

Table of Contents

1. INTRODUCTION	3
2. ROLES AND RESPONSIBILITY	3
3. COMPLIANCE TABLE	3
4. REVIEW	3
5. PROCEDURES	3
5.1 PROTECTION OF SYSTEM TEST DATA.....	3
5.2 ACCESS CONTROL TO PROGRAM SOURCE LIBRARY(CODES)	3
5.3 SEPARATION OF DEVELOPMENT AND OPERATIONAL FACILITIES.....	4
5.4 TEST ENVIRONMENT	5
5.5 SOFTWARE AND SOURCE CODE VERSION CONTROL.....	5
5.6 RESTRICTIONS ON CHANGES TO SOFTWARE PACKAGES.....	5
6. SECURITY IN SOFTWARE DEVELOPMENT LIFECYCLE (SDLC).....	6
6.1 SOFTWARE PLANNING.....	6
6.2 REQUIREMENTS ANALYSIS.....	6
6.3 SECURE SOFTWARE ACQUISITION.....	7
6.4 DESIGN AND DEVELOPMENT	7
6.4.1 ARCHITECTURE AND DESIGN REVIEW	8
6.4.2 SECURE DEVELOPMENT	8
6.5 TECHNICAL REVIEW AND TESTING	9
6.6 SYSTEM CHANGE CONTROL PROCEDURES.....	10
6.7 SECURE DEPLOYMENT	11
6.8 SOFTWARE MAINTENANCE	11
6.9 SOFTWARE DOCUMENTATION	12
--End of Document--	12

DOCUMENT CONTROL**VERSION CONTROL**

Document Control ID	EFR-SECURITY IN SOFTWARE DEVELOPMENT LIFE CYCLE PROCEDURE-022
Issued Date	
Effective Date	
Owner	Information Security & Compliance Department

REVISION TABLE

Date	Version	Brief Description	Author
17/10/2024	1.0	Draft-Security in Software Development Life Cycle	Ms. Ratisha Kukreja
	1.0	Final- Security in Software Development Life Cycle	Ms. Ratisha Kukreja

RELEASE AUTHORIZATION

Task	Author	Title	Date
Prepared By	Ms. Ratisha Kukreja	Consultant	

REVIEWER AUTHORIZATION

Task	Author	Title	Date
Reviewed By	Mr. Antony Arul Selvaraj	Consultant	

APPROVAL AUTHORIZATION

Name	Title	Signature	Date

Important Note: This document is intended solely for the use of the individual or entity to whom it is transmitted to and others authorized to receive it. It may contain confidential or legally privileged information. If you are not the intended recipient you are hereby notified that any disclosure, copying, distribution or taking any action in reliance on the contents of this document is strictly prohibited and may be unlawful. If you have received this document in error, please notify us immediately.

1. INTRODUCTION

Software acquisition, development and maintenance procedure involves the use of a broad range of information security controls to ensure that the software that is acquired or developed passes through a standard process that ensures compliance with functionality, security, performance, and regulatory requirements. The procedure also aims at ensuring timely maintenance of the software and its environment.

2. ROLES AND RESPONSIBILITY

Project Manager shall be responsible for the software acquisition, development, and maintenance.

3. COMPLIANCE TABLE

Policy/Procedure Name	ISO 27001:2022 Control	UAE IA Policy/Procedure Name	UAE IA Control	Requirements
Security in Software Development Life Cycle Procedure	A 8.25	Separation of Development, Test and Operational Facilities	T3.2.5	NIST SP800-144 Guidelines.

4. REVIEW

The procedure needs to be reviewed by EFR once a year, else whenever there are any changes implemented in the existing procedure.

5. PROCEDURES

5.1 PROTECTION OF SYSTEM TEST DATA

Test data shall be protected and controlled.

The use of operational data containing personally identifiable information or any other confidential information for testing purposes shall be avoided. If personally identifiable information or otherwise confidential information is used for testing purposes, all sensitive details and content shall be protected by removal or modification.

The following controls shall be applied to protect operational data, when used for testing purposes:

- The access control policy, which apply to operational application systems, should also apply to test application systems.
- There should be a separate authorization, each time operational information is copied to a test application system.
- Operational information should be erased from a test application system immediately after the testing is complete.
- The copying and use of operational information should be logged.

5.2 ACCESS CONTROL TO PROGRAM SOURCE LIBRARY(CODES)

Access to program source code and associated items (such as designs, specifications, verification plans and validation plans) shall be strictly controlled, to:

- Prevent the introduction of unauthorized functionality Avoid unintentional changes
 - Reduce potential for corruption of computer programs
 - Maintain the confidentiality of valuable intellectual property.
- Where possible, program source libraries shall not be stored on operational systems.
- Programs under development or maintenance should be not stored in operational program source libraries.
- The designated application owner is responsible for the application and should approve accesses to the program source library of the application.
- Only authorized developers will have write access to these libraries. Developers, in general, will have read-only access to the libraries in the testing environment. Support personnel shall not have unrestricted access to program source libraries
- The production archive is where programmers will move modified source code after completion of development, unit testing and integration testing. The production archive will have backup of all source code and executable programs
- Maintenance and copying of program source libraries and source shall be subject to strict change control procedures. The updating of program source libraries and the issuing of program sources to programmers should be performed only by the Program Manager upon authorization from Information Security Head. EFR should consider implementing source code safeguard software, which requires program check in and check out every time a source code is released or submitted to the library.
- Information Security head should periodically review check in and check out details to ascertain whether any unauthorized release or submission of source code is made, and this also assists in analysing the progress of changes being made to the applications.
- A backup copy of the source code will be properly safeguarded in a secure off-site location.
- The source code will be readily available to reproduce application execution code in the event of a disaster or problem with the production environment. The updated source code will accurately reflect the current processing.

5.3 SEPARATION OF DEVELOPMENT AND OPERATIONAL FACILITIES

EFR shall implement separate environment for development, testing and operational facilities to reduce the risk of accidental change or unauthorized access to operational software and business data.

The level of separation that is necessary, between operational, test and development environments, to prevent operational problems shall be considered.

The following controls shall be considered to separate development, test and operational facilities:

- Development and operational versions of the software run on different computer processors, or in different domains or directories.
- The development environment will be accessible only by the application development group.
- Development and testing activities should be separated.

- Compilers, editors and other system utilities should not be accessible from operational systems.
- Different log-on procedures should be used for operational and test systems. Users must use different passwords for these systems, and menus should display appropriate identification messages.
- The production archive is where programmers will move modified source code after completion of development, unit testing and integration testing.
- Development staff will not have knowledge of any operational passwords, unless such passwords are required for the support of live systems.

5.4 TEST ENVIRONMENT

The testing environment is where quality assurance and user acceptance testing will be conducted with an independently controlled library.

If a separate testing environment is not feasible, then the test environment will be created from the development environment by locking the test environment against any modification by programmers.

5.5 SOFTWARE AND SOURCE CODE VERSION CONTROL

Software will be held in secure libraries and the libraries will be qualified using the release and/or version number to distinguish different versions.

The software components will be identified by means of a unique name and assigned a release and version number.

The content of each version will be documented providing a brief description of the system elements that are included.

Source codes will be controlled using an automated version control system. The version control system will ensure that the appropriate versions of source modules are mapped to each application release.

Modified programs will be assigned a higher version number following a change.

Old versions of source programs shall be archived, with a clear indication of the precise dates and times when they were operational, together with all supporting software, job control, data definitions and procedures.

5.6 RESTRICTIONS ON CHANGES TO SOFTWARE PACKAGES

- Modifications to software packages shall be restricted, limited to necessary changes and all changes shall be strictly controlled. Vendor-supplied software packages shall be used without modification. Version control system shall be used to track modifications and updates.
- Where modification is required, consent of the vendor, the risk of built-in controls and integrity processes being compromised and compatibility with other software in use shall be considered before implementing the changes.
- Agreements with the vendor shall be established to provide support and maintenance on the software after the changes.

6. SECURITY IN SOFTWARE DEVELOPMENT LIFECYCLE (SDLC)

6.1 SOFTWARE PLANNING

- EFR will assign a project owner for development and implementation of the software. The project owner will ensure that all the concerned divisions are involved while deciding on the software to purchase or develop.
- A Project Plan detailing steps, responsibilities and project owners will be developed, and a formal Systems Development Life Cycle (SDLC) will be followed.
- The Information Owner of the application will be identified and documented.
- Information to be processed by the new software will be classified according to the EFR's Information Classification procedure. This will help to determine the amount of security controls required.
- The software development and implementation team will include Information Systems Audit officer as assigned by the Inspection and Audit Department. This person will ensure that the required security controls are developed and/or implemented on the system.
- A Security Plan for the project must be developed. This plan will detail the security controls such as segregation of duties and development environments as detailed in the Change and Problem Management procedure. This Security Plan will ensure the integrity of the software being developed.

6.2 REQUIREMENTS ANALYSIS

- Based on the new software's criticality in the Information Classification, an audit plan will also be developed for the software design, development and implementation phases.
- Based on the Information Classification, a contingency plan will be prepared for the software according to the EFR's Business Continuity policy.
- A preliminary security test plan will be developed. This plan will identify all security objectives like evaluation, testing, and audit requirements, relevant EFR security policies and standards.
- Detailed security requirements for the software will be prepared based on Information Classification chart. This will include:
 - Security requirements from the Information Owners
 - Technology controls
 - Audit requirements
- The information security related requirements shall be included in the requirements for new information systems or enhancements to existing information systems.
- Details of the requirements shall be maintained in the Software Requirement Specification document.

6.3 SECURE SOFTWARE ACQUISITION

Software to be acquired shall also be evaluated based on the following-

- Appropriate validation controls (Input and Output) and Information Processing Controls are built into the software.
- Adequate cryptographic & key management controls are constituted in the software to provide appropriate protection of data.
- The software code is assessed internally to check for the vulnerabilities (SQL Injection, Cross Site Scripting, Cross Site Request Forgery etc.) and to ascertain that all fixes have been applied.

Upon vendor finalization, as part of the acquisition checklist, Project Officer shall confirm that the vendor provides the following minimum requirements:

- Software details (Module/Suite purchased).
- Authentication (User ID, LDAP/Active Directory).
- Hardware requirements (Server, RAM, Client Requirements etc.).
- Architecture (Client-server/Web/Virtualization).
- Network Requirements (LAN/WAN, firewall, Port, Protocol details).
- Database requirements (Database – MongoDB/SQL Server etc., versions).
- Supporting documentation (Software Manuals).

Software acquired shall be updated in the asset register, detailing the name, vendor, license details (license number and date of expiry) and support service details (service contact, location and escalation matrix).

Project Officer shall ensure that vendor agreement of the software shall clearly specify all the software and its related components as part of the agreement.

6.4 DESIGN AND DEVELOPMENT

Project Officer department shall facilitate the customization needed for the acquired software. During modification of any third-party product, the project office shall check if

- Any consent of the vendor needs to be obtained for the customization; and
- EFR becomes responsible for the future maintenance of the software for any changes.

Modification shall be initiated only after obtaining affirmative responses for the above points.

Project Officer along with third party IT vendor shall ensure that all changes to the software is made in a controlled and secure manner after referring to the change management procedure. The following change control procedures shall be adhered to:

- Approvals and authorizations from business owners, Project Officer and ISSC shall be sought and documented.
- Version control for all software updates shall be maintained.
- Audit trails to all change requests shall be maintained.

Security specifications for the new software will be documented to provide the development group with specific requirements. The security specification document will be separated from the software design documents. This enables EFR in identifying, reviewing and testing the security functionalities of the software. Security specifications will be reviewed by all the divisions involved in development, maintenance and use of the new software.

6.4.1 ARCHITECTURE AND DESIGN REVIEW

To counter software vulnerabilities upfront, a risk assessment and analysis shall be performed before designing the application.

The objective is to understand and evaluate all modules and tiers of the application to be developed in terms of:

- Connectivity and data flow within the application and through its interfaces
- Use of approved secure protocols and cryptography standards,
- Provisioning for maintaining audit trails.
- Mechanisms for error handling, session management, etc.

Details of the technical and non-technical controls necessary to achieve the requirements shall be maintained.

6.4.2 SECURE DEVELOPMENT

To ensure control & integrity of the source code & component configurations, the applications shall be developed as per the secure coding practices established by the EFR.

Secure programming techniques shall be used both for new developments and in code re-use scenarios where the standards applied to development may not be known or were not consistent with current best practices.

If development is outsourced, EFR shall obtain assurance that the external party complies with these rules for secure development.

To mitigate commonly known software vulnerabilities, secure coding practices shall be used by the developers as listed below, but not limited to:

- Input Validation
- Output Encoding
- Authentication and Password Management
- Session Management
- Access Control

- Cryptographic Practices
- Error Handling and Logging
- Data Protection
- Communication Security
- System Configuration
- Database Security
- File Management
- Memory Management
- General Coding Practices

Testing and code review shall verify their use.

6.5 TECHNICAL REVIEW AND TESTING

When operating platforms are changed, business critical applications should be reviewed and tested to ensure there is no adverse impact on organizational operations or security.

Prior releasing the applications on the production environment, the applications shall be tested from a functionality and security perspective to verify that the application code meets security requirements/specifications.

A quality assurance test of the changes to be implemented will be performed by Quality Assurance team in a test environment prior to implementation in the production environment.

Security test plan - will be implemented and test results will be approved by the concerned divisions. Security test plans will be formalized. Details of test plans, test cases and results if testing shall be documented and maintained.

Unit Testing - The developer will support unit testing in the test environment carried out by the Quality Assurance team. Quality Assurance team may comprise of one or more experienced users / officers depending on the complexity of application and project deadlines. The Development Team Leader will perform an independent review of the test result and will formally sign-off on the test results. If problems are encountered, the developer will document the problem, make appropriate modifications in the development environment and submit it to Quality Assurance team for retesting.

Integration Testing - All modifications, enhancements and installation or implementation of new systems will be subject to "Integration Test" by the appropriate users prior to installation into production. Copies of production data must be used for the testing purposes. For major enhancements and new systems, a formal integration-testing plan will be prepared. The plan will include stress test and volume test, and in some cases, parallel testing may be required. Integration testing will be conducted in a separate, independently controlled test environment and the developers should have no access to this environment. The concerned divisions / users will indicate acceptance through formal sign-off. If problems are encountered, the Quality Assurance team will document the problem, and the developer will make appropriate modifications and submit it to the Quality Assurance team for retesting.

Acceptance Testing - All modifications, enhancements and installation or implementation of new systems will be subject to "Acceptance Test" by the appropriate users prior to installation into production. Copies of production data must be used for the testing purposes. For major enhancements and new systems, a formal "User Acceptance Plan" will be prepared. The plan will include tests of all major functions, processes and

interfacing systems. User acceptance testing will be conducted in a separate, independently controlled test environment and the developers should have no access to this environment. The concerned users will indicate acceptance through formal sign-off. If problems are encountered, the Quality Assurance team will document the problem, and the developer will make appropriate modifications and submit it to the Quality Assurance team for retesting.

Testing shall be performed in a realistic test environment (which are separate systems segregated from the production environment) to ensure that the system will not introduce vulnerabilities to the organization's environment and that the tests are reliable.

Test methodologies - EFR may allow testing methods such as code analysis tools or vulnerability scanners and will verify the remediation of security related defects. The tests shall include test-scenarios over usability, security, effects on other systems and user-friendliness.

The source code of the application may be reviewed for the following (but not limited to):

- Buffer overruns and overflows
- OS Injection
- SQL Injection
- Data Validation
- Cross-Site Scripting
- Cross-Site Request Forgery Issues
- Logging Issues
- Session Integrity Issues
- Race Conditions

The review shall verify the application controls and integrity procedures to ensure that they have not been compromised by the operating platform changes.

6.6 SYSTEM CHANGE CONTROL PROCEDURES

- A. Formal change control procedures shall be documented and enforced to ensure the integrity of system, applications and products, from the early design stages through all subsequent maintenance efforts.
- B. Introduction of new systems and major changes to existing systems shall follow a formal process of documentation, specification, testing, and quality control and managed implementation. The objective of this process is to ensure the following:
 - Including a risk assessment process, analysis of the impacts of changes and specification of security controls needed.
 - Existing security and control procedures are not compromised,
 - Support programmers are given access only to those parts of the system necessary for their work and that formal agreement and approval for any change is obtained.
- C. The change control procedures shall include:
 - a record of agreed authorization levels
 - evidence that changes are submitted by authorized users
 - identification of all software, information, database entities and hardware that require amendment

- formal approval for detailed proposals before work commences and assurance that authorized users accept changes prior to implementation
- version controls for all software updates

D. EFR shall record and maintain audit trail of all change requests.

E. EFR shall ensure that implementation of changes takes place at the right time and does not disturb the business processes involved.

6.7 SECURE DEPLOYMENT

- Before the implementation, the Information Security Team will create and document any new security procedures for the new software based on EFR's Security Policies and Standards.
- For software packages, system default settings must be reviewed prior to installation to determine potential security holes. Settings that could potentially compromise security must be changed prior to the system being placed in a production environment.
- Project Managers will be provided with the listing / hard copies of the prevailing control parameters and default settings to enable them to compare and verify the settings on their respective systems.
- For software packages, all third party supplied default passwords must be changed prior to the system being placed in a production environment.
- Software shall be deployed with the help of the IT Team. All security configurations shall be ensured during installation. Rollback strategy and plan shall be in place prior to implementing the changes to the production environment.
- To ensure minimal issues and security risks during deployment of the application into the production environment, a production readiness activity should be performed to address/accept the following risks/issues:
 - Unhardened production server
 - Application to be deployed without security testing
 - Applications deployed with un-remediated vulnerabilities
 - Performance considerations with respect to bandwidth, platform/OS dependency, concurrent users, network devices, etc.
- The application shall be deployed to the production environment on receiving approvals from the Information security team and the business team.

6.8 SOFTWARE MAINTENANCE

- The entire development environment shall be equipped with latest versions/updates of operating systems, servers and databases. Project Officer shall ensure that patch updates to all software, servers and databases are applied as and when a release happens.
- Software updates (application version upgrade, program libraries etc.) shall be performed by trained administrators upon appropriate authorization from the Project Officer
- Internal Audit Team/Information Security Team shall ensure periodic vulnerability assessments and penetration testing to ensure software resilience. Vulnerability management standard shall be referred for details.

6.9 SOFTWARE DOCUMENTATION

- Project Officer shall maintain and manage all documentation of the software via a list. The list shall capture the document name/ID, location of storage, current version and resource distribution list.

--End of Document--

CONFIDENTIAL