ISSUE HISTORY

Issue	Description of Change	Originator	Effective Date
1	Initial Release	Task Force	February 1, 2024

REFERENCE DOCUMENTS

Document Number	Document Title
ISO9001:2015	Quality Management System - Requirements, Clause XX
ISO 9000: 2015	Quality Management Systems-Fundamentals & Vocabulary

Approval:	Name:	Signature:	Date:

```
FOR DOCUMENT CONTROL
     USE ONLY
```

Approval:	Name:	Signature:	Date:

DEVELOPMENT

1. PURPOSE

The purpose of this work instruction is to establish a standardized approach to software development processes in accordance with ISO 9000 and ISO 9001 standards. It outlines the steps, responsibilities, and quality assurance measures to ensure consistent and high-quality software deliverables.

2. SCOPE

This work instruction applies to all software development activities within Information Network Security Administrator (INSA). It covers the entire software development life cycle, from requirements analysis to testing and deployment.

3. PROCESS OWNER

- Director General

4. DEFINITIONS AND ABBREVIATIONS

4.1 Definitions

Quality Assurance: Detail the quality assurance measures in place, such as code reviews, testing protocols, and adherence to coding standards.

Quality Management System (QMS): A set of processes, policies, and procedures needed for planning and execution in the core business area of an organization.

4.2 Abbreviations

BRS: Business requirement specification

SRS: software requirement specification

RAD: requirement analysis document

UAD: user acceptance document

Approval:	Name:	Signature:	Date:

PID: project initiation document

SAD: system acceptance document

QAMS: Quality Management System

QA: Quality Assurance

XSS: cross-site scripting

5. WORK INSTRUCTION

- The standards apply to projects using JavaScript as the primary programming language, incorporating frameworks and technologies such as Next.js (version 15)**,** Express.js for backend development, and Tailwind CSS for styling, and MongoDB as the database solution. In addition, these standards extend to projects built with Java using the Spring Boot framework, and C# using the .NET framework both of which are approved languages and platforms for backend and enterprise-level application development within this division.

- These standards are mandatory for all development teams working on web applications or services built using any of the above-mentioned technologies: JavaScript with Next.js**,** Express.js**,** Tailwind CSS**,** and MongoDB**;** as well as Java with Spring Boot and C# with **.NET.**

- Any projects or teams using technologies outside of JavaScript, Next.js 15**,** Tailwind CSS**,** Java with Spring Boot, or C# with .NET, or those with unique technical requirements, must seek prior approval from the Architecture Review Board (ARB) before deviating from these standards.

Approval:	Name:	Signature:	Date:

Coding Conventions

- File Naming: Define consistent naming conventions for files and directories.

- Indentation: Specify the preferred indentation style and level for different blocks of code.

- Spacing: Establish rules for spacing around operators, keywords, and parentheses.

- Line Length: Set a maximum line length to enhance readability.

- Comments: Describe conventions for using comments to explain code and its purpose.

- Braces: Define preferences for brace placement and style.

- Naming Conventions: Provide guidelines for naming variables, functions, classes, and modules.

- Variable Declarations: Specify rules for declaring variables, including type, scope, and initialization.

- Control Flow: Outline best practices for using control flow statements (if, for, while, etc.).

- Error Handling: Describe strategies for handling errors and exceptions gracefully.

Documentation Guidelines

- Comments: Explain the use of comments within code for clarity and understanding.

- Documentation Structure: Define the structure and format of external documentation.

- Tools and Templates: Recommend tools or templates for creating and maintaining documentation.

Testing Standards

- Test Case Writing: Outline guidelines for writing effective test cases.

Approval:	Name:	Signature:	Date:

- Test Coverage: Specify requirements for code coverage by tests.

- Test Reporting: Describe standards for test results and reporting.

Version Control

- Branching: Define best practices for branching and merging in version control systems.

- Merging: Establish guidelines for merging code changes back into the main codebase.

- Commit Messages: Provide guidelines for writing clear and concise commit messages.

Security Considerations

- Input Validation: Emphasize the importance of validating user input to prevent security vulnerabilities.

- Output Encoding: Address proper output encoding to mitigate risks like cross-site scripting (XSS).

- Error Handling: Highlight secure error handling practices to avoid sensitive information disclosure.

- Authentication and Authorization: Stress the need for robust authentication and authorization mechanisms.

- Encryption: Discuss guidelines for data encryption, especially for sensitive information.

6. RECORDS

The software development life cycle must the following records.

- BRS
- SRS
- RAD
- UAD

Approval:	Name:	Signature:	Date:

- PID

- SAD

- Test Report

- Test Log

- Audit Certification

7. RELATED DOCUMENTS

Document Number	Document Title
OT/GD/XXX	Business requirement Specification
OT/GD/XXX	Requirement Analysis Document
OT/GD/XXX	Software requirement Specification document
OT/GD/XXX	Software Architecture Document
OT/WI/XXX	DevOps Guideline
OT/GD/XXX	Secure Software Development and Management Standard
OT/GD/XXX	Software development Coding Standard
OT/GD/XXX	User Acceptance Document
OP/GD/XXX	Software development procedure

Approval:	Name:	Signature:	Date: