SOFTWARE DEVELOPMENT PLAN (SDP)
FOR THE
NATO INTEROPERABLE SUBMARINE BROADCAST SYSTEM (NISBS)

NISBS-SDP-01-U-R1C0

Version 1.3

October 32, 1999

Prepared By:

Space and Naval Warfare Systems Center San Diego
Submarine Communications and C4I Systems Division, D83
53560 Hull Street
San Diego, CA 92152-5001

SPAWAR PMW-153	**Program Manager**	**Software Project Manager**
Manager, SSC SD D83	**Quality Assurance**	**Configuration Management**
Systems Engineer	**Hardware Manager**	**Test Manager**

This page intentionally left blank.

TABLE OF CONTENTS

SECTION PAGE

LIST OF FIGURES AND TABLES

Title Page

SECTION 1. SCOPE

1.1 IDENTIFICATION

This Software Development Plan (SDP) establishes the plans to be used during the development of the single Computer Software Configuration Item (CSCI) for the North Atlantic Treaty Organization (NATO) Standard Agreement (STANAG) 5030 Formatter. The Formatter is part of the overall NATO Interoperable Submarine Broadcast System (NISBS) and is referred to herein as the NISBS Formatter. System requirements are specified in references (a) through (d) of 2.1.2.

1.2 SYSTEM OVERVIEW

The primary mission of the NISBS is to provide the U. S. with a NATO-interoperable message preparation, management, format and transmit capability. The NISBS shall be capable of relaying STANAG 5030 formatted broadcasts to submarines in the mid- and north-Atlantic oceans via transmitters located at Naval Radiating Transmit Facility (NRTF) Annapolis, Maryland (both LF and VLF) and NRTF Driver, Maryland. A Submarine Satellite Information Exchange Subsystem (SSIXS) II Message Processing Terminal (MPT) will be located both at the Broadcast Control Authority (BCA), at Commander, Submarine Forces, Allied Command for the Atlantic (COMSUBACLANT) headquarters in Norfolk, Virginia and the alternate BCA located at Commander, Submarine Group 10 (CSG-10) in Kings Bay, Georgia. It is intended that for multilateral (NATO) and US/UK (bilateral) broadcasts, CSG-10 will be the primary BCA. The operational areas supported by these transmitters include the mid- and north-Atlantic and Arctic Ocean areas. In addition, a requirement exists to install an NISBS MPT at Commander, Submarine Forces Pacific (COMSUBPAC) for multilateral and U.S. national broadcasts. A block diagram of NISBS is shown in Figure 1-1.

The purpose of the NISBS Formatter is to provide the U.S. the capability to generate, format, edit, encrypt, modulate, transmit and relay, NATO STANAG 5030 Very Low Frequency/Low Frequency (VLF/LF) submarine broadcasts from both U.S. and NATO-owned Fixed VLF (FVLF) transmitters. The NISBS Formatter will provide COMSUBACLANT with NATO VLF/LF communications interoperability for effective command and control of all NATO submarine forces. A separate development project will provide the U.S. with the capability to detect, demodulate, decrypt, decode, and print STANAG 5030 submarine broadcasts.

The Broadcast Keying Station (BKS) shall be configured with a minimum of two separate message formatters (excluding spares.) One formatter, designated the Black Formatter, shall operate only on encrypted data. The second formatter, designated the Red Formatter, shall operate only on classified clear text data.

1.3 DOCUMENT OVERVIEW

This SDP describes the organization and procedures to be used by the SPAWAR Systems Center, San Diego (SSC SD) in performing software development for NISBS Formatter.
This plan is intended to be used by the sponsor, Space and Naval Warfare Systems Command (SPAWAR) to monitor the procedures, management and contract work effort of the SSC SD D833.

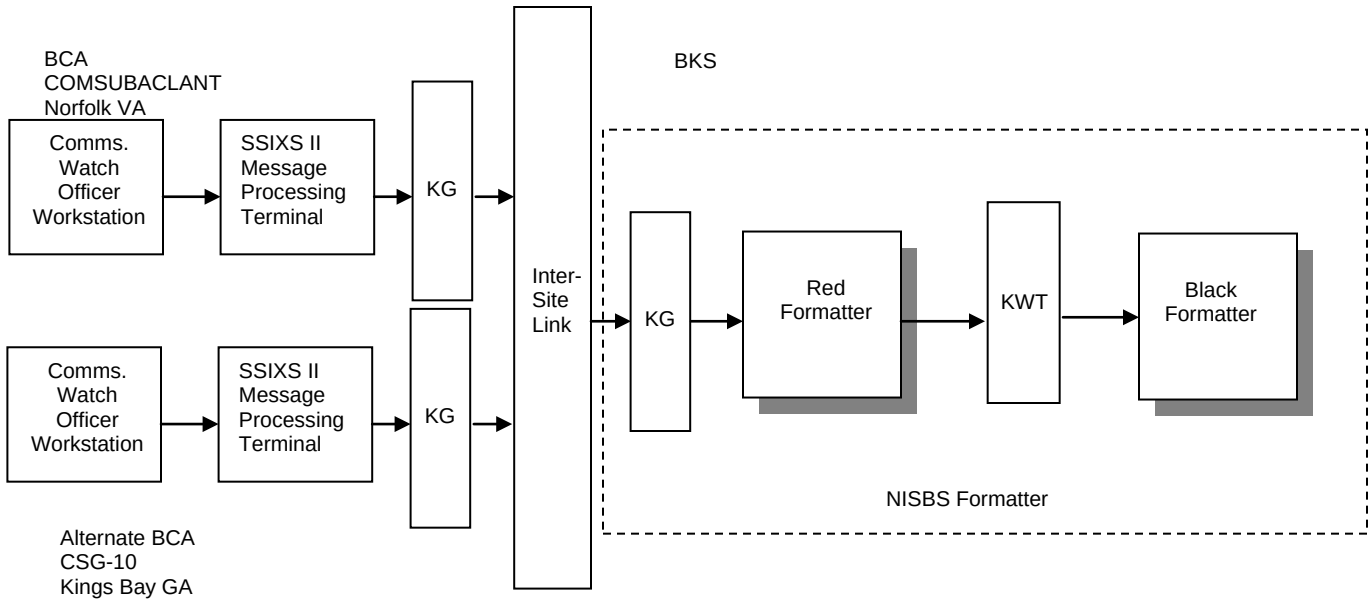


Figure 1-1. NISBS Functional Flow Diagram

This SDP identifies applicable policies, requirements, and standards for NISBS project software development. It defines schedules, organization, resources, and processes to be followed for all software activities necessary to accomplish the development. This SDP contains no privacy considerations pertaining to the NISBS Project.

This document follows the outline of a Software Development Plan contained in the MIL-STD-498 Data Item Description for an SDP: DI-IPSC-8127. It also meets the content requirements for IEEE/EIA 12207.1 Development Process Plan (paragraph 6.5) and Project management plan (paragraph 6.11). The SDP is organized as follows:

Section 2 lists all documents referenced by this SDP and used during its preparation.
Section 3 provides an overview of the required work.
Section 4 describes plans for general software development activities.
Section 5 describes the details of all software planning, design, development, reengineering, integration, test, evaluation, Software Configuration Management (SCM), product evaluation, Software Quality Assurance (SQA), and preparation for delivery activities.
Section 6 defines the project schedule and activity network.
Section 7 describes the project organization and the resources required to accomplish the work.

1.4 RELATIONSHIP TO OTHER PLANS

This SDP and its companion documents, the Software Configuration Management Plan (SCMP) and the Software Quality Assurance Plan (SQAP), serve as the guiding documents to develop the software for the NISBS Project.

SECTION 2. REFERENCED DOCUMENTS

2.1 GOVERNMENT DOCUMENTS

2.1.1 Standards

a. MIL-STD-498, Software Development and Documentation
b. Data Item Description DI-IPSC-81427, Software Development Plan

2.1.2 Other Publications

a. Chief of Naval Operations, NATO STANAG 5030 Format Minimum Shift Key Broadcast Capability, Letter 941D/7U337566 dated 18 May 1997
b. NATO Military Agency for Standardization, STANAG 5030: Single and Multi-channel VLF and LF On-line Broadcast and Off-line OOK Systems, October 1993
c. Operational Concept Document (OCD) for the U.S. NATO Interoperable Submarine Broadcast System, NTX-OCD-01-U-R0C0,1 March 1998
d. System Requirements Specification (SRS) for the U.S. NATO STANAG 5030 Formatter, NTX-SSS-01-U-R0C0, 13 May 1998
e. SSC San Diego Software Engineering Process Policy and associated policies for CM Level 2 and Level 3
f. SSC San Diego Software Project Planning Process
g. SSC San Diego Software Development Plan Template
h. SSC San Diego Software Management for Executives Guidebook

2.2 NON-GOVERNMENT DOCUMENTS

a. IEEE/EIA Standard 12207, Software Life Cycle Processes
b. IEEE Std 1028, Software Reviews and Audits
c. J-STD-016, Software Development – Acquirer-Supplier Agreement
d. The Capability Maturity Model, Version 1.1.
e. Ada 95 Quality and Style: Guidelines for Professional Programmers. Software Productivity Consortium,SPC-94093-CMC, October 1995..

SECTION 3. OVERVIEW OF REQUIRED WORK

Development activities in this plan are subject to the following requirements.

3.1 SYSTEM AND SOFTWARE DEVELOPMENT REQUIREMENTS AND CONSTRAINTS

a. The NISBS formatter software will implement the software requirements identified in the NISBS System Requirements Specification (SRS) and the NISBS Operational Concept Document (OCD).
b. The NISBS Formatter will interface with the SSIXS II Message Processing terminal via the Inter-Site Link. The SSIXS II and ISL are being developed separately from this project.
c. The NISBS Formatter software will be written in the Ada programming language.
d. The MIL-STD hardware is not available at the present, but is expected to be delivered to the development site on time.

3.2 PROJECT DOCUMENTATION REQUIREMENTS AND CONSTRAINTS

Project documents will be developed in accordance with software product descriptions in J-STD-016, Software Development.

3.3 PROJECT POSITION IN THE SYSTEM LIFE CYCLE

NISBS is at the beginning of its life cycle. The software development model selected for NISBS is Once-Through (waterfall) strategy.

3.4 SELECTED PROGRAM/ACQUISITION STRATEGY

The NISBS Formatter is a new development. Software development will be accomplished and managed by Government personnel, with selected activities contracted out using existing contractual agreements.

3.5 PROJECT SCHEDULES AND RESOURCES

Project schedules are dependent on the SSIXS II and ISL systems being completed. Current schedules for these two components is for completion by December 31, 1998, well in time for NISBS integration.

3.6 OTHER REQUIREMENTS AND CONSTRAINTS

None.

SECTION 4. PLANS FOR PERFORMING GENERAL SOFTWARE DEVELOPMENT ACTIVITIES

4.1 SOFTWARE DEVELOPMENT PROCESS

The NISBS Formatter software development is tailored from the guidelines of IEEE/EIA 12207 and MIL-STD-498. Major project development phases and activities related to these standards are shown in Table 4-1.

Table 4-1. Major NISBS Project Activities related to IEEE/EIA 12207 and MIL-STD-498

NISBS Activity	IEEE/EIA 12207.0 Activity	MIL-STD-498 Activity
Project Planning and oversight	5.3.1 Process implementation 7.1 Management Process	5.1 Project planning and oversight 5.2 Establishing a software development environment
Phase 1: Software Requirements	5.3.4 Software reqts analysis	5.5 Software requirements analysis
Phase 2: Software Design	5.3.5 Software arch. design 5.3.6 Software detailed design	5.6 Software design
Phase 3: Software unit development, test, and integration	5.3.7 Software coding and testing 5.3.8 Software integration	5.7 Software implementation and unit testing 5.8 Unit integration and testing
Phase 4: System Qualification Test and Delivery	5.3.9 Software qualification testing 5.3.10 System integration 5.3.11 System qualification testing	5.9 CSCI qualification testing 5.10 CSCI/HWCI integration and testing 5.11 System qualification testing
Phase 5: Support of installation and use	5.3.12 Software installation 5.3.13 Software acceptance support	5.12 Preparing for software use 5.13 Preparing for software transition
Software Configuration Management	6.2 Configuration management process	5.14 Software configuration management 5.17 Corrective action
Software Quality Assurance	6.3 Quality assurance process	5.15 Software product evaluation 5.16 Software quality assurance
Project reviews	6.6 Joint review process	5.18 Joint tech and mgmt reviews
Development activities not applicable to NISBS	5.3.2 System reqts analysis 5.3.3 System arch. design	5.3 System requirements analysis 5.4 System design

The NISBS team will develop the system in accordance with processes defined in Section 5 in the context of the software engineering process model presented above. The software development process is to construct an overall software architecture that will develop software in a single build upon the chosen architecture. The process will integrate reusable software from existing sources with newly-developed software. Software design and coding will be performed using an object oriented design approach and generating class and object process interaction diagrams. Artifacts and evidence of results of software development activities will be deposited in Software Development Files (SDFs) These artifacts, along with pertinent project references will be deposited and maintained in a Software Development Library (SDL) and made available to support management reviews, metrics calculations, quality audits, product evaluations, and preparation of product deliverables.

Following integration of reusable and new software units by the group, Qualification Testing will be performed in accordance with processes defined in Section 5. The test group will prepare

Qualification Test Plans and Procedures and execute test cases. They will generate a Test Report and System Trouble Reports (STRs) to describe results of test analyses.

Software Configuration Management (SCM), Software Quality Assurance (SQA), and support for software delivery will follow detailed processes described in Section 5 of this SDP.

4.2 GENERAL PLANS FOR SOFTWARE DEVELOPMENT

NISBS Formatter software development will follow the guidelines in IEEE/EIA 12207. The development approach will apply selected Level 2 and Level 3 software engineering processes in accordance with the Software Engineering Institute (SEI) Capability Maturity Model (CMM) and product evaluation procedures recommended by SSC San Diego's Software Engineering Process Office (SEPO). The project team has tailored these standards, practices, and processes to NISBS Formatter software development, as described in Section 5 of this SDP.

4.2.1 Software Development Methods

The NISBS Formatter software development will apply the following general methods:

a. Phase 1: Software Requirements. The project will follow the defined processes documented in Section 5 to conduct software requirements analysis and develop the Software Requirements Description (SRD) and preliminary User Documentation Description (UDD). Software requirements will be expressed in language that addresses a single performance objective per statement and promotes measurable verification. An automated data base tool will capture, cross-reference, trace, and document requirements using the database scheme defined in Appendix C. Satisfactory completion of a Software Requirements Review (SRR) will end Phase 1.
b. Phase 2: Software Design. The software architecture will consist of reusable software components and components to be developed. Software requirements will be allocated to one or more components of that architecture. The project will follow the defined processes documented in Section 5 to conduct object-oriented architectural and detailed software design of new software and to capture the design, and reengineer if necessary the software to be reused. Emphasis will be placed on good software engineering principles such as information hiding and encapsulation, providing a complete description of processing, and the definition of all software and hardware component interfaces to facilitate software integration and provide a basis for future growth. The Software Design Description (SDD) and Software Interface Design Description (SIDD) will be produced, and the UDD will be updated. Satisfactory completion of a Software Design Review (SDR) will end Phase 2.

   The project will only design and develop software to meet requirements that cannot be satisfied by reusable software. New software will be based on principles of object-oriented design and exploit object-oriented features associated with the selected high-level language and development environment. New software design will be defined at top-level and detailed design stages of development in the SDD. Software design will promote ease of future growth, specifically new application interfaces. Top-level design will be expressed in a graphical form. Detailed design will also be expressed in a graphical form depicting classes, relationships, operations, and attributes.

   1. The project will develop appropriate standards for design and coding methods for new software by tailoring the "Ada 95 Quality and Style" guidelines (see references in 2.2.)

2. The project will reuse software for requirements that can be satisfied by selected Non-Development Item (NDI) software. Since there may be limited documentation on the design of such software, the development method will involve reengineering the processing design from comments, source listings, user manuals, and hands-on verification of running software. Software designers will select Commercial Off-The Shelf (COTS) reengineering tools and methods best suited to the structure of NDI software and host/target processors. Reengineered design will be modified, as needed, to fully meet allocated requirements, and defined along with new software design, in the SDD.

c. Phase 3: Software Unit Development, Test, and Integration. The project will develop new code (or modify reused code), unit test, integrate, and document software following the processes in Section 5. While reused code will not be expected to conform to a single coding standard, changed source code must be supplemented with sufficient new comments and standard code headers to meet commenting provisions of the coding standard and to promote understandability.
d. Phase 4: System Qualification Test and Delivery. The project will conduct Qualification Testing according to Qualification Test Plans and Procedures, and document results in a Test Report. After successful Software Usability Review (SUR), the software will be ready for installation at the Broadcast Keying Station sites.
e. Phase 5: Support Installation and Use. The NISBS Project team will provide support to software installation, acceptance testing, and user operation. More detail of this phase will be added to future updates of this SDP.

4.2.2 Standards for Software Products

NISBS Formatter development will follow the standards and guidelines listed in Section 2: Referenced Documents. These documents impose standards that are applicable to software requirements, design, coding, testing, and data.
NISBS Formatter software documentation will comply with applicable directions contained in the documents listed in Table 4-2.

Table 4-2. NISBS Documentation Guidelines

NISBS Document	Primary Guideline
Software Development Plan (SDP)	MIL-STD-498 DID: Software Development Plan
Software Requirements Description (SRD)	12207.1 6.22: Software requirements description
User Documentation Description (UDD)	12207.1 6.30: User documentation description
Software Design Description (SDD)	12207.1 6.16: Software design description
Software Interface Design Descr. (SIDD)	12207.1 6.19: Software interface design description
Software Development Files (SDF)	SEPO's Software Development File (SDF) Template V1.1
Integration Test Plan/Procedures	12207.1 6.27/28: Test or validation plan/procedures
Integration Test Report	12207.1 6.29: Test or validation results report
Source Code Record (SCR)	12207.1 6.24: Source code record
Executable Object Code Record (EOCR)	12207.1 6.7: Executable object code record
Qualification Test Plan/Procedures	12207.1 6.27/28: Test or validation plan/procedures
Qualification Test Report	12207.1 6.29: Test or validation results report

4.2.3 Reusable Software Products

This section identifies and describes the planning associated with software reuse during development of the NISBS Formatter project and provisions to promote future reuse of newly-developed software.

NISBS Formatter software components that must be developed will be designed and documented to facilitate their reuse within similar architectures. SSC SD D833 will place reusable software components and associated documentation in the D80 project software development library and provide notification of their availability and development status to SEPO.

4.2.4 Handling of Critical Requirements

Compatibility of interfaces with other systems is important in successful development of the NISBS Formatter software. These programs will be continually monitored by the Software Project Manager to identify, track, and evaluate potential risks.

4.2.5 Computer Hardware Resource Utilization

The Software Project Manager will establish and maintain a detailed schedule for computer hardware resource utilization that identifies anticipated users, purposes, and scheduled time to support analysis, software design, coding, integration, testing, and documentation. It will address sharing of resources by multiple users and workarounds to resolve conflicts and equipment downtime. If computer hardware resource scheduling requires supplementing, potential sources of computer hardware resources including other SSC San Diego projects or commercial vendors will be identified. The Software Project Manager will coordinate resource needs with development, integration, and test groups.

4.2.6 Recording of Rationale

Software development processes for development of the NISBS Formatter Project software described in Section 5 of this SDP identify specific program decision information to be recorded. Additional rationale required for software development will be provided in future SDP updates. Test management decisions and rationale will be recorded in the Test Plans.
Decisions and rationale on software design, coding, and unit testing will be recorded in SDFs as well as system, software, and interface requirements; software engineering, development, and test environments; and system and CSCI test cases.

4.2.7 Access for Acquirer Review

The Software Project Manager will arrange for periodic reviews of NISBS Formatter Project software processes and products at appropriate intervals for the Project Manager. The Software Project Manager will provide representatives of the Project Manager's offices with electronic copies of briefing materials and draft products for review in advance of all reviews, along with discrepancy forms, in accordance with the project's peer review process. The Software Project Manager will direct consolidation of discrepancies and report the status of corrective actions taken in response to reported discrepancies.

SECTION 5. PLANS FOR PERFORMING DETAILED SOFTWARE DEVELOPMENT ACTIVITIES

The overall life cycle to be used for NISBS Formatter development is shown in Figure 5-1. A modified once-through (waterfall) strategy with five major phases and supporting activities will be used. Major documents and management reviews are also shown in the figure. Detailed plans for individual activities are contained in the following subsections.

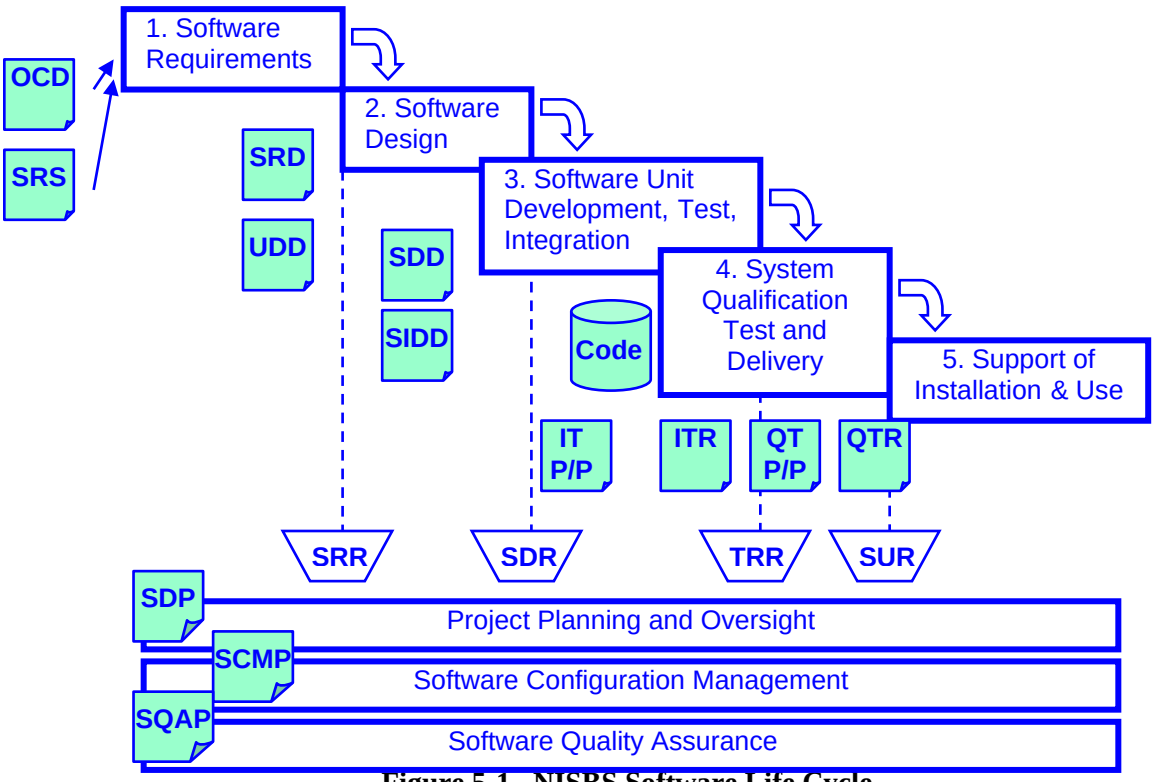


Figure 5-1. NISBS Software Life Cycle

5.1 PROJECT PLANNING AND OVERSIGHT

Good software project management principles shall be followed throughout the NISBS Formatter life cycle. Software development activities shall be planned in advance and monitored carefully throughout the life cycle.

RESPONSIBILITY: the Software Project Manager is responsible for preparing and executing this Software Development Plan.

ENTRANCE CRITERIA: Approved Operational Concept Document (OCD), Systems Requirements Specification (SRS), and project approval received from customer.

INPUTS: OCD, SRS, project approval

TASKS:

1. Clarify the requirements
   - Baseline the requirements documents (OCD and SRS) and control changes
   - Select and tailor standards, processes, and policies to be used (see section 2)
2. Identify the processes
   - Identify the life cycle to be used on the project (see Figure 5-1 and section 4)
   - Establish processes for each phase of the life cycle (see sections 5.2 to 5.6)
3. Document the plans
   - Document software plans in this Software Development Plan. Participation by appropriate parties in the SDP Peer Review and acceptance signatures on the title page demonstrates agreements to commitments in this SDP.
   - Identify work products (see section 6)
   - Establish schedules for the timely completion of tasks (see section 6)
   - Estimate the size of software work products
   - Estimate the project effort, costs, and resources (see section 7.2)
   - Identify project risks (see section 5.10)
   - Establish the project organization (see section 7.1)
4. Track the progress
   - Monitor execution of planned activities (see Project Measurement Plan, 5.13)
   - Analyze status and take action to modify processes, change resources, adjust schedules, or amend plans to satisfy requirements. Obtain agreements from affected groups.
   - Conduct Management Reviews and Status Reviews to determine status of ongoing operations (see section 5.9)
   - Review status of all activities periodically with SSC SD Senior Management
5. Control the products
   - Establish a software configuration management activity to control baselined work products (see Section 5.7)
   - Establish a software quality assurance activity to monitor project products and processes (see section 5.8)
6. Cultivate teamwork
   - Assemble and manage the required project staff and resources
   - Provide training needed by the project staff (see Project Training Plan, 5.14)
   - Include contractors in technical discussions (see section 5.11)
7. Apply appropriate technology
   - Consider reuse during all life-cycle phases (see section 4.2)
   - Satisfy guidelines of Capability Maturity Model maturity levels 2 and 3
   - Strive to continually improve project processes (see section 5.12)

OUTPUTS: this Software Development Plan, operational NISBS Formatter system

EXIT CRITERIA: Satisfactory execution of this Software Development Plan

5.2 PHASE 1: SOFTWARE REQUIREMENTS

The NISBS Team shall establish and document software requirements, including the quality characteristics, specifications, as described below.

RESPONSIBILITY: Software development team, System engineering team

ENTRANCE CRITERIA: Approved Operational Concept Document (OCD) and Systems Requirements Specification (SRS) received from customer.

INPUTS: OCD and SRS

TASKS:

1. Document the software requirements in Software Requirements Description (SRD), including functional and capability specifications, interfaces, safety specifications, security specifications, human-factors engineering, data definition and database requirements, installation and acceptance requirements, user documentation, user operation and maintenance requirements.
2. Normalize software requirements into a database using the scheme defined in Appendix C. Evaluate the requirements against criteria: traceability to system requirements, external consistency with system requirements, internal consistency, testability, flexibility of software design, operation, and maintenance.
3. Develop a preliminary User Documentation Description (UDD) in accordance with 4.2
4. Conduct Peer Reviews on SRD and UDD in accordance with 5.9
5. Conduct Software Requirements Review (SRR) in accordance with 5.9

OUTPUTS: Approved SRD and UDD

EXIT CRITERIA: Successful completion of SRR checklist (see SME Guidebook)

5.3 PHASE 2: SOFTWARE DESIGN

The NISBS Team shall transform the requirements for the software item into an architecture that describes its top-level structure and identifies the software components, and then develop a detailed design for each software component.

RESPONSIBILITY: Software Development Team

ENTRANCE CRITERIA: Successful completion of SRR checklist

INPUTS: Approved SRD and UDD

TASKS:
1. Transform requirements into a top-level architecture
2. Document top-level design for interfaces
3. Document top-level design for the database
4. Develop the design for each component and document it in the Software Design Description (SDD) in accordance with 4.2
5. Develop the design for interfaces and document it in the System Interface Design Description (SIDD) in accordance with 4.2
6. Document the design for database
7. Evaluate architecture against criteria: traceability to requirements, external consistency with requirements, internal consistency between software components, appropriateness of design methods and standards, feasibility of design, operation, and maintenance.
8. Update the UDD as necessary based on the software design.
9. Conduct Peer Reviews on SDD, SIDD, and updated UDD in accordance with 5.9
10. Conduct Software Design Review (SDR) in accordance with 5.9

OUTPUTS: SDD, SIDD, updated UDD

EXIT CRITERIA: Successful completion of SDR checklist (see SME Guidebook)

5.4 PHASE 3: SOFTWARE UNIT DEVELOPMENT, TEST, AND INTEGRATION

The NISBS Team shall transform the software design into executable code, conduct unit testing of each unit, and conduct integration testing of all units.

RESPONSIBILITY: Software Development Team, Testing Team

ENTRANCE CRITERIA: Approved SRD and UDD for Integration Test Plans/Procedures; successful completion of SDR checklist to begin Coding and Unit Testing

INPUTS: Approved SDD and SIDD

TASKS:

1. Develop each software unit and database and document in Software Development Files in accordance with 4.2
2. Develop unit test procedures and data for testing each software unit
3. Conduct unit testing to ensure that each unit satisfies its requirements, and document results in SDFs
4. Evaluate code and test results considering this criteria: traceability to requirements, external consistency with requirements, internal consistency, test coverage of units, appropriateness of coding methods and standards used, feasibility of integration, testing, operation, and maintenance.
5. Prepare Source Code Record (SCR) and Executable Object Code Record (EOCR) in accordance with 4.2
6. Document integration plans and procedures in the Integration Test Plan/Procedures (ITP/P) in accordance with 4.2
7. Conduct integration tests and document results in the Integration Test Report (ITR) in accordance with 4.2
8. Evaluate plans and tests against criteria: traceability to requirements, external consistency with requirements, internal consistency, test coverage of requirements, appropriateness of test standards and methods used, conformance to expected results, feasibility of software qualification testing, operation, and maintenance.
9. Conduct Peer Reviews on software code, SDF, SCR, EOCR, ITP/P, and ITR in accordance with 5.9
10. Conduct Test Readiness Review (TRR) in accordance with 5.9

OUTPUTS: Integrated, executable code; SCR, EOCR, ITP/P; ITR

EXIT CRITERIA: Successful completion of TRR checklist (see SME Guidebook)

5.5 PHASE 4: SYSTEM QUALIFICATION TEST AND DELIVERY

The NISBS Team shall conduct qualification testing for each software item, integrate software configuration items with hardware configuration items, and conduct system qualification testing.

RESPONSIBILITY: Testing Team

ENTRANCE CRITERIA: Approved SRD, UDD for Qualification Test Plans/Procedures; successful completion of TRR checklist to begin Qualification Testing

INPUTS: Code package that has passed integration testing, SCR, EOCR, SRD, UDD

TASKS:

1. Document system qualification plans and procedures in the Qualification Test Plan/Procedures (QTP/P) in accordance with 4.2
2. Integrate the software with hardware configuration items, manual operations, and other systems as necessary, into the NISBS Formatter system.
3. Conduct system qualification tests and document results in the Qualification Test Report (QTR) in accordance with 4.2
4. Evaluate test plans and tests against criteria: traceability to requirements, external consistency with requirements, internal consistency, test coverage of requirements, appropriateness of test standards and methods used, conformance to expected results, feasibility of system qualification testing, operation, and maintenance.
5. Conduct Peer Reviews on QTP/P and QTR in accordance with 5.9
6. Conduct Software Usability Review (SUR) in accordance with 5.9
7. Prepare product for installation

OUTPUTS: Integrated, executable system; QTP/P; QTR

EXIT CRITERIA: Successful completion of SUR (see IEEE Std 1028 guidelines)

5.6 PHASE 5: SUPPORT OF INSTALLATION AND USE

The NISBS Team shall install the completed system at the designated user sites and conduct life-cycle support as necessary.

RESPONSIBILITY: NISBS Project Manager, supported by members of the software development team and testing team

ENTRANCE CRITERIA: Successful completion of SUR checklist

INPUTS: Integrated, executable system, UDD

TASKS:

1. Assist SPAWAR PMW-153 in installation and acceptance test planning
2. Support installation of the NISBS Formatter at the NRTF, Annapolis MD.
3. Support installation of the NISBS Formatter at the NRTF, Driver MD.
4. Support acceptance testing as needed.
5. Provide life-cycle support through implementation of Engineering Change Requests and Software Trouble Reports as directed by SPAWAR PMW-153.
6. Assist with retirement or replacement of the system as needed.

OUTPUTS: Executable system in place through retirement

EXIT CRITERIA: Successful retirement/replacement of system

5.7 SOFTWARE CONFIGURATION MANAGEMENT

The NISBS Project will apply administrative and technical procedures throughout the software life cycle to identify, define, and baseline software items; control modifications and releases of the items; record and report the status of the items and modification requests; ensure the completeness, consistency, and correctness of the items; and control storage, handling, and delivery of the items.

A Software Configuration Control Board (SCCB), led by the Sponsor, will oversee configuration management activities. The Configuration Management team will develop a NISBS SCM Plan with details of the following activities.

5.7.1 Configuration Identification

The CM Team will assign unique identifiers to NISBS program code and related documentation that specify the configuration item and its version.

5.7.2 Configuration Control

Software products will be baselined under CM control when they have completed the Peer Review specified in subsection 5.9. Change requests to baselined products will be identified and recorded by the CM Team. The SCCB will evaluate and approve or disapprove changes. An audit trail shall be maintained by the CM team to trace each modification to baselined products, and its reason and authorization.

5.7.3 Configuration Status Accounting

The CM Team will prepare records and status reports as needed to show the status and history of controlled items.

5.7.4 Configuration Audits

The CM Team will evaluate the functional completeness of the software items against their requirements and the physical completeness of the software item (whether their design and code reflect an up-to-date technical description).

5.7.5 Release Management and Delivery

The release and delivery of NISBS software products and documentation will be formally controlled. Master copies of code and documentation will be maintained for the life of the NISBS system.

5.8 SOFTWARE QUALITY ASSURANCE

The NISBS Project will apply a quality assurance process to provide adequate assurance that the software products and processes in the project life cycle conform to their specified requirements and adhere to their established plans.

The Quality Assurance Team will prepare and execute a NISBS SQA Plan containing quality standards, methodologies, procedures, tools, resources, schedules, and responsibilities for performing the following activities.

5.8.1 Product Assurance

SQA will assure that each required software product exists and has undergone software product evaluations (e.g., Peer Reviews specified in 5.9), testing, and problem resolution, as required.

5.8.2 Process Assurance

SQA will assure that each required process, activity, and task is performed in accordance with this plan and the SQA Plan.

5.8.3 Independence in Software Project Evaluation

Individuals responsible for conducting SQA evaluations shall not be the developers of the software product, performers of the process, or responsible for the software product or process. This does not preclude such individuals from taking part in these evaluations. The individuals responsible for assuring compliance shall have the resources, responsibility, authority, and organizational freedom to permit objective SQA evaluations and to initiate and verify corrective actions. Specifically, the NISBS SQA Team has the authority and freedom to report discrepancies and deviations directly to the Sponsor and to the SSC SD management above the NISBS Project Manager.

5.9 PROJECT REVIEWS

The purpose of peer reviews and management reviews is to provide management with tracking and oversight of the progress of software development undertaken by the NISBS Project and fulfillment of requirements. Timely technical and management reviews at the appropriate level of detail facilitate information reporting and interchange that tracks progress against plans, identify and resolve action items, and verify appropriate expenditure of assigned resources.

5.9.1 Peer Reviews

Peer Reviews (Joint Technical Reviews) will be held on work products using guidelines in the SSC SD Peer Review Process, Version 1.0, 22 June 1998, based on Table 5.9-1.

Table 5.9-1. Peer Reviews for NISBS Work Products

NISBS Work Product	Type of Peer Review
Software Development Plan (SDP)	Formal Inspection
Software Requirements Description (SRD)	Formal Inspection
Preliminary User Documentation Description (UDD)	Formal Inspection
Software Design Description (SDD)	Formal Inspection
Software Interface Design Description (SIDD)	Formal Inspection
Final User Documentation Description (UDD)	Formal Inspection
Code of Software Units	Walkthrough
Software Development Files (SDF)	Walkthrough
Integration Test Plans/Procedures	Technical review
Integration Test Report	Walkthrough
Source Code Record (SCR)	Technical review
Executable Object Code Record (EOCR)	Technical review
Qualification Test Plan/Procedures	Technical review
Qualification Test report	Walkthrough

5.9.2 Management Reviews

The following reviews will be held using guidelines in Annex G of IEEE/EIA 12207.2 and IEEE Std 1028. Attendees shall include SPAWAR PMW-153, the NISBS Project Manager, the Hardware Manager, the Software Project Manager, and others designated by the NISBS Project Manager. The focus will be on completion of review checklists derived from the Software Management for Executives Guidebook, and metrics collected in accordance to the Project Measurement Plan..

- System Requirements Review (SRR)
- Software Design Review (SDR)
- Test Readiness Review (TRR)
- Software Usability Review (SUR)

5.9.3 Status Reviews

Project Status Reviews among all project members will be held at least biweekly. Monthly status meetings with SPAWAR PMW-153 will be held by the NISBS Project Manager and other designated individuals. The focus of these informal reviews will be the status of current processes and completion status of Management Review checklists.

5.10 RISK MANAGEMENT

The project risks are identified in Table 5.10-1. They will be tracked by the NISBS Project Manager and reported in the monthly status meetings and in each Joint Management Review.

Table 5.10-1. Risk Factors for the NISBS Project

Risk Factor (and Reason for Significance)	Metrics (and How Collected)	Risk Reduction/ Prevention Plan	Contingency Plan	Entrance Criteria
1. Requirements stability. (More people and organizations (e.g., Navy, NATO) involved means more time, more requirements, and requirements. instability which leads to cost/schedule slippage)	Track number of requirements: original, new, changed, deleted (Count new, changed, and deleted requirements every month vs. previous month by automated tool which reads requirements database)	Report require-ments stability with written estimated impact assessment to all parties every month and baseline (freeze) requirements at planned time in schedule	Reduce functionality to meet cost and schedule or increase funds and schedule to include new/changed requirements	If number of changed req. and number of new requirements per month exceeds 4
2: New Technology (Learning curves for OOD design techniques and Ada language by the development team will take time; compiler maturity for target hardware (PCs) is low)	Count staff members qualified in each discipline (When staff member gets minimum score on OOD/Ada test or meets other objective criteria)	1) Provide technical training in OOD and Ada; 2) Use and enforce design and coding standards	Employ OOD/Ada expert consultant(s) to assist development	If the design or coding milestones fall behind schedule by more than 15%.
3: Reliability of support organizations. (Non-delivery of required hardware and/or materials will directly impact on our ability to deliver the complete system)	Production, installation, and test milestones of each required outside element (Milestone progress reports from all critical suppliers and vendors)	Provide a bonus structure to each vendor and supplier that is delivering a required item for delivering on time	Identify alternative vendors and suppliers and purchase spares as backup	Delta time threshold over 2 weeks past scheduled delivery date
4: Overall system testability. (No test facility exists that matches the real system from beginning to end as it exists in the field.)	Percentage of Interfaces tested and verified as correct (Overall system interface requirements tracked through bi-weekly status reports until complete)	Develop a simulator and test environment that replicates as nearly as possible the whole system, especially the interfaces	Move testing to field sites	If any required interfaces cannot be tested in a simulator environment by their milestone due date

<<Para. 5.11 Subcontractor Management (page 5-12) and para 5.12 Improvement of Project Processes (page 5-13) not included in this version>>

SUBCONTRACTOR MANAGEMENT

5.11 IMPROVEMENT OF PROJECT PROCESSES

5.12 PROJECT MEASUREMENT PLAN

5.12.1 Measurement Specifications

Effective management of the NISBS project will require ongoing measurement of project status. Table 5.13-1 identifies the issues pertinent to timely oversight, and the corresponding core measurements, data collection sources, and report formats that will be used. Additional details are found in SEPO's *Software Project Tracking and Oversight Process*, Appendix A: Sample Software Measurement Plan.

Table 5.13-1. NISBS Project Status Measurements

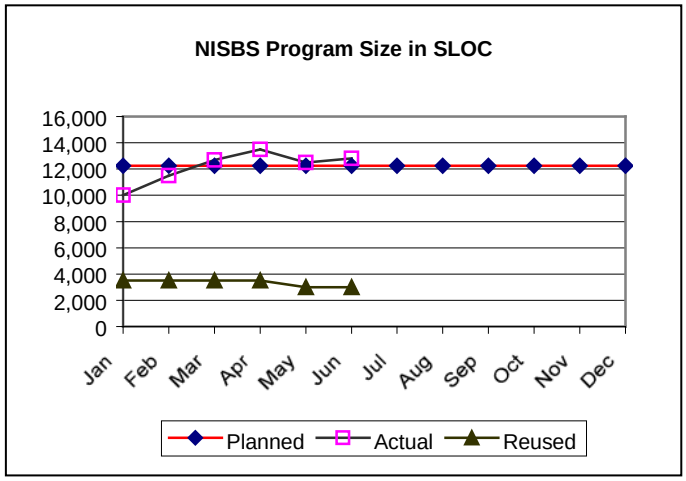
Issues	Core Measurement	Data Collection	Report Format
Program Size	Units/SLOC/Objects planned vs. actual	CM Data Base	Line Graph
Effort	Actual staff size vs. planned	Software Project Manager	Line Graph
Cost	Actual costs expended vs. costs planned	Microsoft Project Plan	Line Graph
Schedule	Actual dates vs. planned dates	Microsoft Project Plan	Gantt Chart
Quality	Trouble Reports open vs. closed	CM Data Base	Line Graph
Stability	Requirements Status/Traceability	RM Data Base	Line Graph
Risks	As required if not covered above	As required	As required

5.12.2 Sample Report Formats

Example report formats are shown below.

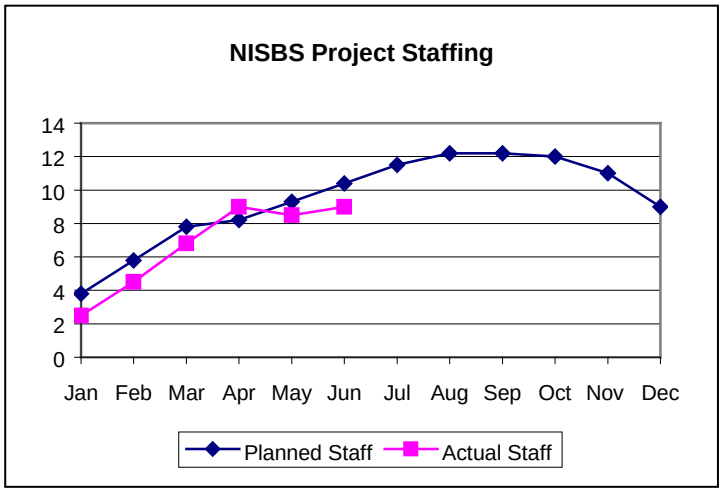
5.13.2.1 Program Size

Size measurements are used to depict the magnitude of deliverable code and the status of code development on the project. Functional size is measured in terms of the requirements. The measure of the code production work necessary to implement the system can be measured in terms of source lines of code(shown here), or total objects, functions points, or software units based on the environment (e.g., language, code generation tools).



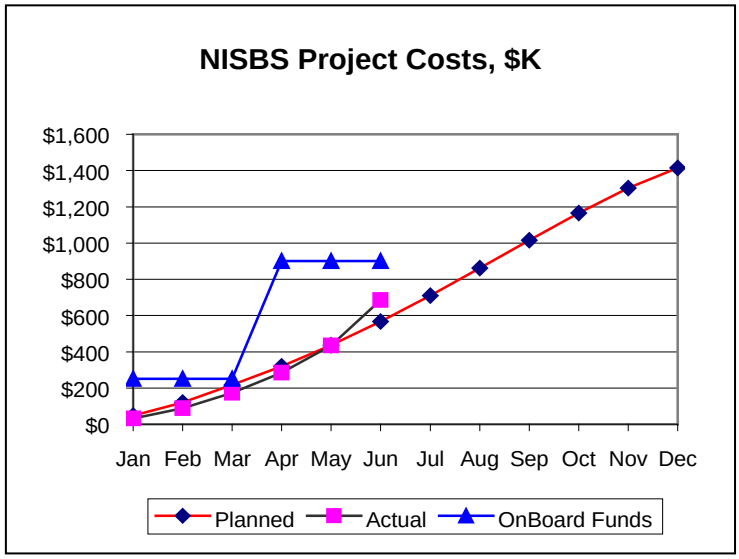
5.13.2.2 Effort

The object of this measurement is to illustrate management success in meeting staffing requirements for the software project. The planned curve is derived from the Software Project Planning effort. This data is plotted against the current staff total supporting the project.



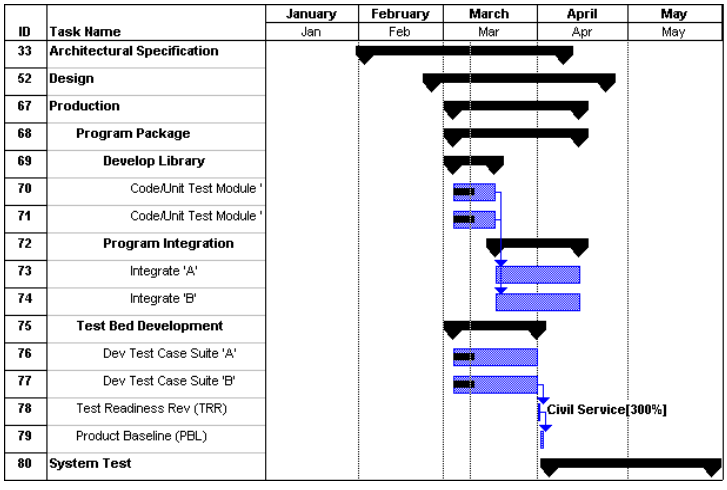
5.13.2.3 Costs

This measurement shows the expenditures of funds relative to the original plan. The graph should include a line indicating current onboard funds to allow visibility of any impending problems that may cause a work stoppage.



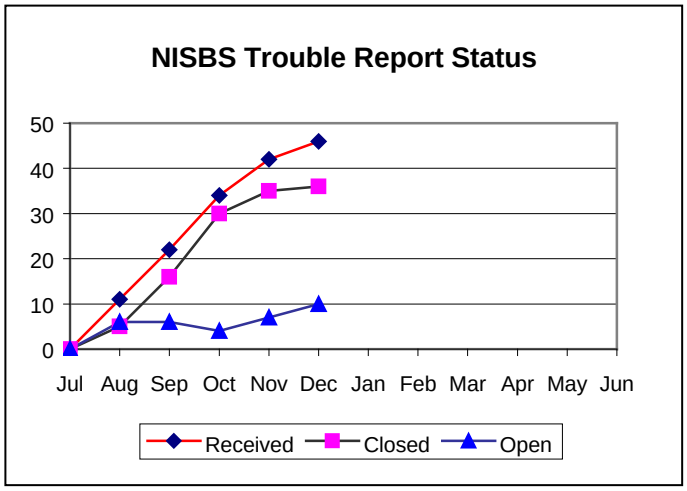
5.13.2.4 Schedule

The Milestones measurement shows tasks scheduled and those that have been accomplished. The information is derived from the Microsoft Project Plan. In order to show a current picture, focus the view of the schedule on activities that are within the current time window using screen snaps from Microsoft Project.



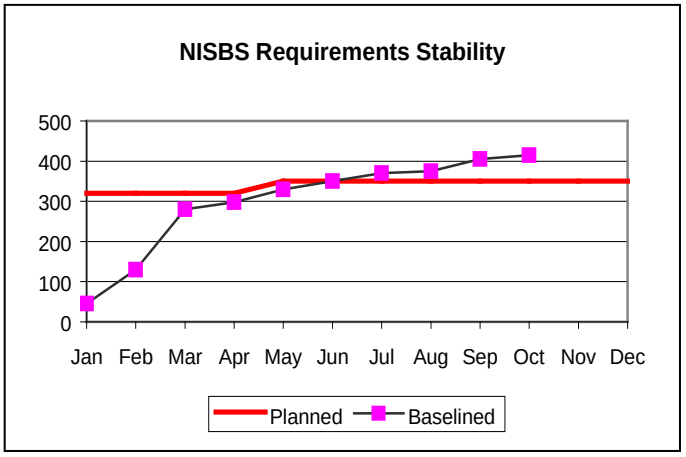
5.13.2.5 Quality

Tracking the status of the program's Trouble Reports (TRs) shows insight into the quality of the product being developed. Plot the TRs received versus those closed, and the difference is TRs still open. An Open TR line that is slanting downward shows successful elimination of problems and increased quality.



5.13.2.6 Stability

The object of measuring the status of requirements is to demonstrate the stability of the implementation effort. The graph should show the planned size of the software effort in terms of total requirements planned and the current number of requirements baselined for the implementation.



5.12.3 Measurement Responsibilities

Project status measurements will be collected under control of the Software Project Manager, with the assistance of the QA Team and inputs from the Software Development Team. Reports will be presented to the NISBS Project Manager and sponsors in the monthly status meetings and at each Joint Management Review. In addition, measurements reflecting problem areas or significant variances will be reviewed at the biweekly Project Status Reviews and with the Project Manager on an as-needed basis.

5.13 PROJECT TRAINING PLAN

Project training will be provided by formal courses, facilitated video, informal "chalk talks," or on-the-job training. Software engineering training will be in accordance with the SSC SD Software Engineering Training Plan. Skills needed by the NISBS Formatter team include:

- Project management/software project management skills (SSC SPM course)
- The Capability Maturity Model and software process improvement (SSC SPIRIT course)
- The role of NISBS in military communications (project-specific)
- Computer program development using the Ada language
- Peer reviews (SSC Peer review Workshop)
- Operation and use of the NISBS software development facility (project-specific)
- Specific processes: planning and oversight, quality assurance, configuration management, etc.

A matrix of training requirements by project organizational group is shown in Table 5.14-1.

Table 5.14-1. NISBS Training Matrix

Category	Name / Course	SPM	SPIRIT	NISBS role	Ada programming	Peer Reviews	NISBS Facility	Planning /Oversight	SQA	SCM	Etc.	Etc.
Project Manager	B. Barlin	√	√	√		√	√	√				
Hardware Manager	A.B. Seaforth	√		√		√	√	S				
Hardware Engineers	D. E. Effort			J		S	P					
	G. H. Eyestrain			J			P					
Software Proj. Mgr.	J. K. Elementary	√	√	√	√	√	P	S				
Software Developers	M. N. O"Leary		√	J	√	√	P	S				
	P. Q. Arbuckle		S	J	√	S	P		S			
	S. T. Unionjack		S	J	√	√	P			S		
	V. W. Exray		S	J	S	S	P					
	H. X Yznaga		S	J	S	√	P					
Quality Assurance	S. Holmes		√	J		√	P	S	√			
Config. Management	K. O. Keys		S	J		√	P			√		
Test Team	I. M. Breakem			J		S	P		J	S		
	U. Checkum			J		√	P		J			

Legend for Individuals:
√ = training completed
S = Required course , SSC SD Training needed this year
J = Required course , On-the-job training needed this year
P = Required course , Project training needed this year

SECTION 6. SCHEDULES AND ACTIVITY NETWORK

6.1 SCHEDULES

NISBS Formatter development activities are shown in Table 6-1. A Schedule and Milestone chart is contained in Figure 6-1.

Table 6-1. Development Activities

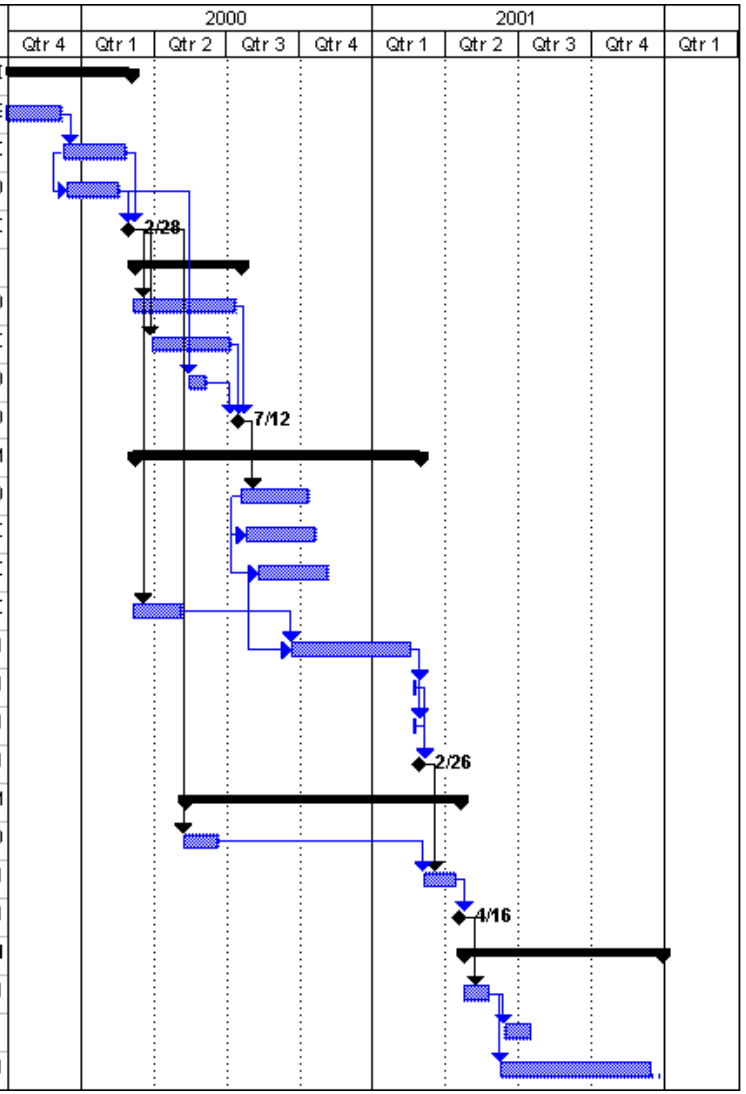
Task Name	Duration	Start	Finish
Phase 1: Software Requirements	95 days	Fri 10/1/99	Fri 3/3/00
Develop Software Development Plan (SDP)	45 days	Fri 10/1/99	Wed 12/8/99
Develop Software Reqts Description (SRD)	45 days	Thu 12/9/99	Fri 2/25/00
Develop preliminary User Documentation Description (UDD)	35 days	Thu 12/16/99	Thu 2/17/00
Conduct Software Reqts Review (SRR)	5 days	Mon 2/28/00	Fri 3/3/00
Phase 2: Software Design	95 days	Mon 3/6/00	Tue 7/18/00
Develop Software Design Description (SDD)	90 days	Mon 3/6/00	Tue 7/11/00
Develop Software Interface Design Description (SIDD)	70 days	Tue 3/28/00	Wed 7/5/00
Update User Documentation Description (UDD)	15 days	Mon 5/15/00	Mon 6/5/00
Conduct Software Design Review (SDR)	5 days	Wed 7/12/00	Tue 7/18/00
Phase 3: Software unit development, test, and integration	240 days	Mon 3/6/00	Fri 3/2/01
Select or Code Software Units	60 days	Wed 7/19/00	Thu 10/12/00
Develop Software Development Files (SDF)	60 days	Wed 7/26/00	Fri 10/20/00
Conduct Unit Tests	60 days	Wed 8/9/00	Fri 11/3/00
Prepare Integration Test Plans/Procedures	45 days	Mon 3/6/00	Fri 5/5/00
Perform Integration Test, write Integ. Test Report	90 days	Thu 9/21/00	Fri 2/16/01
Prepare Source Code Record (SCR)	5 days	Mon 2/19/01	Fri 2/23/01
Prepare Executable Object Code Record (EOCR)	5 days	Mon 2/19/01	Fri 2/23/01
Conduct Test Readiness Review (TRR)	5 days	Mon 2/26/01	Fri 3/2/01
Phase 4: System Qual Test and Delivery	230 days	Mon 5/8/00	Fri 4/20/01
Prepare Qualification Test Plan/Procedures	30 days	Mon 5/8/00	Mon 6/19/00
Conduct Qual Test; write Qual Test Report	30 days	Mon 3/5/01	Fri 4/13/01
Conduct Software Usability Review (SUR)	5 days	Mon 4/16/01	Fri 4/20/01
Phase 5: Support Installation and Use	163 days	Mon 4/23/01	Fri 12/28/01
Install at Annapolis MD	25 days	Mon 4/23/01	Fri 5/25/01
Install at Driver VA	25 days	Tue 6/12/01	Tue 7/17/01
Provide Life-Cycle support	138 days	Tue 5/29/01	Fri 12/28/01

6.2 ACTIVITY NETWORK

An activity network, depicting sequential relationships and dependencies among activities and identifying those activities that impose the greatest time restrictions on the project, is shown in Appendix A.

Figure 6-1. NISBS Schedule and Major Milestones

ID	Task Name	Duration	Start	Finish
1	**Phase 1: Software Requirements**	**95 days**	**Fri 10/1/9**	**Fri 3/3/0**
2	Develop Software Development Plan (SDP)	45 days	Fri 10/1/9	Wed 12/8/9
3	Develop Software Reqts Description (SRD)	45 days	Thu 12/9/99	Fri 2/25/0
4	Develop preliminary User Documentation Description (UDD)	35 days	Thu 12/16/99	Thu 2/17/00
5	Conduct Software Reqts Review(SRR)	5 days	Mon 2/28/00	Fri 3/3/0
6	**Phase 2: Software Design**	**95 days**	**Mon 3/6/00**	**Tue 7/18/00**
7	Develop Software Design Description (SDD)	90 days	Mon 3/6/00	Tue 7/11/00
8	Develop Software Interface Design Description (SIDD)	70 days	Tue 3/28/00	Wed 7/5/0
9	Update User Documentation Description (UDD)	15 days	Mon 5/15/00	Mon 6/5/00
10	Conduct Software Design Review(SDR)	5 days	Wed 7/12/0	Tue 7/18/00
11	**Phase 3: Software unit development, test, and integration**	**240 days**	**Mon 3/6/00**	**Fri 3/2/01**
12	Select or Code Software Units	60 days	Wed 7/19/0	Thu 10/12/00
13	Develop Software Development Files (SDF)	60 days	Wed 7/26/0	Fri 10/20/0
14	Conduct Unit Tests	60 days	Wed 8/9/0	Fri 11/3/0
15	Prepare Integration Test Plans/Procedures	45 days	Mon 3/6/00	Fri 5/5/0
16	Perform Integration Test, write Integ. Test Report	90 days	Thu 9/21/00	Fri 2/16/01
17	Prepare Source Code Record (SCR)	5 days	Mon 2/19/01	Fri 2/23/01
18	Prepare Executable Object Code Record (EOCR)	5 days	Mon 2/19/01	Fri 2/23/01
19	Conduct Test Readiness Review(TRR)	5 days	Mon 2/26/01	Fri 3/2/01
20	**Phase 4: System Qual Test and Delivery**	**230 days**	**Mon 5/8/00**	**Fri 4/20/01**
21	Prepare Qualification Test Plan/Procedures	30 days	Mon 5/8/00	Mon 6/19/00
22	Conduct Qual Test; write Qual Test Report	30 days	Mon 3/5/01	Fri 4/13/01
23	Conduct Software Usability Review(SUR)	5 days	Mon 4/16/01	Fri 4/20/01
24	**Phase 5: Support Installation and Use**	**163 days**	**Mon 4/23/01**	**Fri 12/28/01**
25	Install at Annapolis MD	25 days	Mon 4/23/01	Fri 5/25/01
26	Install at Driver VA	25 days	Tue 6/12/01	Tue 7/17/01
27	Provide Life-Cycle support	131 days	Thu 6/7/01	Fri 12/28/01



SECTION 7. PROJECT ORGANIZATION AND RESOURCES

SSC SD is organized by departments into technology areas. D80 is the Communications and Information Systems Department. Within D80 is the Submarine Communications and C4I Systems Division, D83, which includes the Operational Systems Branch, Code D833. The Operational Systems Branch shall be responsible for the software requirements analysis, design, coding, integration and testing of the delivered NISBS Formatter software.

7.1 PROJECT ORGANIZATION

The project organization for the NISBS Formatter development is shown in Figure 7-1.

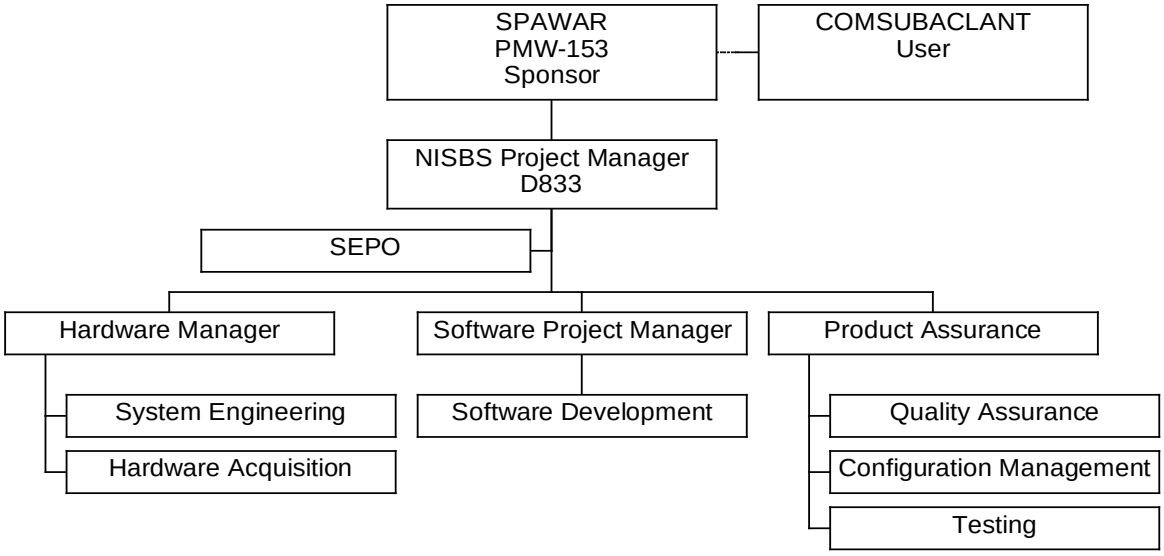


Figure 7-1. NISBS Project Organization

The organizations involved and their roles in this project are shown in Table 7-1.

Table 7-1. Project Roles and Responsibilities

Role	Responsibilities
Sponsor (PMW-153)	• Oversee project execution • Provide approved OCD and SRS • Chair SCCB
User (COMSUBACLANT)	• Review OCD, SRS, UDD • Conduct Qualification Testing • Accept system
NISBS Project Manager	• Manage NISBS development • Chair management reviews and project reviews
NISBS Hardware Manager	• Oversee system engineering and hardware
System Engineering Team	• Develop SRD, UDD
Hardware Acquisition team	• Acquire/develop hardware and equipment
Software Project Manager	• Oversee software development activities

Role	Responsibilities
	• Develop and maintain SDP
Software Development team	• Prepare SRD, SDD, SIDD, SDFs • Code, integrate, and unit test the software • Support testing and delivery
Quality Assurance team	• Prepare and execute SQAP
Configuration Management team	• Prepare and execute SCMP
Testing team	• Prepare Test Plans/Procedures and Test Reports • Conduct Integration testing • Support Installation and Qualification testing
SEPO	• Support process development and improvement

7.2 PROJECT RESOURCES

7.2.1 Personnel Resources

The staff loading of personnel by categories and by quarters is shown in Table 7-2. A Work Breakdown Structure (WBS) with cost and effort estimates is contained in Appendix B.

Table 7-2. Personnel Staffing Plan

	1999	2000				2001				2002	
Average Staff Size	4Q	1Q	2Q	3Q	4Q	1Q	2Q	3Q	4Q	1Q	2Q
Program Management	0.5	0.8	0.8	1.0	1.0	1.0	1.0	1.0	1.0		
Hardware/System Engineers		2.0	2.0	2.0	1.0	1.0	1.0	1.0			
Software Project Manager	0.5	1.0	1.0	1.0	1.0	1.0	0.5				
Software Development Team		1.0	3.0	5.0	4.0	2.0	1.0	1.0	1.0		
QA Team			1.0	1.0	1.0	1.0	1.0	0.5	0.5		
CM Team		0.5	0.5	0.5	1.0	1.0	1.0	0.5	0.5		
Testing Team		0.5	1.0	2.0	2.0	2.0	1.0	0.5	0.5		
Totals	1.0	5.8	9.3	12.5	11.0	9.0	6.5	4.5	3.5	tbd	tbd

7.2.2 Personnel Requirements

The required qualifications of the personnel are:

Program Manager - A minimum of ten (10) years experience in the management of government contracts is required for this position. Familiarity with current versions of MIL-STD-480/482/483/490/1521 and IEEE/EIA 12207 is also required.

Hardware/Systems Engineers - Qualified individuals shall have a minimum of five years experience in real time data processing, higher order computer languages, familiarity with the NATO STANAG 5030 and VLF communications.

Test Engineer - A minimum of five years experience testing software systems at the unit, computer software component (CSC), CSCI and system level is required.

Software Development Team and Manager - Two years hands on experience with the Ada programming language is required. One year of experience with machine

level implementations experience working with IEEE/EIA 12207, MIL-STD-498 or equivalent is also required.

Quality Assurance - The product assurance function shall be staffed as necessary to support the project. A minimum of two years experience in quality assurance on government projects is required.

Configuration Management - Two years minimum experience is required with configuration data management for government projects. Familiarity with DOD-STD-2168 is also required.

All project personnel require SECRET clearances. Additionally, NATO access shall be required for the programmers, test personnel and the system engineers.

7.2.3 Facilities

SSC SD is considered the prime contractor to SPAWAR for NISBS Formatter. Therefore, the Formatter will be developed and tested using Government Furnished Equipment (GFE) and facilities. SSC SD D833 may provide office space for contractors as well as the test environment. Coding and initial module integration will also be performed using GFE. Module integration and testing will use SSC SD VLF/LF TEMPEST secured screen room.

7.2.4 Government Furnished Equipment, Software and Services

SSC SD will provide the INTEL Pentium computers and the software development tools required for the coding, testing, and integration phases. These tools include, but are not limited to, the editor, compiler, linker, and library resources required for the development environment. The NISBS software will migrate to the Pentium II environment. The INTEL Pentiums will be upgraded to Pentium II by C3, Inc. During the integration and test phase SSC SD will provide the following GFE.

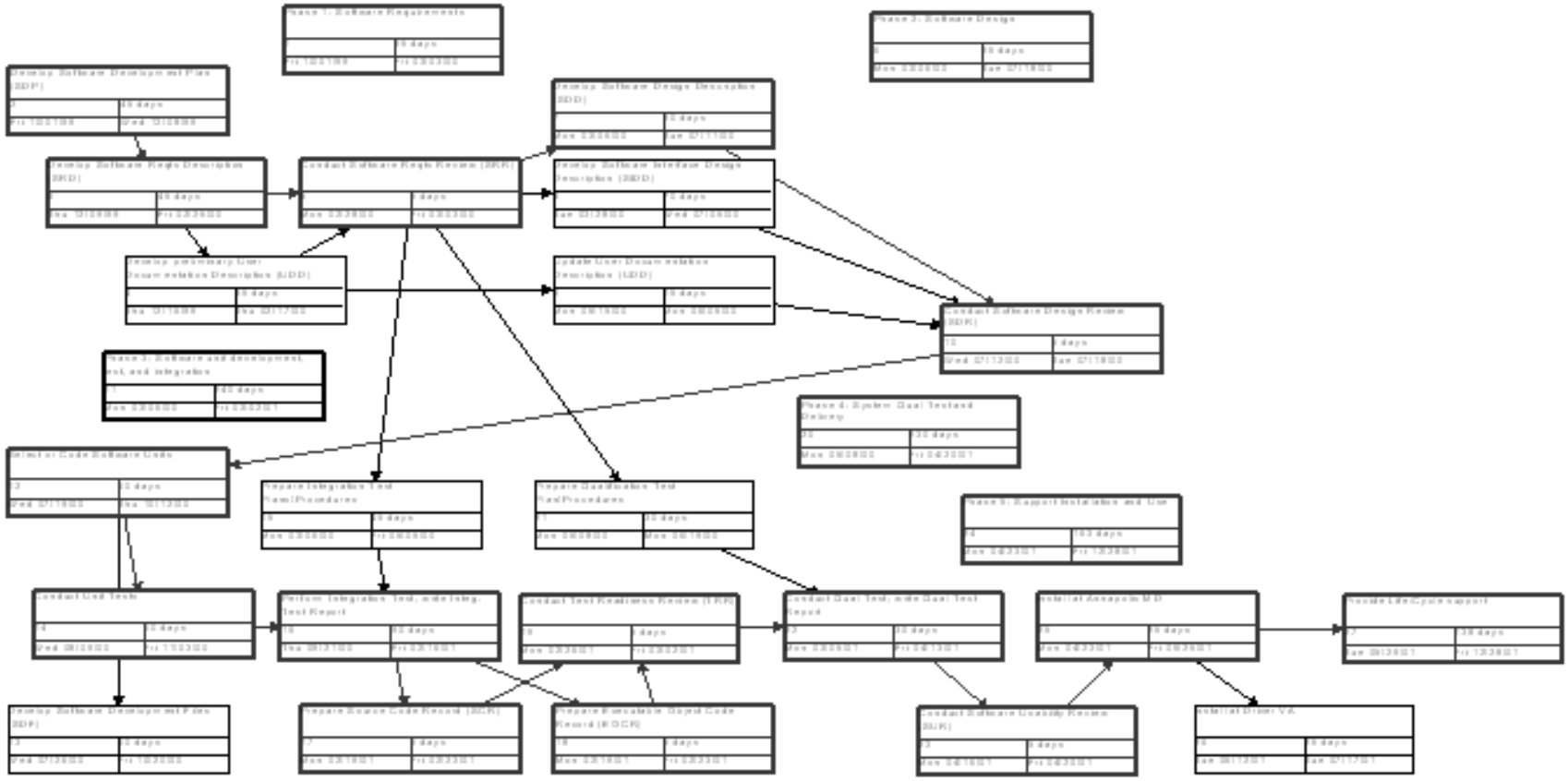
a. Vallor cryptographic equipment KWT/KWR-46
b. Jason cryptographic equipment KWT/KWR-37
c. SSC SD-developed NATO Transmit Simulator
d. VERDIN transmit terminal AN/URC-62 and receive terminal AN/WRR-7
e. Frequency time Standard 0-1695/U
f. KG-84C cryptographic equipment

Several commercial software packages have been made available by SSC SD for the NISBS Formatter design and development effort. These packages include:

a. A Meridian Ada compiler, used during development and testing of NISBS Formatter.
b. A Disk Operating System (DOS) for each development system.
c. A Borland C compiler and Assembler for I/O board software development.
d. The Meridian Pretty Printer for printing source code and program design language (PDL) output.

The Navy will provide timely access to appropriate Government personnel and facilities in support of the development.

Appendix A. NISBS Project Activity Network



APPENDIX B. NISBS WORK BREAKDOWN STRUCTURE

WBS	Description	Cost ($K)	Effort (mo)	Schedule	Risk *	POC
1.0	**NISBS Formatter Project**	$1,984.5	189.0	27 mo	(all)	PM
1.1	**Project Management**	$252.0	24.0	27 mo		PM
1.1.1	Project Planning	$42.0	4.0	27 mo	Reqts	PM
1.1.2	Project Tracking and Control	$52.5	5.0	27 mo		PM
1.1.2.1	Metrics Collection and Analysis	$52.5	5.0	27 mo		SQA
1.1.3	Cost Estimation	$31.5	3.0	6 mo	Reqts	SPM
1.1.4	Contract Acquisition & Monitoring	$31.5	3.0	20 mo	Support	PM
1.1.5	Travel	$10.5	1.0	4 mo		PM
1.1.6	Training (internal)	$31.5	3.0	12 mo		PM
1.1.7	Risk Management	$21.0	2.0	27 mo		PM
1.1.8	Reviews	$31.5	3.0	27 mo		PM
1.1.8.1	Technical	$15.8	1.5	24 mo		SPM
1.1.8.2	Management	$15.8	1.5	27 mo		PM
1.2	**Systems Engineering**	$126.0	12.0	27 mo		HM
1.2.1	Requirements Analysis	$94.5	9.0	6 mo	Reqts	HM
1.2.2	Computer Security	$31.5	3.0	27 mo		HM
1.3	**Software Engineering**	$756.0	72.0	20 mo		SPM
1.3.1	Software Requirements Analysis	$84.0	8.0	6 mo	Reqts	SPM
1.3.2	Software Design	$189.0	18.0	9 mo	Tech, Reqts	SPM
1.3.3	Coding and Unit Testing	$157.5	15.0	6 mo	Tech, Test	SPM
1.3.4	CSCI Testing	$273.0	26.0	12 mo	Test	SPM
1.3.5	Software Tools and Compilers	$52.5	5.0	12 mo	Tech,Support	SPM
1.4	**Hardware Engineering**	$105.0	10.0	6 mo		HM
1.4.1	Design	$63.0	6.0	4 mo		HM
1.4.2	Procurement	$42.0	4.0	3 mo	Support	HM
1.5	**Facilities**	$52.5	5.0	27 mo		HM
1.5.1	Physical Security	$21.0	2.0	27 mo		HM
1.5.2	Facilities Management	$31.5	3.0	27 mo	Support	HM
1.6	**Integrated Logistics Support**	$31.5	3.0	5 mo		HM
1.7	**System Level Testing**	$299.3	28.5	12 mo		Test
1.7.1	Integration Testing	$299.3	28.5	12 mo	Test	Test
1.8	**Software Configuration Mgmt**	$173.3	16.5	24 mo		SCM
1.8.1	Configuration Identification	$47.3	4.5	6 mo		SCM
1.8.2	Configuration Control	$105.0	10.0	18 mo		SCM
1.8.3	SCCB	$21.0	2.0	18 mo		SCM
1.9	**SQA**	$189.0	18.0	20 mo		SQA
1.9.1	Audits	$157.5	15.0	20 mo		SQA
1.9.2	Requirements Traceability	$31.5	3.0	4 mo	Reqts	SQA

* Risk code: Reqts = Requirements stability
Tech = New Technology
Support = Reliability of support organizations
Test = Overall System testability

APPENDIX C. NISBS REQUIREMENTS DATABASE SCHEME

The following attributes will be collected on requirements:

Attribute Name	Type	Meaning/Values
Requirement key	Alphanumeric	An assigned number that uniquely identifies each requirement in the database.
Requirement Text	Text	Text of requirement.
Requirement Status	Numeric	Status of the requirement as follows: (1) Build Assigned (3) In Draft (Change) (5) Deleted (2) CCB Baselined (4) In Draft (New) (6) Consolidated
Comment	Text	Amplifying information such an SCP Number spawning the requirement, the requirement number that an In Draft Change is addressing, etc
Origin Date	Alphanumeric	Date requirement first entered into system (e.g. 9/10/99).
Last Modified	Alphanumeric	Date of last change to a particular requirement.
Build	Alphanumeric	Identification of software build that the requirement is targeted for, or was implemented.
Priority	Numeric	Numeric value from 1-5, based on MIL STD 498 P/CR priority values
Source	Alphanumeric	Document and paragraph identification of higher level document spawning this requirement (e.g. SSS para 3.1)
Implementation	Alphanumeric	Lower level document and paragraph identification implementing the requirement (i.e. SSD), or the ID of an implementing module, or name of a COTS product resolving the requirement.
V&V Method	Numeric	Type of method used to verify and validate implementation of the requirement: (1) Integration Test Case (3) Unit Test (2) IV&V (4) Inherent via Qualification Test
Test Case ID	Alphanumeric	If V&V Method = 1 then enter Test Case ID
Test Status	Numeric	The state of the verification and validation of implementation of the requirement: (1) Test in Development (3) Test Failed (2) Test available/not executed (4) Test Passed

NISBS stated “requirements” may actually be a set of requirements within a single statement of controlling document. If left as stated, these multiple requirements could create misunderstandings as to the customer’s intent. Thus, it becomes important to restate (or “normalize”) a requirement so as to eliminate potential misunderstandings. Requirements normalization, then, is the process whereby ambiguous or unclear requirements are restated as discrete entities into a database. This process is key because it also:

- breaks general requirements statements into discrete requirements
- facilitates checks for testability
- facilitates checks for clarity
- creates a focal point for requirement tracking and management
- provides a basis for traceability

APPENDIX D: CROSS REFERENCE FROM CMM TO SDP

KPA	Common Feature	SDP Para.
RM	CO1: Follow RM policy	4.2, 2.1.2
	Ab1: Set responsibility	7.1
	Ab2: Document reqts.	4.2.1
	Ab3: Provide resources	7.2
	Ab4: Train practitioners	5.14
	Ac1: Review reqts	4.2.1
	Ac2: Use reqts for plans	3.1a
	Ac3: Process changes	
	M1: Use measurements	5.13
	V1: Review w/sr. mgmt.	5.1
	V2: Review w/Proj. Mgr.	5.9
	V3: SQA review	5.8
SPP	CO1: Designate Mgr.	7.1
	CO2: Follow SPP policy	4.2, 2.1.2
	Ab1: Use SOW	1.1
	Ab2: Set responsibility	7.1
	Ab3: Provide resources	7.2
	Ab4: Train practitioners	5.14
	Ac1: Support proposal	5.1
	Ac2: Plan early	5.1
	Ac3: Plan often	5.1
	Ac4: Review commitments	5.1
	Ac5: Identify life cycle	4.1
	Ac6: Follow procedure	1.3
	Ac7: Document plans	1.1
	Ac8: Identify products	5.1
	Ac9: Estimate size	5.1
	Ac10: Estimate effort, costs	5.1
	Ac11: Estimate resources	7.2
	Ac12: Derive schedule	6.1
	Ac13: Identify risks	5.10
	Ac14: Plan facilities	7.2
	Ac15: Record plan data	
	M1: Use measurements	5.13
	V1: Review w/sr. mgmt.	5.1
	V2: Review w/Proj. Mgr	5.9
	V3: SQA review	5.8
PTO	CO1: Designate manager	7.1
	CO2: Follow PTO policy	4.2, 2.1.2
	Ab1: Document SDP	1.1
	Ab2: Assign manager	7.1
	Ab3: Provide resources	7.2
	Ab4: Train practitioners	5.14
	Ac1: Use SDP for tracking	5.13
	Ac2: Keep SDP current	
	Ac3: Review commitments	
	Ac4: Communicate changes	
	Ac5: Track software size	5.13
	Ac6: Track effort, costs	5.13
	Ac7: Track resources	5.13
	Ac8: Track schedule	5.13
	Ac9: Track activities	5.13
	Ac10: Track risks	5.13
	Ac11: Record planning data	
	Ac12: Review progress	5.9.3
	Ac13: Hold formal reviews	5.9.2
	M1: Use measurements	
	V1: Review w/sr. mgmt.	5.1
	V2: Review w/Proj. Mgr.	5.9
	V3: SQA review	5.8

KPA	Common Feature	SDP Para.
SSM	CO1: Follow SSM policy	4.2, 2.1.2
	CO2: Appoint sub. Manager	
	Ab1: Provide resources	7.2
	Ab2: Train practitioners	
	Ab3: Orient managers	
	Ac1: Plan subcontract work	
	Ab2: Select able subs	
	Ac3: Flow down reqts.	
	Ac4: Review sub's plans	
	Ac5: Track with sub's plan	
	Ac6: Resolve changes	
	Ac7: Review mgmt. status	
	Ac8: Review technical status	
	Ac9: Review sub's work	
	Ac10: Review sub's SQA	
	Ac11: Review sub's SCM	
	Ac12: Test sub's products	
	Ac13: Review performance	
	M1: Use measurements	
	V1: Review w/sr. mgmt.	5.1
	V2: Review w/Proj. Mgr.	
	V3: SQA review	
SQA	CO1: Follow SQA policy	4.2, 2.1.2
	Ab1: Appoint SQA group	7.1
	Ab2: Provide resources	7.2
	Ab3: Train practitioners	5.14
	Ab4: Orient project on SQA	5.14
	Ac1: Prepare SQA plan	5.8
	Ac2: Follow SQA plan	5.8
	Ac3: SQA part of planning	
	Ac4: SQA review activities	5.8
	Ac5: Audit work products	5.8
	Ac6: Report SQA results	5.8
	Ac7: Document deviations	
	Ac8: Review with customer	
	M1: Use measurements	
	V1: Review w/sr. mgmt.	5.1
	V2: Review w/Proj. Mgr	5.9
	V3: Review SQA activities	
SCM	CO1: Follow SCM policy	4.2, 2.1.2
	Ab1: Establish SCCB	5.7
	Ab2: Appoint SCM group	7.1
	Ab3: Provide resources	7.2
	Ab4: Train SCM group	5.14
	Ab5: Train others in SCM	5.14
	Ac1: Prepare SCM plan	5.7
	Ac2: Follow SCM plan	5.7
	Ac3: Use CM library	5.7
	Ac4: Identify work products	5.7
	Ac5: Use change requests	5.7
	Ac6: Baseline changes	
	Ac7: Control releases	
	Ac8: Record config. status	
	Ac9: Report SCM activities	
	Ac10: Conduct audits	
	M1: Use measurements	
	V1: Review w/sr. mgmt.	5.1
	V2: Review w/Proj. Mgr	5.9
	V3: Audit baselines	
	V4: SQA review	5.8