CONFIGURATION MANAGEMENT PLAN

(CMP) for

Next Gen Archives and Records Center Information System (ARCIS)

Version Number: 1.0

Status: Final

Date: 05/15/24

Document Revision History

Version Number	Implemented By	Revision Date	Revision Description
0.1	CM Team	03/05/24	Initial Draft
1.0	Seema Dheman	05/15/24	Final Version

Configuration Management Plan

For *NextGen ARCIS*

SIGNATURE PAGE

Signature acknowledges that you agree with the Configuration Management Plan and will abide by the details as stated within.

Recommendation for Approval:

_____ _____

Name Falguni Jani Date
Title: Project Manager

_____ _____

Name Edward Graham Date
Title: Application Manager

_____ _____

Name Jay Trainer Date
Title: System Owner

Table of Contents

1.0 Introduction

Configuration Management (CM) is essential throughout a product's life cycle, from inception through its end-of-life. The intensity of CM activity; however, varies according to different technical processes being invoked during a system's life cycle. The CM process in systems and software engineering comprises four primary processes under CM Planning, all designed to manage Configuration Items (CIs) within a given system, beyond the life of a single system.

These four processes are Configuration Identification, Configuration Change Control, Configuration Status Accounting, and Configuration Auditing.

Maintenance support for the licensing of the Oracle/Seibel platform on which ARCIS resides will expire by July 2022. Without support and security patching, as an HVA system containing Personally Identifiable Information (PII) for millions of former military and Federal employees, ARCIS will be operating at risk once the Oracle support has ended. Without ARCIS, the FRCP ability to fulfill the program's primary mission will be critically impacted and risk of loss of Federal records is high. Additionally, the ability for the FRCP to meet its contractual services/obligations to its Federal Agency customers will be affected.

NARA is looking to migrate ARCIS from NARA's Integrated Siebel Platform (NISP) to its Salesforce-based Enterprise Customer Relationship Management (ECRM) system. This call order is to conduct the necessary analysis of the ARCIS Next Generation requirements as provided in the System Requirements Specification (SyRS) and the Requirements Verification Traceability Matrix (RVTM), as well as the Process Descriptions Document (PDD). Once the analysis is complete, those requirements will then be implemented within ECRM. The outcome for this call order is to migrate the ARCIS functionality into the ECRM as a final state solution that will fulfill the business processes of the current and future state of ARCIS.

1.1 Purpose

The purpose of developing a Configuration Management Plan (CMP) is to ensure that the integrity of all Configuration items is maintained throughout the system's life cycle.

This Plan specifies the CM activities and resources required to perform CM for the ARCIS system. These activities and resources will establish and define CM across the life cycle of the ARCIS system.

1.2 Scope

Configuration Management (CM) is a collection of activities focused on establishing and maintaining the integrity of systems, through control of the processes for initializing, changing, and monitoring the configurations and baselines of those systems. The key principles of CM ensure that all components of the ARCIS system can be uniquely identified, managed, and that any previous version of the system can be readily reproduced.

CM support for various cloud environments is performed differently. The cloud provider maintains the configuration of its infrastructure devices, but a customer is responsible for configuring their own

operating systems, databases, and applications. The CM team supports all types of cloud service models as listed below:

In the **Software as a Service (SaaS)** distribution model, applications are hosted by a vendor or service provider and made available to customers, usually over the internet. Software configuration may be enabled for the user so that they can make some types of changes to customize a locally hosted application. In other cases, there may be a single configuration offered. The provider's configuration responsibilities are listed in the service-level agreement (SLA).

In the **Platform as a Service (PaaS)** model, operating systems and associated services are delivered over the internet without downloads or installation. The platform for a developer could include, for example, an operating system, a programming language, an execution environment, a database, and a web server. Configuration and management of all elements are the responsibility of the provider. The vendor is responsible for developing the application.

The **Infrastructure as a Service (IaaS)** quickly scales up and down with demand, letting us pay only for what we use. Like the PaaS service model, configuration of all elements is the responsibility of the provider. It avoids the expense and complexity of buying and managing physical servers and other datacenter infrastructure. Each resource is offered as a separate service component, and we can lease a specific one for as long as we need it. A cloud service provider manages the infrastructure, while we can purchase, install, configure, and manage our own software—operating systems, middleware, and applications.

In the **Function as a Service (FaaS)**, also known as "Serverless Computing" model, software developers design applications and then deploy an individual "function," piece of business logic, or action without maintaining a server. This model takes away low-level infrastructure decisions and server management from the developers. The application architect need not deal with the allocation of resources as it is managed by the cloud service provider. In this model, configuration management is typically invisible to the end user.

The **ARCIS** is hosted as PaaS. Any other ARCIS instances hosted under cloud will follow this CMP.

1.3 Assumptions

A list of possible assumptions that may impact the ability to perform CM activities for the ARCIS system are as follows:

- Access to the cloud environment, and work products, will be provided to whoever requires access as needed to allow reliable monitoring of activities.
- Interface with other NARA governance organizations (e.g., the Enterprise Change Advisory Board (ECAB) or working groups) will be available as needed to support ARCIS system CM activities.
- Sufficient time will be allocated to allow required CM activities to be performed.

1.4 References

Table 1 provides a list of reference documents that are used for developing the CMP.

Note: A list of acronyms is found in Appendix A and key terms can be found in Appendix B.

Reference Documents	
Document	**Description**
IEEE Standard 828-2012, Standard for Configuration Management in Systems and Software Engineering	Establishes the minimum requirements for processes and activities for CM in systems and software engineering. It addresses what activities are to be done, when they are to be done in the life cycle, and what resources and planning are required.
ISO/IEC/IEEE 24765:2017, Systems and software engineering — Vocabulary	Provides a common vocabulary applicable to all systems and software engineering work. It is developed to collect and standardize terminology.
NARA Cloud Computing CM Framework, version 1.0, dated March 28, 2017	Cloud computing (CC) focuses on delivery of reliable, secure, sustainable, and scalable infrastructures for hosting Internet-based application service
NARA System Development Life Cycle (SDLC) Methodology, version 1.6, Date 11/27/13	Software development lifecycle (SDLC) is a framework that development teams use to produce high-quality software in a systematic and cost-effective way. The SDLC methodology is used by both large and small software organizations
NIST Special Publication 800-128	Guide for Security-Focused Configuration Management of Information Systems
SP 800-160 Vol. 1	Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems
SP 800-160 Vol. 2	Developing Cyber Resilient Systems: A Systems Security Engineering Approach
NARA IT Security Methodology for Configuration Management	Guide for the implementation of Configuration Management security controls required by NARA

Table 1 - Reference Documents

2.0 CM Planning

The purpose of CM planning is to determine the level of CM support required for the ARCIS system. This section of the Plan specifies the organization, and resources required to support CM for this system.

2.1 Project Organization

The intended audience for this plan are the CM Team, all Program Team members, and the vendor for the ARCIS system.

2.2 CM Roles and Responsibilities

Below listed are all the roles and responsibilities associated with the project for CM activities:

Roles	Responsibilities
All Project Team Members	Responsible for implementing and complying with the CM guidelines and procedures set forth in this document.
CM Lead	Responsible for: (a) coordinating and implementing CM for On Prem and Cloud Computing; (b) identifying and controlling configuration items (CIs) (c) establishing and managing baselines; (d) providing CM reports as required by the system/project; and (e) ensuring adequate resources, including the tools that are necessary to support system/project CM activities.
CM Analyst	Assists CM lead in performing CM functions

Project Manager (PM)	Interfaces with other functional areas during the life of the project (e.g., reviewing the CM CIL reports monthly) and acts as a liaison between the CM Team and other functional areas (e.g., Security, Operations, etc.).
System Owner	Control and prioritize all the business requests with the PM. Chair the project steering committee to resolve issues that significantly impact the system/project. Elevates issues to the higher-level management whenever needed.
Enterprise Change Advisory Board (ECAB)	Provides governance reviews of system baselines and approves changes to baselines.
Test Team	Responsible for ensuring that the testing processes are performed according to the Test Plan and the documented test processes and procedures.
Release Manager (RM)	Responsible for ensuring that all changes associated with discrete system releases are managed, documented, integrated, verified, validated, and approved prior to release.
ISSO	Ensures processes and activities of established security requirements of the system are maintained. The ISSO completes the security focused areas of the CMP.

Table 2 - CM Roles and Responsibilities

3.0 CM Activities

A list of the major CM activities is provided below.

- Configuration Identification
- Configuration Change Control
- Configuration Status Accounting
- Configuration Audits

3.1 Configuration Identification

3.1.1 Purpose

The purpose of Configuration Identification is to determine the discrete CIs that are established by the system so that they can be tracked and managed throughout the life cycle of the system.

3.1.2 Activities and Tasks

Configuration Identification is associated with selecting the CIs for a system, assigning unique identifiers, and naming conventions, and recording the functional and physical characteristics of CIs in technical documentation. This section documents activities to be performed for establishing and maintaining configuration identification for the ARCIS system.

Naming conventions, combined with labels, are used to uniquely identify CIs that are placed under CM control for the system. The ARCIS follows the Quality Assurance(QA) naming convention that uniquely identifies each CI and enables its different versions to be tracked and managed. All CIs (physical and electronic) are placed under CM control within their respective repositories in accordance with their level of control.

Another important activity related to configuration identification is establishing baselines. A baseline is a specific release of a CI or its aggregate and provides a logical basis for comparison. A baseline is a

reference point in software development that is marked by completion or delivery of one or more software configuration items.

3.1.3 SecCM Configuration Identification

<To be completed by system ISSO>

The CI is identified, labeled, and tracked during its life cycle – the CI is the target of many of the activities within SecCM, such as configuration change control and monitoring activities. A CI may be a specific <System/Project Name> component (e.g., server, workstation, router, application), a group of information system components (e.g., a group of servers with similar operating systems; a group of network components such as routers and switches; or an application or suite of applications), a non-component object (e.g., firmware, documentation), or an information system. CIs give organizations a way to decompose the information system into manageable parts whose configurations can be actively managed.

Configuration Item Labeling

Assets contained within the <System/Project Name> authorization boundary are labeled to include unique hostname and software items:

<Add a list of identified assets including hostnames and software items within the system boundary>

A baseline configuration is a set of specifications for <System/Project Name>, or Configuration Item (CI) within <System/Project Name>, that has been formally reviewed and agreed on at a given point in time, and with <System/Project Name> which can be changed only through change control procedures. The baseline configuration is used as a basis for future builds, releases, and/or changes.

The baseline configuration of <System/Project Name> may evolve over time depending on the stage of the system development life cycle (SDLC). Early in the SDLC when <System/Project Name> is being initiated and acquired, the baseline may be a set of functional requirements. As <System/Project Name> is developed and implemented, the baseline may expand to include additional configuration items such as the technical design, the software load, the architecture, and configurations of <System/Project Name> and its individual components. A baseline configuration may also represent different information computing environments such as development, test, and production.

When a new baseline configuration for <System/Project Name> is established, the implication is that all the changes from the last baseline have been approved. Older versions of approved baseline configurations are maintained and made available for review or rollback as needed.

Identification of Applicable Common Secure Configurations

Common secure configurations are created based on requirements contained within established benchmarks. For all assets, a combination of scripts, batch files and manual configuration will be performed to implement the initial secure state. All cloud components contained within <System/Project Name> will be configured in accordance with the approved benchmark below:

Information System Component CI Baselines

Configurations represent the possible states in which <System/Project Name> can be arranged. Secure configurations are designed to reduce the organizational security risk from operation of <System/Project Name>, and may involve using trusted or approved software loads, maintaining up-to-date patch levels, applying secure configuration settings of the IT products used, and implementation of endpoint protection platforms. Secure configurations for <System/Project Name> are most often achieved through the application of secure configuration settings to the IT products (e.g., operating systems, databases, etc.) used to build the information system. For example, a secure configuration for selected IT products used within <System/Project Name> could incorporate the principle of least functionality. Least functionality helps to minimize the potential for introduction of security vulnerabilities and includes, but is not limited to, disabling, or uninstalling unused/unnecessary operating system (OS) functionality, protocols, ports, and services, and limiting the software that can be installed and the functionality of that software.

The secure baseline is represented in the System Security Plan Hardware/Software List and Architecture Diagram. During assessment, the documented baseline is compared against the assessed baseline. This process is performed at least annually.

The <System/Project Name> baselines have been developed against <insert benchmarks applicable to each CI here>. The following Table(s) documents <System/Project Name> baselines including authorized deviations and exceptions:

Index	Configuration Requirement	Status	System Setting	Justification/Risk Management

Table 3 <System/Project Name> Baselines for <include CI>

3.2 Configuration Change Control

3.2.1 Purpose

The purpose of configuration change control is to systematically control changes to CIs and their baselines throughout the system life cycle and to maintain the integrity of project work products and CIs. Configuration control is also associated with ensuring that all changes to these items are properly specified, evaluated, approved, tracked, and documented.

3.2.2 Activities and Tasks

Configuration control is the process for evaluating, coordinating, and deciding on the disposition of proposed changes to the CIs and implementing the approved changes to system baseline and associated documentation and data. The configuration control process ensures that the changes that have been proposed are evaluated and dispositioned (approved or disapproved), and the changes that are approved are implemented, tested, verified, and incorporated into a new baseline.

The Project Manager (PM) along with the CM lead, must ensure that an adequate configuration control process is in place to evaluate all changes for impact and urgency and shall include a process to handle

emergency changes; also responsible for developing a Configuration Control Board (CCB) Charter for the NextGen ARCIS if needed.

Configuration Control process ensures that proposed changes are rigorously evaluated and dispositioned. Approved changes are then implemented, tested, verified and integrated into a new baseline. Depending on the scope, schedule, or cost there will be a hierarchy of people who can approve the changes. This hierarchy is defined in the ECRM CCB Charter and ECAB charter.

The NextGen ARCIS leverages ECRM support requests to initiate tickets for changes within the NextGen ARCIS system, seamlessly integrating with ECAB for DNS entry and infrastructure adjustments. Any ticket initiated with a NARA support request, identified by IO as an NextGen ARCIS application issue, will be directed to the ECRM CCB as seemed appropriate for resolution.

3.2.3 Security Impact Analysis

<To be completed by ISSO>
Prior to the approval of a change to <System Name>, the ECAB in consultation with the NARA Cyber Security & Information Assurance Division (IS) and <System/Project Name> ISSO analyzes changes to the information system to determine potential security impacts. The security impact analysis for <System/Project Name> is conducted to determine the extent to which the change impacts the security posture of the system.

3.2.4 SecCM Monitoring

<To be completed by system ISSO>

Configuration monitoring involves activities to determine whether <System/Project Name> is configured in accordance with the organization's agreed-upon baseline configurations, and whether the IS components identified within <System/Project Name> are consistent with the IS component inventory being maintained by the organization.

Configuration monitoring helps to ensure that SecCM controls are operating as intended and are providing effective security while supporting adherence to SecCM policies and procedures. Configuration monitoring may also help to motivate staff members to perform SecCM activities in accordance with policies and procedures. Additionally, configuration monitoring supports organizations in their efforts to conform to the Risk Management Framework.

Information gathered during configuration monitoring can be used to support overall continuous monitoring activities including ongoing assessments of specific security controls and updates to security documentation such as System Security Plans, Security Assessment Reports, and Security Status Reports. Automated capabilities including system scanning, can be used for configuration monitoring activities.

Configuration monitoring is part of the Monitoring phase of SecCM and supports the implementation of all NIST SP 800-53 controls in the CM Family.

SecCM Monitoring Requirements and Frequencies

The following types of monitoring activities are performed monthly:

1. Scanning to discover components not recorded in the inventory.
2. Scanning to identify disparities between the approved baseline configuration and the actual configuration for an information system.
3. Scanning to identify vulnerabilities in an information system.

3.3 Configuration Status Accounting

3.3.1 Purpose

The purpose of Configuration Status Accounting (CSA) is to record, track and report on critical information and status of project CIs to management, the project team, and other stakeholders.

3.3.2 Activities and Tasks

A few of CM's standard reports are listed below.

3.3.2.1 Configuration Item List (CIL) Report

The CIL Report provides a status of the system CIs as placed under CM control. The CM team compiles the CIL report and is issued to all project stakeholders.

Frequency: A CIL Report is published on bi-monthly or as needed basis.

3.3.2.2 Project Closeout Report

A Project Closeout Report is used to reconcile the final project deliverables and system CIs at the time of project closeout.

Frequency: A Project Closeout Report is published at the end of the project, a phase of the project, or a logical end point of the project. The ARCIS system does not have a close out report instead completed CIL report is considered as Project Closeout Report.

3.3.2.3 Change Request (CR) Report (CRR)

A CRR is generated to provide information regarding the status of changes to the system. A CRR report includes information such as: status of pending and open CRs; number of CRs submitted, approved, closed, etc.; number of CRs deferred or rejected; number of emergency fixes; category or priority of CR (e.g., emergency, expedited, normal, standard, etc.). The CRR report allows for analysis of metrics, such as average turnaround time to process a CR, etc.

Frequency: The CR Report is published as needed by the PM.

3.4 Configuration Audit

3.4.1 Purpose

A Configuration Audit is an evaluation of a product to ascertain compliance to CM requirements and CM standards. Configuration audits are generally used to assess the system's baseline or release status and determine whether the system is complete as delivered. Audits of CM activities are performed to assess effectiveness of the system's CM procedures and to identify areas for improvement.

3.4.2 Activities and Tasks

There are four (4) standard types of CM audits conducted by the IQ CM team for all projects under CM support.

3.4.2.1 Functional Configuration Audit (FCA)

The FCA shall be conducted to verify that the development of a CI has been completed satisfactorily, that the item has achieved the performance and functional characteristics specified in the functional or allocated configuration identification, and that it is operational and support documents are complete and satisfactory. The FCA allows validation of the system against the Requirements. (ISO/IEC 24765:2009)

The FCA is not needed for this system as this is developed in agile methodology. In an Agile environment the evidence for a sign-off on the audit is embedded within the collaborative nature of the process itself. At the end of each iteration, typically lasting between one to four weeks, working software is delivered, potentially ready to be deployed to production, which is designed, tested and requirements in form of user stories are already met.

3.4.2.2 Physical Configuration Audit (PCA)

The PCA shall be conducted to verify that a CI, as built, conforms to the technical documentation that defines it. (ISO/IEC 24765:2009)

PCA is not applicable because it's using agile methodology. At the end of each iteration, typically lasting between one to four weeks, working software is delivered, potentially ready to be deployed to production, which is designed, tested and requirements in form of user stories are already met. The CMDB will be provided by the O&M team.

3.4.2.3 Baseline Audits

A baseline audit shall be conducted to assess the integrity of a baseline for: completeness, correctness, functionality compared to requirements, documentation compared to requirements, and generating an audit report. A baseline audit may be either scheduled or unscheduled. Physical examination to verify that the configuration item(s) (CI) "as built" conform to the technical documentation which defines the item. Approval by the government program office of the CI product specification and satisfactory completion of this audit establishes the product baseline

The following documents are provided to establish a project baseline:

- **Version Description Document (VDD)**

- The VDD serves as the key change control document, pinpointing all Configuration Items (CIs) linked to a system release for verification, integration, or deployment. It offers a concise overview of the features and contents of the CIs comprising a specific release.

  Frequency: A VDD is submitted with each new system release.

- **Configuration Management Database (CMDB)**

  A CMDB is a database used to store configuration records of the system CIs throughout the system's life cycle. A CMDB may be created in a spreadsheet format if a database tool is not available. The CMDB is prepared for IaaS and PaaS projects but not SaaS.

  Frequency: A CMDB is submitted for each new system release and subsequent code changes as needed.

- **Deployment Source and Object Code Package**

  A "Deployment Source and Object Code Package" in electronic format is required for each new system release that involves code changes. The package shall consist of all source code files, arranged in build directory structure, and include all instructions, build script, make files and utilities necessary to rebuild the application from the source code, along with instructions for their use. The code is submitted for IaaS and PaaS systems but not SaaS.

  Frequency: A Deployment Source and Object Code Package is submitted for each new system release.

3.4.2.4 Project Closeout

A Project Closeout Audit is conducted by the CM Lead at the end of the project, a phase of the project, or a logical end point of the project. A Closeout Audit is conducted against all CIs and controlled work products to verify and assess the integrity of final versions and dates (as required) in accordance with QA's CM guidance, procedures, and standards; and all results, findings, and discrepancies are documented in a report. This report is submitted to all relevant contractors and any functional unit involved in the technical documentation or development of any system CIs for final reconciliation and resolution of outstanding findings.

4.0 Release Management

Release management refers to the process of planning, designing, scheduling, deploying, and controlling software releases. It ensures that release teams efficiently deliver the applications and upgrades required by the business while maintaining the integrity of the existing production environment. The ARCIS NextGen will follow the Release Management procedure for any upcoming releases.

5.0 CM Resources

The below listed sections reflect the information about the CM personnel, CM products and tools and training necessary to implement CM activities for the project.

5.1 CM Personnel

The CM team is divided into two (2) roles in support of CM operations: the CM Lead and the CM Analyst. The CM Analyst is responsible for supporting the CM Lead in the day-to-day operation and planning of CM activities for these projects.

5.2 CM Processes and Tools

The CM team uses PVCS Version Manager as the CM repository for all CIs and work products that are placed under CM control for projects. PVCS provides access for authorized users to check-in and check-out items for use. PVCS provides version control capability to manage the versioning of changed files, basic reporting capability, the ability to review previous versions, and information about revisions made to an item stored within PVCS.

5.3 CM Training

The training will be provided as needed.

6.0 CM Plan Maintenance

The PM along with CM Lead is responsible for the development and maintenance of the CMP. At a minimum, the CMP will be reviewed annually, and updated as needed throughout the entire project lifecycle. It ensures the relevance and adequacy to plan and manage CM activities. The CMP will be kept under CM control.

Appendix A: Acronyms

SYSTEM NAME	Full Name
CCB	Configuration Control Board
CC	Cloud Computing
CI	Configuration Item
CIL	Configuration Item List
CM	Configuration Management
CMDB	Configuration Management Database
CMP	Configuration Management Plan
CR	Change Request
CSA	Configuration Status Accounting
CR	Change Request

CRR	Change Request Report
ECAB	Enterprise Change Advisory Board
FCA	Functional Configuration Audit
IaaS	Infrastructure as a Service
IEEE	The Institute of Electrical and Electronics Engineers, Inc
ISO/IEC	International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC)
ISSO	Information System Security Officer
IT	Information Technology
NARA	National Archives and Records Administration
PaaS	Platform as a Service
PCA	Physical Configuration Audit
PM	Project Manager/Project Management
QA	Quality Assurance
RFC	Request for Change
RM	Release Manager/Release Management
SaaS	Software as a Service
SDLC	System Development Life Cycle
SLA	Service Level Agreement
VDD	Version Description Document

Appendix B: Key Terms and Definitions

Baseline is a specific release of a work product and each of its component CIs that are aggregated as a composite set of related files, under version control and configuration management (usually in a CM tool) and staged for release. A baseline also contains any staging instructions that describe how to assemble the work product from its component parts and verify the assembly.

Configuration Item (CI) is any document, diagram, software component, file, model, hardware element, database, or procedure developed by the <system/project name> that requires version control and access control and must be managed throughout the life cycle of the system.

Configuration Management is the process of managing the individual CIs that constitute a composite work product (i.e., *configuration*). Configuration management is used to manage the *collection and integration* of CIs that comprise a work product, with each component CI being separately maintained and managed under version control.

Release Management is the process of disseminating, tracking, and controlling a composite set of work products (e.g., a software release or a sub-system build) that are under configuration management.

Releases are formalized updates to work products that are: (a) assigned specific release identifiers, (b) given official release dates, (c) formally reviewed and vetted, (d) widely and formally disseminated, and (e) usually the outcome of specific project activities that are prescribed by the Project Management Plan.

All versions of a work product are not necessarily releases (i.e., numerous intermediate versions of a work product (and its component CIs) may be developed prior to an official release).

Staging (build or assembly) is the activity of combining and integrating composite work products from their component CIs and rendering those work products in whatever formats are appropriate for the distribution of a release. In software development parlance, this is analogous to the build, and packaging of a software release. In systems engineering parlance, this is analogous to assembly, integration, and packaging of system elements.

Version Control is the process of tracking and controlling the changes to a specific CI or work product.

Versions are iterations of individual CIs or composite work products that reflect changes in structure or content from the previous version.

Work products are aggregates of numerous CIs such as documents with embedded diagrams, composite software systems containing numerous software modules, or composite system elements containing hardware, software, and process components.

Appendix C: ARCIS Instances

- NextGen ARCIS