AD-A052 060 ARINC RESEARCH CORP ANNAPOLIS MD F/G 9/2

QUALITY ASSURANCE SUPPORT FOR SYSTEM VALIDATION MODEL - TARGET --ETC(U)

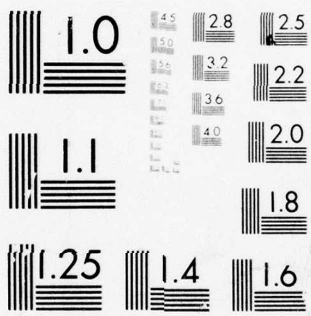
AUG 75 N00123-73-C-1698

UNCLASSIFIED 1616-11-6-1440 NL

| OF |

AD
A052 060

END
DATE
FILMED
5-78
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

Final Report

# QUALITY ASSURANCE SUPPORT FOR SYSTEM VALIDATION MODEL — TARGET DATA PROCESSOR/ COMMUNICATION PROCESSOR (SVM/TDP/CP) PROGRAM

August 1975

Prepared for

NAVAL UNDERSEA CENTER
San Diego, California

Under Contract N00123-73-C-1698





Publication 1616-11-6-1440

ARINC RESEARCH CORPORATION

13

Final Report

# QUALITY ASSURANCE SUPPORT FOR SYSTEM VALIDATION MODEL — TARGET DATA PROCESSOR/ COMMUNICATION PROCESSOR (SVM-TDP/CP) PROGRAM

August 1975

Prepared for

NAVAL UNDERSEA CENTER
San Diego, California

Under Contract N00123-73-C-1698



ARINC RESEARCH CORPORATION

CORPORATE HEADQUARTERS
2551 Riva Road
Annapolis, MD 21401

FLEET SUPPORT OFFICE
3656 Kenyon Street, Suite 7
San Diego, CA 92110

Publication 1616-11-6-1440

# ABSTRACT

Quality assurance tasks relating to computer program development for the System Validation Model of the Target Data Processor/Communication Processor are described. Conclusions and recommendations are presented concerning quality assurance provisions developed for software procurement specifications, and quality management and test planning activities.

# ABBREVIATIONS

| | |
|---|---|
| CCB | — Configuration Control Board |
| CM | — Configuration Management |
| CPCI | — Computer Program Configuration Item |
| CPDS | — Computer Program Design Specification |
| CPOS | — Communications Processor Operating System |
| CPPS | — Computer Program Performance Specification |
| CPTPL | — Computer Program Test Plan |
| CPTPR | — Computer Program Test Procedure |
| DBM | — Data Base Management |
| DRG | — Design Review Group |
| EC | — Evaluation Center |
| ECP | — Engineering Change Proposal |
| FCDSSA | — Fleet Combat Direction System Support Activity |
| IDS | — Interface Design Specification |
| L&T/CORR | — Localization and Tracking/Correlation |
| MIP/MOP | — Message Input Processor/Message Output Processor |
| MODS | — Modifications |
| NAVFAC | — Naval Facility |
| NUC | — Naval Undersea Center |
| OUF | — Operational Utility Function |
| OUR | — Operational Utility Routines (old name for OUF) |
| QA | — Quality Assurance |
| SOSUS | — Sound Surveillance System |
| SVM | — System Validation Model |
| TDP/CP | — Target Data Processor/Communication Processor |

# SUMMARY

In support of computer program development activities for the System Validation Model of the Target Data Processor/Communication Processor, ARINC Research Corporation developed quality assurance documentation and performed various other QA tasks.

Documentation developed included an SVM Master Test Plan, QA Monitoring Plan, and the QA sections of computer program performance specifications.

Quality assurance activities performed included, among others, participation in design reviews, conduct of documentation reviews, consulting services on QA management, and integration assurance studies.

The software QA activities described in this report have contributed to the establishment of control elements early in the SVM-TDP/CP program. Realization of total quality assurance will be dependent upon the implementation of these controls throughout the program.

# CONTENTS

# 1 INTRODUCTION

Under Contract N00123-73-C-1698 with the U.S. Naval Undersea Center, ARINC Research Corporation conducted quality assurance activities in support of operating and data management system software associated with the System Validation Model — Target Data Processor/Communication Processor (SVM-TDP/CP) program. These activities comprised four tasks, as identified below:

| Task | Short Title | Period of Performance |
|---|---|---|
| 11 | SVM System Quality Assurance | 11/18/74 — 6/30/75 |
| 13 | SVM Communication Software QA | 3/16/75 — 6/30/75 |
| 14 | SVM Application Software QA | 3/16/75 — 6/30/75 |
| 15 | SVM Operating Software QA | 3/16/75 — 6/30/75 |

Objectives of Task 11 were to:

a. Develop quality assurance provisions for TDP/CP computer program performance specifications.

b. Prepare a TDP/CP Master Test Program Plan.

c. Investigate the diagnostic/fault-isolation capability required for the TDP/CP.

d. Assist in the development of the SVM Integration and Integration Test Plan.

Tasks 13 through 15 had identical objectives:

a. Prepare plans and procedures to assure the quality of communications software.

b. Prepare, review, and/or maintain software quality documentation.

c. Assist in test planning and evaluation for communication system software.

d. Perform quality design review, both formally and informally, to assure that internal quality standards are maintained for communication software.

The four tasks had several common areas of activity — design review, documentation review, specification development, and test planning. Results of work performed in these areas are discussed in Section 2. Further activities specifically related to each task are described in Section 3. Recommendations arising from the overall study are presented in Section 4.

# 2
# COMMON TASK ELEMENTS

## 2.1 DESIGN REVIEW

As a member of the Design Review Group/Configuration Control Board (DRG/CCB), ARINC Research Corporation participated in the design review activities described in Appendix A of the TDP/CP Configuration Management Plan. The Corporation's activities principally involved reviewing computer program configuration items (CPCIs).

During the Corporation's participation on the DRG/CCB, the computer program performance specifications (CPPSs) for most programs in the SVM project were reviewed and established as baselines for product development. Also, certain of the system interface documents, computer program design specifications (CPDSs), and common data base design documents (CDBDDs) were reviewed.

All errors, inconsistencies, and comments were documented and, if adequate lead time was available, the documentation was presented to the Software Configuration Management Office before the formal review meeting was held to aid the originator of the document in preparing for the review. Since the Corporation's major concern was software quality assurance, the subject material was reviewed with emphasis on consistency with previously reviewed documents and adherence to specified formats rather than on a mathematical analysis of algorithms. The specified format for most of the documents was WS-8506, Requirements for Digital Computer Program Documentation.

CPPSs and interface documents were checked to verify that they covered all requirements specified in the program definition material.* The documents were

*Which included:

(1) NAVELEX PME 124-22, SVM Functional Description (U), SECRET, 16 December 1974.

(2) NAVELEX PME 124-22, System Design, Phase I SOSUS Backfit (MEC/SVM) (U), SECRET, 29 January 1975.

(3) NAVELEX, System Operating Concept, Phase I SOSUS Backfit (MEC/SVM) (U), SECRET, 20 December 1974.

also checked against other CPPSs and interface design specifications (IDSs) referenced in the program definition material to verify that all interfaces were consistent.

The CPPSs were reviewed for internal consistency using WS-8506 as a guideline. The sections presenting the functional descriptions of the programs were checked for input/output traceability; i.e., all specified inputs to the functions were used to produce the specified outputs.

Terminology in the CPPS description of functions was checked for accuracy and consistency. Also, since the documents were to be used in establishing contractual requirements for SVM computer programs, the clarity of the functional descriptions was evaluated. If a requirement was described so that a design approach could not easily be defined or that a method of testing was not obvious, a comment to that effect was made.

The Quality Assurance sections of the CPPSs were examined for adequacy of test methodology and design requirements.

## 2.2 PROJECT MANAGEMENT DOCUMENTATION REVIEW

ARINC Research was also responsible for reviewing other documents relating to SVM project management, such as the Verification and Validation Plan, Master Test Plan, and Standards and Guidelines Document. These documents were checked to determine if they adequately covered their stated goals. The Corporation's comments and criticisms were forwarded directly to project management.

## 2.3 SPECIFICATION DEVELOPMENT

### 2.3.1 General

ARINC Research was responsible for writing the Quality Assurance sections of certain computer program performance specifications used in the SVM project. This responsibility included the QA sections for the entire documentation packages for the data base management program and some of the applications programs. The content of these sections is defined in WS-8506.

Since software quality and reliability are presently a matter of general concern, the importance of the Quality Assurance portions of the specifications is evident. No requirement is meaningful in any specification unless it can be verified. As an independent group, the Corporation attempted to present an objective approach to assuring that the software being developed met the goals stated in the system definition documents.

### 2.3.2 Performance Specifications

ARINC Research worked closely with the SVM teams developing the CPPSs. The Corporation's major role was to write the Quality Assurance section for these documents. WS-8506 limits this section to a discussion of test methodology for the program. More specifically, the requirements for this section are the definition of:

a. Levels of testing

b. Test requirements for each level

c. Acceptance test requirements.

Since the test approach was to be similar for all programs in the system, a general (baseline) test methodology description was generated for inclusion in all CPPSs. This baseline specifies the requirements for computer program test plan and test procedures documents to be written in accordance with WS-8506. These documents cover three of the four levels of testing — computer subprogram, computer program, and computer program acceptance. The QA section states that system integration testing is to be conducted by NUC and an independent verification/validation agency, and not to be covered in individual test plans and procedures.

The baseline also presents requirements for subprogram and program testing, including test methodology, tools and facilities, and reporting. Similar requirements are given for acceptance testing, including a table specifying the acceptance criteria for each requirement in the detailed functional description portion of the CPPS. This format complies with all WS-8506 requirements, and was approved by the Design Review Group/Configuration Control Board and by project management.

After the baseline was developed, the Corporation's activity centered on generating acceptance criteria tables for each CPPS. Each requirement specified in the functional description was listed, together with a statement as to how it had to be verified to pass acceptance testing. The basic approach was to state what must be input to the function and to verify that the expected output occurred. Also, for requirements that could not be thoroughly tested by this approach (i.e., requirements "buried" in subfunctions), the statement was made that the formal results of subprogram testing would be required as part of acceptance testing. For the data base management program, which performs services for other programs, the approach was to combine sets of service requests and to verify that the expected combination of events occurred. For each CPPS the wording of the baseline was modified to be appropriate and the table of acceptance criteria was appended. This section was then delivered to the developing group for inclusion in the publication.

### 2.3.3 Design Specifications

ARINC Research was tasked to write the Quality Assurance sections of the following CPDSs: data base management system, localization and tracking program, correlation program, and operational utility functions. The activity in this area was terminated before any of these sections were delivered, and therefore the resultant recommendations (see Section 4) concerning this area are based on intended rather than actual accomplishments. These suggestions should prove helpful in the development of design specifications containing adequate quality assurance provisions.

### 2.3.4 Test and Evaluation

Under Tasks 14 and 15, work had been initiated to develop an approach to the establishment of computer program test plans (CPTPLs) and computer program test procedures (CPTPRs). These CPTPLs and CPTPRs were to address the testing of those computer programs developed at NUC. This activity did not proceed much beyond the planning stage when the tasks were terminated. However, because some planning had been completed and various pertinent discussions had taken place, it was agreed that documentation of the study rationale and approach might prove beneficial to the SVM program or on some future project. Appendix A contains a discussion of test and evaluation as a software assurance tool.

# 3
# SPECIFIC TASK ACTIVITIES

Specific activities conducted under Tasks 11, 13, 14, and 15 are discussed in this section.

## 3.1 TASK 11 — PROJECT SUPPORT

### 3.1.1 QA Provision for CPPSs

The following CPPS Quality Assurance sections were prepared and submitted under the indicated titles.

a. EXEC — AN/UYK-7 (SHARE 7 MODS)
b. DBM — AN/UYK-7
c. EXEC — AN/UYK-20 (CPOS)
d. DBM — AN/UYK-20 (CPOS)
e. L&T and CORR/CLASS (L&T CORR)
f. Resource Allocation Program
g. COMM Processor (C/P NAVFAC)
h. Interactive Display (EC Display)
i. EC MIP/MOP
j. Operation utility routines.

### 3.1.2 QA Monitoring Plan

A Quality Assurance Monitoring Plan was prepared and submitted on 13 March 1975.

### 3.1.3 TDP/CP Master Test Program Plan

The TDP/CP Master Test Program Plan was submitted in draft form in December 1974, and in final form in February 1975.

### 3.1.4 Diagnostic/Fault Isolation

The requirement to investigate the diagnostic/fault isolation capability required for the TDP/CP was verbally deleted from the task by SVM TDP/CP management since the information needed to perform the task was not available. The effort was redirected to participating as a member of the Design Review Group, and to the preparation of event calendars, schedules, and a progress reporting system.

### 3.1.5 SVM Integration and Integration Test Plan

ARINC Research participated in the activities associated with the development of the SVM Integration and Integration Test Plan.

## 3.2 TASK 13 — COMMUNICATIONS/SCHEDULING SUPPORT

Task 13 was initially directed to the preparation of quality assurance documentation for communications software; redirected to the development of an automated scheduling system; then further redirected to participation as a member of the Design Review Group and to the review of project documentation.

## 3.3 TASK 14 — APPLICATIONS SOFTWARE SUPPORT

Activities under Task 14 are summarized below.

a. Prepare plans and procedures to assure the quality of applications software. A Quality Assurance Monitoring Plan was prepared for the computer application programs, and submitted informally.

b. Prepare, review, and maintain software quality documentation. Complete revisions of all CPPSs for the application programs were prepared, coordinated, and submitted.

c. Assist in test planning and evaluation. This activity was terminated shortly after it was undertaken. No deliverable items were produced.

d. Perform quality design review. Informal design assistance and review were performed on several application programs. The main activity was on the Resource Allocation CPPS.

e. Prepare reports periodically on software quality. Monthly letter reports were prepared and submitted covering the software quality activity.

## 3.4 TASK 15 — SYSTEM SOFTWARE SUPPORT

Activities under Task 15 are summarized below.

a. Prepare plans and procedures to assure the quality of system software. A Quality Assurance Monitoring Plan was prepared for the system software and submitted informally. A quality assurance checklist for in-house contractors was developed and submitted on 27 March.

b. Prepare, review, and maintain software quality documentation. A complete revision of the DBM Quality Assurance section was prepared.

c. Assist in test planning and evaluation. This task was terminated shortly after it was undertaken.

d. Perform quality design review. Informal design reviews were conducted during meetings on the DBM program.

e. Prepare reports on system software quality. Monthly letter reports were prepared and submitted covering the software quality activity.

## 3.5 TASK FULFILLMENT

Required contract tasks were completed and accepted by NUC. Documents delivered are listed in Appendix B.

The software quality assurance activities described in this report have contributed to the establishment of control elements early in the SVM-TDP/CP program. Realization of total quality assurance will be dependent upon the implementation of these controls throughout the program.

Quality assurance provisions and program control will be of particular importance during SVM system integration and SOSUS backfit. For this reason, certain recommendations are presented in Section 4 that may be of value to the program in the future.

# 4
# CONCLUSIONS AND RECOMMENDATIONS

Many recommendations concerning quality assurance were offered by ARINC Research during its participation in the SVM-TDP/CP program. These recommendations were documented in design review and various program meetings. Some of the more important recommendations, together with their rationale, are provided in the following paragraphs.

## 4.1 PERFORMANCE SPECIFICATION QUALITY ASSURANCE

ARINC Research observed during this study that the QA section was not usually considered an important part of a CPPS. Generally, little time was allowed between the completion of a functional description and the scheduled time for its publication. Since the requirements stated in the functional description were necessary for the acceptance criteria, there was thus inadequate time to prepare comprehensive test requirements. Further, the review process paid little attention to the Quality Assurance sections. The comments on these sections were found to deal more with level of detail rather than substance.

The Quality Assurance section, if adequately prepared, is of benefit to program development in some other ways. First, it is an aid to reviewing requirements and determining if a requirement is well enough defined that a test can be constructed for it. The acceptance criteria tables are also very useful guidelines in the development of test plans and procedures. Where requirements generally specified normal or nominal operation, the acceptance criteria included abnormal and limit conditions that might be seen in actual operation. Far from being an unimportant part of a specification, the Quality Assurance section should be considered as important as the statment of requirements.

## 4.2 DESIGN SPECIFICATION QUALITY ASSURANCE

The requirements for the Quality Assurance section of the design specification, as stated in WS-8506, are to "define the review processes for verification of the computer program design". The formal method by which the contracting agency reviews the design specification is described in the Configuration Management Plan. The QA section of the design specification should contain at least a brief summary of these requirements, with reference to that management plan. However, such a minimal approach seems to leave the designers open to the possibility of major rewrites of the document. If less formal reviewing procedures are not required during the design process, the DRG/CCB review is likely to find more major problems — or, worse yet, not find them.

Many of these problems can be avoided with an internal design review that would follow the concept of the "structured walk-through". The Quality Assurance section of the CPDS should specify that the design group conduct these reviews periodically during the design of a computer program. The review team would consist of representatives of both the design group and quality assurance team. The reviews would cover a portion of the design work of one individual in the design group. A formal presentation of the elements of the design, to an appropriate level of detail exposing known areas of uncertainty, would be made by the individual.

Error avoidance and error detection would be the major concern of the reviewers, who would prepare for the review by studying the section of the design being covered. The reviewee would be responsible for following up on suggestions and reporting to the reviewers the actions taken. Internal Quality Assurance would chair the review, record the proceedings, and monitor actions. In this way a large percentage of the minor design errors and interface inconsistencies would be discovered and corrected before the lengthy formal review process begins. A firm basis for the formal design reviews would thereby be developed.

Individual design requirements could be verified by requiring specific sub-program level tests if necessary. Critical processing steps and results should be checked. Visual review of computer codes, desk checking, and error analysis should be documented.

A complete design description document should result from the CPDS development effort, with the design logic and the method of verifying that logic disclosed.

## 4.3 DESIGN REVIEW QUALITY ASSURANCE

Active involvement by ARINC Research in the design review procedures allowed the observation of some problems associated with the approach taken in document review. First, many of the documents were not ready for a formal project-level review and the DRG/CCB had to suggest many major changes to the material. In many instances, insufficient time was allowed for the developing group to produce an organized and accurate document. The rush to get documents signed off may have actually cost time because of the many complete rewrites required to generate an adequate product.

It is generally agreed that if more time is spent in the specification and design phase of a project, less time will be spent in coding and debugging and fewer problems will occur during integration. One possible way to improve this situation would be to require the developing group to review its product more extensively before it is distributed to the rest of the project members. The extra time and effort put into the design documents would be to the benefit of the project over the long run.

Another problem was observed with regard to the strict use of WS-8506 as a formatting guide when the system functions were partitioned into the various specifications. WS-8506 is inadequate for specifying an overall system architecture that would describe how all system components fit together. That is, while the format is sufficient for describing the processing of the functions, problems arise describing interfaces to other functions within the system. Quite often during the documentation review, contradictory information was found in the external-interface areas of the specifications.

A preferable approach to the use of WS-8506 would have been to initially develop a system requirements document from the program definition documents, after which the individual functions could be more easily specified. By this approach, the system as well as the functions would have a top-down design, and the review group would have a more organized review schedule. Functions subordinate to others would have their specification developed and reviewed after those of the others, which would greatly reduce the interface confusion.

It was observed that each reviewer had his own approach to reviewing the documents. For example, some concentrated only on editorial problems, while others were interested only in technical matters. This situation may have caused some problems to be overlooked. To assure that all reviewers cover all aspects of the document, a checklist of review procedures could be developed and the reviewers required to follow its guidance. Alternatively, reviewers could be assigned areas of concentration with appropriate checklists to assure in-depth review of all areas.

# APPENDIX A

# ROLE OF TEST AND EVALUATION IN SOFTWARE ASSURANCE

This appendix contains a general discussion of test and evaluation as a software assurance tool. The thoughts expressed are based both on observations made during this program, and past experience of ARINC Research personnel. Topics covered are benefits of testing (Section A.1), test planning (Section A.2), test procedures (Section A.3), test evaluation (Section A.4), and data collection (Section A.5).

## A.1 BENEFITS OF TESTING

Quality assurance can comprise many steps in product monitoring during the production cycle. However, unless the final product is tested in accordance with a well organized plan for both standard and nonstandard conditions, the acquiring agency has no assurance as to its performance.

Two factors commonly mitigate against the effectiveness of testing. First, the test function is often viewed by programmers as an annoying obstacle that must be endured as a prerequisite to the delivery of the product. Second, management is often working against a time constraint in getting the completed product into the hands of the user, and the test cycle is viewed as a costly and time consuming effort that causes delays and possible late deliveries.

In actuality, a well run test program can have several benefits:

a. It gives management confidence in the product. Although there may be short-run effects in terms of delays, the long-range benefits are a reputation for delivery of a good product, fewer return visits to fix program errors, and more time for development of new programs because less manpower is required for maintenance.

b. The delivery team has confidence in the product, and consequently is better able to isolate software and hardware problems at the site.

c. The user gains confidence in the product, based upon the confidence and assurance evidenced by the delivery team. This confidence is further reinforced after the team's departure because the program continues to perform as expected.

## A.2 TEST PLANNING

### A.2.1 Purpose

The purpose of test plans and procedures is to develop a set of test items which, if completed satisfactorily, will guarantee that the program will not only meet acceptance standards but ensure a low error rate after delivery. (The term "program", as used here, can be either a program, a module, or a function — that is, a clearly defined and testable unit.)

The program test plan must be detailed enough to ensure that the acceptance criteria can be met; must guarantee that the program functions work, no matter how varied the input within the specified limits; and must ensure that the program is as close to fail-safe as possible. At this stage the test plan fulfills three functions:

a. Defines external interface needs
b. Helps to identify faults in documentation by forcing a closer scrutiny than normally required by review
c. Documents for management review the intent of the test procedures so that management may add or delete tests before they actually begin.

Thus a concensus is reached as to test methodology, allocation of computer time, manpower requirements, simulation planning, and the schedule of testing. As the final step, program acceptance test demonstrates to the acquiring agency that all functions work properly.

### A.2.2 Methodology

ARINC Research identified the following areas of concern in relation to SVM test planning:

a. Test methodology
b. Documentation requirements necessary to support the methodology
c. Method of test partitioning
d. Operating system interfaces
e. Whether the quality assurance group would review-code
f. Scheduling constraints, including system availability.

A few comments on the test methodology for this project are in order. The code would be developed according to the project's top-down design methodology, and then a bottom-up procedure would be used for subprogram and program testing.

Bottom-up testing starts at the lowest level of the program heirarchy, with each module tested separately and independently. Testing would then proceed by building upon the tested modules until the entire program had been tested.

Inherent to this approach is the high probability that serious problems will not be discovered until late in the test cycle, and as a consequence any redesign necessary could have a "ripple" effect in terms of previously tested items as well as the delivery schedule. The fact that top-down design was employed minimizes the probability to a certain extent; however, it is still higher than if top-down testing had been employed.

Top-down testing parallels the development of the program while concurrently testing the program at each stage of development. The advantages of this approach are that complex problems are usually revealed early in the test cycle rather than at a time which would necessitate extensive redesign and consequent overall program delivery delays. Each module is tested in the total program environment, and thus interface problems and design concept errors are more readily revealed.

Once the test philosophy was decided upon, the programs were analyzed, and the methodology outlined in Table A-1 was established. ARINC Research felt that this structured approach would minimize the probability of program errors occurring late in development.

The algorithms mentioned under "Subprogram" in Table A-1 involve such functions as the various screening calculations crucial to the proper performance of the function itself and essential to the operation of the system. These calculations can be tested much more thoroughly at the subprogram level, and the output of each measured directly. As noted in Table A-1, the algorithms are first tested individually, and then combined and tested as a unit through the major functions. Because their individual outputs have been previously tested and are known to work at this point, fault isolation is simplified.

### A.2.3 Documentation Requirements

A minimum requirement for writing the SVM test plan is a computer program performance specification. The CPPS should specify the equipment requirements, interfaces (both software and hardware), and acceptance criteria. The CPPS should be available prior to its final version to provide details on limits, alerts, message formats, etc.

TABLE A-1. LEVEL OF TESTING AND METHODOLOGY

| Level | L&T | Correlation | OUR |
|---|---|---|---|
| Subprogram | Subtest algorithms | Test algorithms. Test the testing subfunctions (i.e., score derivation). | Algorithm testing |
| Program | 12 major functions, exercising all subfunctions | 3 functions: 1) Test scores 2) Linking 3) Decorrelation | 21 functions while exercising subfunctions |
| Acceptance | Same as above | Same as above | Same as above |

After completion of the preliminary test plan, external interface requirements can be identified and design work can begin. After the design specification becomes available, the next task is to compare it against the test plan to determine if any test sequences need to be added or modified; at what level this is to take place (subprogram or program); and how the simulation requirements are impacted.

The design specification provides amplification as to:

a. Legality checks (i.e., validation of functional input)

b. Flow charts (i.e., amplification as to how the processing will be handled)

c. Message formats

d. Functional partitioning

Items a and b usually result in only minor additions or modifications. Item c is important primarily in terms of data extraction requirements for the simulator, but can also be useful in determining the level at which an item can be most efficiently tested.

Item d can have major ramifications by allocating items considered to be at the functional level to the subfunctional level, or by dividing what was considered to be an independent function in the CPPS into two separate functions. Although these ramifications might cause a major reorganization in certain areas of the test plan, ARINC Research decided that it would not be prudent to wait until this point before starting the plan. As mentioned earlier, the test plan is primarily an outline, and thus reorganizing it at this point to meet the requirements was considered to be far more efficient in terms of scheduling. The major test design work would be completed, and therefore although the design specification could force the reorganization or deletion of a particular test sequence the major portion of the test plan and simulation design work would be completed at this point.

### A.2.4 Scheduling

Three primary variables are involved in scheduling. In order of criticality they are:

a. Timely delivery of documentation (CPPSs and CPDSs)
b. Availability of simulator and accompanying documentation
c. Availability of time for documentation rewrites and updates.

All of these items are necessary in writing a comprehensive test procedure. The need for the documentation is apparent. The time required to revise the test plan based either upon changes to the CPPSs and CPDSs or procedural shortcomings revealed in the testing cycle is often overlooked. An allowance should be made to cover the probability of this occurrence. This can be done by allowing two to three weeks after the completion of testing until the start of delivery. In this manner it does not have to be included in the cost of the contract unless needed. It is easier to move a delivery date up than back.

### A.2.5 Test Driver Considerations

As soon as the test plan is well enough defined that test driver needs can be identified, work should be started on the test driver program.

The input driver can utilize selected target data collected from actual operation, prepacked data generated locally, or modifiable real-world data.

#### A.2.5.1 Operational Data

Operational data has the advantage of representing inputs as they actually occur. The source (i.e., classification and confidence, etc.) of the data is known, and therefore the expected output can readily be determined. This type of data represents the closest to real-world conditions, and this is a good test of the system.

Unfortunately, operational data has distinct disadvantages. First, it would require numerous valid traces, each representing a given input condition (e.g., time trace, classification and confidence, frequency, etc.). Second, these traces would have to be both selectable and identifiable so that the tester would be able to isolate problems. Third, the bulk of the resultant driver would be prohibitive and unwieldy.

A.2.5.2 Locally Generated Data

A modifiable driver, whereby the tester can structure his inputs to satisfy a given test sequence, would be ideal. The tester then knows exactly what is being input. He can generate invalid conditions to satisfy legality checks, and since he is doing the formatting rather than relying upon preformatted messages, the bulk of the simulator is considerably reduced.

Some testers contend, though, that the inputs provided by this type of simulator are too pure to represent the real world; and the fact that the program responds correctly under pure conditions does not mean that it will perform as well under the impure conditions of real-world testing.

A.2.5.3 Modifiable Real-World Data

The third type of driver utilizes real-world inputs that represent the basic conditions and which can be modified to create any of the other input conditions required. This would be bulkier than the message generator above, but would more closely approximate the conditions encountered in the field. The difficulties would be 1) establishing the degree of cleanliness that these messages must meet to adequately represent the "typical" field input, and 2) based upon this standard, selecting the representative messages from field data that meet the criteria.

The choice is clearly between the second and third option, and has not yet been resolved. Once the decision is reached, then other concurrent simulation requirements must be defined. One such requirement is extraction. To maintain a log of events tested, and the input and output, there must exist some means of extraction and reduction. This enables the tester to examine the data at each critical point (as determined during the test plan phase) of the processing to ensure that the output is as expected. The hard copy provides useful documentation for logging test shots as well as providing programmers a useful aid for debugging purposes.

## A.3 TEST PROCEDURES

Test procedures are detailed instructions based on the test plan for the program. These procedures specify manpower and equipment requirements, the type of simulation to be used, and the expected output of each test step. Each test procedure comprises a series of steps that refer to a requirement specified in the CPPS. The level of detail needed for the test procedures necessitates that as a prerequisite the simulation documentation be written (including the operators manual), the applicable CPDSs and CPPSs be available, the interfaces and message formats be clearly defined, and the test plans be written.

The experience level to which the test is to be written must be specified. It is often assumed that those writing the procedure will be running them. Naturally, this is not always so. The degree of tester experience could very well vary over the life-span of the test procedure. Another consideration in the structure of the procedures is that the test sequences must be as independent as possible to allow for test interruptions (e.g., equipment or software problems), retest for patching, and for selective retest of fixes. If these constraints are met, then it becomes merely a matter of adding detailed operational steps to the test plan. The detail should specify the number of people required, the location of the input and output (i.e., console, teletype, card deck, etc.), and the steps (both simulator and console) needed to complete the test sequence. A sample portion of a test procedure appears in Table A-2. From this table it is evident that a simple CPPS requirement may have several implications and may require several reiterations in the test plans and procedures.

## A.4 EVALUATION OF TESTS

For the SVM project, ARINC Research Corporation's anticipated participation during the test cycle was to review subprogram test results, perform document program tests, and observe acceptance tests. However, ARINC Research was able to provide only limited assistance in these areas since the drivers and program listings were not available, equipment needs and availability were not established, and the Corporation's participation in the SVM program ended while these activities were still in progress or not yet undertaken.

Since the scope of subprogram testing was small, and since ARINC Research was thoroughly familiar with the test requirements (having written the subprogram test plan), the Corporation's involvement was limited to reviewing the test results.

ARINC Research anticipated full participation in program level testing. The intent was to further verify the testing that took place at the subprogram level, and at the same time assure that all program tests were conducted in accordance with the test plan. However, the value of such quality assurance participation in the development of test plan and procedures was never resolved, since the Corporation's effort was terminated prior to the commencement of test plan development.

## A.5 DATA COLLECTION

Test documentation should describe encountered problems (both hardware and software), an evaluation of the seriousness of the problems, and any fixes accepted. These test notes should be numbered and annotated for historical purposes.

In addition to the test notes, a careful record in the form of a software trouble (or incident) report should be kept of the program discrepancies that occur. Not only are these critical from the historical standpoint, but they are also necessary for problem regeneration, problem isolation by the programmer, and patch evaluation by the tester. In view of this function, the trouble report should detail the following:

a. Run time (in all cases)
b. Program version and patches in core
c. Whether or not the problem was repeatable (after restart? after reload?)
d. Steps necessary to recreate the problem
e. Simulation required
f. In the event of a stop, the type of stop and critical register readings
g. In the event a core dump is taken, the type of dump and core limits.

These reports should be evaluated and filed, both serially and by the nature of the problem. These historical data are valuable in terms of updating test plans/ procedures, in retest after a recompile, or in fault isolation in future builds.

Evaluation is usually based upon the degree that operational capability is impaired. Usually there are three to four levels, ranging from "unable to fulfill mission" to "negligible".

Equipment failure logs are important from the management standpoint for determining how much time was lost due to equipment failure and consequently determining the true cost of the test effort.

This system enhances program status evaluation in that the problems can be listed by short title under the various evaluation categories. Thus a weekly report can be given to the project manager so that he can determine how close to delivery he is by the number of serious outstanding problems. Furthermore if this is done for each program, then the project manager can determine if one program support effort appears to need more attention than another. A final benefit would be that summary reports are simplified in that cost can be determined in terms of peak effort, time lost due to hardware problems, time lost due to software problems (development costs), and overall costs.

TABLE A-2. SAMPLE TEST PLAN (Sheet 1 of 6)

| Seq. No. Step No. | Requirement |
|---|---|
| A. CPPS PARA. 3.3.4 | |
| — | System List. This subfunction shall specify the listing of star target files based upon one of six options. The options include:<br>1. All targets<br>2. All except the constituents of specified EC files<br>3. All except specified star target files<br>4. All targets from specified station arrays<br>5. All targets from specified station<br>6. Targets held or held and lost during a given time span.<br><br>Two additional options are available: 1) number of most recent reports to be presented (cannot be used with option six above); 2) start or start/stop time of interval of interest (cannot be used with option one). |
| 010.000 | Limit Checks. These tests check the subfunction's rejection of entries which exceed the upper or lower limits established for a given input. |
| 010.010 | Option indicator |
| 010.011 | Input a selection of 0 or of the equivalent lower limit. Verify ERROR message generated. |
| 010.012 | Input a selection of 7 or the equivalent of one greater than maximum entry. |
| 010.020 | All targets option |
| 010.021 | Make an input that is any single character entry other than the one specified (character to be defined). Verify ERROR message generated. |
| 010.030 | FC file number limits (Option 2) |
| 010.031 | Specify ANEC file number of 00000. Verify ERROR message sent. |

TABLE A-2. (Sheet 2 of 6)

| Seq. No. Step No. | Requirement |
|---|---|
| | A. CPPS PARA. 3.3.4 (Continued) |
| 010.032 | Specify EC file number that exceeds the upper limit for EC file numbers. |
| 010.033 | Specify Option 2 and Input 10 file numbers. Verify ERROR message generated. (This check might be a function of display; this is not clear at this time.) |
| 010.040 | Star target option limits test star target file entries in the same manner as EC file procedure 010.030. |
| 010.050 | Station array limits |
| 010.051 | Test station array data as specified in Step 010.030. |
| 010.060 | Station entries (options) |
| 010.061 | Test station entries as specified in Step 010.030. |
| 010.070 | Time span option (Option 6). |
| 010.071 | Specify entry of 0 hours, 0 minutes. Verify ERROR message sent. |
| 010.072 | Specify a time span of XX hours, 60 minutes, where XX is a legal entry. Verify ERROR code sent. |
| 010.073 | Test for maximum entry (to be specified). |
| 010.074 | Test for maximum entry plus one. Verify ERROR message sent. |
| 010.080 | Number of most recent reports. |
| 010.081 | After selecting entry, specify zero recent reports. |
| 010.082 | Specify maximum number of recent reports. |
| 010.083 | Specify maximum number of recent reports plus one. Verify ERROR message sent. |
| 010.090 | Time limits |
| 010.091 | Select Option 6 (time span) and select start/stop time option. Verify ERROR message sent. |
| 010.092 | Select time limit option and specify month 00. Verify ERROR message sent. |

TABLE A-2. (Sheet 3 of 6)

| Seq. No. Step No. | Requirement |
|---|---|
| A. CPPS PARA. 3.3.4 (Continued) | |
| 010.093 | Repeat above, specifying month 13. Verify ERROR message sent. |
| 010.094 | Specify legal month and day 00. Verify ERROR message sent. |
| 010.095 | Specify legal month and day 32. Verify ERROR message sent (note: if ERROR routine is refined to the month/day relationship, try to select illegal month-day combination, e.g., 2/30). |
| 010.096 | Specify legal month/day but specify time 2360. Verify ERROR message sent. |
| 010.097 | Specify legal month/day and time of 0000.0. Verify entry is accepted. |
| 010.100 | Stop time |
| 010.101 | Specify start time and then test stop time in accordance with plan specified in 010.090. |
| 020.000 | EC file input |
| 020.021 | Specify nonexistent file numbers as candidates for omission. Should generate ERROR message. |
| 020.022 | List valid EC file numbers for omission and ensure (via extraction) that they are omitted. |
| 020.030 | Repeat 020.022, specifying the number of most recent reports. |
| 020.031 | Structure so that less than the number of recent reports specified are available. Should list those available. |
| 020.032 | Structure so that more than the number specified are available. Should list only the number specified and in order of recency. |
| 020.040 | Start time. |
| 020.041 | Repeat 020.022, specifying start time of interval of interest. |
| 020.042 | Specify the input such that no reports exist after the time specified. |
| 020.043 | Specify the input such that some are previous to the time specified and some are after start time. Ensure that only those with times later than start time are listed. |

TABLE A-2. (Sheet 4 of 6)

| Seq. No. Step No. | Requirement |
|---|---|
| | A. CPPS PARA. 3.3.4 (Continued) |
| 020.050 | Repeat Step 020.022, specifying start and stop time. |
| 020.051 | Specify time constraints such that stop time is earlier than start time. Should get ERROR message. |
| 020.052 | Specify time constraints such that all targets are outside the time constraints specified. Verify negative report. |
| 020.053 | Specify time constraints such that only a limited number are within the time constraints specified. Verify only those targets extracted. |
| 020.054 | Specify a start/stop time which crosses the month limit, e.g., 06/30/2359.9 — 07/01/2200.0. Verify ERROR message is not sent. (Verifies month check.) |
| 020.055 | Specify time constraints which cross midnight, e.g., 06/29/2300.0 — 06/30/0001. Verify ERROR message not sent. (Checks day differential.) |
| 030.000 | Star target files as input. |
| 030.001 | Repeat Seq. No. 020.000 in its entirety, specifying star target files vice EC files. |
| 040.000 | Station arrays as input. |
| 040.010 | Repeat Seq. No. 020.000 utilizing station arrays vice EC files. |
| 050.000 | Station number as input. |
| 050.010 | Repeat 020.000 utilizing station numbers vice EC files. |
| 060.000 | Targets held or lost during a given time span. |
| 060.010 | Repeat Sequence 020 utilizing time span. Verify ERROR message procedure 020.030, as these two options are mutually exclusive. |
| | B. CPPS PARA. 3.3.5 |
| — | Classification search. The classification search routine will search 1) star target files, 2) EC target files, or 3) EC abstracts for targets of a given classification/confidence. |
| 010.000 | Limit checks. |

TABLE A-2. (Sheet 5 of 6)

| Seq. No. Step No. | Requirement |
|---|---|
| B. CPPS PARA. 3.3.5 (Continued) | |
| 010.010 | File selection. |
| 010.011 | Test for inputs of 0 or 4 (or whatever values are used for source selection). |
| 010.020 | Classification/confidence entry |
| 010.021 | Specify an input of 00X where X is a valid confidence level. Verify ERROR message sent; no classification specified. |
| 010.022 | Specify an input of XXY (assuming Y to be an invalid confidence level, XX being a valid classification). Verify ERROR message generated. |
| 010.023 | Specify (XX+1)X, where (XX+1) is one greater than valid classification limit, and X = valid confidence level. Verify ERROR message sent (assuming (XX+1) = 99). |
| 020.000 | Search processing |
| 020.010 | Star target files |
| 020.011 | Specify input such that none of the file data meets the classification/confidence specified. |
| 020.012 | Specify input such that classification is satisfied but confidence does not meet specified criteria. If function screens for both classification and confidence, then a "no data found" message should result. |
| 020.013 | Repeat above step, specifying input such that confidence level criteria are met but classification criteria are not satisfied. Result should be same. |
| 020.014 | Specify a general classification (e.g., no confidence level specified). Verify that star targets meeting that classification level are listed (irrespective of confidence level; may be done via data extraction/reduction). Verify EC header retrieved from DBM. May be done via data extraction/reduction. |
| 020.015 | Specify a specific classification/confidence level such that data does exist within the area specified. Verify via extraction that data are made available for list. Verify that EC headers are extracted for each file selected. May be done via data extraction/reduction. |
| 020.020 | EC target file search |

TABLE A-2. (Sheet 6 of 6)

| Seq. No. Step No. | Requirement |
|---|---|
| | B. CPPS PARA. 3.3.5 (Continued) |
| 020.021 | Repeat 020.010 series utilizing EC targets. |
| 020.030 | Repeat 020.010 series specifying EC abstracts. |

NOTE:

This sample test plan is based upon selected excerpts from the requirements specified in the SVM Operational Utilities Performance Specification. Some of the limit checks might be assigned to the input function at design time and thus would be tested under that function vice the OUR (OUF) function. In all cases the results can be verified by extraction/reduction.

# APPENDIX B
# DELIVERED DOCUMENTS

1. Quality Assurance Sections
   a. EXEC — AN/UYK-7 (SHARE 7 MODS)
   b. DBM — AN/UYK-7
   c. EXEC — AN/UYK-20 (CPOS)
   d. DBM — AN/UYK-20 (CPOS)
   e. L&T and CORR/CLASS (L&T CORR)
   f. R/A Program (Resource Allocation)
   g. COMM Processor (C/P NAVFAC)
   h. Interactive Display (EC Display)
   i. EC MIP/MOP
   j. Operation utility routines
2. Quality Assurance Monitoring Plan
3. Master Test Program Plan
4. Quality Assurance Monitoring Plan for Applications Programs
5. Quality Assurance Monitoring Plan for System Software
6. Checklist for In-House Contractor Quality Assurance
7. Monthly letter reports
8. Technical comments on CPPSs and interface specification, delivered informally during the program.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

# REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

**1. REPORT NUMBER:** (14) 1616-11-6-1440

**2. GOVT ACCESSION NO.:**

**3. RECIPIENT'S CATALOG NUMBER:** Final rept.

**4. TITLE (and Subtitle):** (6) QUALITY ASSURANCE SUPPORT FOR SYSTEM VALIDATION MODEL - TARGET DATA PROCESSOR/COMMUNICATION PROCESSOR(SVM/TDP/CP) PROGRAM.

**5. TYPE OF REPORT & PERIOD COVERED:**

**6. PERFORMING ORG. REPORT NUMBER:** 1616-11-6-1440

**7. AUTHOR(s):** NOT LISTED

**8. CONTRACT OR GRANT NUMBER(s):** (15) N00123-73-C-1698

**9. PERFORMING ORGANIZATION NAME AND ADDRESS:**
ARINC Research Corp.
2551 Riva Road
Annapolis, Maryland 21401

**10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:** (12) 35p.

**11. CONTROLLING OFFICE NAME AND ADDRESS:**
NAVAL UNDERSEA CENTER
San Diego, California

**12. REPORT DATE:** (11) Aug[redacted]75

**13. NUMBER OF PAGES:** 20

**14. MONITORING AGENCY NAME & ADDRESS (If different from Controlling Office):**
NAVAL UNDERSEA CENTER
SAN Diego, California

**15. SECURITY CLASS. (of this report):** UNCLASSIFIED

**15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:**

**16. DISTRIBUTION STATEMENT (of this Report):**
UNCLASSIFIED/UNLIMITED

**17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report):**

**18. SUPPLEMENTARY NOTES:**

**19. KEY WORDS (Continue on reverse side if necessary and identify by block number):**

**20. ABSTRACT (Continue on reverse side if necessary and identify by block number):**

Quality assurance tasks relating to computer program development for the System Validation Model of the Target Data Processor/ Communication Processor are described. Conclusions and recommendations are presented concerning quality assurance provisions developed for software procurement specifications, and quality management and test planning activities.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

400 247