N	P				3	1			8	4	2	10			
42G	A	N	P				4	1			8	4	4	30	7	4	2
42H	A	N	P						4					20	[illegible]	4	3
42J	A	N	P	1			3	1						10			
42M	A	N	P	1			3		3					5	1		
42N	A	N	P	1			3							8			
44A	A	N	F				8	2			10	6	3	8	7	5	3
44B	A	N	F	5	1	4	2	5						6	10	10	10
44C	A	N	P	1	2	2	8	2			10	6	3	12	8	6	4
44F	A	N	F				4	2			10	6	3	10	10	10	10
44G	A	N	F	1	2	2	4	1			10	6	3	18	8	6	4
44L	A	N	F		1	2	2	1			10	10	10	6	10	10	10
44X	A	N	F											[illegible]			
46A	A	N	P	1			6	2	4	1	8	5	3	6	8	6	4
46B	A	N	F		1	2	2	1			10	10	10	6	10	10	10
46C	A	N	F	1			8	2	4		7	4	2	12	8	6	4
46H	A	N	F	1			6	2	4	1	8	6	4	6	8	7	6
46J	A	N	F	1	1	2	4	1	4	1	8	6	4	15	8	6	4
46P	A	N	F	5	1	4	2	5			10	10	10	6	10	10	10
46R	A	N	F	1	2	4	2	2			5	3	2	10	5	3	1
46S	A	N	F	1			2	1	2		5	2	1	6	5	3	1
48A	A	N	F	1			3	1			5	4	3	6	5	3	1
48F	A	N	P				6	1						20			
48G	A	N	P						4	1				20			
48H	A	N	P						4	1				20			
48I	A	N	P				4	1						20			
50A	A	N	P	2	1		4	2			3	5	3	30	5	3	1
50C	A	N	P						4	2	6	5	4	15	5	3	2
50H	A	N	P	1			3	1						[illegible]			
52C	A	N	F						4	1	6	4	2	20	6	4	2
52G	A	N	F	1			3	1	3	1	5	4	2	5	4	2	0
52H	A	N	F	1			2	1			5	5	5	10	5	3	2
52M	A	N	F	3			1 2	1	4	2				1			
52P	A	N	F						4	2				3			
52R	A	N	F	1			2		2	1				2			
527	A	N	F						1					1			
53A	A	N	F	5					2					300			
53C	A	N	F	5					3					340			
53G	A	N	F	2	2	4	6	2						20	1		
54A	A	N	P	2			8	2			6	4	2	20	8	6	1

Table B-2 (Concluded)

54D	A	N	F	.5	1	4	2	.5				6	6	5	10	8	6	5	
54H	A	N	F	2			8	2	6	2		6	4	2	20	5	3	2	
54L	A	N	F	.5	1	4	2	.5				6	4	2	8	7	6	6	
54P	A	N	P	1			2		2	1					3				
54Q	A	N	F	1			3	2	3	1		8	5	2	6	7	5	2	
54T	A	N	F	2			4	2				6	4	2	25	6	3	1	
54U	A	N	F						4	1					20	6	4	2	
54W	A	N	F	.5	1	4	2	.5							6				
60A	H	N	F	.5	1	2		4							40				
60B	H	N	F					2							10				
62A	H	N	P	.5	1	2	1	4							40				
62B	H	N	F					2							10				
62C	H	F	F			1	3	1							10	5	3	2	
66B	A	N	F	.4	.5			1	2	1	1				5				30
66D	A	N	F	2.1	9		1	1	8	2	1				4				
70A	A	N	P	2	2			4							40	1			
70B	A	N	P	3	1.2			4				7	5	3	20	7	5	2	
70C	A	N	F						4	2	1	7	5	3	25	7	5	2	5
72A	A	N	P	2	1		3								10				
72C	A	N	P	2	.5										15				
74A	A	N	F	2			1	1	2	2		8	5	2	12	3	2	1	
74B	A	N	F	1			.5	.5				6	4	2	6	4	2	1	
80A	A	N	F						6		2				20				5
80E	A	N	F	1	1	1	1	3	2			5	4	3	5	5	3	1	
80L	A	N	P						2			5	5	5	10	5	3	2	
80N	A	N	P	2	2	2		4				5	5	5	20	5	3	1	5
80P	A	N	F						2	2	2	5	5	5	15	6	4	2	5
82A	A	N	F	2	2	1		1							3				3
82E	A	N	F						8	2	2				3				
82G	A	N	F	2	2				8	2	2				2				
82J	A	N	F	1	2	1	1	1	6	2	2				3				3
82P	A	N	F	1	1			1	2	1	1				2	5	2	0	
82Q	A	N	F	1	1			1							1	5	5	5	
82S	A	N	F						2	1	1				1				
82V	A	N	F						1						1				
82W	A	N	P	2	3	6	2	2				8	6	4	30	5	3	1	
82X	A	N	F						4	1	1	8	6	4	10	5	3	2	

SOFTWARE ACQUISITION PROCESS MODEL

9/10/82

TABLE B-3 DECISION BOX PARAMETER DATA GAP VERSION

MODEL 1 DATA

BOX NAME	BOX TYPE	BOX GROUP	YES EXIT PROBABILITIES 1ST	2ND	3RD	4TH FF.	WAIT
04G	B	N	80	100	100	100	0
04L	B	N	80	90	90	90	0
06E	B	N	20	40	60	100	0
06I	B	D	90	95	100	100	0
06P	B	D	50	100	100	100	0
08C	B	D	90	100	100	100	0
10E	B	D	60	80	93	97	0
10J	B	D	60	80	93	97	0
12C	B	D	80	90	90	95	0
12G	B	D	90	90	90	100	0
18F	B	D	95	100	100	100	0
18I	B	D	00	05	15	70	0
20E	B	D	90	95	97	100	0
24E	B	D	90	95	97	100	0
40E	B	N	50	80	90	100	0
42E	B	N	40	60	75	90	0
42L	B	N	40	60	75	90	0
44E	B	N	30	50	80	90	0
44H	B	N	25	35	60	80	0
46E	B	N	20	35	60	80	0
46L	B	N	25	40	75	90	0
46T	B	N	80	90	95	100	0
50E	B	N	0	45	70	80	0
52E	B	N	70	80	100	100	0
52J	B	N	75	95	100	100	0
54F	B	N	65	80	92	95	0
54K	B	N	45	65	85	93	0
54M	B	N	35	100	100	100	0
54S	B	N	30	60	90	100	0
54V	B	N	20	50	80	100	0
70E	B	N	50	70	90	100	0
74C	B	N	60	80	90	100	0
80J	B	N	80	90	100	100	0
82T	B	N	75	80	90	100	0
82Y	B	N	75	90	95	100	0

TABLE B-4. EVENT BOXES PARAMETER DATA
(MILESTONE, COUNTER, AND REINITIALIZER TYPES)

MODEL 1 DATA

BOX NAME	BOX TYPE	BOX GROUP	EVENT LABEL	PARAMETER	
06J	C	D			
06L	M	D	PDR		
08E	C	D			
10F	C	D			
12E	C	D			
12F	M	D	CDR		
12H	C	D			
16B	M	D	START CODING		
18H	C	D			
18P	C	D			
18R	M	D	CCI & C END		
18T	M	D	CPCI T & I END		
44D	M	N	FQT START		
46U	R	N		FO	1
46W	M	N	FQT - END		
52F	M	N	FCA - START		
54G	M	N	SYS DT & E START		
54R	M	N	SYS DT & E END		
80D	R	N		FO	1
80F	M	N	PCA - START		
82Z	M	N	PCA - END		

SOFTWARE ACQUISITION PROCESS MODEL

TABLE B-5: PERSONNEL BOX PARAMETER DATA

MODEL 1 DATA | 10/5/82 GAP VERSION

BOX NAME	BOX TYPE	BOX GROUP	TRIGGER	CONTRACTOR PERSONNEL SYST ENG	DSGN ENG	PGMR	TEST	SPRT	MGMT
01Y	P	N	N	2	2		2	2	3
06Y	P	N	N	2	4			1	
08Y	P	D	F	2	2	5			1
10Y	P	D	F	1	7	4		2	1
			C		-3	-4	1		
16Y	P	D	F		-5	24	1	1	
			C	-1	-1	-4			
18Y	P	D	F			6			
			C		-3	-20			
22Y	P	N	N	1	0	-8	3		
42Y	P	N	N				4		
46Y	P	N	N	-1	0	0	-1		
47Y	P	N	C	-1	-1	0	-4		
48Y	P	N	N	-2		0	-2	-3	-1
50Y	P	N	N	1	1	2	0	1	
54Y	P	N	N	-1	-1	-2	-3	-1	-1
60Y	P	N	N	1	2	3	1	3	1
62Y	P	N	N	-1	-2	-3	0		
84Y	P	N	N		1	3			-3

SOFTWARE ACQUISITION PROCESS MODEL

TABLE B-6 SUBNETWORK TITLES

MODEL 1 DATA

SUBNET#	TITLE
1	REQUIREMENTS DEFINITION
2	COMPUTER PROGRAM DESIGN
3	CODING THROUGH CPT&E
4	FORMAL TEST PREPARATION
5	FORMAL TEST CONDUCT/REPORT
6	SYSTEM TEST
7	PRODUCT SPECIFICATION PREP
8	USERS' MANUALS & CDRL ITEMS
9	FORMAL REVIEWS & AUDITS
10	SUPPORT & MANAGEMENT

C = counter decision box. This is similar to a type B box, except that the exit is determined by whether an incrementing counter has reached its limit.

H = helping box (i.e., a support activity box). See figure A-4.

M = milestone box. This provides for displaying milestones, at designated locations in the process flow.

P = personnel box. This will establish and adjust the manpower assigned to the project.

R = remote action box. This box provides for resetting counters, changing parameter values, and providing for as yet undefined future needs.

B.2.2 General Data

a. Transformation class (TRANS.CLASS)

A set of 15 transformation classes has been established to provide for combinations involving: activity type, documentation, and growth pattern.

The classes are identified by three letters "AGD" as follows:

Basic Activity (see 5.2 of text) (A): D = design; P = programming; T = test; G = general.

Growth Pattern (see 5.3 of text) (G): F = fragmented; I = integrated; K = constant.

Documentation Task (D): Letter "D" identifies documentation, if present; otherwise, it is omitted.

For example, "TFD" indicates a test activity, with fragmented growth, involving documentation.

Fourteen of these classes are derived as combinations of the seven basic activities and two of the growth patterns (i.e., F and I). The fifteenth class includes all boxes that have type K (constant) growth. This latter class is provided to allow any project-unique boxes to retain their assigned data.

b. Doer: This defines the agency or agencies assigned to perform the activity or to make the decision.

   A = government (e.g., Air Force)

   C = contractor

   B = both

   N = does not apply

c. Box Grp: This defines the box's membership (if any) within an integration group (see section 3.4.2 of text).

   D = developmental integration group (DIG)

   N = no integration group

d. Burst: This defines the box's status as to whether it is a burst box or not and if it is, its status within the burst group.

   N = non-bursting

   R = non-bursting and recurrent (see 3.6 of text)

   S = start of burst

   C = continue burst

   E = end of burst

e. Subnet: A user may assign the box to any one of up to 15 subnetworks by entering a number in the range 1-15 in this column. The simulator will develop aggregate timing and cost data for each subnetwork as well as for the entire network.

B.2.3 <u>Successors</u>

a. Box: This is the box name (see paragraph B.2.1a above) of the successor box. If a box has more than one successor, the data for its second and any subsequent successors are stored in corresponding columns of successive lines.

b. Exit: This is the box's exit used to reach this successor box.

Y = "YES" exit or single exit

N = "NO" exit

R = remote activation exit

c. Group: The box's group control parameter, used to maintain group (i.e., DIG) number continuity and incrementation during network flow.

N = no group involvement

D = increment DIG number

G = retain predecessor's group number

d. Progression Mode: A parameter used to indicate the direction of the box-to-box progression.

F = normal forward progression

I = iterative (i.e., backward) progression

C = continue progression mode of predecessor

e. Start Logic: Defines the combination of predecessors that must before this box may start.

A = "AND" relationship. This predecessor's completion is a necessary but not a sufficient condition for starting the box.

R = "OR" relationship. Completion of only one type R predecessor is necessary to start the box. Predecessors of other types, if specified, are also required.

S = Start immediately. This predecessor's completion by itself is sufficient to start this box. All iterative progression uses immediate start.

B.3 TABLE B-2

Table B-2 contains the parameter data for each activity box (box types A and H) in table B-1. Every activity box must have a table B-2 entry. Tables B-3, B-4, and B-5 contain the parameter data for the other box types.

B.3.1 Box Data

a. Box Name: This is the box's name, which must be identical to its table B-1 box name (see paragraph B.2.1a).

b. Box Type: This must be the same as the table B-1 entry's box type (see paragraph B.2.1b).

c. Box Group: Identical to the table B-1 entry's box group (see paragraph B.2.2c). See note d. for category F.

d. When box group is set to "F" (used only for activity boxes), it indicates that the box is in a DIG but the activity duration on each access is fixed, i.e., not altered to reflect the quantity of DIGs.

e. SST: This "successor selection timing" field enables each activity box in the Model to require "full" or "partial" completion before it flows; i.e., enables its successor boxes. "Partial" (P) boxes will "flow" after a defined percentage (initially set at 70%) of their assigned time durations has elapsed. "Full" (F) boxes will flow only after their total assigned time duration has elapsed.

B.3.2 Manpower

Manpower is subdivided into five categories of work for contractor personnel, and three for government personnel, as explained below. Note that management personnel are not assigned to specific activities. Instead, manpower and dollar costs representing a given management structure are sustained for the project as a whole, or for designated parts of it. Management personnel effort is not shown for specific boxes, even if the work is largely done by such persons.

The table provides a column to indicate quantity of persons (to one decimal place) for each manpower category; i.e.:

a. Contractor

Sys = system engineers and analysts

Dsgn = designers (junior and senior)

Prgm = computer programmers

Test = software test engineers

Sprt = support personnel; e.g., writers, operators, maintenance persons

b. Government

Dev = developing command (e.g., ESD)

Usr = using command (e.g., TAC)

Sprt = supporting command (e.g., AFLC)

c. Iterate Factor

Many tasks may need to be repeated because the results achieved on the first pass were not adequate to meet subsequent needs or review criteria. Since the work required on subsequent passes usually involves fewer persons, these three columns each contain a factor (from 0 to 10) representing the number of tenths by which the original number of persons in each of the manpower columns (as specified for the first pass) must be multiplied to obtain the manpower needed respectively on the second, third, and fourth or later iteration of the activity. If blank, the task never requires iteration, or multiplier value is equivalent to 10.

B.3.3 Durations

a. Days: The first duration column contains the mean duration of the activity, in work days to the nearest tenth.

b. It Fctr: The next three columns each contain a factor (from 0 to 10) representing the number of tenths of the first iteration's duration (i.e., days column) required to complete the second, third, and fourth or later iteration, respectively; a blank in these columns is the same as a "10." Some tasks have responses to iterative entry indicated by a negative digit as follows:

-2: Used to signal "impossible" situations that, if encountered, will cause the whole simulation run to halt.

-1: Used to indicate that certain network paths are not to be followed iteratively. Any path so entered is automatically terminated.

B.3.4 Wait

This field may contain a mean waiting time (in days) before the activity may begin. The action may begin only after the wait period is completed; the wait itself starts after all predecessor conditions are satisfied. If blank, no wait is required.

B.4 TABLE B-3

Each table B-3 entry contains the parameter data for a normal decision box (box type B) with an entry in table B-1. Every normal decision box in table B-1 must be represented by a table B-3 entry.

B.4.1 Box Data

These fields are defined in paragraphs B.3.1a-c.

B.4.2 YES Exit Probability

These four columns contain the probabilities (in percent) of taking the YES exit on the first four iterative passes through the decision box; see paragraph B.3.2c. The leftmost column provides first pass probability. The rightmost column probability will be halved repeatedly if the box is iterated more often than four times.

B.4.3 Wait

See paragraph B.3.4.

B.5 TABLE B-4

This table contains an entry for each counter decision box (type C), each special event box milestone box (type M), and each remote action box (type R). Every such box must be represented by a table B-4 entry.

B.5.1 Box Data

These fields' functions are given in paragraph B.3.1a-c.

B.5.2 Type

Two types of function have thus far been allocated to special event boxes:

M = milestone. The contents of the event label column (a milestone name) will be output on schedule reports for each special event box entered.

R = remote action. This action applies to the box override (FO) field so that a box can be skipped over, or "pinched off" or reset to normal. Only "pinch" (FO=1) is currently used.

B.5.3 Event Label

This contains the characters to be output as the milestone name for a milestone-type special event box.

B.5.4 Parameter

This column identifies the parameter that is to be changed by a reset (type R) special event.

B.6 TABLE B-5

This table contains an entry for each personnel box (type P) defined in table B-1.

B.6.1 Box Data

These fields' functions are given in paragraph B.3.1 a-c.

B.6.2 Contractor Personnel

This contains seven columns, a trigger, and six manpower categories. If more than one trigger is used, additional lines are used.

a. Trigger: A parameter used to indicate the point(s) in the process at which the P-box is activated to alter the personnel pool levels.

F - causes the P-box to act when the first DIG enters.

C - causes the P-box to act when the last DIG arrives.

N - activates the P-box on first entry for all non-DIG related boxes.

b. Manpower Categories: The type of manpower used.

Syst Eng = system engineers and analysts

Dsgn Eng = design engineers (junior and senior)

Pgrm = computer programmers

Test = software test engineers

Sprt = support personnel; e.g., writers, operators, maintenance persons

Mgmt = management persons

B.7 TABLE B-6

This table identifies the subnetworks by name and number.

B.7.1 Subnet#

This is the subnetwork number.

B.7.2 Title

This is the name given to the subnetwork.

APPENDIX C

SWAP OUTPUT REPORTS

APPENDIX C

TABLE OF CONTENTS

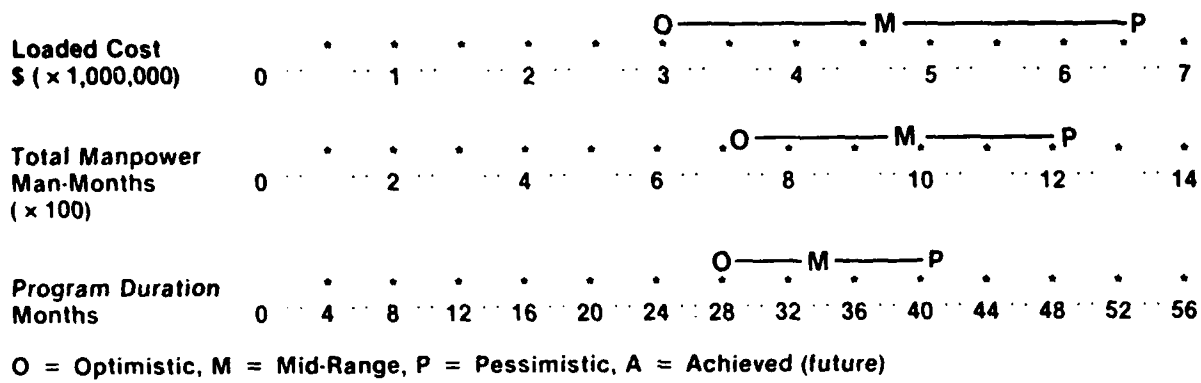


Figure C-1 Summary Forecast

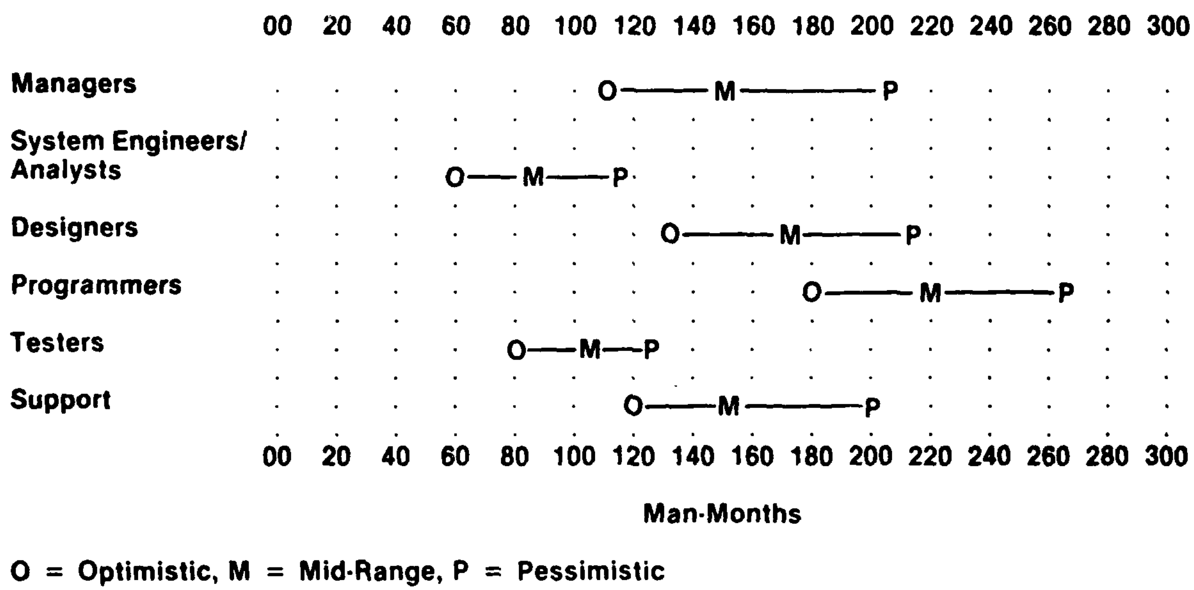


Figure C-2 Total Contractor Manpower Usage by Type

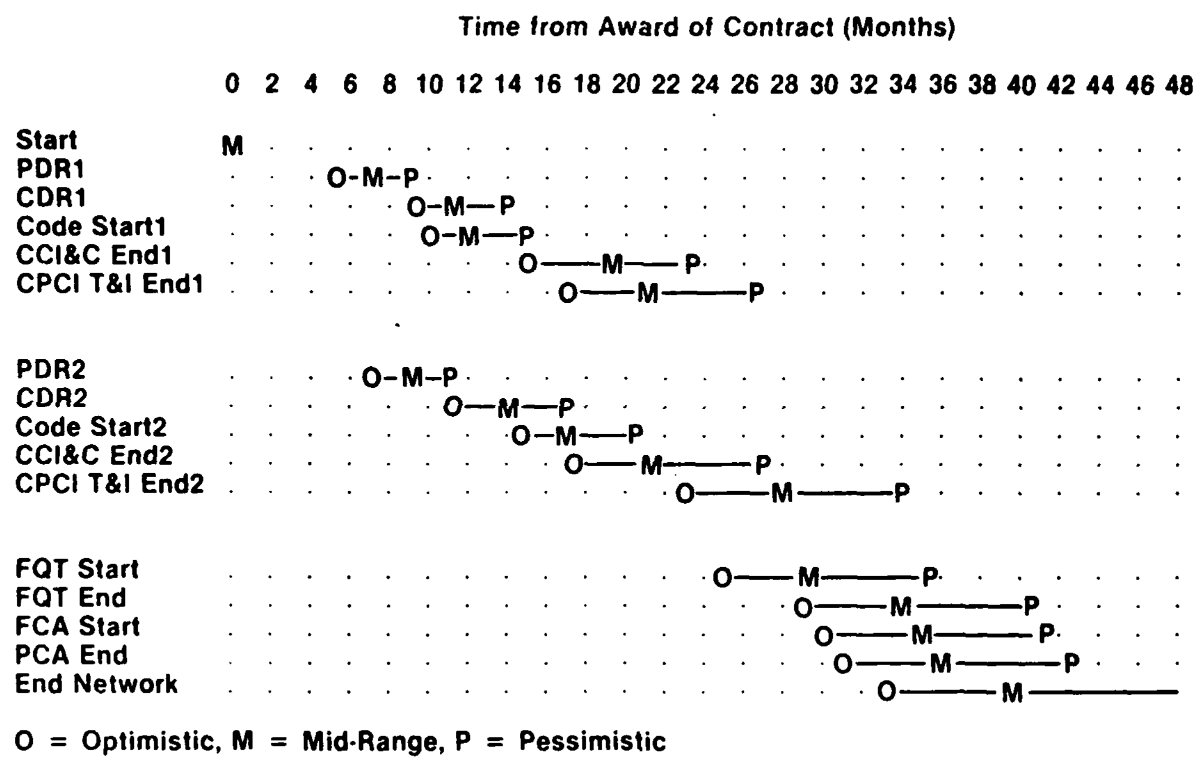


Figure C-3 Milestone Schedule

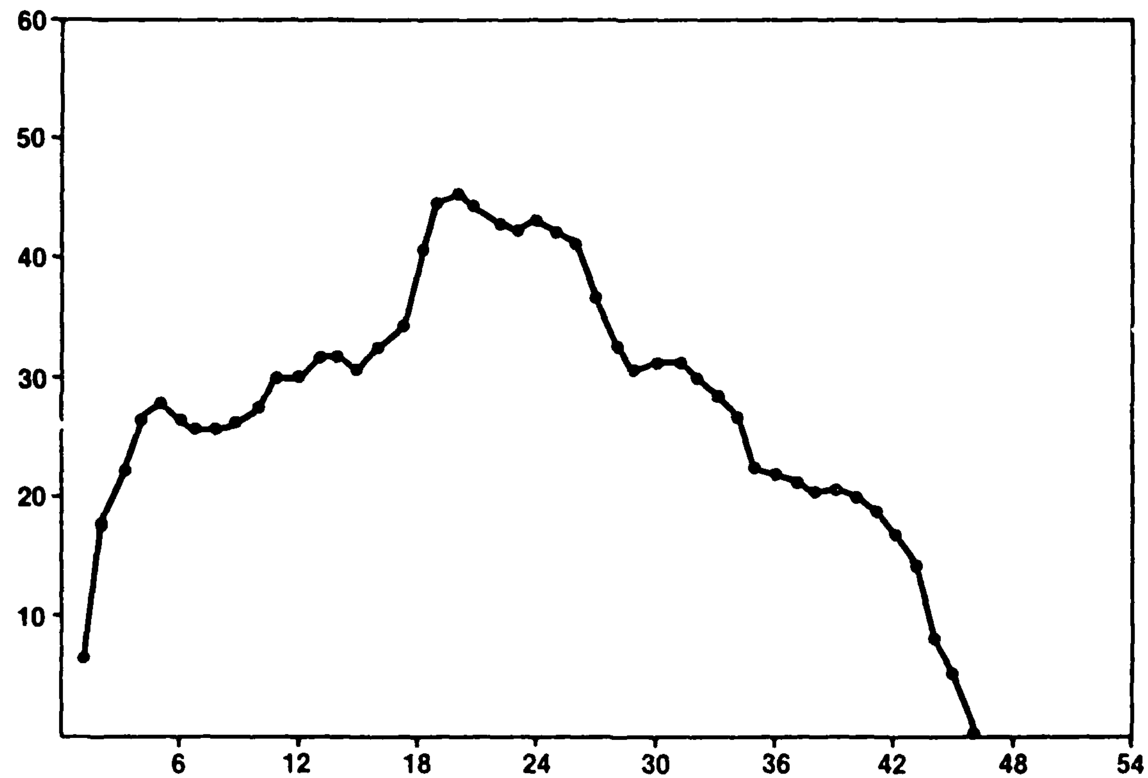


Figure C-4 Monthly Manpower Usage Summary

SOFTWARE ACQUISITION PROCESS MODEL REPORT

CONTRACTUAL EXPENDITURE SUMMARY
FULL NETWORK

PROJECT: FY81 DEMO-SLASH2

PAGE: 1
FULL SCALE DEVELOPMENT PHASE

OPTIMISTIC REPETITIONS: 5
MID-RANGE REPETITIONS: 20
PESSIMISTIC REPETITIONS: 6

SIMULATOR IDENTIFICATION: FY81 VERSION
SIMULATOR VERSION: MODEL ONE
DATA BASE TIME/DATE: 10/30/81 09.57.08
SIMULATION CONDUCT TIME/DATE: 10/30/81 09.59.02
REPORTS PRODUCED TIME/DATE: 10/30/81 13.22.54

LOADING FACTORS: OVERHEAD = 98.50% G&A = 14.12% FEE = 12.53%
LABOR RATES: LEVEL AT BASE YEAR RATES SHOWN

	- CONTRACTOR -							- GOVERNMENT -				- ALL -
	MANAGERS	SYSTEMS ENGINEERS	DESIGNERS	PROGRAMMERS	TEST ENGINEERS	SUPPORT	CONTRACT TOTAL	DEVEL CMD (ESD)	USING CMD (TAC)	SPRT CMD (AFLC)	GOV'T TOTAL	GRAND TOTAL
LABOR RATES ($)	3510	2770	2760	1870	2450	2085						
OPTIMISTIC												
MEAN MAN-MONTHS	87.5	46.1	68.4	90.4	41.0	132.0	465.4	55.4	12.4	7.1	74.9	540.3
% IT	----	7	7	3	21	2		6	6	2		
LOADED COSTS (X $1000)	691	288	425	381	226	620	$2630					
MID-RANGE												
MEAN MAN-MONTHS	91.7	45.3	69.4	90.2	40.8	135.6	473.0	55.0	12.5	7.2	74.7	547.7
% IT	----	7	8	4	22	2		6	7	3		
LOADED COSTS (X $1000)	724	282	431	380	225	637	$2680					
PESSIMISTIC												
MEAN MAN-MONTHS	98.6	47.0	70.8	95.1	43.6	144.4	499.5	57.1	12.9	7.5	77.5	577.0
% IT	----	8	8	5	24	3		7	7	3		
LOADED COSTS (X $1000)	779	293	440	400	240	678	$2831					

Figure C-5 Contractual Expenditure Summary

SOFTWARE ACQUISITION PROCESS MODEL REPORT

MILESTONE SCHEDULE
FULL NETWORK

PROJECT: FY81 OEMD-SLASH2

PAGE: 1
FULL SCALE DEVELOPMENT PHASE

OPTIMISTIC REPETITIONS: 5
MID-RANGE REPETITIONS: 20
PESSIMISTIC REPETITIONS: 6

SIMULATOR IDENTIFICATION: FY81 VERSION
SIMULATOR VERSION: MODEL ONE
DATA BASE TIME/DATE: 10/30/81 09.57.08
SIMULATION CONDUCT TIME/DATE: 10/30/81 09.59.02
REPORTS PRODUCED TIME/DATE: 10/30/81 13.22.54

MILESTONE		DTN	TIME OF OCCURRENCE		
			OPTIMISTIC	MID-RANGE	PESSIMISTIC
	START NETWORK	---	0M 0D	0M 0D	0M 0D
06L	PDR	1	4M 1D	4M 8D	4M 6D
		2	4M 11D	4M 13D	4M 15D
12F	CDR	1	9M 10D	10M 11D	10M 11D
		2	11M 15D	12M 5D	11M 16D
16B	START CODING	1	9M 17D	10M 19D	11M 1D
		2	12M 17D	14M 0D	14M 3D
18R	CCI & C END	1	13M 12D	14M 13D	14M 19D
		2	15M 9D	16M 12D	16M 18D
18T	C CI T & I END	1	14M 12D	15M 11D	15M 16D
		2	16M 2D	17M 7D	17M 19D
44D	FQT START	1	18M 9D	19M 18D	21M 9D
46W	FQT - END	1	21M 12D	23M 3D	24M 18D
52F	FCA - START	1	22M 10D	24M 6D	25M 19D
54G	SYS DT & E START	1	23M 11D	24M 14D	26M 9D
54R	SYS DT & E END	1	24M 17D	25M 18D	27M 15D
80F	PCA - START	1	22M 15D	24M 10D	26M 10D
82Z	PCA - END	1	23M 3D	24M 18D	26M 18D
	END NETWORK	---	26M 0D	27M 0D	29M 0D

Figure C-6 Milestone Schedule

SOFTWARE ACQUISITION PROCESS MODEL REPORT

MONTHLY MANNING PROFILE
FULL NETWORK

PROJECT: FINAL GO THRU
SIMULATOR ID: BASE.F407L
SIMULATOR VERSION: RPC LAST

FULL SCALE DEVELOPMENT PHASE
MID-RANGE
PAGE: 1

REPETITIONS: 6
BASED ON BOX: 822 FOR DIG/TIG: 1
RANGE: 39M 19D THRU 42M 12D
TIME SPAN: 0M 0D -> 45M 5D

DATA BASE DATE/TIME: 10/11/82 12.36.05
SIMULATION CONDUCT DATE/TIME: 10/11/82 12.41.37
REPORTS PRODUCED DATE/TIME: 11/10/82 10.35.15

	- CONTRACTOR -							- GOVERNMENT -				- ALL -
CON-TRACT MONTH	MANA-GERS	SYSTEMS ENGI-NEERS	DESI-GNERS	PROG-RAM-MERS	TEST ENGI-NEERS	SUP-PORT	CON-TRACT TOTAL	DEVEL CMD (ESD)	USING CMD (TAC)	SPRT CMD (AFLC)	GOV'T TOTAL	GRAND TOTAL
						MAN - MONTHS						
1	3.1	.7	.7	0.0	1.2	1.2	6.9	0.0	0.0	0.0	0.0	6.9
2	3.9	3.8	5.3	1.6	.8	4.5	20.1	2.5	.6	.6	3.6	23.8
3	4.0	4.4	7.0	2.0	.3	4.4	22.1	3.8	.6	.6	5.0	27.1
4	4.6	4.8	8.4	3.6	2.6	5.5	29.5	2.0	.3	.1	2.4	31.9
5	5.0	4.8	8.8	4.3	2.7	5.8	31.3	.5	0.0	0.0	.5	31.8
6	5.0	3.8	7.5	2.7	1.6	5.7	26.3	2.1	.4	.3	2.9	29.1
7	5.5	3.4	10.6	3.4	.2	4.6	27.9	5.0	1.1	1.1	7.3	35.2
8	6.0	3.6	13.8	3.1	.1	4.8	31.4	4.6	1.1	.9	6.5	38.0
9	6.0	3.5	12.5	3.0	.3	4.7	30.1	4.5	1.1	.8	6.4	36.5
10	6.0	3.7	10.8	2.0	.9	5.8	29.0	3.1	.7	.4	4.2	33.2
11	6.0	3.4	10.9	1.2	1.0	6.0	29.0	3.6	.8	.5	4.9	34.0
12	6.0	3.1	11.2	1.9	.6	5.8	28.7	4.4	1.1	.8	6.3	35.0
13	6.0	4.4	11.3	2.3	2.1	5.8	31.9	6.2	1.5	1.1	8.8	40.7
14	6.0	4.6	10.0	1.7	2.7	6.2	31.1	4.7	1.1	.7	6.5	37.6
15	6.0	4.4	7.7	2.4	3.6	7.1	31.2	6.2	2.0	1.3	11.5	42.6
16	6.0	2.8	6.5	16.1	1.8	6.4	39.6	3.3	.8	.6	4.6	44.2
17	6.0	3.2	6.1	27.8	1.8	7.4	52.4	4.1	.9	.4	5.4	57.8
18	6.0	3.7	6.7	22.9	3.6	8.6	51.6	2.0	.4	.2	2.5	54.1
19	6.0	3.6	6.0	15.5	4.3	7.9	43.3	5.0	1.1	.6	6.8	50.1
20	6.0	2.4	5.1	11.8	3.6	6.7	35.6	3.8	.8	.4	5.0	40.7
21	6.0	1.7	5.4	25.2	3.9	6.6	49.0	4.1	.9	.3	5.2	54.2
22	6.0	1.6	5.7	22.6	5.9	7.1	49.1	1.5	.4	.1	2.0	51.1
23	6.0	1.3	2.0	7.0	4.4	6.1	27.4	3.5	.7	.3	4.5	32.0
24	6.0	1.3	1.2	1.8	5.6	6.3	23.3	1.8	.2	.1	2.2	25.4

Figure C-7 Monthly Manning Profile

SOFTWARE ACQUISITION PROCESS MODEL REPORT

CONTRACTOR MONTHLY EXPENDITURE PROFILE
FULL NETWORK

PROJECT: FY81 DEMO-SLASH2

PAGE: 1
FULL SCALE DEVELOPMENT PHASE

OPTIMISTIC REPETITIONS: 5
MID-RANGE REPETITIONS: 20
PESSIMISTIC REPETITIONS: 6

SIMULATOR IDENTIFICATION: FY81 VERSION
SIMULATOR VERSION: MODEL ONE
DATA BASE TIME/DATE: 10/30/81 09.57.08
SIMULATION CONDUCT TIME/DATE: 10/30/81 09.59.02
REPORTS PRODUCED TIME/DATE: 10/30/81 13.22.54

LOADING FACTORS: OVERHEAD = 98.50 %; G&A = 14.12%; FEE = 12.53%
INFLATION RATE: 10.00%, STARTING IN MONTH 3
COSTS ARE IN $1000 INCREMENTS

	OPTIMISTIC				MID-RANGE				PESSIMISTIC			
	THIS MONTH		CUMULATIVE		THIS MONTH		CUMULATIVE		THIS MONTH		CUMULATIVE	
CONTRACT MONTH	BASE YEAR	THEN YEAR	BASE YEAR	THEN YEAR	BASE YEAR	THEN YEAR	BASE YEAR	THEN YEAR	BASE YEAR	THEN YEAR	BASE YEAR	THEN YEAR
1	53.6	53.6	53.6	53.6	49.4	49.4	49.4	49.4	49.0	49.0	49.0	49.0
2	119.8	119.8	173.5	173.5	112.6	112.6	162.0	162.0	113.6	113.6	162.6	162.6
3	131.6	144.7	305.0	318.2	130.7	143.8	292.7	305.8	130.2	143.2	292.8	305.8
4	113.1	124.4	418.1	442.5	113.5	124.9	406.2	430.7	111.4	122.6	404.2	428.4
5	132.7	145.9	550.7	588.5	116.0	127.6	522.2	558.2	125.1	137.6	529.3	566.0
6	119.0	130.8	669.7	719.3	111.1	122.2	633.3	680.5	118.6	130.5	648.0	696.5
7	108.8	119.7	778.5	839.0	107.2	117.9	740.6	798.4	113.5	124.8	761.5	821.3
8	115.0	126.5	893.6	965.6	117.2	128.9	857.7	927.3	110.4	121.4	871.8	942.7
9	90.8	99.9	984.4	1065.5	101.9	112.1	959.6	1039.4	94.4	103.8	966.2	1046.6
10	107.0	117.7	1091.4	1183.2	91.4	100.5	1051.0	1139.9	92.5	101.8	1058.7	1148.3
11	142.7	157.0	1234.1	1340.2	97.9	107.7	1148.9	1247.6	109.9	120.9	1168.7	1269.3
12	156.5	172.2	1390.6	1512.3	138.7	152.5	1287.6	1400.1	122.0	134.2	1290.7	1403.5
13	134.4	147.8	1525.0	1660.1	144.5	159.0	1432.1	1559.1	144.9	159.4	1435.5	1562.8
14	159.9	175.8	1684.8	1836.0	130.9	144.0	1563.0	1703.1	136.6	150.3	1572.1	1713.1
15	126.3	152.8	1811.1	1988.8	147.4	178.3	1710.4	1881.4	141.0	170.6	1713.2	1883.7
16	82.6	99.9	1893.7	2088.7	126.9	153.6	1837.3	2035.0	131.1	158.6	1844.2	2042.3
17	81.8	98.9	1975.5	2187.6	92.0	111.4	1929.3	2146.4	106.0	128.3	1950.3	2170.6
18	76.5	92.5	2051.9	2280.2	87.2	105.5	2016.5	2251.9	82.6	99.9	2032.9	2270.6
19	91.3	110.5	2141.3	2390.7	76.5	92.5	2093.0	2314.4	76.0	92.0	2108.9	2362.6
20	87.6	106.0	2230.8	2495.6	80.2	97.1	2173.2	2441.5	82.3	99.6	2191.2	2462.2
21	64.6	78.2	2295.4	2574.8	88.1	106.5	2261.3	2548.0	73.4	88.8	2264.6	2551.0
22	84.7	102.5	2380.2	2677.3	70.7	85.6	2332.0	2633.6	77.1	93.3	2341.7	2644.3
23	68.0	82.3	2448.2	2759.6	72.1	87.2	2404.1	2720.8	87.4	105.7	2429.1	2750.0
24	81.7	98.9	2529.9	2858.5	78.7	95.2	2482.8	2816.0	70.1	84.9	2499.2	2834.9
25	55.8	67.6	2585.7	2926.0	[illegible]	85.5	[illegible]	[illegible]	67.0	81.1	[illegible]	2916.0
26	25.2	30.3	2610.9	[illegible]	[illegible]	65.1	[illegible]	[illegible]	75.5	91.4	2641.8	3007.3
27	9.4	12.5	2620.4	[illegible]	[illegible]	54.5	[illegible]	3045.2	69.6	92.6	[illegible]	3099.9
28	8.2	10.3	[illegible]	2980.0	[illegible]	14.3	[illegible]	3059.5	69.1	92.0	2780.5	3192.0
29	1.6	2.2	[illegible]	[illegible]	[illegible]	1.1	[illegible]	[illegible]	35.2	46.8	2815.6	3238.8
30	0.0	0.0	2630.2	[illegible]	0.0	0.0	[illegible]	3060.5	15.5	20.6	2831.1	3259.4

Figure C-8 Contractor Monthly Expenditure Profile

SOFTWARE ACQUISITION PROCESS MODEL REPORT

MILESTONE SCHEDULE
SUBNETWORK 2: COMPUTER PROGRAM DESIGN
SUBNETWORK 3: CODING THROUGH CPT&E
SUBNETWORK 4: FORMAL TEST PREPARATION

PROJECT: FY81 DEMO-SLASH2

PAGE: 1
FULL SCALE DEVELOPMENT PHASE

OPTIMISTIC REPETITIONS: 5
MID-RANGE REPETITIONS: 20
PESSIMISTIC REPETITIONS: 6

SIMULATOR IDENTIFICATION: FY81 VERSION
SIMULATOR VERSION: MODEL ONE
DATA BASE TIME/DATE: 10/30/81 09.57.08
SIMULATION CONDUCT TIME/DATE: 10/30/81 09.59.02
REPORTS PRODUCED TIME/DATE: 10/30/81 13.22.54

MILESTONE		DTN	TIME OF OCCURRENCE OPTIMISTIC	MID-RANGE	PESSIMISTIC
	START NETWORK	---	0M 0D	0M 0D	0M 0D
16B	START CODING	1	9M 17D	10M 19D	11M 1D
		2	12M 17D	14M 0D	14M 3D
18R	CCI & C END	1	13M 12D	14M 13D	14M 19D
		2	15M 9D	16M 12D	16M 18D
18T	CPCI T & I END	1	14M 12D	15M 11D	15M 16D
		2	16M 2D	17M 7D	17M 19D
	END NETWORK	---	26M 0D	27M 0D	29M 0D

Figure C-9 Milestone Schedule - Subnetwork

APPENDIX D

COMPUTER PROGRAM OPERATION

APPENDIX D

TABLE OF CONTENTS

LIST OF FIGURES

D.1 OVERVIEW

This appendix provides instructions for operating the Release 1 version of the SWAP Model simulator program. The SWAP Model is currently designed to be operated from a cathode ray tube terminal that is tied into the MITRE Bedford Computer Center using the Time-Sharing Option (TSO) facility. Some familiarity with TSO is assumed.

As shown in appendix B, a set of six tables define the software acquisition process logic and parameter values to the simulator. Data values for these tables may be entered or altered via a terminal (as explained in section D.3.2 of this appendix) to reflect the particular software being estimated. With the tables complete, a user then obtains cost and schedule estimates with the simulator by performing three successive steps.

In the first step some presimulation processing is performed on the input tables. The table layout was designed to make the creation/alteration task easy for the user. However, that format cannot be used directly by the simulator. The tables must first be converted into a data base that is easier for the simulator to manipulate.

As part of this conversion process, the tables are subjected to a series of error checks. Each table is examined to see that its format is correct and its information consistent with the other tables. Diagnostic messages inform the user of any detected errors and the severity (i.e., one of three degrees) of those errors. Since the quality of the input data impacts the quality of the SWAP Model's output, the user should (and for one degree is required to) resolve errors before completing the input data conversion process.

Once this presimulation processing has been successfully completed, the next step is to conduct the actual simulation of the software acquisition process. In this step the simulator follows the network (as defined in the processed input tables) box by box by resolving each decision box and keeping track of time and resources used. This information is later used to create the output forecasts.

The user is allowed some control over a simulation run. For example, the number of passes to be made through the network can be specified, as can other simulator parameters (e.g., number of work days in a month) to reflect different acquisition environments. Certain input data values (e.g., the yes probability of a box) may also be changed at this step, eliminating the need to edit input tables and then repeat the data input and conversion step.

After the simulation run, the results are provided in a number of output reports. In this third and final step, the simulator produces those output reports requested by the user. Section 4 of this report discusses the output reports available, and examples are shown in appendix C. Included are milestone schedules, cost/manpower summaries, and monthly cost/manpower profiles.

When producing output reports, the user can optionally change the percentiles used in determining the mid-range, optimistic, and pessimistic estimates. Contractor cost information (e.g., overhead, G&A, etc.) may also be changed. Since this information is used only in the third and final step, the first two steps need not be repeated to obtain cost and schedule estimates for different sets of contractor costs or output data partitionings.

The simulator has a separate routine for performing the tasks of the above three steps. So that a user may fully utilize the simulator's capabilities, a simple yet sufficiently detailed discussion of each routine's use is provided in the next section.

D.2 USING THE SIMULATOR

The SWAP Model simulator program consists primarily of three routines that the user normally executes sequentially to produce cost and schedule estimates. The presimulation processing is handled by a routine termed the Data Input Processor (DIP), the actual simulation is executed by a routine termed the Simulation Conduct Processor (SCP), and the output reports are created by a routine termed the Output Report Generator (ORG). A user interface provides the means by which an operator can access and control the three routines or create/alter input tables. These three routines are discussed below.

D.2.1 Data Input Processor (DIP)

DIP reads the data contained in the network tables. It runs format and consistency checks on this data, and produces warning messages when errors are found. Finally, DIP transforms the tables into a data base the simulator can easily manipulate.

D.2.1.1 Input Table Error Checking. Network table errors are categorized into three classes. Warning errors (signified WW) suggest corrections that are needed to maintain the network within more consistent rules. Normal errors (signified XX) will not prevent the simulation from running because changes are made internally (not to the input tables) to correct these errors.

However, these changes may not reflect the user's intended logic. Severe errors (signified YY) must be corrected by the user before DIP will perform the input data conversion.

The table data fields defined in appendix B are checked to determine that the following conditions are correct.

(1) Developmental Integration Group (DIG) spread percentages total 100%.

(2) Each box is described by defined field designators: valid box types, DIG participation, organization performing the action (i.e., contractor or government), and burst membership.

(3) Tables are consistent with each other:

- boxes described in tables B-2, B-3, B-4, or B-5 must all be included in table B-1;

- boxes described in table B-1 appear in one and only one of tables B-2, B-3, B-4, or B-5;

- group and type designators match between table B-1 and either table B-2, B-3, B-4, or B-5.

(4) Successor data is complete and consistent:

- all three network progression fields are present and legitimately designated;

- box group membership is consistent with successor group labels;

- each successor box identification (ID) also appears as a box ID entry;

- progression modes and start logic are consistent with each other;

- box burst memberships and successor progression modes are consistent with each other.

(5) Numeric values are within prescribed limits, e.g., many cannot be negative.

D.2.1.2 Reports Produced. Each execution of DIP produces an output report containing box breakdowns and a listing of any input errors (including the corrections made to type XX errors). A sample report is shown in figure D-1. An optional network report, which is shown in figure D-2, is also available. It includes a list of all successor and predecessor boxes for each box in the network, and is a valuable diagnostic aid for any error reports obtained during SWAP operation.

D.2.2 Simulation Conduct Processor (SCP)

SCP performs the actual simulation by progressing through the network a given number of repetitions, using the data base created by DIP and the control card file provided by the user as input. The simulation results (such as manpower used and occurrence time of a box) are stored for use by the next package (ORG).

D.2.2.1 Control Card File. The control card file allows the user to supply SCP with simulation run information (e.g., number of repetitions), to change standard simulator program values (e.g., number of working days in a month), and to make input data changes without re-executing DIP (e.g., change the yes probability of a decision box). The file consists of one-line change instructions termed "control cards." Each control card consists of a code (e.g, N 1) followed by the desired value or change. For cards that change the value of a box parameter, the ID of the affected box is also specified (just before the new value). Possible control cards are described below. The control card file must have card N 1 or N 7 for the first card. All other cards are optional. Some examples of the content on the control cards are shown in figure D-3.

Control Card Code	Purpose
N 1	Number of repetitions; should not occur with card N 7.
N 3	Starting time of simulation; defaults to 0; cannot be negative.
N 4	A play-by-play report is to be printed for the pass number specified. The queue analysis will not be printed on this pass.
N 6	The number of personnel (of a designated type) to be assigned at the project's starting point. In the third field is the manpower type code (1 = systems engineers, 2 = designers, 3 = programmers, 4 = testers, 5 = support

```
     DATA INPUT AND CONVERSION SECTION FOR THE SOFTWARE ACQUISITION
                PROCESS MODEL SIMULATION PROGRAM MODEL 1

    IN THIS SECTION THE DATA IS READ IN FROM SEVEN TABLES AND CONVERTED
INTO USABLE SIMSCRIPT FORMAT.  ERROR MESSAGES ARE PRINTED AS THEY OCCUR
AND AN ATTEMPT IS MADE TO CONTINUE PROCESSING IN SPITE OF THE ERRORS TO
ALLOW ALL ERROR CORRECTION IN ONE PASS.

    THERE ARE 3 CATEGORIES OF ERRORS:
  WW - WARININGS.        THE SCP WILL RUN. NO CHANGES MADE TO INPUT
  XX - ERRORS.           THE SCP WILL RUN. CHANGES HAVE BEEN MADE TO INPUT
  YY - SEVERE ERRORS.    THE SCP WILL FAIL. NO CHANGES MADE TO INPUT


    DATA INPUT PROCESSOR DATE:  12/29/82 ,   AND TIME:  10.15.35


FOR  2 DIGS, THE BREAKDOWNS ARE:  60%   40%
FOR  2 TIGS, THE BREAKDOWNS ARE:  60%   40%


      FOR  193 BOXES, THE BREAKDOWNS ARE AS FOLLOWS:

             116 ACTIVITY BOXES           50 IN DIG PARTICIPATION
              35 DECISION BOXES            0 IN TIG PARTICIPATION
               7 COUNTER BOXES           143 PARTICIPATE IN NEITHER
               5 HELPER BOXES
              12 MILESTONE BOXES         121 NOT IN BURSTS
              16 PERSONNEL  BOXES          8 START BURSTS
               2 REMOTE CHANGER BOXES     44 CONTINUE BURSTS
                                          16 END BURSTS
                                           4 RECUR INDEFINITELY

           0 ERRORS OF TYPE WW ARE SUSPECTED AS DETAILED ABOVE
           0 ERRORS OF TYPE XX ARE SUSPECTED AS DETAILED ABOVE
           0 ERRORS OF TYPE YY ARE SUSPECTED AS DETAILED ABOVE

 *** THE PROGRAM HAS COMPLETED DATA INPUT AND CONVERSION ***
```

Figure D-1. Data Input Processor Output Report

SOFTWARE ACQUISTION PROCESS MODEL
INPUT NETWORK PAGE 1

NAME	TYPE	GROUP	BOX CLASS	BURST	SST	RECUR	SNET	:	SUCCESSORS ID	SUCCESSORS TYPE	SUCCESSORS GROUP	SUCCESSORS PROG	SUCCESSORS SLA	:	PREDECESSORS ID	PREDECESSORS GROUP	PREDECESSORS PROG	PREDECESSORS SLA
01Y	P	N	GI	N	F	0	0	:	02A	Y	N	F	S	:				
02A	A	N	GI	N	P	0	10	:	04A	Y	N	F	A	:	01Y	N	F	S
									04S	Y	N	F	S					
04A	A	N	GI	N	P	0	1C	:	04C	Y	N	F	S	:	02A	N	F	A
									06A	Y	N	F	A		04S	N	F	A
									06Y	Y	N	F	S					
									53A	Y	N	F	S					
									53C	Y	N	F	S					
04C	A	N	GI	N	P	0	10	:	04E	Y	N	C	S	:	04A	N	F	S
															04G	N	I	S
04E	A	N	GI	N	F	0	10	:	04G	Y	N	C	S	:	04C	N	C	S
04G	B	N	GI	N	F	0	10	:	04J	Y	N	F	S	:	04E	N	C	S
									04C	N	N	I	S					
04J	A	N	GI	N	F	0	10	:	04L	Y	N	F	S	:	04G	N	F	S
															04M	N	I	S
04L	B	N	GI	N	F	0	10	:	66B	Y	N	F	R	:	04J	N	F	S
									04M	N	N	I	S					
04M	A	N	GI	N	F	0	10	:	04J	Y	N	I	S	:	04L	N	I	S
04S	A	N	GI	N	P	0	10	:	04A	Y	N	F	A	:	02A	N	F	S
									60A	Y	N	F	S					
									62A	Y	N	F	A					
									60Y	Y	N	F	S					
06A	A	N	DI	N	P	0	2	:	06D	Y	N	C	S	:	04A	N	F	A
									06F	Y	N	F	S		06E	N	I	S
06D	A	N	DI	N	F	0	2	:	06E	Y	N	C	S	:	06A	N	C	S
06E	B	N	DI	N	F	0	2	:	06G	Y	N	F	A	:	06D	N	C	S
									06A	N	N	I	S					
									08Y	Y	N	F	S					
06F	A	N	DI	N	P	0	2	:	06G	Y	N	F	A	:	06A	N	F	S
									14A	Y	N	F	S					
06G	A	D	DI	S	P	0	2	:	06H	Y	G	C	S	:	06E	N	F	A
															06F	N	F	A
															06I	G	I	S
															06J	D	F	A
															06P	G	I	S
06H	A	D	DI	C	F	0	2	:	06I	Y	G	C	S	:	06G	G	C	S
06I	B	D	DI	C	F	0	2	:	20A	Y	G	F	A	:	06H	G	C	S
									06G	N	G	I	S					
									06J	Y	G	F	S					
06J	C	D	DI	E	F	0	2	:	06G	N	D	F	A	:	06I	G	F	S
06L	M	D	DI	N	F	0	9	:						:	08C	G	F	S
06M	A	D	DI	N	P	0	2	:	08E	Y	G	F	S	:	08C	G	F	S

Figure D-2. Data Input Processor Network Report

```
CONTROL CARDS USED FOR THIS RUN:
*COMMENTS APPEAR BY A CHARACTER IN COLUMN 1
*
   N   1        10                  NUMBER OF REPETITIONS
*  N   6    5   10                  MANPOWER
   A   2    BASE.P407L              PROJECT NUMBER
   A   3    01Y                     STARTING BOX
   M   4    50H       7
   A   1    NEW B3, IT CT           PROJECT NUMBER
   A   4    NEW FLOW Y/N            VERSION
   N   10      10                   DEVIATIONS
   N   11      20                       "
   N   12      15                       "


  RUN OF SCP STARTED ON 12/10/82 AT 13.42.22
  USING DATA BASE GENERATED ON 12/10/82 AT 11.16.03
  PROJECT NUMBER: NEW B3, IT CT
  SIMULATION ID: BASE.P407L
  AMOUNT OF REPETITIONS REQUESTED IS:    10
  ITERATION LIMIT IS:    10
  STARTING BOX IS: 01Y
  STARTING TIME IS:         0

1041.0    END PASS    1   @#
 959.0    END PASS    2   @#
 959.0    END PASS    3   @#
 986.0    END PASS    4   @#
1007.0    END PASS    5   @#
 963.0    END PASS    6   @#
1052.0    END PASS    7   @#
 923.0    END PASS    8   @#
 894.0    END PASS    9   @#
1078.0    END PASS   10   @#
```

Figure D-3. Simulation Conduct Processor Output Report personnel, 6 = managers), and the fourth field has the number of people. Only one card is allowed per type, with a maximum of six cards allowed in total.

N 7	Only run this specific pass number, printing the play-by-play and the array and box information dump-out. The output data base is not affected. This card overrides the 'N 1' card. It is usually used as a diagnostic aid after an error has been detected in a pass.
N 8	Maximum number of months allowed for the simulation; default is 60.
N 9	Number of working days in a month; default is 20.
N 14	Pass to print the queue analyses. The play-by-play will not be printed on the pass.
A 1	Project number; default is "NO PROJECT # IS GIVEN."
A 2	Simulation ID; default is "NO SIM ID GIVEN."
A 3	Starting box; defaults to first box in the network that is without any predecessor boxes.
A 4	Simulation version; default is "MODEL ONE."
M 0	Change subnetwork of a box.
M 1	Change successor selection timing (SST) designation of a box.
M 2	Change doer for a box.
M 3	Enter a flow override for the box.
M 4	Change the initial wait of a box.
M 5	Change the yes-probability of the first iteration for a decision box.
M 6	Change the yes-probability of the second iteration for a decision box.
M 7	Change the yes-probability of the third iteration for a decision box.
M 8	Change the yes-probability of the fourth iteration for a decision box.

D.2.2.2 Reports Produced. Each execution of SCP produces an output report containing a listing of the control cards used, some general information about the simulation, and a listing of the realized number of work days from start to completion of project for each repetition. A sample report is shown in figure D-3.

D.2.3 Output Report Generator (ORG)

ORG statistically analyzes, formats, and prints the simulation results based on the data provided by SCP and the information contained in the following three input files:

1. Report Selection

2. Wage & Inflation Information

3. Pessimistic/Optimistic/Mid-range (P/O/M) Selection

These three files are described below.

D.2.3.1 Report Selection. The Report Selection file tells the simulator which of the following reports are to be produced:

TABULAR FORMAT

1) CONTRACTUAL EXPENDITURE SUMMARY
2) MILESTONE SCHEDULE
3) MONTHLY MANNING PROFILE
4) MANPOWER CALIBRATION REPORT and PERSONNEL BOX SUMMARY
5) ACTIVITY BOX SUMMARY -- INPUT ORDER
6) ACTIVITY BOX SUMMARY -- EST SORT
7) MONTHLY MANPOWER

PSEUDOGRAPHIC FORMAT

8) MONTHLY MANNING CHART (relates to Report 3)
9) CUMULATIVE COST GRAPH (relates to Report 7)
10) MONTHLY COST GRAPH (relates to Report 7)
11) CHART OF MILESTONE SCHEDULES (relates to Report 2)

These reports can be requested for the full network or any combination of subnetworks, except for report 4, which can not be requested at the subnetwork level.

Figure D-4 shows a sample report selection file. Desired reports are listed by number, one per line, followed by the corresponding subnetwork or combination of subnetworks, if applicable. Full network reports result if no subnetwork is given. Sample output reports are provided in appendix C.

D.2.3.2 Wage & Inflation Information. The Wage & Inflation Information file contains contractor expense parameters used in producing the cost reports. Inflation (in percent) is accounted for once a year on the first month in which it is to begin and every 12 months thereafter, compounding on a yearly basis. The first month of inflation can be negative, inflating wages before the project starts. Figure D-5 shows a sample Wage & Inflation Information file.

D.2.3.3 P/O/M Threshold Selections. The P/O/M Selection file provides the information for partitioning simulation results into the divisions (pessimistic, optimistic, and mid-range) for reporting purposes. Figure D-6 shows a sample P/O/M Selection file.

D.3 PROGRAM OPERATING INSTRUCTIONS

The following instructions describe how to:

- Access SWAP and identify the desired project (par. D.3.1),
- Edit or create the network tables (par. D.3.2),
- Check and process the new or revised tables (par. D.3.3),
- Conduct the simulation (par. D.3.4),
- Generate output reports (par. D.3.5), or
- Exit from SWAP programs (par. D.3.6).

All automatic program responses are indicated by CAPITAL letters in the examples shown below.

D.3.1 Program Start-up

After the standard MITRE log-on procedure, the SWAP Model program is accessed by typing: exec swap. The user is then presented with a display such as the one shown in the following example.

WHICH PROJECT ARE YOU WORKING ON?

1) JTIDS

ORG REPORT SELECTION

MANPOWER SUMMARY REPORT		1
MILESTONE SCHEDULE		2
MONTHLY MANPOWER CHART		3
DSN, PGM, TESTERS & P-BOXES		4
ACTIVITY BOX SUMMARY		5
ACTIVITY BOX SUMMARY BY EST		6
MONTHLY COST CHART		7
MONTHLY MANPOWER GRAPH	(3)	9
MONTHLY CUMULATIVE COST GRAPH	(7)	11
MONTHLY COST GRAPH	(7)	12
MILESTONE SCHEDULE CHART	(2)	13

REPORT TYPE	SUBNETWORK(S)
1	
2	
5	
9	
1	2
1	2 3 4

Figure D-4. Output Report Generator Report Selection File

```
*             SOFTWARE ACQUSITION PROCESS MODEL
*                                                       9/3/81    RPC
*                  CONTRACTOR COST TABLE
*
* FILL IN THE INFORMATION TO THE LEFT OF THE HEADING ON THE ROW
* THE FIRST TWO GROUPS REQUEST DIRECT COST PER PERSONNEL TYPE.
*  THESE ARE WITHOUT ANY OVERHEAD
*  DO NOT INCLUDE THE $ SIGN
*  DO NOT INCLUDE ANY CENTS  (WHOLE DOLLARS ONLY)
*
* THE LAST TWO GROUPS DEAL WITH OVERHEAD AND INFLATION RATES
*  INPUT THE NUMBERS WITHOUT THE % SIGN.
*  DECIMALS ARE ALLOWED'
* THE START MONTH IS AN INTEGER THAT REPRESENTS WHICH MONTH INTO
* THE SIMULATION THE FULL INFLATION RATE OCCURS.
* EVERY 12 MONTHS THEREAFTER THE RATES WILL INCREASE BY THE INFLATION RATE
*
*
*   DIRECT MONTHLY RATE                PERSONNEL TYPE
*   ==================                 ==================
*
        2770                            SYSTEMS ENGINEER
        2760                            DESIGNER
        1870                            PROGRAMMER
        2450                            TEST ENGINEER
        2085                            SUPPORT PERSONNEL

        3510                             MANAGER

*====================================================================

         98.5                            OVERHEAD
         14.12                           G & A
         12.53                           FEE

          10.0                           INFLATION RATE
          3                              MONTH INFLATION BEGINS
```

Figure D-5. Output Report Generator Wage & Inflation Information File

```
822          NAME OF BOX TO BASE PARITITIONS ON
 1           AND ITS DIG/TIG  (ENTER ONE IF NONE)

1.0          .THE RANGE OF THE PARITITION (% OF SIGMA IN HUNDREDTHS)
*             THUS (1 WOULD BE 100%  AND .5 WOULD BE 50%)
```

Figure D-6. Output Report Generator P/O/M Selection File

2) E3-A

3) COMBAT GRANDE

4) PAVE PAWS

?

Responding with the desired project line number declares the project simulation to be exercised (and thus, what project files are to be accessed) and produces the main menu display.

At any time later in this simulation session, the user can switch to a different project simulation by typing P in response to the main menu display, causing the above display to reappear.

THE MAIN MENU

As shown in figure D-7, the main menu is effectively the hub of the SWAP program. It is from this point that one accesses all program functions. The first three lines of the main menu display inform the user of what the last task performed in simulating the project was (e.g., updating network tables, running DIP, etc.), the date and time that the task was performed, and what is logically the next task to perform. The balance of the main menu display is as follows:

```
N = NETWORK TABLES
D = DATA INPUT PROCESS                    (DIP)
S = SIMULATION CONDUCT PROCESSOR          (SCP)
R = OUTPUT REPORT GENERATOR               (ORG)

P = ANOTHER, ALREADY EXISTING, PROJECT
C = CREATE A NEW PROJECT
F = FINISHED, EXIT SWAP SIMULATOR

?
```

D.3.2 Input Data Creation

D.3.2.1 Creating Network Tables. To establish network tables for a new project simulation, the user could build "from scratch" the six data files. An easier method is to copy the network tables of a

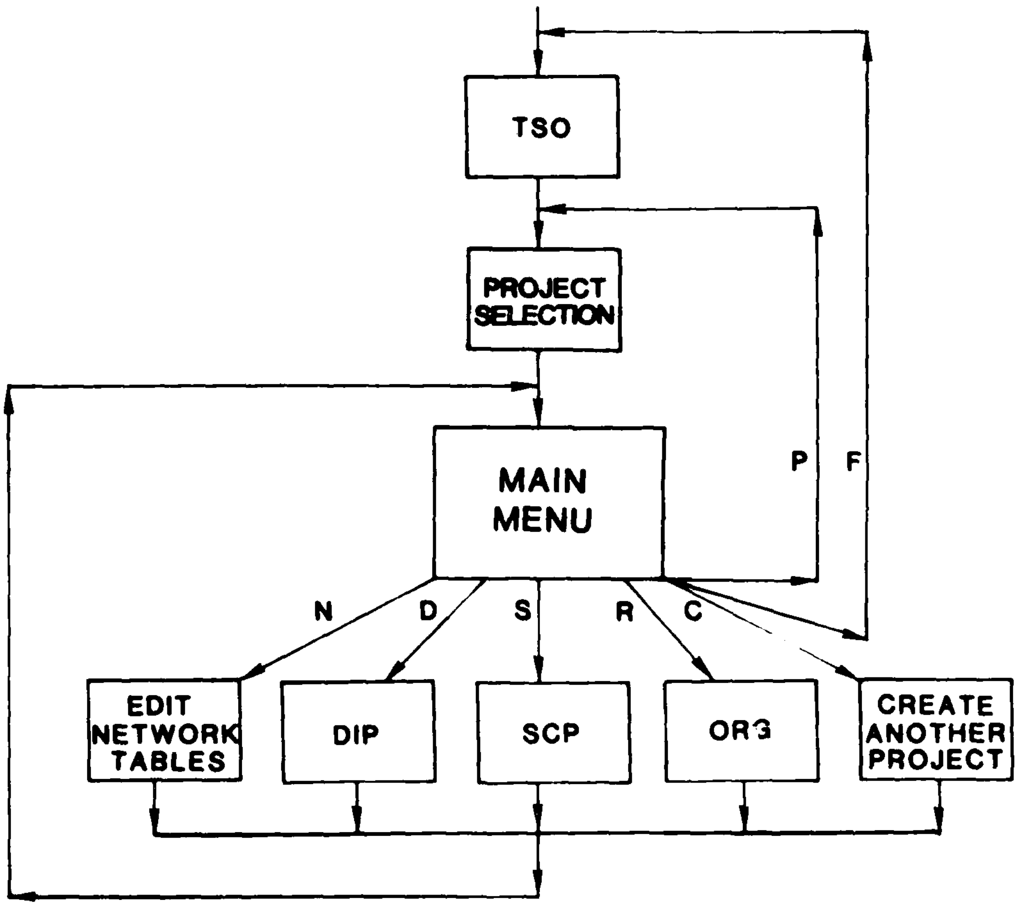


Figure D-7. Simulator Configuration

similar project simulation and then tailor (by editing as per section D.2.3) those copies to reflect the new project simulation. The steps required to create a duplicate set of network tables are described below.

To create a new project simulation, the user responds to the main menu display by typing C. This results in the following display:

ENTER THE PROJECT NAME

THE NAME MUST BE IN GROUPS OF 8 LETTERS OR LESS SEPARATED BY '.' ONLY LETTERS AND DIGITS CAN BE USED AND THE FIRST CHARACTER OF ANY GROUP MUST BE A LETTER THE TOTAL OF CHARACTERS CAN BE NO MORE THAN 19

?

After entering a name for the new project simulation, the following is displayed:

ENTER THE BASE PROJECT NAME:

All base projects must have "base" as the first component of their name. When responding to this display, the "base" component of the name is omitted. After responding with the base project name, the following is displayed:

THE GAP PARAMETERS AND THE SCP AND ORG INPUTS WILL BE COPIED FROM

EITHER:

ONE OF YOUR ALREADY EXISTING PROJECTS

OR:

THE DEFAULT FOR EACH

ENTER THE NUMBER CORRESPONDING TO YOUR CHOICE

YOUR PROJECTS:

1) JTIDS

2) E3-A

3) **COMBAT GRANDE**

4) **PAVE PAWS**

OR:

5) **SYSTEM DEFAULT**

After responding with the desired number, the following is displayed:

YOU ARE CREATING PROJECT <Project's name>
WITH A BASE PROJECT OF <Base Project's name>
AND THE INPUT IS CONSTRUCTED FROM <Input Source>

PROJECT CREATION WILL TAKE A FEW MINUTES
AND CANNOT BE INTERRUPTED
THIS OPPORTUNITY ALLOWS YOU NOT TO CREATE THE PROJECT AT THIS TIME

ENTER THE LETTER CORRESPONDING TO YOUR CHOICE

Y = CREATE PROJECT
N = DON'T CREATE PROJECT (reproducing Main Menu display)
O = GO UP ONE MENU DISPLAY

Responding Y to this display causes creation of the project files and produces the following display sequence:

FILE CREATION IS NOW TAKING PLACE
THIS WILL TAKE A LITTLE WHILE

followed by:

ALL THE FILES HAVE BEEN CREATED FOR PROJECT project name

The main menu display will appear, signifying the file creation is complete. Any operations now performed will be using the newly created files (i.e., a change in projects takes place when creating a new project).

D.3.2.2 Editing the Network Tables. To edit the network tables, the user responds to the main menu display by typing N. This produces the following display:

WHICH TABLE# DO YOU WANT TO WORK ON

JUST ENTER THE NUMBER

TABLE 1 - THE NETWORK CONNECTIVITY
TABLE 2 - ACTIVITY BOXES
TABLE 3 - DECISION BOXES
TABLE 4 - SPECIAL EVENT BOXES
TABLE 5 - PERSONNEL BOXES
TABLE 6 - SUBNETWORK TITLES
TABLE 7 - DIGS AND TIGS

JUST <ENTER> TO RETURN TO MAIN MENU

To edit any of these files, the user responds to this display with the desired table's number, invoking the TSO full screen editor. Full screen editor commands are used to modify the file contents. Terminating the editing session (with the SE command) will reproduce the above display. To retrieve the main menu display, the user hits return.

D.3.3 Input Data Processing

To run DIP, the user responds to the main menu display by typing D. This results in the following display:

WHAT WOULD YOU LIKE TO DO?

WRITE THE APPROPRIATE LETTER

JUST <ENTER> TO RETURN TO MAIN MENU

R = RUN DIP
L = LIST RESULTS OF PREVIOUS RUN OF DIP

Typing R in response to this display produces the display:

IF YOU WOULD LIKE THE NETWORK REPORT, ENTER <Y>:

Typing Y (if the network report is desired) or hitting return causes execution of DIP. The display will now show:

PREPARING TO RUN PROGRAM . . .

and soon the line:

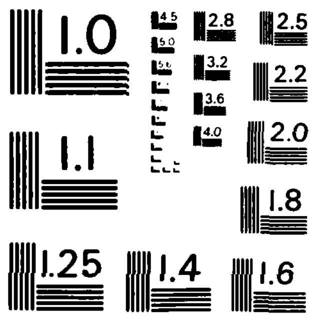
THE DIP PROGRAM IS NOW RUNNING

will be added to the display.

AD-A145 413 DESCRIPTION OF THE SOFTWARE ACQUISITION PROCESS (SWAP) MODEL(U) MITRE CORP BEDFORD MA O SHAPIRO AUG 84 MTR-9005 ESD-TR-84-151 3/3

UNCLASSIFIED F/G 9/2 NL

END
DATE FILMED
10 84
DTIC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

When the DIP program completes execution, the program output file is displayed (via the TSO LIST command). Terminating this display (with the END command) results in the display:

IF YOU WANT A HARDCOPY PRINTOUT OF THIS LIST, ENTER <Y>

Typing Y (if a hardcopy printout is desired) or just hitting return, reproduces the main menu display.

RE-LISTING DIP OUTPUT

To view the output results from the last running of DIP, the user responds L (instead of R) to the display:

WHAT WOULD YOU LIKE TO DO?

WRITE THE APPROPRIATE LETTER

JUST <ENTER> TO RETURN TO MAIN MENU

R = RUN DIP
L = LIST RESULTS OF PREVIOUS RUN OF DIP

This causes the program output file to be displayed (via the TSO LIST command). Terminating the display (with the END command) results in the display:

IF YOU WANT A HARDCOPY (PRINTOUT) OF THIS LIST, ENTER <Y>:

Typing Y (if a hardcopy printout is desired) or hitting return, will retrieve the display:

WHAT WOULD YOU LIKE TO DO?

WRITE THE APPROPRIATE LETTER

JUST <ENTER> TO RETURN TO MAIN MENU

R = RUN DIP
L = LIST RESULTS OF PREVIOUS RUN OF DIP

Hitting return will reproduce the main menu display.

D.3.4 Conducting the Simulation

To run SCP, the user responds to the main menu display by typing S. This results in the following display:

WHAT WOULD YOU LIKE TO DO?

WRITE THE APPROPRIATE LETTER

JUST <ENTER> TO RETURN TO MAIN MENU

R = RUN SCP
L = LIST RESULTS OF PREVIOUS RUN OF SCP

Typing R in response to this display produces the display:

ENTER <Y> IF YOU WOULD LIKE TO LOOK AT/EDIT THE CONTROL CARD FILE:

Typing Y in response to this display invokes the TSO full screen editor. Full screen editor commands are used to view/modify the file. Terminating this editing session (with the SE command) or hitting return in response to the display:

ENTER <Y> IF YOU WOULD LIKE TO LOOK AT/EDIT THE CONTROL CARD FILE:

produces the display:

S = RUN SCP NOW
O = MOVE BACK TO THE PREVIOUS MENU
M = RETURN TO THE MAIN MENU NOW WITHOUT RUNNING SCP

?

Typing S causes execution of SCP. The display will show:

PREPARING TO RUN PROGRAM . . .

THE SCP PROGRAM IS NOW RUNNING; THIS MAY TAKE AWHILE

When the SCP program completes execution, the program output file is displayed (via the TSO LIST command). Terminating this display (with the END command) results in the display:

IF YOU WANT A HARD COPY (PRINTOUT) OF THIS, ENTER <Y>:

Typing Y (if a hardcopy printout is desired) or hitting return, reproduces the main menu display.

RE-LISTING SCP OUTPUT

To view the output results from the last running of SCP, the user responds L (instead of R) to the display:

WHAT WOULD YOU LIKE TO DO?

WRITE THE APPROPRIATE LETTER

JUST <ENTER> TO RETURN TO MAIN MENU

R = RUN SCP
L = LIST RESULTS OF PREVIOUS RUN OF SCP

This causes the program output file to be displayed (via the TSO LIST command). Terminating the display (with the END command) results in the display:

IF YOU WANT A HARDCOPY (PRINTOUT) OF THIS, ENTER <Y>:

Typing Y (if a hardcopy printout is desired) or hitting return, reproduces the main menu display.

D.3.5 Output Report Generation

To run the ORG, the user responds to the main menu display by typing R. This results in the following display:

WHAT WOULD YOU LIKE TO DO?

WRITE THE APPROPRIATE LETTER

JUST <ENTER> TO RETURN TO MAIN MENU

R = RUN ORG
L = LIST REPORTS FROM THE PREVIOUS RUN OF ORG

Typing R in response to this display produces the display:

WHICH INPUT TO THE ORG PROCESSOR DO YOU WANT TO EDIT?

R = REPORT SELECTION
P = P/O/M SELECTION
W = WAGE & INFLATION INFORMATION

B = BEGIN RUNNING ORG NOW

O = AVOID RUNNING ORG

?

To edit any of the ORG input files (i.e., Report Selection, P/O/M Selection, Wage & Inflation Information), the user responds to this display with the desired file's letter (R, P, or W), invoking the TSO full screen editor. Full screen editor commands are used to modify the file contents. Terminating the editing session (with the SE command) reproduces the above display.

Typing B in response to the above display causes execution of ORG. The display will show:

PREPARING TO RUN PROGRAM . . .

and soon the line:

THE ORG PROGRAM IS NOW RUNNING; THIS WILL TAKE A SHORT TIME

will be added to the display.

When the ORG program completes execution, the program output file is displayed (via the TSO LIST command). Terminating this display (with the END command) results in the display:

IF YOU WANT A HARDCOPY (PRINTOUT) OF THE REPORTS, ENTER <Y>:

Typing Y (if a hardcopy printout is desired) or hitting return, reproduces the main menu display.

RE-LISTING ORG OUTPUT REPORTS

To view the output reports from the last running of ORG, the user responds L (instead of R) to the display:

WHAT WOULD YOU LIKE TO DO?

WRITE THE APPROPRIATE LETTER

JUST <ENTER> TO RETURN TO MAIN MENU

R = RUN ORG
L = LIST REPORTS FROM THE PREVIOUS RUN OF ORG

This causes the program output file to be displayed (via the TSO LIST command). Terminating the display (with the END command) results in the display:

IF YOU WANT A HARDCOPY (PRINTOUT) OF THE REPORTS, ENTER <Y>:

Typing Y (if a hardcopy printout is desired) or hitting return, will retrieve the display:

WHAT WOULD YOU LIKE TO DO?

WRITE THE APPROPRIATE LETTER

JUST <ENTER> TO RETURN TO MAIN MENU

R = RUN ORG
L = LIST REPORTS FROM THE PREVIOUS RUN OF ORG

Hitting return will reproduce the main menu.

D.3.6 Exiting the SWAP Model Program

To terminate a simulation session (and exit from the SWAP Model program), the user responds to the main menu display by typing F.

GLOSSARY

AFLC	Air Force Logistics Command
CDR	critical design review
CDRL	contract data requirements list
CER	cost estimating relationship
CI	critical item
CPC	computer program component
CPCI	computer program development item
CPDP	computer program development plan
CPFF	cost plus fixed fee
CPT&E	computer program test and evaluation
CRISP	computer resources integrated support plan
DID	data item description
DIG	development integration group
ECP	engineering change proposal
ESD	Electronic Systems Division
FIFO	first-in, first-out
FOC	full operational capability
FQT	formal qualification testing
FSD	full scale development
FSD	function sequence diagram
HIPO	hierarchical input/output
I&C	integration and checkout
ICE	independent cost estimate

GLOSSARY (Concluded)

I/O	input/output
IOC	initial operational capability
JCL	job control language
MOL	machine oriented language
OSD	operational sequence diagram
PCA	physical configuration audit
PDL	program design language
PDR	preliminary design review
PERT	program evaluation and review technique
PM	progression mode
PMR	program management review
PQT	preliminary qualification testing
PRF	problem reporting form
PSD	product specification document
QA	quality assurance
SEMP	system engineering management plan
SPO	System Program Office
SST	successor selection timing
SWAP	Software Acquisition Process (Model)
TAC	Tactical Air Command
TBD	to be determined
TEMP	test and evaluation management plan
TSO	time sharing option
USI	user interface