SOFTWARE CONFIGURATION MANAGEMENT PLAN FOR THE <SYSTEM ID>

CI #: <SYSTEM ID>-SCMP-#

PREPARED FOR:
DEFENSE LOGISTICS INFORMATION SERVICE
(DLIS)

PREPARED BY:
Deborah K. Clark
DLIS CM Administrator

Approved By:

________________________ ______________
<CONTRACTOR> Project Manager Date

________________________ ______________
DLIS CCB Chairperson Date

________________________ ______________
<SYSTEM ID> Program Manager Date

________________________ ______________
DLIS CM Program Manager Date

1. INTRODUCTION

This document is the Software Configuration Management Plan (SCMP) for software development of Military Engineering Drawing Asset Locator System <SYSTEM ID> on the Web. This SCMP is to be implemented during the <SYSTEM ID> development effort by Northrop/Grumman Data Systems (<CONTRACTOR>. This plan is prepared in compliance the Capability Maturity Model (CMM). This SCMP presents a methodology by which Software Configuration Management (SCM) is maintained throughout the software development life cycle. Benefits provided by the implementation of the SCMP are:

- Ensures fulfillment of the <SYSTEM ID> software requirements during the software design, development, integration, testing, and deployment phases of the life cycle.

- Enables changes to the software requirements to be made under controlled conditions.

- Provides a historical reference for the life cycle of the software product.

The Defense Logistics Information Service (DLIS) shall maintain and update this plan as required throughout the development effort. All changes to this plan must be approved by the DLIS CCB in coordination with the DLIS <SYSTEM ID> Program Manager.

1.1 Purpose
This SCMP describes how DLIS and <CONTRACTOR> shall develop, implement, and maintain SCM for <SYSTEM ID> development efforts. This SCMP lays a framework within which the four primary CM functions (configuration identification, control, status accounting, and audits and reviews) shall be performed. Additionally, this SCMP forms the basis for an automated CM system by which baselines are stored and changes to those baselines are tracked as they are implemented. It also addresses the involvement of ADP Security in the development of any software application. The requirements for meeting DLA’s DII/COE/SOE are outlined in Section 4 of this document. The configuration management metrics is outlined in this document along with known Year 2000 requirements.

1.2 Scope
The <SYSTEM ID> SCMP sets forth and/or incorporates by reference, the CM policies and procedures for software development of <SYSTEM ID>. It describes how the CM elements of configuration identification, change control, status accounting, and audits and reviews, as shown in Table 1.2-1, shall be applied to establish and maintain the <SYSTEM ID> software configuration throughout the development effort.

Table 1.2-1 Configuration Management Functions

Configuration Identification	Configuration Identification is the identification and definition of documents and related CSCIs and all affected changes. Documents include all those necessary to provide a full technical description of the characteristics of the CSCIs that require control. Configuration identification is: a. Founded on baseline management b. Maintained through control of software specifications and developed software for each identified baseline c. Based on software requirements through specifications and plans
Configuration Control	Configuration Control is the system coordination, approval or disapproval, and implementation of all approved changes in configuration of a CSCI after establishment of its baseline which includes: a. CCB/SCCB administration in relation to software changes b. Classification of changes (Class I or II) c. AWR/SCR/ECP control and implementation d. ACSN preparation, control, and distribution
Configuration Status Accounting	Configuration Status Accounting is used as a discipline to establish the listing of all CSCIs, the status of all proposed changes, and the implementation status of approved changes. CSA is: a. Maintained by a system which defines each software component by identification number and records all proposed, scheduled and implemented changes b. Provided with computer program definitions and identification by SVDs which accompany CSCI deliveries and document all version and release numbers

Audits and Reviews	Audits and Reviews are inspection and review of records and procedures. SCM shall: a. Conduct internal audits (as required) to ensure requirements are in accordance with approved SCM plans b. Provide support for FCA and PCA

Legend

ACSN	Advanced Change Study Notice	FCA	Functional Configuration Audit
AWR	ADP/T Work Request	PCA	Physical Configuration Audit
CSA	Configuration Status Accounting	SCCB	Software Configuration Control Board
CSCI	Computer Software Configuration Item	SCM	Software Configuration Management
ECP	Engineering Change Proposal	SCR	Systems Change Request
		SVD	Software Version Description

1.3 Acronyms and Glossary

Acronyms and Glossary referenced in this document are detailed in Appendix A.

1.4 References

(1) J-Std-016, Standard for Information Technology software Life Cycle Process Software Development Acquirer-Supplier Agreement, Sep 1995.

(2) IEEE Std 828-1990, Standard for Software Configuration Management Plans, Sep 28, 1990.

(3) IEEE Std 1042-1987, Guide to Software Configuration Management, Sep 10, 1987.

(4) DLISI 8000.1, DLIS Application Development Standards (Draft).

(5) MIL-STD-973, Configuration Management, Apr 17, 1992.

(6) DLAR 4730.3, Defense Logistics Agency Automated Data Processing/Telecommunication (ADP/T) Configuration Management Program, Feb 20, 1991.

(7) IEEE Std 610.12-1990, Glossary of Software Engineering Terminology.

(8) CMU/SEI-93-TR-25, Key Practices of the Capability Maturity Model, Version 1.1, Feb 1993.

(9) DoDD 5000.1, Defense Acquisition, March 15, 1996.

(10) DoD Regulation 5000.2-R, Mandatory Procedures for Major Defense Acquisition Programs and Major Automated Information Systems, March 23, 1998.

(11) DLAR 5200.17, Security Requirements for Automated Information and Telecommunications Systems, Jun 1993.

(12) DLIS Year 2000 Management Plan, May 19, 1998.

(13) DLA Architecture Guidelines, November 1, 1998
(14) Joint Technical Architecture, V 2, May 1998.

2. MANAGEMENT

2.1 Organization

This section outlines the <CONTRACTOR> organizational relationships, roles and responsibilities for SCM during the <SYSTEM ID> software development effort. Figure 2.1-1 represents <CONTRACTOR>’s Method of Operation. The software project team for <SYSTEM ID> shall be composed of the appropriate staff across the various offices. The SCM functional staff from <CONTRACTOR> shall support the DLIS CCB and shall be responsible for directing developmental SCM for the <SYSTEM ID> development effort.

<Provide contractor method of operation>

Figure 2.1-1 <CONTRACTOR> Method of Operation

2.2 ADP Security

The DLIS configuration management processes and procedures must be appropriate for controlling changes to a system's hardware and software structure for the purpose of ensuring that such changes will not lead to decreased data security. DLIS ADP Security shall provide guidance, insight, and testing of application software to ensure any changes have not introduced security hazards or jeopardized the integrity of the software and data.

2.3 SCM Responsibilities

SCM shall be done in accordance with DLIS Standards and procedures. The <CONTRACTOR> <SYSTEM ID> System Engineer is responsible for the implementation of SCM procedures during the development phase of <SYSTEM ID>. Additional responsibilities of the <CONTRACTOR> System Engineer include:

(1) Configuration control
(2) Status accounting
(3) Configuration identification
(4) Assistance in the implementation and maintenance of the Software Configuration Management Plan
(5) Establishment and maintenance of development baselines
(6) Assisting in both formal and internal audits
(7) Participation in software development reviews

2.3.1 Configuration Identification

Configuration identification shall be applied to all developed software including both code and associated documentation. Associated documentation (i.e., specifications, design documents, program/procedure listings, etc.) along with the actual produced software makes up the configuration item. The <CONTRACTOR> System Engineer implements the agreed-upon identification schema in concert with the DLIS SQA, the <CONTRACTOR> <SYSTEM ID> Project Manager and the DLIS <SYSTEM ID> Program Manager.

2.3.2 Configuration Control

All documentation and software entities are released to and maintained by Software Configuration Management in a controlled library. Changes to a controlled baseline (i.e. Integration, Functional, Allocated, or Product) must be initiated through the DLIS <SYSTEM ID> Program Manager in the form of a System Change Request (SCR) or Task Order (TO). These changes must be reviewed and approved by the DLIS CCB. When a change has been approved and incorporated into the project plan, the <CONTRACTOR> System Engineer may check out with a lock the dev version of the CSCI element(s) from the archive using an automated CM tool. The CM Administrator has responsibility for releasing the CSCI element(s) from the controlled libraries to <CONTRACTOR> engineering development groups for the purpose of applying the change. After the engineering development group makes and unit tests the change, the code is checked in to the automated CM tool where it is held until the <SYSTEM ID> CCB approves the implementation of the change. At that time the candidate for update to the Product Baseline is made available to DLIS CM.

2.3.3 Status Accounting

The <CONTRACTOR> shall maintain a Configuration Management System (CMS) to provide status accounting and reporting for all configuration items. The DLIS CCB may authorize use of the DLIS CMS for <CONTRACTOR> use upon <CONTRACTOR> request. The configuration items shall include all work products produced, licensed, or interfaced for <SYSTEM ID>. The <CONTRACTOR> Project Manager shall use this information to provide CSA reports.

2.3.4 Audits

The <CONTRACTOR> Project Manager is responsible for assisting DLIS on all formal audits. In addition, <CONTRACTOR> shall conduct periodic internal audits of the <SYSTEM ID> effort in accordance with established audit procedures.

2.3.5 Configuration Control Board (CCB)

DLIS has the overall responsibility for CM control for the <SYSTEM ID> project. <CONTRACTOR> is responsible for SCM control during the development phase of the project.

2.4 Interface Control

The initial changes to interface systems are controlled through SCR implementation at each interfacing system. The interfacing systems are ***<LIST SYSTEMS>***. DLIS CCB is responsible to ensure that changes to systems that impact these interfaces are coordinated with the DLIS <SYSTEM ID> Program Manager.

2.5 SCMP Implementation

The SCMP shall be implemented when it has been approved by DLIS and placed under CM. Any unresolved issues found once the SCMP is approved must be resolved prior to any baselines being established. When change packages are developed, the DLIS CCB schedules test and change implementation. DLIS CM Administrator shall be responsible for the movement of the software from the development environment to the quality assurance environment. After successful independent, third party assurance testing, DLIS shall be responsible for the movement of the software from the quality assurance environment to the production environment.

2.5.1 Configuration Control Board

The <CONTRACTOR> controls the development phase for the <SYSTEM ID> project.

2.5.2 Configuration Baselines

Development baselines shall be established and controlled by <CONTRACTOR>. Formal baselines such as the Functional, Allocated, and Product Baseline must be approved by the DLIS CCB. The specific baselines that shall be used are described in section 3.1.3.

2.5.3 Schedules and Procedures for SCM Reviews and Audits

Reviews and audits shall be held as defined by the <SYSTEM ID> Software Development Plan (SDP) and as described in section 3.4 of this document.

2.5.4 Configuration Management of Software Development

Non-deliverable support software (e.g., compilers, operating systems, CASE tools, etc.) used in the development and test for <SYSTEM ID> software shall be managed and controlled in accordance with DoD 5000.1-R.

2.5 Applicable Policies, Directives, and Procedures

The complete SCM policies, directives, and procedures that apply to <SYSTEM ID> are in accordance with DLIS standards and procedures.

3. SCM ACTIVITIES

3.1 Configuration Identification

This SCMP documents the configuration identification numbering schema for the Configuration Items (CI) upon their selection and allocation as required for the <SYSTEM ID> software development project. The specific items that shall be placed under SCM are listed in Appendix B.

3.1.1 Documentation

All support documentation generated for <SYSTEM ID> shall comply with J-STD-016. This documentation shall be identified by use of an agree-upon schema. This schema shall be documented in DLISI 8000.1.

The System Change Request (SCR)/Task Order (TO) shall be assigned and tracked by the DLIS SCR Administrator. Problem Reports (PRs) shall be assigned and tracked by the DLIS PR Administrator.

In addition, a unique number shall be assigned by the automated CM tool used for the Configuration Status Accounting (CSA) and SCM reporting for <SYSTEM ID>.

3.1.2 Software Parts.

Identification of CSCIs in <SYSTEM ID> is as follows:

<describe all components that make up the system such as development tools, compilers, etc.>

The identification schema shall follow the standards and procedures described in DLISI 8000.1.

3.1.3 Configuration Identification of <SYSTEM ID> Baselines

Baselines shall be established and controlled in accordance with DoD 5000.1-R. Formal baselines such as the Functional, Allocated, and Product Baseline shall be approved by DLIS CM. An internal “baseline” known as the Developmental Configuration shall be established to control the development of the software and shall become the basis for the Product Baseline. The Developmental Configuration may be subdivided into the following environments as required: Development, Integration Test, and Quality Assurance Test.

The Developmental Configuration shall be maintained and controlled by <CONTRACTOR>. All <CONTRACTOR> software maintenance or development projects shall utilize at a minimum the following baselines: Functional, Developmental Configuration, and Product Baseline. The specific baselines and related items that shall be established are identified in Appendix B.

3.2 Configuration Control

Software configuration management and change control is applied to all documents and code including the non-deliverable <SYSTEM ID> operating system and support software. Control is accomplished through the implementation of configuration identification, the <SYSTEM ID> CCB, change control, and status accounting functions in accordance with DoD 5000.1-R.

3.2.1 Function of the Configuration Control Boards

3.2.1.1 DLIS Configuration Control Board (CCB)

The DLIS CCB in concert with the DLIS <SYSTEM ID> Program Manager maintains overall CM control for the <SYSTEM ID> project. <CONTRACTOR> shall be responsible for SCM control during the <SYSTEM ID> development phase. The DLIS CCB approves, disapproves, or tables all changes to the formal baselines. The mechanism for submitting changes to the software or documentation is the System Change Request (SCR), Task Order (TO) or Problem Report (PR).

The <SYSTEM ID> CCB shall review proposed changes for assuring compliance with approved specifications and designs and determine the impact on existing software. The CCB members provide their recommendations to the chairperson in accordance with the <SYSTEM ID> CCB charter. If the proposed change does not affect multiple products, the <CONTRACTOR> shall recommend to approve, disapprove, or table all changes that affect the Developmental Configuration and forward its recommendation to the DLIS <SYSTEM ID> Program Manager for submission to the DLIS CCB for changes that affect formal baselines. Table 3.2.1.3-1 reflects (but does not limit) the primary and consulting members of the <SYSTEM ID> CCB.

Primary Members	Consulting Members
Chairperson	Security Representative
Functional Requirements Representative	Training Representative
System Administration Representative	Customer Representative
Database Administration Representative	Contractor(s) Representative
Quality Assurance (QA) Representative	
CM Administrator	

Table 3.2.1.3-1, <SYSTEM ID> SCCB Membership

3.2.2 The System/Software Change Request (SCR)/Task Order (TO).

SCRs/TOs originate with the <SYSTEM ID> Program Manager in the DLIS <SYSTEM ID> Program Office. The SCR is forwarded to the <CONTRACTOR> <SYSTEM ID> Project Manager who returns System Change Workload and Cost Estimate spreads of the hours/costs to perform the work to the DLIS <SYSTEM ID> Program Manager. The SCR is submitted to the DLIS CCB for review, approval, and scheduling. Upon completion of the software change, appropriate unit and system tests are performed by <CONTRACTOR> to ensure that change requirements of the SCR have been met and that the change has not in any way adversely affected proper system operation. At this point the new version of <SYSTEM ID> application which incorporates the change is migrated to the DLIS staging environment. DLIS shall install the change for Acceptance/QA testing of the new release.

3.2.3 Software Change Authorization

When an SCR/TO is received from the DLIS <SYSTEM ID> Program Manager and receives approval from the DLIS CCB, the <CONTRACTOR> System Engineer copies the current <SYSTEM ID> Production software to the development environment so that the changes may be incorporated. The CM tool provides an automated method to maintain multiple versions that represent the same system at different stages of development, testing, and production.

When an object is changed in one version, Version Control can transfer the changed object to other copies of the software. For example, an object changed in the <SYSTEM ID> development version shall be transferred to the QA version and upon approval by the DLIS CCB to the production version. A migration must be completely successful or it is rolled back in its entirety. This ensures the version is never left in an invalid state.

Version Control Software shall capture both source and executable items. DLIS CM and DLIS SQA shall audit, verify, and validate CSCIs prior to movement into the quality assurance environment. Detailed reporting of all changes (i.e., DLIS Software Update Form) that are included in the migration of software from the development environment to the quality assurance environment shall be documented and provided to DLIS CM.

3.2.4 Change Control Automated SCM Tools

The file structures, datasets, directories, and/or folders of the <SYSTEM ID> project are used to control all files containing the specifications, documentation, test plans, test procedures, and source and executable code. The non-developmental support software shall be placed under software configuration management by the <CONTRACTOR>. The specific file structures and tools used by the <SYSTEM ID> project are identified in Appendix B.

3.3 Configuration Status Accounting

Records shall be prepared and maintained of the configuration status of all entities that have been placed under project-level or higher configuration control. These records shall be maintained for the life of the project. They shall include, as applicable, the current version/revision/release of each entity, a record of changes to the entity since being placed under project-level or higher configuration control, and the status of problem/change reports affecting the entity.

3.4 Audits and Reviews

The <CONTRACTOR> System Administrator shall assist DLIS CM, DLIS SQA, and the <CONTRACTOR> Project Manager in performing the formal Functional Configuration Audit (FCA) and the Physical Configuration Audit (PCA). A schedule of planned audits and reviews is provided in Appendix B.

3.4.1 Functional Configuration Audit

The FCA is performed on the software configuration items when the Functional Test has been completed. The audit is performed on the formal test plans, descriptions, and procedures and compared against the official test data. The results are checked for completeness and accuracy and verified against the Test Report. Deviations are documented in accordance with MIL-STD-973, along with date(s)of resolutions.

3.4.2 Physical Configuration Audit

The Physical Configuration Audit for software is intended to verify that the software product conforms “as-built” to its technical documentation including the operation and user manuals. Although for software development, the PCA could be conducted in conjunction with the FCA; typically it is completed in concert with the Environment Test (ET) or Initial Operating Capability (IOC) tests. The <CONTRACTOR> System Engineer assembles and makes available to the PCA team at the time of the audit all data describing the deliverable Configuration Items (CIs) as well as the ET/IOC Test Reports. This includes a current set of listings and the final draft of manuals. The results are checked for completeness and accuracy and verified against the Test Report. Deviations are documented in accordance with MIL-STD-973. Completion dates for all deviations are clearly established and documented. Upon completion and written acceptance of the PCA by the DLIS CCB, the Product Baseline shall be established.

3.4.3 Internal Audits

In addition, <CONTRACTOR> shall conduct internal CM audits of the <SYSTEM ID> project during the development phase in accordance with <CONTRACTOR> internal procedures.

3.4.4 Reviews

The <CONTRACTOR> System Engineer and <CONTRACTOR> Project Manager participates in all formal reviews with DLIS as specified by the <SYSTEM ID> Software Development Plan.

4. DEFENSE INTEGRATED INFRASTRUCTURE (DII)/COMMON OPERATING ENVIRONMENT (COE)/STANDARD OPERATING ENVIRONMENT (SOE)

All software, hardware, telecommunications, operating systems, and the like that is created, purchased, altered, or subcontracted for DLIS shall meet the interoperability guidelines as described in the Defense Integrated Infrastructure (DII)/Common Operating Environment (COE) or the Standard Operating Environment (SOE), depending on the target platform, in accordance with the DLA Architecture Guidelines.

5. CONFIGURATION MANAGEMENT METRICS

Metrics shall be captured on each application release for each production platform. Metrics shall consist of the release identification, the number of projects included in the release, and the number of modules impacted by the release. If a module(s) fails after implementation, the module(s) shall be identified and a reason for failure annotated in the metric. Each month, the metrics shall be rolled up to indicate the percentage of successful releases implemented for each production platform.

6. YEAR 2000 REQUIREMENTS

The contractor shall certify all AISs developed or purchased for DLIS management as Year 2000 compliant.

This plan shall apply to all AISs supported by DLIS. Support includes hardware, firmware, Commercial Off-The-Shelf (COTS) and Non-Developmental Items (CANDI), Government Off-The-Shelf (GOTS), developed software, and data. Software includes CANDI/GOTS packages, operating systems, third and fourth generation language compilers and interpreters, functional applications, system utilities, translators, and database management systems. Data includes databases and other data storage structures and mechanisms, data and system interfaces, Electronic Data Interchange (EDI) transaction sets and implementation conventions, and other messages or forms of data exchange.

The DoD standard for system interfaces is the four-digit year. Military standards transactions are an exception to this and will continue to use two-digit year interfaces because of the 80-column limitation.

The Year 2000 Checklists shall be prepared for each AIS in accordance with the DLIS Year 2000 Management Plan. Each checklist shall be forwarded to the DLIS Year 2000 Project Manager for processing of certification requests.

6.1 Contractor-Developed Software.

The structure, storage, and communications method of transport of date data within the application shall determine compliance. All dates used within the software shall be structured and stored in the eight-digit format (YYYYMMDD). The program managers of the systems shall determine the communication vehicle of dates between the systems. Components that use the two-digit year for interfaces are required to fund bridges or translators to other systems using four-digit year interfaces. The use of a four-digit year interfaces is acceptable unless both partners agree to keep existing formats. In this case, when no changes are required, an MOU must be established between the two parties describing the interface agreement.

6.2 Commercial-Off-The-Shelf and Non-Developmental Items (CANDI)/ Government-Off-The-Shelf (GOTS).

The contractor shall obtain a compliance letter from the vendor of CANDI/GOTS software. The compliance letter shall address the license agreement compliance as well as the software compliance. The compliance letter shall accompany the Year 2000 Checklist for the software package.

7. TOOLS, TECHNIQUES, AND METHODOLOGIES

7.1 Version Control Tools [AUTOMATED CM TOOL]

7.1.1 Version Control Tools

The basic Configuration Management tool for the <SYSTEM ID> CSCIs is the Platinum CCC/Harvest configuration management tool. This tool is also used for version control of documentation. Harvest supports change management by:

(1) Providing the means to track proposed changes, e.g. System Change Requests (SCRs)/Task Orders (TOs)

(2) Providing an automated revision/version manager.

(3) Providing visibility and control over the entire development life cycle

7.1.2 Library Control

The DLIS CM Administrator shall establish controlled libraries to manage the changes to support CSCIs prior to migration to the production machine.

Development Environment. The development environment is used by the systems engineers as they develop the code. The individual engineers control the units and components. Code that has been unit tested and certified by the SQA group shall be promoted into the integration environment by DLIS CM.

Integration Environment. The integration environment is used by the DLIS CM Administrator to capture and build the modules that are designated for system and functional test. This environment shall contain the source code and executable load modules created as a result of a system build. Criteria for release from this environment is determined and enforced by the CM Administrator.

Production Environment. The production environment contains the master copies of all the CSCIs used in <SYSTEM ID>.

Software Repository. All <SYSTEM ID> software resides on the development machine until it meets the requirements of <CONTRACTOR> SQA and is ready for release. The software is migrated by DLIS to the quality assurance environment. After successful completion of the quality assurance testing, DLIS shall move the software to the production environment.

7.2 Software Change Authorization

When an SCR is received from the DLIS <SYSTEM ID> Program Manager, and receives approval from the <SYSTEM ID> CCB, the <CONTRACTOR> System Engineer copies the current <SYSTEM ID> Production version to the development environment so that the changes may be incorporated. Harvest provides an automated method to maintain multiple versions that represent the same system at different stages of development, testing and production.

When an object is changed in one environment, Version Control can transfer the changed object to other copies of the environment. For example, an object changed in the <SYSTEM ID> development environment shall be transferred to the quality assurance environment and upon approval by the DLIS CCB to the production environment. A migration must be completely successful or it is rolled back in its entirety. This ensures the environment is never left in an invalid state.

Version Control Software shall capture all controlled objects and both source and executable items. Detailed reporting of all changes (i.e., DLIS Software Update Form) that are included in a migration of software from the development environment to the quality assurance environment shall be documented and provided to DLIS CM.

8. SUPPLIER CONTROL

8.1 Sub-Contractor Software

Sub-Contractor-provided software to be used by the <SYSTEM ID> application must conform to good business practice SCM. Any Sub-Contractor wishing to do business with <CONTRACTOR> must provide a copy of their SCMP to the applicable project manager for evaluation. <CONTRACTOR> must find the Sub-Contractor SCMP to be adequate. If the Sub-Contractor’s SCMP is found inadequate, or if no Sub-Contractor SCM Plan is available, the Sub-Contractor can be disqualified from providing software for this project in accordance with DoD 5000.1-R.

8.2 Sub-Contractor Audit

Periodic audits shall be performed in accordance with DoD 5000.2-R.

9. RECORDS COLLECTION AND RETENTION

All formal documentation produced for and by this <SYSTEM ID> Project is retained and safeguarded for the life of the project. All documentation shall be provided in softcopy as described in DLISI 7900.4 and shall conform to J-STD-016. All documentation shall be placed under configuration management control in the DLIS Documentation Center.

10. PROJECT DEVELOPMENT BOARD (PDB)

All requirements for new hardware or software must be presented to the Project Development Board (PDB) for approval, coordination, and purchasing. DLSC Form 1840 must be filled out *in full* and sent to the Configuration Management mailbox for PDB scheduling. Taskings for new contractor support must also be documented on a DLSC Form 1840 for coordination through the appropriate Contract Officer Representative (COR).

APPENDIX A: ACRONYMS AND GLOSSARY

A.1 Acronyms

ADP	Automated Data Processing
AIS	Automated Information System
CANDI	Commercial and Non-Developmental Items
CASE	Computer Aided System Engineering
CCB	Configuration Control Board
CCBD	Configuration Control Board Directive
CI	Configuration Item
CM	Configuration Management
CMP	Configuration Management Plan
COR	Contract Officer Representative
COTS	Commercial Off-the-Shelf
CSA	Configuration Status Accounting
CSCI	Computer Software Configuration Item
DBL	Developmental Baseline
DLA	Defense Logistics Agency
DLIS	Defense Logistics Information Service
DoD	Department of Defense
FBL	Functional Baseline
FCA	Functional Configuration Audit
FD	Functional Description
OCD	Operational Concept Description
PBL	Product Baseline
PCA	Physical Configuration Audit
PDB	Project Development Board
PR	Problem Report
QA	Quality Assurance
SCM	Software Configuration Management
SCMP	Software Configuration Management Plan
SCR	System Change Request
SDF	Software Development File
SDL	Software Development Library
SE	System Engineering
SQA	Software Quality Assurance
SVD	Software Version Description
TO	Task Order

A.2 Glossary

Acquirer (customer): The organization that imposes this standard and the associated contract on a developer in order to procure software products for itself or another organization. [J-STD-016]

Allocated Baseline (ABL): In configuration management, the initial approved specifications governing the development of configuration items that are a part of a higher level configuration item. [IEEE Std 610.12]

Audit: An independent examination of a work product or set of work products to assess compliance with specifications, standards, contractual agreements, or other criteria. [IEEE Std 610.12]

Check in: A term used by the CM tool which allows users to provide a new version of the software to the tool. [Platinum CCC/Harvest User's Manual]

Check out with lock: A term used by the CM tool to indicate that only one copy of the software has been made available to the programmer for update outside the control of the tool. [Platinum CCC/Harvest User's Manual]

Computer Software Configuration Item (CSCI):

a. An aggregation of software that satisfies an end use function and is designated for separate configuration control by the acquirer. CSCIs are selected based on trade-off among software function, size, host or target computers, developer support concept, plans for reuse, criticality, interface considerations, need to be separately documented and controlled, and other factors. [J-STD-016]

b. An aggregation of software that is designed for configuration management and treated as a single entity in the configuration management process. [IEEE Std 610.12]

Configuration Control: An element of configuration management, consisting of the evaluation, coordination, approval or disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification. [IEEE Std 610.12]

Configuration Control Board (CCB): A group of people responsible for evaluating and approving or disapproving proposed changes to configuration items, and for ensuring implementation of approved changes. [IEEE Std 610.12]

Configuration Identification:

a. An element of configuration management, consisting of selecting the configuration items for a system and recording their functional and physical characteristics in technical documentation. [IEEE Std 610.12]

b. The current approved technical documentation for a configuration item as set forth in specifications, drawings, associated lists, and documents referenced therein. [IEEE Std 610.12]

Configuration Item (CI):

a. An aggregation of hardware, software, or both, that is designated for configuration management and treated as a single entity in the configuration management process. [IEEE Std 610.12]

b. An aggregation of hardware, software or both that satisfies an end use function and is designated for separate configuration control by the acquirer. [J-STD-016]

Configuration Management (CM): A discipline applying technical and administrative direction and surveillance to: identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record, and report change processing and implementation status, and verify compliance with specified requirements. [IEEE Std 610.12]

Configuration Management Plan (CMP): The Configuration Management Plan defines the implementation (including policies and methods) of configuration management of a particular program/project. [DoD-HDBK-287A]

Configuration Status Accounting (CSA): An element of configuration management, consisting of the recording and reporting of information needed to manage a configuration effectively. This information includes a listing of the approved configuration identification, the status of proposed changes to the configuration, and the implementation status of approved changes. [IEEE Std 610.12]

Database: A collection of related data stored in one or more computerized files in a manner that can be accessed by users or computer programs via a database management system. [J-STD-016]

Deliverable Software Product: A software product that is required by the contract to be delivered to the acquirer or other designated recipient. [J-STD-016]

Dev: A term related to the CM tool indicating an environment within the tool.
[Platinum CCC/Harvest User's Guide]

Functional Configuration Audit (FCA):
An audit conducted to verify that the development of a configuration item has been completed satisfactorily, that the item has achieved the performance and functional characteristics specified in the functional or allocated configuration identification, and that its operational and support documents are completed and satisfactory.
[IEEE Std 610.12]

Physical Configuration Audit (PCA): An audit conducted to verify that a configuration item, as-built, conforms to the technical documentation that defines it.
[IEEE Std 610.12]

Product Baseline (PBL): In configuration management, the initial approved technical documentation (including, for the software, the source code listing) defining a configuration item during the production, operation, maintenance, and logistics support of it's life cycle.
[IEEE Std 610.12]

Software Configuration Management (SCM) Plan: The document defining how configuration management shall be implemented (including policies and procedures) for a particular software acquisition or program. The plan may be a separate document or included within the configuration management section of the Software Development Plan.
[H. Ronald Berlack, "Software Configuration Management"]

Software Development (SD): A set of activities that results in software products. Software development may include the development, modification, reuse, reengineering, maintenance, or any other activities that result in software products. [J-STD-016]

Software Development File (SDF): A repository for material pertinent to the development of a particular body of software. Contents typically incude (either directly or by reference) considerations, rationale, and constraints related to requirements analysis, design, and implementation; developer-internal test information; and schedule and status information. [J-STD-016]

Software Development Library (SDL): A software library containing computer readable and human readable information relevant to a software development effort. [IEEE Std 610.12]

APPENDIX B MILITARY ENGINEERING DRAWING ASSET LOCATOR SYSTEM <SYSTEM ID> ON THE WEB

B.1 Configuration Items

B.1.1 Computer Software Configuration Items.

<list components of software>

B.1.2 Computer Hardware Configuration Items.

<list components of hardware used/required>

B.1.3 Documentation

System/Subsystem Specification
Software Development Plan
Software Configuration Management Plan
Project Plan
SQA Plan
Security Plan
Interface Requirements Specification
Database Design Description
Interface Design Description
Software Requirements Specification
Software Design Description
Software Test Plan
Software Test Description
Software Test Report
Software Installation Plan
Test Results

B.1.4 Non-developmental Items

<list components of software not written by programmer>

B.2 Configuration Control

B.2.1 Change Documentation

Various change documents (forms) shall be used by SCM to serve the different purposes for change. Table B.2.1-1 identifies the documents (forms) that may be used to propose and control changes during the <SYSTEM ID> software development effort. When change packages are developed, the DLIS CCB schedules test and change implementation. DLIS shall move the software from the development environment to the quality assurance environment. After successful independent third-party assurance testing, changes are ready for implementation. DLIS shall move the software from the quality assurance environment to the production environment.

Form Number	Form Acronym	Form Name
DD Form 2616	ACSN	Advance Change Study Notice
DD Form 1692/1693	ECP	Engineering Change Proposal
DLA Form 558	AWR	ADP/T Work Request
DD Form 2021	SCR	System Change Request
DLSC Form 1202	PTR	Problem Trouble Report

Table B.2.1-1, Change Documentation

<SYSTEM ID> UPDATE PROCEDURE

	Activity	Action
1.	DLIS <SYSTEM ID> PM	Change Document (SCR/TO) is generated and signed by the DLIS <SYSTEM ID> Program Manager.
2.	DLIS <SYSTEM ID> CCB	Receives the Change Document. Change is approved by board.
3.	DLIS <SYSTEM ID> COR	Change document is accepted by COR and forwarded to <CONTRACTOR> PM.
4.	<CONTRACTOR> System Engineer	If change to existing source software, checks out with lock, the affected source software module(s) from the CM tool. Otherwise, a new module is created following DLIS naming conventions.
5.	<CONTRACTOR> System Engineer	Makes required change(s) to <SYSTEM ID> source software, performs unit test(s), and system test on the changed system.
6.	<CONTRACTOR> System Engineer	Checks affected source software into CM tool, entering the change document number and the nature of the change.

7.	<CONTRACTOR> System Engineer	Completes documentation and move sheet(s). Forwards to <SYSTEM ID> PM.
8.	<SYSTEM ID> PM	Performs function acceptance testing. Completes the move sheet(s). Forwards to DLIS CM.
9.	DLIS CM	Prepares QA environment. Notifies DLIS QA testing of completion. Forwards move sheet(s) to DLIS QA testing.
10.	DLIS QA Testing	Performs quality assurance testing of changes. Determines need for further testing (integration, customer, etc.). If testing fails, process returns to step 4. If testing passes, completes move sheet and testing documentation. Forwards to DLIS CM.
11.	DLIS CM	Prepares software release for production update. Forwards tasking for release to Operations.
12.	DLIS Operations	Implements release as instructed in release documentation. Notifies DLIS CM of completion.
13.	DLIS CM	Sends notification of release implementation to designated recipients including COR.

B.2.2 Configuration Identification of <SYSTEM ID> Baselines

The identification of items included in these baselines are stored and maintained in an electronic media format once changes are approved in accordance with DLIS standards and procedures. A current baseline consists of the previous baseline plus all approved changes to that baseline. Table B.2.2-1 depicts the composition of the baselines including software and related documents.

Functional Baseline	Operation Concept Description (OCD) System/Subsystem Specification (SSS) Software Development Plan (SDP) SCM Plan (SCMP) Software Quality Assurance Plan (SQAP)
Allocated Baseline	Functional Baseline *plus* approved changes System/Subsystem Design Description (SSDD) Software Requirements Specification (SRS) Interface Requirements Specification (IRS)
Development Configuration	Allocated Baseline *plus* approved changes Software Design Description (SDD) Interface Design Description (IDD) Database Design Description (DBDD) Software Test Plan (STP) Software Test Description (STD) Software Test Report (STR) CSCI Source Code/Executable Code CSCI Database CSCI Non-developmental Items
Product Baseline	Developmental Configuration *plus* approved changes Software Product Specification (SPS) Software Installation Plan (SIP) Software Transition Plan (STrP) Software Version Description (SVD) Software User Manual (SUM) Software Input/Output Manual (SIOM) Software Center Operator Manual (SCOM) Computer Operation Manual (COM) Computer Programming Manual (CPM) Firmware Support Manual (FSM) CSCI Source Code/Executable Code CSCI Database CSCI Nondevelopmental Items

Table B.2.2-1, Configuration Identification of Baselines

B.2.3 Change Control Developmental Configuration Libraries and Tools

The following libraries shall be used to control the development of the <SYSTEM ID> project. Table B.2.3-1 identifies both the libraries and approval required to promote the CI.

Library	Authorization
Functional Baseline	DLIS CCB
Allocated Baseline	DLIS CCB
Development	Programmer/System Engineer
System Test	Technical Lead/Project Manager
Integration Test	DLIS CM/DLIS SQA
Product Baseline	DLIS CM/DLIS SQA

Table B.2.3-1, Developmental Configuration Libraries

B.3 Audits and Reviews

Table B.3-1 identifies the specific software development reviews and CM audits that shall be utilized to control and establish the baselines for the <SYSTEM ID> project.

Baseline	Purpose	Reviews & Audits
Functional	Functions established	System Design Review (SDR)
Allocated	Requirement defined	Software Requirements Review (SRR)
Developmental Configuration	Top level design complete	Preliminary Design Review (PDR)
Developmental Configuration	Detailed design complete	Critical Design Review (CDR)
Product	Approval of product and documentation	Functional Configuration Audit (FCA) Physical Configuration Audit (PCA)

Table B.3-1, Baseline Reviews & Audits

Table B.3-2 identifies scheduled/completed reviews for the <SYSTEM ID> Project. (Formal indicates that the customer attended and Informal indicates attendance by only <CONTRACTOR>).

Code	Audit or Review	Formal or Informal	Scheduled	Completed
SRR	Systems Requirements Review	formal		
SDR	Systems Design Review	formal		
SSR	System Software Review	informal		
PDR	Preliminary Design Review	formal		
CDR	Critical Design Review	formal		
TRR	Test Readiness Review	informal		
FQR	Formal Qualification Review	formal		
DRR	Deployment Readiness Review	formal		

Table B.3-2, Scheduled Reviews & Audits