1-1-1991

TESTS Software Development Plan: Preliminary

University of Central Florida Institute for Simulation and Training

PREPARED UNDER CONTRACT NUMBER N61339-91-C-0100
for
NAVAL TRAINING SYSTEMS CENTER
and
NAVAL AIR TEST CENTER

TACTICAL ELECTRONICS SIMULATION TEST SYSTEM

Software Development Plan: Preliminary

December 23, 1991

Institute for Simulation and Training
12424 Research Parkway, Suite 300
Orlando, FL 32826

and

Department of Electrical Engineering
University of Central Florida
Orlando, FL 32816

University of Central Florida
Division of Sponsored Research

IST-CR-91-16

TESTS Software Development Plan

Preliminary

December 23, 1991

Prepared Under Contract Number 61339-91-C-0100

for

Naval Training Systems Center

and

Naval Air Test Center

Institute for Simulation and Training
12424 Research Parkway, Suite 300
Orlando, FL 32826

University of Central Florida
Division of Sponsored Research

LIST OF EFFECTIVE PAGES

NOTE:

This list of effective pages maintains a record of submittals against a specific Contract Data Requirements List (CDRL) Sequence Number. Original issues, revisions, and changes of data items are recorded in the following log.

In the event that this page accompanies a complete revision, destroy the superseded document and replace, in its entirety, with the latest revision.

Date of issue for original and change pages are:

Original . 04 Jan 91

Total number of pages in this publication is 71 consisting of the following:

Page No.	Change/Revision No.
Title	0
A	0
i to viii	0
1-1 to 1-3	0
2-1	0
3-1 to 3-9	0
4-1 to 4-8	0
5-1 to 5-2	0
6-1 to 6-3	0
7-1 to 7-6	0
8-1	0
9-1	0
A-1 to A-2	0
B-1 to B-4	0
C-1 to C-16	0
D-1 to D-2	0
E-1 to E-3	0

NOTE:

Zero in the "Change/Revision No." column indicates an original page.

FOREWORD

This Software Development Plan (SDP) is provided as a rough draft for the Tactical Electronics Simulation Test System (TESTS) Project, Phase I: Requirements Analysis and Feasibility Assessment, Contract No. N61339-90-C0125. The TESTS project is being developed by the University of Central Florida Institute for Simulation and Training and the Department of Electrical Engineering, hereafter called "the Contractor" in concert with NAVTRASYSCEN (NTSC) and NAVAIRTESTCEN (NATC). TESTS is to be delivered to and used by NATC.

This SDP has been prepared with the formats specified in Data Item Description (DID) DI-MCCR-80030A and DOD-STD-2167A as guidelines. Also the Military Handbook, MIL-HDBK-287, of 11 Aug 89, A Tailoring Guide for DOD-STD-2167A, Defense System Software Development, is being utilized substantially to streamline the software acquisition process for TESTS. This approach is justified since the TESTS project is early in the feasibility assessment stage of development. The tailoring approach is in accordance with policy directed in DODD 5000.43, Acquisition Streamlining, and MIL-HDBK-248, Guide for Application and Tailoring of Requirements for Defense Materiel Acquisitions. Also as stated in MIL-HDBK-287,

> "The Software Development Plan (SDP) is the primary mechanism for describing the contractor-tailored development process in response to the tailored set of requirements."

Therefore, although this SDP follows the format of DI-MCCR-80030A, as paragraphs are encountered that are to be tailored, the phrase "tailored/deleted" or "tailored/replaced" will be indicated.

The SDP delivery has been divided into three submittals: Draft, Preliminary, and Final. The draft is being submitted as part of the early efforts of the feasibility study, the preliminary at the end of Phase I, and the final version at the end of Phase II. To identify any items that are not included in any submittal, i.e., to be determined, a TBD will be inserted, e.g. <<TBD>>. These TBDs will be addressed in later submittals and/or in project Progress Reviews (PRs).

TABLE OF CONTENTS

TABLE OF CONTENTS - Continued

TABLE OF CONTENTS - Continued

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ACETEF	Air Combat Environment Test and Evaluation Facility
APSE	Ada Programming Support Environment
CASE	Computer Aided Software Engineering
CDA	Critical Design Activity
CDR	Critical Design Review
CDRL	Contract Data Requirements List
CFAB	Composite Functional/Allocated Baseline
CIN	Configuration Identification Number
CM	Configuration Management
COTS	Commercial-Off-the-Shelf
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CSU	Computer Software Unit
DBMS	Data Base Management System
DCA	Development Configuration Activities
DEECS	Department of Electrical Engineering and Communications Sciences
DID	Data Item Description
DoD	Department of Defense
ECP	Engineering Change Proposal
FD	Functional Description
FQT	Formal Qualification Testing
GFE	Government Furnished Equipment
HWPDR	Hardware Preliminary Design Review
IBB	Incremental Build Baseline
IDD	Interface Design Document
IFF	Identification Friend or Foe
IRS	Interface Requirements Specification
IST	Institute for Simulation and Training
ITM	Integration and Test Manager
ITT	Interrogator Stimulators
IV&V	Independent Verification and Validation
LAN	Local Area Network
NA	Not Applicable
NATC	NAVAIRTESTCEN
NDS	Non-developmental Software
NTSC	NAVTRASYSCEN
OS	Operating System
PC	Physical Configuration Audit
PCR	Problem/Change Report
PDA	Preliminary Design Activity
PDR	Preliminary Design Review
PM	Program Manager
POM	Programmer's/Operator's Manual
PPB	Prototype Product Baseline
RCB	Risk Control Board
RFD	Request for Deviation
RTM	Requirements Traceability Matrix
RTSS	Real-Time Software System

SCM	Software Configuration Management
SDC	Software Development Center
SDD	System Design Document
SDF	Software Development Folder/File
SDL	Software Development Library
SDP	Software Development Plan
SDR	System Design Review
SDS	Software Development System
SEI	Software Engineering Institute
SIT	Systems Integration Test
SPS	Software Product Specification
SQA	Software Quality Assurance
SRA	Software Requirements Analysis
SRR	Software Requirements Review
SRS	Software Requirements Specification
SSDD	System Segment Design Document
SS	System Specification
SSR	Software Specification Review
SSSR	System/Software Specification Review
STD	Software Test Description
STP	Software Test Plan
STPPRR	Simulator Test Plan/Procedures Results/Report
STR	Software Test Report
STRR	System Test Readiness Review
SWE	Software Engineer/Software Engineering
SWEG	Simulated Warfare Environment Generator
SWPDR	Software Preliminary Design Review
TESTS	Tactical Electronics Simulation Test System
TTT	Transponder Stimulators
UCF	University of Central Florida
VDD	Version Description Document

1.0 SCOPE

1.1 Identification

This Software Development Plan (SDP) establishes the software engineering effort to be used to develop all the Computer Software Configuration Items (CSCI's) for the Tactical Electronics Simulation Test System (TESTS). This SDP applies to the following CSCI's:

a. The Software Development System (SDS) CSCI
b. The Real-Time Software System (RTSS) CSCI.

1.2 System Overview

The TESTS shall be a computer-based system tool to simulate prototype models of advanced tactical electronic systems. The goal is to determine the feasibility of simulation to test and train Naval personnel on the operation of advanced tactical electronic systems. The simulation environment selected as a test case involves the Identification Friend or Foe (IFF) with emphasis on the Mark XII and Mark XV IFF systems. The TESTS project is a multi-phased effort with objectives as defined in the TESTS Work Plan Report, CDRL A001, 14 Sept 90.

It is proposed that the TESTS tool be designed to interface with various platforms under test (PUT) through simple, potentially off-the-shelf, connections. For the IFF test case, the TESTS tool would consist conceptually of three parts: interrogator stimulators (ITTs), transponder stimulators (TTTs), and a software development center (SDC). The ITTs and TTTs are the simulation prototype models containing the software and hardware to stimulate the respective PUT's interrogator or transponder receivers and transmitters. The SDC is the proposed environment to develop the TESTS software and may be ultimately used to support the delivered CSCIs in the field.

A summary of the preliminary computer hardware environment proposed for the TESTS includes the following Commercial-Off-the-Shelf (COTS) equipment and hardware:

a. 3 SUN SPARC Workstations (networked),
b. one SUN SPARC Server to control the network,
c. 3-4 Skystation Sun compatible accelerator systems
d. an Operator's Console, including <<TBD>>,
e. <<TBD>> PUT Interfaces,
f. <<TBD>> Signal Generators,
g. <<TBD>> Receivers,
h. Interface to the Simulated Warfare Environment Generator (SWEG),
i. Interface to the other parts of the Air Combat Environment Test and Evaluation Facility (ACETEF).

The proposed TESTS software is a combination of COTS software, Simulator Operational software, and potentially SWEG, software that would be Government Furnished Equipment (GFE). There may also be Reusable Software as the prototypes evolve from simple to more complex IFF features and modes, and from the Mark XII to the Mark XV simulations.

The types of COTS software identified at this time include:

a. The Operating System (OS) or Run-time Kernel for each computer system,
b. The Ada Compiler and its Ada Programming Support Environment (APSE),
c. The Local Area Network (LAN) software,
d. The graphics software,
e. The Data Base Management System (DBMS) software,
f. The Configuration Management (CM) software,
g. Other language compilers (C, C++ and/or assemblers),
h. The Mark V Systems Objectmaker, Ada language module, and C language module Computer Aided Software Engineering (CASE) tools.

The Simulator Operational software identified at this time is the RTSS CSCI including:

a. The software for the ITTs and the TTTs,
b. The software for the Operator's Console,
c. The software to interface to COTS,
d. The software to interface with SWEG and/or ACETEF,
e. The diagnostic software to ensure daily readiness of the system.

The need and usage of SWEG software, GFE or otherwise, is <<TBD>> at this time. It is expected that early in the development of TESTS will utilized only the data base structures of SWEG. Later in the development, TESTS will use the actual SWEG software augmented by TESTS specific data input routines.

Although the SDS CSCI consists primarily of the COTS software listed above, this CSCI also includes the following additional support software:

a. Job control or command strings to build the operational software loads for the various processors,
b. Test software drivers or stubs,
c. Off-line software necessary to integrate COTS software development products (CM, CASE, APSE).

1.3 Document Overview

This SDP outlines the Contractor's plans to develop TESTS software utilizing DOD-STD-2167A as a guideline. The SDP also describes the organization and methods that will be utilized by the Contractor to perform software development. This SDP applies to all developed software as identified above in the Simulator

Operational category.

1.4 Relationship to Other Plans

This SDP is linked to the TESTS Work Plan Report, CDRL A001 for Phase 2A: System Requirements Specification.

2.0 REFERENCED DOCUMENTS

The following documents are referenced within this SDP, the list does not necessarily represent all TESTS referenced documents.

ANSI/MIL-STD-1815A	Ada Language Standard
DOD-STD-2167A	Defense System Software Development
DOD-STD-2168	Defense System Software Quality Program
MIL-STD-483A	Configuration Management Practices for Systems, Equipment, Munitions and Computer Software
MIL-STD-490A	Specification Practices
MIL-STD-1521B	Technical Review and Audits for Systems, Equipment, and Computer Software
DOD-STD-480A	Configuration Control Engineering Changes, Deviations and Waivers
MIL-STD-1472C	Human Engineering Design Criteria for Military Systems, Equipment and Facilities

Contractor Generated Documents

CDRL A001	Work Plan Report, 14 July 91

3.0 SOFTWARE DEVELOPMENT MANAGEMENT

3.1 Project Organization & Resources

3.1.1 Contractor Facilities

The TESTS project development will be conducted at the University of Central Florida's (UCF) Institute for Simulation and Training (IST) and the Department of Electrical Engineering and Communication Sciences facilities in Orlando, Florida, including the TESTS program management, subcontract management (if needed), procurement, and the system, software, and hardware engineering activities. The initial facilities consist of two Sun Sparc 2 workstations plus one Sky Computers Skystation accelerator system. The software development environment at IST is located within a secure facility which is designed to meet DoD standards for classified data processing. The Contractor facility and equipment resources will support the requirements of the TESTS systems, software, and hardware engineering development activities.

3.1.2 Government Furnished Equipment, Software, and Services

There are <<TBD>> GFE, software or services identified at this time to accomplish the software engineering effort:

a. SWEG and its support environment,
b. Mark XII IFF equipment,
c. Updated IFF equipment,
d. <<TBD>>.

3.1.3 Organizational Structure

The overall TESTS project organizational chart is shown in Figure 3.1.3-1.

3.1.4 Personnel

Table 3.1.4-I provides the peak number or personnel required for the TESTS project team.

3.2 Schedules and Milestones

3.2.1 Activities "tailored/replaced"

The Program Master Milestone Schedule, which highlights the TESTS schedule is provided in Appendix A. The TESTS system/software development activities, including the following discussion, is summarized in Figure 3.2.1-1:

a. System/Software Requirements Analysis Activity (SSRAA)

This activity includes requirements analysis and design activities, as documented in the Functional Description (FD),

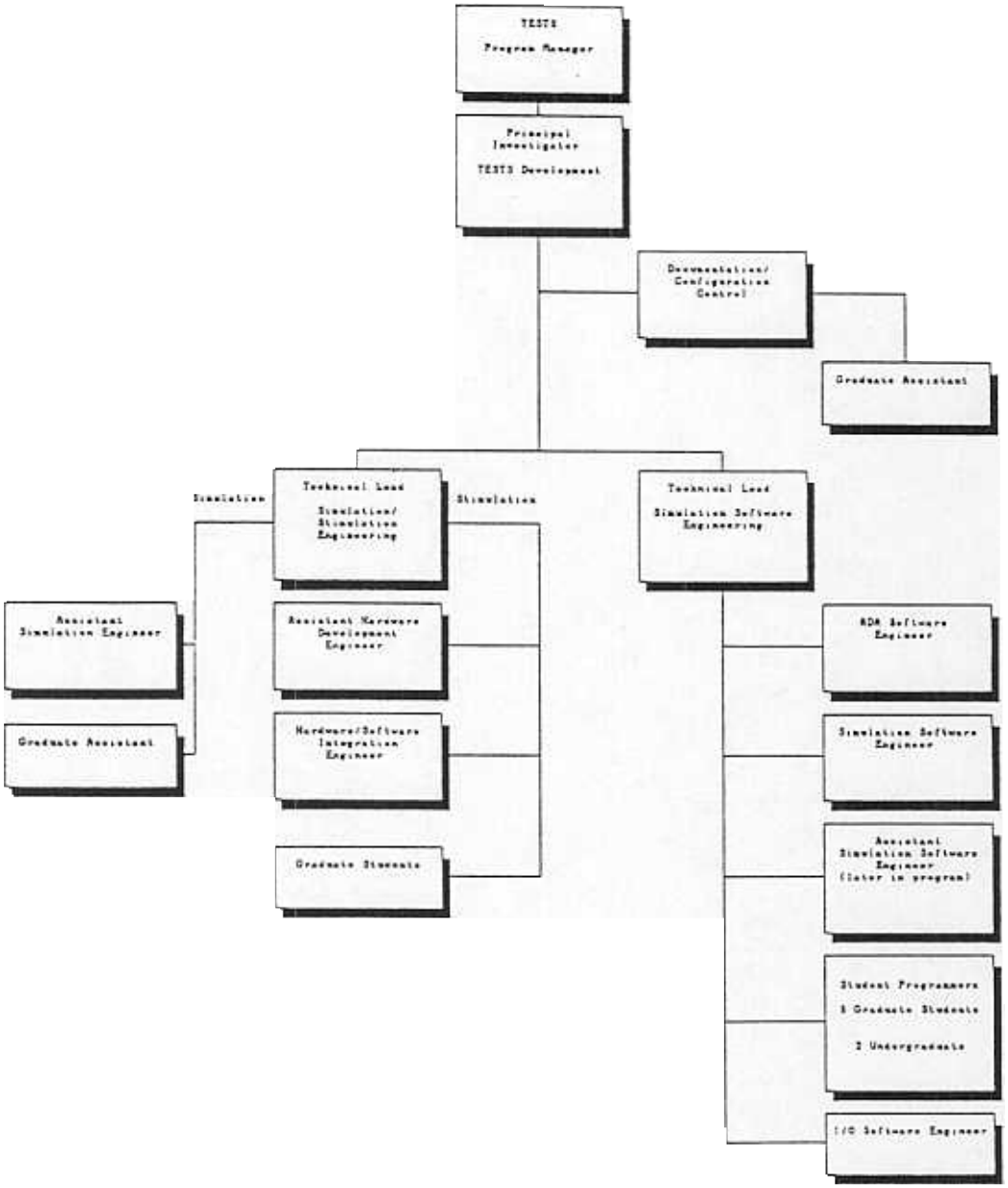

Figure 3.1.3-1. Preliminary TESTS Project Organization Chart

TABLE 3.1.4-I. Personnel Requirements

Category	Peak Number
Project Management	1
Principal Investigator	2
Principal Engineer	5
Senior Engineer	2
Simulation Engineer	2
Graduate Student	5
Undergraduate Student	2
Administrator	1

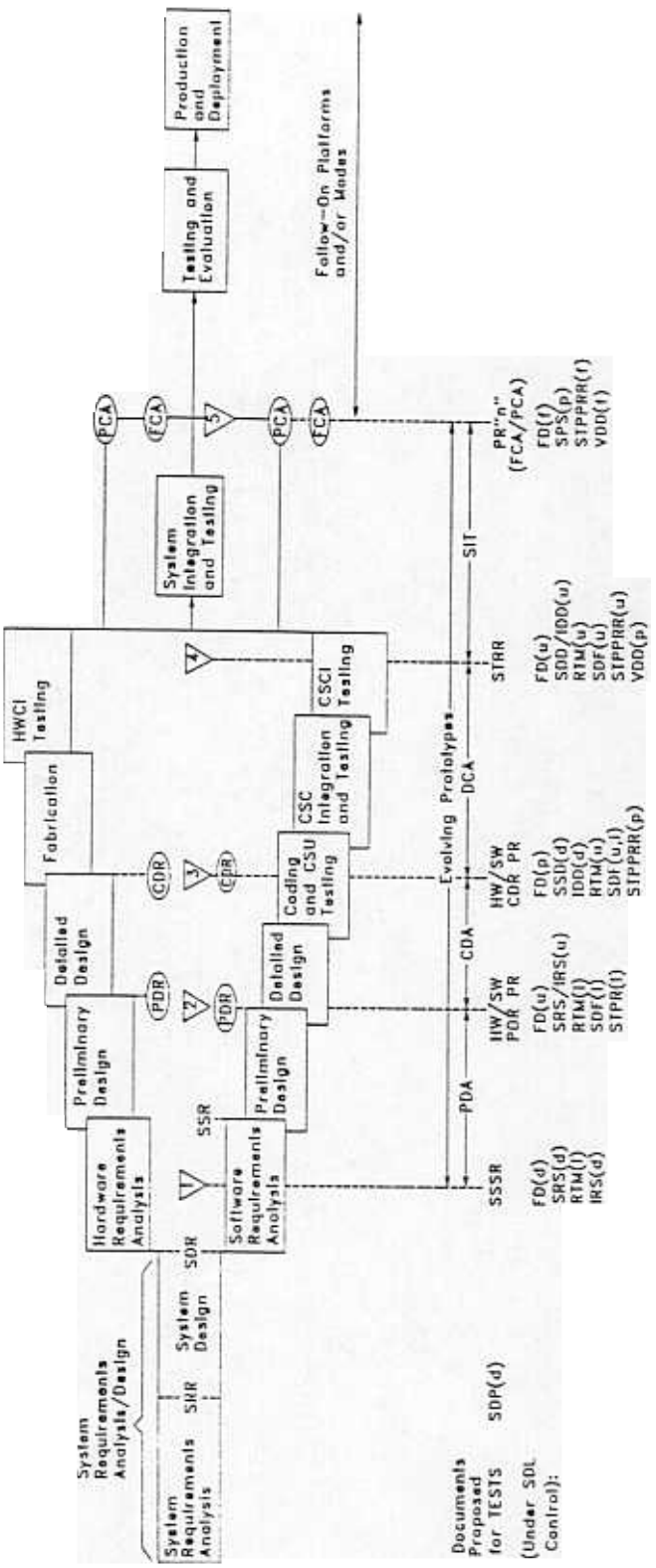

Figure 3.2.1-1. TESTS System/Software Development Activities

DI-E-30104B, which replaces the Systems/Segment Specification and the Systems/Segment Design Document (SSDD). Based on the allocation of systems requirements to the configuration items (HW and SW), software analyzes and derives engineering and interface requirements for the CSCIs. These requirements are documented at first in Software Development Folders/Files (SDFs) and then later in the Software Requirements Specification (SRS) and in the Interface Requirements Specification (IRS). Rough draft versions of the SRSs and IRS, along with supporting engineering analysis, the updated SDP, an internal version of the Requirements Traceability Matrix (RTM), and the a draft version of the FD are presented at the System/Software
Specification Review (SSSR).

The SSSR combines essential elements of the MIL-STD-1521B System Requirements Review (SRR), the System Design Review (SDR), and the Software Specification Review (SSR). The system and software requirements will continue to be refined in later phases as different prototypes are evolved.

b. Preliminary Design Activity (PDA)

This activity begins with an establishment of the preliminary components of the software design. The engineering and interface requirements are allocated to these components. The preliminary software architecture for each of the processors is also established. The preliminary software architecture includes identification of interfaces to COTS software (Operating Systems, DBMSs, LANs, etc.), GFE (SWEG) and hardware (I/O systems, graphics boards, etc.) and identification of operational capabilities that are to be provided by each software element. This preliminary software design is updated in SRSs, the IRS, and the SDFs. The system/hardware design is updated in the FD.

During this activity, the Computer Software Components (CSCs) are defined and SDFs are created for each CSC. As features of the prototypes are implemented, further delineation of the CSCs into Computer Software Units (CSUs) will occur and SDFs will also be generated for CSUs. The process of defining CSCs and CSUs will continue throughout this activity and later activities as more prototyping is accomplished and as more information about the Mark XV becomes available.

The internal RTM is updated and is utilized to help generate a rough draft of a Simulator Test Plan/Procedures Results/Report (STPPRR) document. The STPPRR is similar to the MIL-STD-1644B Trainer Device Computer Program Test Procedures/Results document, DI-T-25852 (this DID will serve as a guideline), but tailored to the needs of TESTS during this activity. The STPPRR replaces the DOD-STD-2167A Software Test Plan (STP), the Software Test Description (STD), and the Software Test Report (STR). The STPPRR includes system level test plans, descriptions, procedures, and reports instead of just the software

counterparts. The activity culminates in PR "n" and the updated documents are included in the review. The status of the evolving prototypes (system, hardware, and software) is discussed and any demonstrations that may be ready are conducted. This PR replaces the MIL-STD-1815B HW and SW Preliminary Design Reviews (HWPDR and SWPDR).

c. Critical Design Activity (CDA)

The HWE and SWE efforts of the evolving prototypes begun in PDA are continued during this activity. As more and more features of the Mark XII are prototyped and as the Mark XV becomes available and its features and modes are prototyped, then these TESTS models become "builds". This implementation approach is also called an "incremental build" approach. As each prototype or feature gets implemented, it is tested in accordance with an incremental build plan for the project. A prototype may make up a build or prototypes may be grouped to form a build. The Contractor continues in the process commonly referred to "build a little, test a little" until the different features, modes, PUTs of each IFF for the project have been implemented and each build has been tested. Users manuals are generated for each of the prototype models or builds that will ultimately be a TESTS tool piece used at ACETEF. Presentations and/or demonstrations at various PRs of the builds are conducted and status is discussed.

Throughout this incremental build process, the development team is updating the SDFs, the FD, the RTM and the STPPRR. The SRSs and the IRS help the team develop detailed interfaces and software design and to continue to define CSCs and CSUs. These efforts are documented in the SDFs and draft versions of the Interface Design Document (IDD) and the Software Design Document (SDD).

At a mutually agreed upon PR (shown on the TESTS schedule), a collection of builds, their progression as documented in the SDFs, the RTM, the FD, the SDDs, the IDD, the STPPRR, the users manuals, related demonstrations and results are presented. This more extensive PR replaces the MIL-STD-1521B CDRs for HW and SW.

d. Code and CSU Test, CSC Integration and Testing and CSCI Testing Activities ... "tailored/replaced" by Development Configuration Activities (DCA)

Throughout the incremental build/evolving prototype process CSCs and CSUs are defined, coded, tested, and integrated with their respective builds. Results are compiled in the SDFs. This collection of activities involves system HW and SW integration (HSI), the placing of the CSCIs under a formal Software Configuration Management (SCM) system, and internal, confidence-level testing at a system level. The confidence-level tests are the tests that were specified in the STPPRR. At a <<TBD>> PR, a system level test readiness review (STRR) will occur. Some typical examples of system level tests might be:

1. Testing of the whole Mark XII, or
2. Testing the Mark XV for a particular platform, or

3. Testing all ITTs up to a certain mode, etc.

This level of testing groups large parts of the RTSS CSCI and the respective prototype models for demonstration. As previously done, users manuals for test groups or builds are written, as applicable.

Upon successful completion of CSCI and system level testing, the CSCI software version is documented in a Version Description Document (VDD). The SDDs and IDD are updated to reflect this testing also.

e. System Integration and Testing (SIT) Activity

The HSI and the confidence-level STPPRR testing efforts generally point out changes and tuning of the system that need to occur. Changes are incorporated and tracked by the SCM system and an updated VDD is generated. The updated SDDs and IDD with the magnetic media are compiled to generate the Software Product Specification (SPS). The users manuals for the prototypes and the SDC are compiled into the programmer's and operator's manual. This document replaces the DOD-STD 2167A Computer System Operator's Manual (CSOM), the Software User's Manual (SUM), the Software Programmer's Manual (SPM), and the Computer Resources Integrated Support Document (CRISD), but utilizes essential elements from those documents' DIDs as guidelines.

This activity is culminated by a PR to conduct the functional and physical configuration audits (FCA and PCA) with format specified by the Contractor.

3.2.2 Activity Network

The activity network depicting TESTS activities is provided in Appendix A, Schedules.

3.2.3 Source Identification

The software and hardware resources necessary for the software development effort are identified and described, including the need dates in Table A-4, Software Development Resources in Appendix A.

3.3 Risk Management ... "tailored/replaced"

Risk management for software will be accomplished as a function of the overall project risk management discussions at PRs as needed. A special functional group within the program organization, the Risk Control Board (RCB), is not needed and is deleted from this paragraph.

3.4 Security

The security requirements at this time require the ability to handle data and develop software at the secret level. The

Contractor has facilities to handle classified data at the secret level. A software development evironment has been established at IST to comply with DoD requirements for classified software development. Any activities which can not be accommodated within IST facilities will be conducted at the Naval Training Systems Center.

3.5 Interface with Associate Contractors

No associate contractors have been identified at this time.

3.6 Interface with Software IV&V Agents

IV&V agents are not required at this time.

3.7 Subcontractor Management

There are no subcontracts anticipated at this time.

3.8 Formal Reviews

The reviews for the TESTS project are described in the text of section 3.2.1 above.

3.9 Software Development Library (SDL)

The SDL serves as a controlled repository for software products generated throughout the software development effort. The SDL will be maintained using commercially available or government furnished configuration management tools provided as part of the APSE. The products of and the procedures used to maintain the SDL are <<TBD>>.

3.10 Corrective Action Process

The corrective action process consists of a project implemented system based upon a Problem/Change Report (PCR) described in section 3.11 below. The PCR is utilized to effect change to CM controlled software (documentation and/or source code after the STRR is held), whether initiated by an ECP, test discrepancy, or a corrective action. Utilizing the PCR assures that the change is reviewed, coordinated and tracked.

3.11 Problem/Change Report

The project reporting system utilizes the PCR form, Figure D-9 provided in Appendix D, Forms. The PCR will be maintained using configuration management tools. The PCR is used in the following manner:

a. To document software deficiencies in as much detail as possible using the PCR and submit to SCM.

b. SCM assigns a control number and submits the PCR to the Lead Software Engineer for evaluation.

c. The PCR and the proposed corrective action are submitted to the Principle Investigator for review and approval. If disapproved, the PCR is returned for re-evaluation. If approved, the change is implemented and the PCR is closed. If approved and the change impacts an established prototype, then approval for proposed corrective action requires review with the customer approval at the next PR.

   Customer submitted deficiencies are also documented and tracked via the PCR form. The PCR remains open until the entire deficiency is addressed.

4.0 SOFTWARE ENGINEERING (SWE)

4.1 Organization and Resources - SWE

4.1.1 Organizational Structure - SWE

The overall software engineering organization and associated support organizations are discussed in 3.1.3 and the following. The TESTS software engineering organization is responsible for all developed software and the integration with non-developmental software (NDS) into TESTS.

The TESTS software engineering organization consists of a lead software engineer, 2-3 software engineers, including 5 graduate students and 2 undergraduate students, an engineering aide and a part-time software librarian. The position descriptions are given below.

> Lead Software Engineer - <<TBD>> title/job description - The Lead Software Engineer is a member of the project's technical staff and leads the overall SWE technical and administrative (cost and schedule) activities of software requirements analysis, design, coding, and test.
>
> Software Engineers - <<TBD>> title(s)/descriptions - A Software Engineer is responsible for designing, coding, documenting/updating SDFs, and testing of assigned component(s). Also, the software engineer is responsible for providing source data inputs to the data items. Graduate students may also perform many of these functions dependent on skill level.
>
> Configuration Manager - <<TBD>> title/description - The Configuration Manager generates SCM schedule and status reports and performs procedures to maintain the software library and version control of the TESTS software. The Configuration Manager may be responsible for helping to coordinate source data generation for deliverable data items, reviewing SWE source data inputs for the deliverable data items and for the production and evaluation of deliverable data item. The Configuration Manager may also help compile and unit test CSUs. A graduate student could perform these functions.

4.1.2 Personnel "tailored/replaced" by 3.1.3 and 4.1.1

4.1.3 SWE Environment

The SWE Environment identifies the hardware and software elements proposed for the TESTS development. A description of each element is provided below.

4.1.3.1 Software Items

Table 4.1.3.1 I, The TESTS APSE, CASE, and Software Tools,

Preliminary TESTS APSE, CASE and Software Tools

Vendor	Product No.	Quantity	Description
Mark V Systems Ltd.	OBJ-Sparc	1	Object Maker CASE tool
	CLM-Sparc	1	C and C++ Language Module Compilers
	MC-Sparc	1	Maintenance Extension
Meridian Software	Ada-Z	4	Ada Language Compiler & Professional Development System
Sun Microsystems	(Ada)	1	Ada Language Compiler
	(C)	1	C Language Compiler
Sky Computers	Sky Vec C	1	Sky Vectorizing and Optimizing C Compiler

TABLE 4.1.3.1-I. TESTS APSE, CASE, and Software Tools

describes the software items to be used to perform the software engineering activities. The table provides the vendor, product number, and a description of each item. This table will be updated as appropriate.

4.1.3.2 Hardware & Firmware Items

Table 4.1.3.2 - I, The TESTS APSE Hardware, describes the initial hardware and firmware items to be used in the software engineering environment. The table includes vendor, product number, and a description. This table will be updated at the end of Phase I and subsequent PRs. The initial configuration will consist of a single SUN workstation and Sky Computers Skystation accelerator. The complete development system will be implemented as the project progresses.

4.1.3.3 Proprietary Nature & Government Rights

Table 4.1.3.3-I, Software Item Government Rights, and Table 4.1.3.3-II, Hardware and Firmware Item Government Rights, describes the Government rights and restrictions associated with each item to the software engineering environment. This table will be updated at the end of Phase I and at the PR for the PCA/FCA.

4.1.3.4 Installation, Control, Maintenance

The APSE support group responsible for the installation, test and maintenance of the software engineering environment will be comprised of project engineering personnel. Each element of the support environment will be identified and logged into the project's configuration management data base.

The APSE support group will assemble and document the hardware, install/document the software, and run bench mark and checkout procedures to certify the fitness of the various products for use in the APSE. The APSE will then be made available for use by the project team.

Each item will be purchased with vendor support and update options. When updates are received, the changes will be logged in and evaluated before they are provided to the development team.

4.2 Software Standards and Procedures

The software standards and procedures are contained in a separate appendix of this document in order to facilitate its distribution to project personnel. Generally speaking, the topics for Software Standards and Procedures to be developed are as follows:

Software Development Techniques and Methodologies
Software Development Files
Design Standards
Coding Standards

Preliminary TESTS APSE Hardware

Vendor	Model No.	Quantity	Description
Sun Microsystems	4/75 GX-16-P40	2	SPARC station 2 Computer Workstation
	X 300 H	2	USA Keyboard
	SPRN-400	2	SPARC Printer
	S4SL2	2	SUN System Software
	SS2-07	2	SUN OS 4.1
	SX-09	2	SUN OS 4.1 Documentation
Sky	Series B110-P	1	Skystation Application Accelerator

TABLE 4.1.3.2-I. TESTS APSE Hardware

[GOVERNMENT SOFTWARE ITEM RIGHTS WILL
BE EXTRACTED FROM CONTRACT PROVISIONS]

TABLE 4.1.3.3-I. Software Item Government Rights

[GOVERNMENT HARDWARE AND FIRMWARE RIGHTS WILL
BE EXTRACTED FROM CONTRACT PROVISIONS]

TABLE 4.1.3.3-II. Hardware and Firmware Item Government Rights

4.3 Non-Developmental Software

Non-developmental software (NDS) consists of commercially available, Government furnished, and/or reusable software. In the interest of cost effectiveness, reliability and maintainability, the use of non-developmental software will be maximized. Use of NDS products will be as follows.

NDS shall be reviewed with respect to project software requirements. Selected NDS shall be documented within the appropriate CSCI's Software Design Document (SDD) as either a CSC or CSU and in their respective SDF. Where more than one CSCI utilizes the NDS, the appropriate CSCI is determined as that CSCI which utilizes the NDS most completely. Other CSCIs utilizing the same NDS in the same manner shall reference the documenting CSCI's SDD. If another CSCI utilizes the same NDS in a different manner then it is only necessary to document the unique usage and make the appropriate references.

The identification of the NDS as a CSC or CSU is determined by the level of requirement that the NDS satisfies. NDS that satisfies a high-level requirement should be considered as a CSC or sub-level CSC. NDS that satisfies a low-level requirement should be considered as a CSU.

Table 4.3-I, Non-Developmental Software, lists the non-developmental software which will be used on the TESTS. The table includes a title, NDS type, description and rationale for use for each item. This table will be updated at PRs that discuss prototype or the incremental build process and at the PR prior to the STRR. As with the developed SW, the users manuals will include NDS documentation.

[TO BE DETERMINED]

TABLE 4.3-I. Non-Developmental Software

5.0 FORMAL QUALIFICATION TESTING (FQT)

5.1 Organization & Resources - FQT

5.1.1 Organizational Structure - FQT

The formal qualification and test (FQT) organization are members of the TESTS project organization and a representative from NTSC and NATC. The PM appoints a project team member to be the Integration and Test Manager (ITM). The ITM is also responsible for the oversight of the STPPRR preparation and to conduct the system level confidence tests and the SIT activity.

The TESTS FQT organization consists of the Integration and Test Manager and 2-3 number of project engineers.

5.1.2 Personnel - FQT ... "tailored/replaced"

Refer to section 4.1.1.

5.2 Test Approach/Philosophy - FQT - "tailored/replaced"

Philosophy: Formal testing will be accomplished by project personnel who have completed their respective software development responsibilities. In order to satisfy the independence requirement of 2167A the designated personnel will be assigned and report to the ITM for this effort. In addition, every attempt will be made to select engineers for testing software that was not developed by them. FQT will be accomplished during the SIT software development activity.

The development of the test cases and procedures will be a joint effort of the ITM and software/hardware engineering team members. The standards and criteria for the development of the cases and procedures will be established by the ITM. Final approval of all test products will rest with the ITM. However, the actual work will be performed by the development team.

Method: Preparation for testing will be initiated during the SSRAA. The PM, Principle Investigator, and Lead Software Engineer will review SRS and IRS requirements for completeness and testability. The PM will appoint the ITM to oversee the generation of the STPPRR. The software engineers will provide assistance to document the plans for FQT of each CSCI. The plan will be reviewed and approved at the PR that completes the CDA, refer to section 3.2.1 above.

The FQT will start with the STRR and CSCI/system level tests and will continue through HSI and SIT culminating with PCA/FCA. The ITM will be responsible for scheduling, testing, and recording all test activities for audit by the customer.

5.3 Test Planning Assumptions & Constraints - FQT

There are no assumptions or constraints identified at this time.

Further test planning is <<TBD>> at the end of Phase I and Phase II.

6.0 SOFTWARE PRODUCT EVALUATIONS (SPE) "tailored"

6.1 Organization and Resources - SPE

6.1.1 Organizational Structure - SPE

SPE for TESTS is designed to address quality issues through all software development activities. Controls are enforced by assigning specific responsibilities and authority to different members of the project team. Program Management is responsible for overall project management and development of system/ software products and associated documentation. The PM assigns Configuration Management responsibility and monitors the effort. CM is responsible for maintaining a system of configuration identification, configuration control, configuration accounting, and configuration reviews for all deliverable elements of a system. The PM also assigns Software Quality Assurance responsibility and monitors that effort. SQA assures the performance of the software quality evaluation procedures throughout all software development activities.

The project team members that hold these responsibilities form the project SPE team. The specific responsibilities of each representative are <<TBD>>.

6.1.2 Personnel - SPE ... "tailored"

The number and skill level of all personnel requirements have been previously defined in sections 3.1.3 and 4.1.1.

6.2 SPE Procedures and Tools ..."tailored"

6.2.1 Procedures

The procedures for SPE fall into three categories: software engineering product evaluation procedures, software quality assurance product evaluation procedures, and deliverable product evaluation procedures. The procedures for each category are provided in the following paragraphs.

6.2.1.1 Software Engineering Product Evaluations

Specific procedures for engineering evaluations are provided in Appendix C, Software Standards and Procedures.

6.2.1.2 SQA Product Evaluations

SQA procedures for product evaluations are part of the <<TBD>> duties of the PM, the PI, the whole SWE team and specifically the PM's designee for SQA.

6.2.1.3 Deliverable Product Evaluations

Deliverable product evaluations will be accomplished in the following manner:

a. A memo will be generated by the lead software engineer to identify the personnel responsible for the evaluation and specific review tasks.

b. The memo, a copy of the product, a product evaluation checklist and a sign-off sheet will be routed to the assigned personnel for review and sign off.

c. The product evaluation checklist, comments and deficiencies will be returned to the author for review and a written response generated for each critique of the product. The review is considered complete when all comments and deficiency dispositions have been agreed upon by the author and the reviewers, the software product has been updated and tested, and placed back under CM.

d. Upon successful completion of the review process, the document is signed off, shipped, and the updated original of the document is stored in secured cabinets.

6.2.2 Tools

There are no evaluation tools identified for use by software engineering at this time.

6.3 Subcontractor Products

There are no current plans for UCF/IST to subcontract software development work on this program.

6.4 SPE Records ... "tailored"

6.4.1 SQAE Evaluation Records

Heavy usage will be made of checklists that will become part of the SDFs. The checklists are <<TBD>> but contain results of evaluations, and audits of engineering work. The SDFs are available for review and are discussed as needed at the PRs.

6.4.2 SWE Evaluation Records

SWE records will be maintained for all formal and informal reviews and audits performed by the software engineering staff. The evaluation records will be entered into the appropriate SDF as each review and test is accomplished. Informal CSC level review and test records will be maintained in the CSC level SDF while formal review and test records will be maintained in the CSCI level software development folder. The specific format and

form of the evaluation records and the procedures for using and maintaining the records is <<TBD>>.

6.5 Activity Dependent Product Evaluations ... "tailored"

Deliverable software and documentation will be evaluated by the assigned evaluation personnel to assure adherence to the specific checklists. Checklists will be generated and distributed to the development team. All checklists and records will be placed in the respective SDFs and are available for audit.

6.5.1 SPE - SSRAA ... "tailored"

The Software Development Plan (SDP), SRS, IRS, and FD will be inspected in accordance with the <<TBD>> checklist in Appendix E and results placed in the SDFs.

6.5.2 SPE - PDA ... "tailored"

The main effort during this activity is toward allocating SRS and IRS requirements to CSCs for each CSCI and for prototype cases that evolve during this activity, CSUs, developing test plans and procedures for the STPPRR, and performing preliminary design for the external interfaces and the CSCs of each CSCI.

6.5.3 SPE - CDA ..."tailored/replaced"

The primary effort during this activity is toward establishing the detailed level CSCI, CSC, and CSU design. The SDDs, IDD, and SDFs are evaluated to assure the software designs and test cases are being developed and placed in the SDFs as prototyping and the incremental build process proceeds. Checklists are completed and placed in the SDFs.

6.5.4 SPE - DCA "tailored/replaced"

The primary efforts during this activity are toward completing the "build a little, test a little" incremental build effort, performing HSI, placing SW under SCM, and conducting the system level confidence testing. The following product evaluations will be done in accordance with the checklists provided in Appendix E.

a. The source code for each CSU as documented in the SDFs

b. The test results for each CSU, CSC, & CSCI as documented in the SDFs

c. The test procedures as documented in the SDFs

d. The VDD is reviewed to verify that the SW is under CM

6.5.5 SPE - SIT "tailored/replaced"

All software and associated documentation will be updated and to reflect modifications made during SIT. The updated design documents and source code for each CSU will be inspected and evaluated according to the relevant criteria listed in sections 6.5.2, 6.5.3, and 6.5.4 to support any necessary retesting.

7.0 SOFTWARE CONFIGURATION MANAGEMENT (SCM) "tailored/ replaced"

The specific tools, forms, procedures, and personnel responsibilities that will be used to accomplish SCM will not be completely defined until the APSE has been defined and/or at least not before the end of TESTS Phase I. However, there are certain broad statements that can made at this time:

a. Formal configuration control of the code (that is the formal filling out of forms and approvals) will not occur until just prior to the start of HSI. All CSUs, CSCs, and incremental builds have been successfully tested and evaluated prior to placing the SW under formal SCM.

b. The <<TBD>> CM tool will be utilized to form audit trails and to enter and log the Problem/Change Report (PCR). All changes to documents and magnetic media will be initiated via the PCR and the CM tool.

c. Changes for SW under formal CM will have to be evaluated by a review board including <<TBD>> members of the development and customer's team.

d. The developmental configuration baselines described in DOD-STD-2167A have been tailored. Since the overall objective is to evaluate the feasibility of TESTS, the tested evolving prototypes may yield a product that could serve as a model system specification for future simulators. That is, the successful completion of SIT would normally yield a "Product Baseline" as defined in 2167A. TESTS, however, would yield working prototypes that could also serve as a blueprint for a "Functional Baseline" for future projects.

   For TESTS, the baseline that will exist at the SSSR will be a composite of the 2167A Functional and Allocated Baselines. Since the Mark XV is still under development at this time, the Contractor has chosen an Incremental Build approach utilizing evolving prototypes. Since all requirements are not known at the time of Phase I (and perhaps during Phase II also), requirements are analyzed, allocated to HW and SW, made part of a "build", and implemented in a prototype. As new equipment is made available and/or as new requirements are defined, the added features are mapped into prototypes and the Contractor will update the RTM. An objective is to have the Mark XII and the Mark XV requirements for simulation defined, mapped into builds, and implemented into model prototypes by the end of CDA or by <<TBD>>, whichever comes first.

e. CM would monitor the tracking of requirements in the RTM to the contents of the SDF during the PDA, CDA, and

DCA, informally but utilizing the <<TBD>> CM tool to the fullest extent possible. Once under formal CM, the VDD, the CM tool, and all SW documents and files will be monitored and be controlled.

7.1 Organization and Resources - SCM ... "tailored/replaced"

7.1.1 Organizational Structure - SCM ... "tailored/replaced"

The responsibility for TESTS CM will be assigned by the PM from the development team. The title and job description are <<TBD>>.

7.1.2 Personnel - SCM ... "tailored/replaced" by 7.1.1

7.2 Configuration Identification

7.2.1 Developmental Configuration Identification

The Contractor shall establish Developmental Configuration Identification or tailored 2167A baselines for major milestones or Phases of TESTS. A "Composite Functional/Allocated Baseline" (CFAB) based on <<TBD>> prototypes will be established at the end of the PDA. An "Incremental Build Baseline" (IBB) will be established at the end of the DCA, and the "Prototype Product Baseline" (PPB) will be established at the end of SIT. The methods to establish and the contents of the baselines are <<TBD>>.

7.2.2 Identification Methods

Software configuration control begins with the addition of CSCI(s), CSCs, CSUs, and documentation (or version) to the project's SDL and the establishment of the SDP and SDFs. These SWE products shall be named/identified in a <<TBD>> manner with respect to the CSCI they are a member. All versions of developed SWE products under this SDP are assigned version numbers of the form Version X.YZ, where:

X - Indicates a major program version
Y - Indicates a program update
Z - Indicates a sub-version created for the purpose of testing or to support a peculiar requirement

A sample of a Configuration Identification Number (CIN) document identification and version follows:

ccccxx-nnn-a, where ...

ccccc	is	the root SWE id no.,
xx	is	the CSCI identifier,
nnn	is	the document code,
a	is	the alpha revision level.

examples: I. 6133901-102-B may mean

61339	root SWE id for TESTS,
01	the Real-Time Software System CSCI
102	the SRS (for the RTSS CSCI)
B	revision B

II. 6133902-108-A may mean ...

61339	root SWE id for TESTS,
02	the Software Development System CSCI
108	the SPS (for the SDS CSCI)
A	revision A

A sample list of documents recommended for TESTS follows:

DOC/CODE	DESCRIPTION
100 FD	Functional Description
101 SDP	Software Development Plan
102 SRS	Software Requirements Specification
103 IRS	Interface Requirements Specification
104 SDD	Software Design Document
105 IDD	Interface Design Document
106 STPPRR	System Test Plan/Procedures Results Report
107 VDD	Version Description Document
108 SPS	Software Product Specification
109 POM	Programmer's/Operator's Manual.

7.3 Software Configuration Control "tailored/replaced"

As a minimum, changes to software baselines shall be documented in the source code in accordance with the design and coding standards described herein. Software changes will be initiated on PCRs and reviewed (reference 3.11). When the DCA and SIT testing are complete, a VDD and release is made, two copies of the new baseline software will be forwarded to the project/PM for review and release to the customer at a PR after PCA/FCA.

7.3.1 Flow of Configuration Control

Figure 7.3.1-1 illustrates the change process for source code software documents under CM.

[TO BE DEVELOPED]

Figure 7.3.1-1. Configuration Control Flow

7.3.2 Reporting Documentation

The change reporting procedures describe the forms used and instructions for their use are <<TBD>>. The forms are included in Appendix D. Engineering Change Proposals (ECP) for modifications to accepted baselines, adjustments to contract funding and/or schedule shall be formally submitted. The ECP utilizes DOD-STD-480A as a guideline for its format and preparation.

7.3.3 Review Procedures "tailored/replaced"

As stated in the Activity section above, re 3.2.1, the TESTS project shall utilize PRs to brief the customer on status and to discuss changes and requirements that evolve out of prototyping. Internally, the project team would meet at <<TBD>> intervals (at least monthly) to discuss progress, refinements, and changes. These technical meetings shall serve as software and system review boards for the project. Minutes are kept and distributed, the RTM is updated and distributed, and SDFs are updated accordingly.

7.3.3.1 Review Board Procedures "tailored/deleted"

7.3.4 Storage, Handling, and Delivery of Project Media

Although the details are <<TBD>>, the storage and handling of the SW media is facilitated by the SDL and controlled by SCM. Storage of project media will be in DoD/GSA approved cabinets in accordance with industrial security requirements.

After the SDC is established, at least weekly backups to tape magnetic media will occur. Backups utilize three sets of media that are rotated so that data loss in the case of a system failure is kept to a minimum. During the testing activities, the backup procedure may occur more frequently than weekly.

Once selected, the SCM tool will help define procedures and capabilities. Delivery of released software would also be on the selected magnetic media per contract requirements.

7.3.5 Additional Controls

None identified at this time.

7.4 Configuration Status Accounting ... "tailored to Contractor format"

This is <<TBD>> at this time, but the reports will be used to assist in configuration audits.

7.5 Configuration Audits ... "tailored/replaced"

The PCA and FCA will use MIL-STD-1521B as a guideline but will conduct the audits to Contractor defined format and agendas.

7.6 Preparation for Specification Authentication

The successful conducting of the PCA after the successful completion of the SIT is the final authentication of the system. The following summarizes the efforts throughout the development to prepare for the authentication. SCM will perform walkthroughs with the development SWEs to verify that the SDFs, the RTM, the SRSs, the IRS, the SDDs, the IDD, and the Incremental Build Plan track requirements and are in agreement. Walkthroughs will be held at least monthly during the CDA, biweekly during the DCA, and weekly during SIT. The PCR will be utilized to track and initiate changes. All updates will be reflected in the SPS; a VDD will be generated for the tested builds; all change documents will be compiled to prepare for the PCA.

7.7 SCM Major Milestones

Although the list is to be refined, the following are major milestones for TESTS and for SCM:

Phase I completion,
Completion of SSSR,
Phase II completion,
PRs to review Preliminary and Detailed Design,
PRs to define the Incremental Builds,
Phase "N" initiation and completion,
Acquisition of Mark XV data and/or hardware,
Completion of the STRR / placing of SW under formal CM,
VDD completion and subsequent SW release,
PCA/FCA completion.

8.0 Other SW Development Functions

None at this time.

APPENDIX A

TESTS MAJOR MILESTONE SCHEDULE

<<TBD>>

APPENDIX B

SDL DEVELOPMENT LIBRARY

11.0 SDL DEVELOPMENT LIBRARY

11.1 Introduction

The SDL serves as a controlled repository for the source programs, documentation and vendor support software products. The function of the SDL is to receive, document, identify, maintain, and control the configuration of the products. The SDL configuration is maintained by a member of the project team who performs the librarian duties and who ensures that the SDL products are only updated via change procedures. The SDL is the source of all code and data used for the test loads and baseline definitions.

11.2 Product Identification

Figure B-1 identifies the SDL products that will be generated during each design activity.

11.3 SDL Description

The SDL for the TESTS is the library scheme under which software associated specifications and data are developed, debugged, integrated, and tested. It may be manual or a <<TBD>> automated method and contains files and procedures for handling files. The SDL is under project control and monitored by the team member designated to do SCM duties.

The SDL ensures integrity of the software throughout the development of the project. The SDL will aid in:

a. Uniquely identifying all software items
b. The release process during and after the testing cycle
c. The build process to support testing and the test load process
d. Controlling the change process for all software items
e. Giving visibility into the project development activities by SQA and CM team members.

The planned utilization of the <<TBD>> automated CMS streamlines a great deal of the SDL and CM activities. A summary of the CMS features follows:

a. CMS manages the generation and documentation of changes. This provides management with control
b. CMS provides status accounting and maintains an audit trail of changes.
c. CMS stores and tracks all parts of an application through its development life
d. CMS controls the evolution of the application from release to release, forming baselines from which further controlled development and maintenance occurs
e. CMS automatically maintains different versions of information, allowing parallel development and maintenance on a single set of information.
f. CMS generates reports to provide managers and software

engineers with control and visibility throughout the software life cycle.

[TO BE DEVELOPED]

Figure B-1. Software Development Library Products

APPENDIX C

SOFTWARE STANDARDS AND PROCEDURES

12.0 SOFTWARE STANDARDS AND PROCEDURES

12.1 Software Development Techniques and Methodologies

The following paragraphs define the software development standards and procedures which will be applied to the TESTS project software development using DOD-STD-2167A, Defense System Software Development, as a guideline.

The tasks are presented, to the extent possible, in a chronological order. However, there is not a requirement for the tasks to be completed in that order or that the tasks be completed once and for all. In most cases it will be necessary to iterate upon the tasks to complete them. For each of the tasks identified there is a one line summary of the task and a paragraph which describes the task. The one line task summaries are also included in a checklist for each activity. The check lists will be utilized to track the progress of the tasks' completion. The Software Development File/Folder (SDF) for each CSCI will contain the checklists associated with that particular CSCI.

a. System/Software Requirements Analysis Activity (SSRAA).
Tasks to be performed:

(1) Assign Software Engineers
The lead SWE and the PI shall assign a responsible SWE for each CSCI.

(2) Begin initial SSRAA tasks
The responsible SWEs shall perform the following for each CSCI:

(a) Establish SDF
The responsible SWE shall establish a SDF for the CSCI. The form of the SDF may be either electronic or paper. If suitable automated resources are available, all or part of the SDF maybe on electronic media. See the SDF table of contents/ checklists for required contents of the SDF.

(b) Obtain CIN
The responsible SWE shall obtain the CIN for the CSCI and enter it in the SDF.

(c) Initiate allocated requirements review
The responsible SWE shall extract all relevant information and allocated requirements from the FD, SRS (draft), and the IRS (draft) and place in SDF. The responsible SWE shall review allocated requirements for consistency and completeness. Identify any weaknesses (i.e. missing/implied requirements) and place in the SDF. If changes

are necessary to the FD, SRS, or IRS, submit a change request, placing a copy in the SDF.

(d) Establish "make/buy" decisions and reqs
If not predetermined in the FD, the SWE shall review allocated requirements for a "make/buy" decision and record any "buy" decisions in the SDF including derived requirements. Each derived requirement shall have a name, project unique identifier, purpose, performance in measurable terms, associated inputs and outputs and parent requirement.

(e) Review allocated and derived reqs. against FD
The SWE shall analyze the CSCI's set of allocated and derived requirements for completeness and document any identified inconsistencies in the SDF.

(f) Review CSCI's external interfaces
The SWE shall establish for each external I/F, a formal name, project identifier, short description and purpose, source, destination, and (if any) associated data elements and record in SDF.

(g) Identify necessary data elements
The SWE shall establish for each identified (and necessary) data element a formal name, project unique identifier and (as applicable) the following information and record in SDF:

1. communication protocol
2. priority level
3. concurrent or sequential
4. units of measure
5. limit/range
6. accuracy
7. precision/resolution
8. rate

(3) Internal review of initial SSRAA tasks
The LSE (or designee) shall conduct an internal review of the tasks completed thus far. The review should determine that the allocated and derived reqs and associated analysis specify the external interfaces of the CSCI.

(4) Begin remaining SSRAA tasks
Upon successful completion of internal review, the SWEs shall complete the remaining tasks:

(a) Identify necessary internal derived requirements.
The SWE shall identify the derived requirements that are necessary to bridge the external I/Fs from input to output. The SWE shall establish a

formal name, etc. as above for derived reqs. and record in SDF.

(b) Identify necessary internal data elements
The SWE shall identify data elements necessary to support the allocated or derived requirements. The SWE shall establish a formal name, etc. as above for data elements and record in SDF.

(c) Identify necessary internal interfaces
The SWE shall identify the necessary internal I/Fs between capabilities. The SWE shall establish a formal name, project identifier, description and a summary of information transmitted and record in SDF.

(d) Determine requirement testability and test method
The SWE shall review all allocated and derived reqs and determine the testability, the method and the level of testing to be performed for each req and record in SDF. If a requirement is not testable the requirement should either be reworded or deleted.

(e) Establish timing and sizing requirements
The SWE shall identify the timing and sizing reqs for memory and processing for the CSCI and record in SDF.

(f) Identify safety requirements
The SWE shall identify any necessary safety reqs and record in SDF.

(5) Internal review of SSRAA tasks
The LSE shall conduct an internal review of the completed tasks. The review should determine that the allocated and derived requirements and associated analysis present a complete, testable set of engineering and interface requirements for the CSCI.

(6) Generation of SRS and IRS portion
Upon successful completion of the internal review the SWE shall obtain from CM the CIN for the SRS and IRS and have them (relevant portion of the IRS) generated.

(7) Review of SRS and portion of IRS
Upon receipt of the SRS and portion of the IRS the SWE shall schedule with the PI a review of the SRS and IRS. Delegates from the engineers responsible for interfacing hardware shall be a part of the review. The customer shall be invited to attend the review but he may choose to review the results at the next PR.

(8) Software Specification Review preparation
Upon successful completion of the reviews of all the SRSs and the IRS, an evaluation shall be made of their completeness and technical value. The evaluation shall include any risks with an abatement plan and the individual status of each CSCI. Based on this evaluation, a decision shall be made whether the design effort can continue or whether rework is necessary. The evaluation, individual statuses and an overview of the CSCIs, including any support materials, will be presented at the System/Software Specification Review.

b. Preliminary Design Activity (PDA)

There are two separate efforts that take place during the PDA. The first effort involves the top-level design of each of the CSCIs and the interface of these CSCIs with each other and with other CIs (hardware). The first effort evolves from the CSCI-CSC structure to CSC and CSU structure and definition that defines the evolving TESTS prototypes. These prototypes are grouped into builds for the next step of implementation. Emphasis is placed on requirements definition and creating a SDF for the software elements. The second effort consists of outlining the overall system/software test plan and preliminary procedures for all the CSCIs and TESTS and concludes with the generation of the draft of the Simulator Test Plan/Procedures Results/Report (STPPRR) document.
Tasks to be performed:

(1) Establish a preliminary software architecture
Using the TESTS FD as the stimulator baseline document, the LSE, SWEs, and PI shall generate a system level diagram of the simulator/software architecture to depict the relationships between the CSCIs and the other CIs in the system based on the requirements in the SRS. Computer Software Components (CSCs) that were identified in the SRS should be noted, as applicable, in the architectural diagram.

A description of the organizational structure of the CSCIs, including the any internal structure represented, shall be generated by the SWE group to supplement the simulator/ software architectural diagram. Together, the diagram and the description, shall provide the basis for the top-level design until further analysis or design identifies an inconsistency that was not evident in the systems level representation. Any design inconsistency shall be evaluated against the requirements to determine if the requirements were misunderstood or if the requirements were incomplete or incorrect. In the latter case, a problem/change report identifying the discrepancy shall be submitted and recorded in the CSCI SDF. If further work is at risk, because of the discrepancy, the LSWE shall redirect the tasks and report

the risk to the project and the risk will be discussed at a future PR.

(a) Evaluate requirements and identify the CSCs
The SWE group shall identify the CSCs and sub-level CSCs necessary to meet a requirement or set of reqs in the SRS and/or IRS such that all the requirements are allocated to CSCIs. The SWE group shall generate a summary of each CSC's purpose and record this in the CSCI SDF.

(b) Identify any non-developmental software CSCs
The SWE group shall determine any non-developmental software (NDS) that will be incorporated to meet the requirements of a CSC. Refer to paragraph 4.3, Non-developmental Software, in the Software Development Plan for the development requirements regarding incorporation of NDS.

(c) As TESTS prototypes are defined, the above steps are repeated for CSUs, SDFs are created for CSUs, and diagrams are refined to show sub-elements and functions.

(2) Assign Software Engineers
The LSWE shall assign the responsible software engineer(s) (SWEs) to the identified CSCs and CSUs as they evolve.

(3) Begin initial PDA tasks
The SWEs shall perform the following for each CSC:
(a) Establish SDF

The SWE shall establish a SDF for each currently identified CSC. The form of the SDF may be either electronic or paper. If suitable automated resources are available, all or part of the SDF maybe on electronic media. See the SDF table of contents/checklists for the required contents of the SDF.

(b) Identify parent specification(s)
The SWE shall identify the higher-level specification(s) containing the requirements from which the design of the CSCI was derived and record same in the CSC SDF.

(c) Establish the programming language of the CSC
Based on the allocated requirements the SWE shall determine the implementation programming language of the CSC and record this in the CSC's SDF. If Ada is not feasible, the design shall follow the same approach as presented for Ada noting any additional direction given for development "other than Ada". If a requirement or standard can not be met note this in the SDF and the LSWE will discuss the item at a future PR.

(d) Establish preliminary Object-based Architecture
The SWE shall review the Object Oriented Architecture

(OOA) identifying the components of the architecture necessary to implement the CSC.

1. Identify Objects
   a) Identify Object's attributes/characteristics
   b) Identify Object's behavior
2. Identify Managers
   a) Identify control flow of Objects
   b) Identify data flow of Objects
3. Identify Aggregate
4. Generate OOA diagram (depends on <<TBD>> CASE tool, methodology, and graphic representation to be used)

(e) The SWE shall review the components of the OOA and identify which components will be utilized in the development of the CSC. The use of a language other than Ada does not negate the advantages of an object-oriented approach. The language may not support the approach directly or may not have been implemented traditionally in an object-oriented manner. However, the structure and the relationships of the components of the OOA are applicable in other languages.

(4) Review CSC OOA approaches.
The LSWE shall perform an internal review of the CSC OOA approaches for consistency within and across the CSCs. The responsible SWE shall then generate the specific Ada packages in accordance with the <<TBD>> Design and Coding Standards

****
The following notes are possible guidelines for examination and as such are marked fore and aft by the line of asterisks.

The following summarizes <<TBD>> tasks that require further definition for each CSC and also depends on the final CASE tools. CSCs may have sub-level CSCs and, as the prototyping evolves, CSUs will emerge. The following description relates to CSCs and their lower elements as applicable.

(a) Generate a CSCI top-level architecture to graphically illustrate the CSCs and their relationships.

(b) Describe the relationships among the CSCs by identifying and stating the purpose of each CSC-to-CSC interface and the data transmitted via the interface.

(c Identify each system state and mode in which the CSCI operates and the CSCs that execute in each state and mode. A state/CSC table may be provided to illustrate the system states and modes that each CSC executes.

(d) Describe the general flow of both execution control and data between CSCs while operating in the different states and modes. A flow diagram(s) may be used to illustrate the execution control and data flow in each

state and mode.

(e) Allocate the memory and processing time to the CSCs. The allocation may be illustrated by a memory/processing time table.

(f For each CSC, sub-level CSC, and CSU generate the following information:

1. Name
2. Project unique identifier
3. Description of its purpose.
4. Identify the reqs allocated to the CSC from the applicable requirements specification(s). If the CSC is composed of sub-level CSCs, some or all of this information may be referenced and provided by the sub-level CSC or CSU description.
5. Identify the preliminary design of the CSC in terms of execution control and data flow. If a CSC is composed of sub-level CSCs, this description shall identify the relationships among the sub-level CSCs. Repeat this process for CSUs.
6. As applicable, identify each CSCI internal I/F documented in the Software Requirements Specification, that is to be addressed by the CSC and its sub-level CSCs. This information may be referenced rather than duplicated for each sub-level CSC. The process is repeated for CSUs.
7. Identify the derived design requirements for the CSC and any given constraints imposed on or by the CSC. If the CSC is composed of sub-level CSCs, some or all of this information may be referenced and provided by the sub-level CSC description. The process is repeated for CSUs.

(g) Provide traceability of the requirements allocated down to each CSC back to the requirements of the Software Requirements Specification and Interface Requirements Specification. Traceability is updated in the RTM.

(h) Update the FD from CM for the refined CSCIs. Note that CSCs and CSUs are not included in the FD for they will be described in later activities in the SDD and IDD.

(i) Identify and describe the role of the CSCI within the system.

(j Identify and state the purpose of each external I/F of the CSCI.

(k) Include any general information that aids in understanding this document (e.g. background information glossary, formula derivations).

(l) Generate an alphabetical listing of all acronyms, abbreviations, and their meanings as used in this

document.

(m) Utilize appendixes to provide information published separately for convenience in document maintenance (e.g. charts, classified data). As applicable, reference the applicable appendix in the main body of the document where the data would normally have been provided.

****

(5) Generation of the draft STPPRR

Obtain identification number, title, and abbreviation of the system to which this STPPRR applies. It shall also identify the CSCIs, CSCs, and the CSUs by title, abbreviation, and identifier. Briefly state the purpose of the system and the CSCIs, summarize the purpose and contents of this document, describe the relationship to related TESTS documents.

Identify and describe the plans for implementing and controlling the resources (software, firmware, and hardware) necessary to perform formal qualification testing (FQT). To reduce duplication, references may be made in the paragraphs below to the software engineering environment previously described in the SDP for those resources that are used in both environments.

(a) Identify the software items (e.g., operating systems, compilers, code auditors, dynamic path analyzers, test drivers, preprocessors, test data generators, post-processors) necessary to perform the FQT activities. This paragraph shall describe the purpose of each item and shall identify any classified processing or security issues associated with the software items.

(b) Identify the computer hardware, interfacing equipment, and firmware items that will be used in the software and system integration test environment. This paragraph shall describe the purpose of each item and shall identify any classified processing or security issues associated with the hardware or firmware items.

(c Identify the proprietary nature and Government rights associated with each item of the software test environment.

(d) Identify the contractor's plans for installing and testing, controlling, and maintaining each item prior to its use.

(e) Identify each FQT and to describe the FQT requirements for each CSCI, CSC, and CSU to which this STPPRR applies.

1. Identify a CSCI, CSC, or CSU by name and project-unique identifier and describe the scope of testing for the software element.
2. Describe requirements that apply to the FQT or to a group of tests.
3. Describe the types or classes of FQTs to be executed (e.g., stress tests, timing tests, erroneous input tests, maximum capacity tests).
4. Describe the levels at which FQT will be performed
5. Identify and describe each FQT to be conducted on the software element.

(f) Provide the information specified below for the test. Some or all of this information may be provided graphically.

1. Test objective
2. Any special requirements (e.g. 48 hours of continuous facility time, weapon simulation, SWEG, etc.)
3. Test level
4. Test type or class
5. Qualification method
6. Cross reference to the CSCI engineering requirements in the SRS addressed by this test
7. Cross reference to the CSCI interface requirements in the IRS addressed by this test
8. Type of data to be recorded
9. Assumptions and constraints.

(g Describe the data reduction and analysis procedures to be used during and following the tests identified. Describe how information resulting from data reduction and analysis will be retained. The results of data recording, reduction, and analysis activities shall be documented in such a way that the resulting information will clearly show whether the test objectives have been met.

c. Critical Design Activity (CDA)

The tasks accomplished during this activity are a continuation of the PDA with iterations of the evolving prototype efforts. The incremental build definition and the process of "build a little, test a little" continues resulting in refinement of CSCs and completion of the CSU delineation. The main products of this activity are the SDD and the IDD, updates to the SDFs, STPPRR, and the FD and RTM, if needed. Tasks to be performed:

(1) Preliminary IDD
Obtain the identification number, title, and abbreviation of the system(s), CSCI(s), and interface(s) to which the IDD applies. Briefly state the purpose of the system and identify and describe the role of the interfaces, to which this IDD applies, within the system.

Specify for each CSCI to which this IDD applies, its relationship to the HWCIs, CSCIs, or critical items (CI) with which it interfaces. Generate one or more interface diagrams to depict the CI level interface relationships. For each interface state its purpose and describe the design of the interface including the data elements transmitted across he interface. For each data element provide the following information, as applicable:

(a) A project unique identifier for the data element

(b) A brief description of the data element

(c The CSCI, HWCI, or critical item that is the source

(d) The CSCI(s), HWCI(s), or critical item(s that are the users of the data element

(e) The units of measure required for the data element, such as seconds, meters, kilohertz, etc.

(f) The limit/range of values required for the data element (for constants provide the actual value)

(g) The accuracy required for the data element

(h) The precision or resolution required for the data element in terms of significant digits

i The frequency at which the data element is calculated or refreshed, such as 10 KHz or 50 Msec

(j Legality checks performed on the data element

(k) The data type, such as integer, ASCII, fixed, real, enumerated, etc.

1. The data representation/format
2. The priority of the data element

Identify the messages transmitted across the interface by name and project unique identifier, and describe the assignment of data elements to each message. Generate a cross-reference between messages and data elements in both directions. Specify the relative priority of the interface and of each message transmitted across the interface.

Identify and describe the commercial, military, or proprietary communications protocols associated with the interfaces addressing the following communications specification details, as applicable:

(a) Fragmentation and reassembly of messages

(b) Message formatting

Error control and recovery procedures, including fault tolerance features

d Synchronization, including connection establishment, maintenance, termination, and timing

(e) Flow control, including sequence numbering, window size, and buffer allocation

(f) Data transfer rate, whether it is periodic or aperiodic, and minimum interval between transfers

(g) Routing, addressing, and naming conventions

(h) Transmission services, including priority and grade

(i Status, identification, notification, and any other reporting features

(j) Security, including encryption, user authentication, compartmentalization, and auditing.

(2) Preliminary SDD
Obtain the identification number, title, and abbreviation of the system(s), CSCI(s), and interface(s) to which the SDD applies. Briefly state the purpose of the system and identify and describe the role of the interfaces and the hierarchical decomposition of CSCI requirements to CSCs and then to CSUs to which this SDD applies, within the system. Include the <<TBD>> graphical representation for the software architecture and lower level flow diagrams to show requirement flowdown from CSCI to CSC to CSU as applicable. Use the SWE methodology described in the SDP to perform further OOA analysis and describe the product. Make estimations of timing and sizing of the software elements. For each CSCI, CSC, or CSU update or provide the information, as applicable from the PDA, task "4-f". (Note that descriptions will not be repeated if described in detail at a lower CSU level, but may be referenced.)

Typically, the following lists items to be supplied that are more detailed than those mentioned in the PDA:

a) Hierarchical decompositions of the CSCs and sub-level CSCs and charts or flows to depict

(b) Updates of states and modes for the elements

(c The listing of derived requirements and allocation to elements. The RTM is then updated.

(d) The listing of I/O and interfaces is updated

) Man-machine interface considerations are listed

(f) System events, interrupts, error or exceptions are described

(g Processing and algorithms are described

Error handling is described

i Conversions and restrictions of processing or data is included

j Descriptions of the logic flow is included. This could be in the form of PDL and / or charts.

(k) Data base and file information is included.

(l Preliminary test descriptions, procedures, and user considerations are written.

(3) Capture results in the SDFs
Prototypes and incremental builds have been continuing during the CDA and data concerning these efforts are inserted in the SDFs:

(a) All CSCI, CSC, and CSU descriptions, code, prototype test results are captured

(b) Observations of potential risks and feasibility of the build and its effect on TESTS is included

(c) Notes on usage of a software element or prototype are compiled for later inclusion in users manuals

d. Development Configuration Activities (DCA)

This activity includes the Code and Unit Test and SW Integration (SWI) efforts (CSC and CSCI testing) guidelines that are <<TBD>> until decisions on compilers, APSE, SCM are made.

A summary of the Coding Standards, however, will include Appendix B from DOD-STD-2167A as a guideline:

1 These standards apply to all source code developed under the contract.

(2 The Presentation style for the source code shall include standards for:

(a) Indentation and spacing

(b) Use of capitalization

(c Uniform presentation of information in the source code, e. g., the grouping together of all data declarations

(d) Use of headers

(e) Layout of source code listings

f) Conditions for comments and their format

(g) Size of code aggregates by average and a "not to exceed" statement shall be included

(3 Naming conventions shall be described and the reserve words and keywords that are restricted shall be listed

(4 Restrictions on the implementation language due to project or machine-dependent characteristics shall be described

(5 The allowed use of language constructs and features shall be described for all intended implementation language/s

(6) The coding standards shall include controls and restrictions on the complexity of code aggregates.

In addition to these minimum code standards from 2167A, TESTS source code will have the following:

1) All source listings (CSUs and data file listings, as applicable) shall have a Preamble including:

(a) CSU name/identifier and linkages to parent software elements, like CSC, CSCI.

(b) Interfaces, external and internal, identifiers

(c) Usage restrictions

(d) Error handling, if applicable

(e Version information of the software element

(f) Version information of the compiler/assembler

(g) Security related declarations

(h) RTM/requirements related to this element

i) Calling and/or execution parameters

(j) Results parameter descriptions and elements called or invoked by this element

(k) <<TBD>>

(l <<TBD>>

(2) a copy of the listing in the SDF

(3) The software element will be graphically shown on <<TBD>> CASE diagrams/flows/hierarchical/state representations. The software element shall be shown graphically with linkages to other CSUs, CSCs, and CSCI as applicable.

Other standards during the DCA that are <<TBD>> are:

1 CSU testing

(2) String or "threaded string" testing per evolving prototypes of CSU to CSU and related CSC elements

(3) CSC to CSC and incremental build tests

(4) CSCI and large build tests

(5) Test data capture for all the above

(6 The contents of the SCM tool and placing elements under SCM

e. The System Integration and Testing (SIT) Activity

The standards for this activity are minimal since the previous activities primarily contain all developmental steps. The products of SIT are governed by the previously defined STPPRR, SCM procedures, and by the <<TBD>> format for the PCA and FCA.

12.2 Software Development Files/Folders (SDFs)

Software Development Files/Folders (SDFs) shall be established for all developed software on the project. SDFs are utilized to track the various software products (i.e., documents, reports, source code, schedule, test cases, test results, etc.) that are generated in the development of the software. The structure of the SDFs will follow the break-out of CSCIs, CSCs and CSUs. A SDF shall be established for each CSCI. The LSE) or LSE's designee shall be the custodian of the CSCI SDFs. The LSE shall maintain a composite list of all SDFs established and the associated custodians. Lower level SDFs will be established for CSCs and CSUs based on the work packages identified by the LSE and PM/PI. The lower level

SDFs shall be maintained by the assigned software engineer.

SDFs consist of a cover page which includes the title, abbreviation, and the CIN of the CSCI, CSC, or CSU(s) contained within. Also included on the cover page is the project name, the development location, the date the SDF was initiated, the completion dates of the internal and formal reviews, the names and CINs of the associated CDRLs and their completion dates, and the names of the LSE, assigned SWEs and other associated personnel (e.g., QA auditor, customer, etc). Individuals may utilize their initials when signing off on a particular entry. Additionally, if the SDF is for a CSC or CSU, the parent CSCI or CSC title and CIN are to be included as well. At the bottom of the page is a designated area for QA auditor initials and date.

For each level of SDF there is a checklist in the form of a table of contents which identifies what is required within the SDF. The table of contents shall immediately follow the cover page in the SDF. Following the line items in the table of contents are columns for sign-off and date of completion. The assigned engineer initials and dates the item when it is completed. Under each line item there is a line to designate the system name, directory path and file name for information that is stored on electronic media or the page number(s) within the SDF or the location, title and page numbers of the document containing the information when the information is in hard copy.

The SDF cover page and check lists are located in Appendix D, Forms and are <<TBD>>. However, the following is a summary of the types of information that will be captured in the SDF:

(1 A statement of the software element's purpose and requirements, the RTM parts that relate to the element

(2) The graphical representation (flow or structure chart or OOA) showing the element and its related CSUs, CSCs, and CSCI (For TESTS, this should also show the evolving prototype and/or incremental build that contains the element.)

(3) The source code listing, as available

4 Test related items: plans, procedures, testware, results

(5) Users notes, these may later get compiled to form users manuals

(6) SCM items, version, time/date of rev, change reports

7 Meeting notes, agreements, minutes

(8) <<TBD>>

APPENDIX D

FORMS

<<TBD>>

APPENDIX E

SOFTWARE PRODUCT EVALUATION CHECKLISTS

ROUGH DRAFTS

14.1 SPE Checklists

Although the details are <<TBD>> there will be SPE Checklists for the following:

(1 The corrective action process

(2) The program change process

(3) The SDF and its contents

(4 The SSRAA and its products:

a) The RTM

(b) The SDP

(c) The FD

(d) The SRS

(e) The IRS

f The data element information

(g) The related Project Review/s (PR) agenda/s, including a special agenda for the SSSR

(h) An informal walkthru agenda for internal reviews

(5) The PDA and its products:

(a The RTM

(b) The updated SDFs

(c) The updated SRSs and IRS

(d) The STPPRR and related test checklists

(e) The OOA diagrams and other graphical presentations

(f) The CSCI and CSC checklists

(g) The incremental build plan

(h) The updated FD including simulator HW descriptions

(i) The SDC and its environment

j The PR agendas

(6) The CDA and its products:

(a Updates of RTM, SRS, IRS, FD, and STPPRR as needed

(b) The IDD checklist

(c The SDD checklist

(d) The updated SDFs

(e) The incremental build/evolving prototype status

(f The related PR agendas

(g) The CSC and CSU checklists

7 The DCA and its products:

(a) The CSU checklists including the coding standards

(b) The updated SDDs and IDD

(c) The SCM checklist

(d) The users manuals

(e) STPPRR related items, e. g., test requirements, test cases, test descriptions, test procedures, test results

(f The updated SDFs

(g) The incremental build status

(h) The related PR agendas

(8) The SIT and its products:

(a) Updated SDDs and IDD with media to generate the SPS

(b) The PCA and FCA checklists and related PRs

(c) The P/O Manual

(d) The VDD

(e) The test results and reports of the STPPRR

(g) The updated FD for follow on of other TESTS platforms