Secure Software Development Lifecycle Policy

|  |  |
| --- | --- |
| Policy Area | IT Policy Library |
| Approved Date | December 31, 20XX |
| Approved By | Policy Committee |
| Effective Date | January 1, 20XX |
| Current Version | 1.0 |

# I. Overview

Implementing sound software development techniques with proven security strategies helps ABC Company manage its risks throughout the entire software lifecycle from initial inception through final disposition of an application.

# II. Purpose

This policy describes the requirements for securely developing, implementing, acquiring, and managing the software development life cycle.

# III. Scope

This policy applies to all Staff that provide and manage software development services.

# IV. Policy

A. Software Development Lifecycle

ABC Company’s IT Department management is responsible for developing, maintaining, and managing a Software Development Life Cycle (SDLC). All software developed in-house which runs on production systems must be developed according to the SDLC. At a minimum, SDLC should address the following ten areas:

1. Feasibility study
2. Risk assessment
3. User requirements
4. Systems design
5. Programming or acquire software (development/acquisition)
6. Quality assurance
7. Documentation
8. Systems testing and acceptance
9. Installation
10. Maintenance through disposal

Information security must be integrated into new application and systems development from their inception and throughout the development lifecycle. The development lifecycle is defined as a period that begins with conception of a new development project and ends with retirement or removal of the developed software from all active use.

B. Roles and Responsibilities

The Chief Security Officer (CSO) are responsible for:

* Ensuring that security controls are developed to manage and reduce software related risks throughout its lifecycle.
* Ensuring software development projects include security in all stages of the lifecycle phases.
* Assist application developers and data owners in identifying security requirements for each lifecycle phase.

Data Owners are responsible for:

* Identifying sensitive information.
* Making application development staff aware of sensitive information and where it is used.

Application Developers are responsible for:

* Understanding and defining the security requirements for sensitive data.
* Understanding each phase of the software development lifecycle and the security controls required at each phase.
* Documents security controls required at each phase.

C. Secure Software Development Lifecycle Procedures

Specific security requirements shall be followed at each phase of the software development lifecycle.

1. Feasibility

The Chief Security Officer shall review requests that deal with sensitive information to ensure that proper security controls are considered during the feasibility study.

2. Risk assessment

The Risk Management Officer shall perform a preliminary risk assessment to ensure all material risks have been identified. The risk assessment shall include a sensitivity assessment that evaluates the sensitivity and criticality of the information to be collected, stored, and processed. The risk assessment shall consider privacy issues, compliance requirements, and system continuity.

3. User requirements

Systems Analysts shall review system requirements and prepare user requirements, including the need to implement security controls that maintain information confidentiality, integrity, and availability.

4. Systems design

Systems Analysts will design the proposed system, incorporating security controls such as identification and authentication, access rights, logging and monitoring systems, encryption controls, and other specific security methodologies.

5. Programming or acquire software (development/acquisition)

Applications shall be designed in accordance with industry accepted security standards (i.e., OWASP for web applications) and comply with applicable regulatory and business requirements. Programmers and software developers shall code applications using secure coding techniques that validate input fields, encrypt sensitive data, restrict access by need to know, etc. Data input and output integrity routines (i.e., reconciliation and edit checks) shall be implemented for application interfaces and databases to prevent manual or systematic processing errors or corruption of data. If the software was acquired, the IT Department shall ensure the application meets security requirements and, as much as possible, adheres to internal security development standards.

Developers shall be trained in secure coding techniques, including how to avoid common coding vulnerabilities, and understanding how sensitive data is handled in memory. Applications shall be based on secure coding guidelines (see Open Web Application Security Project) that protect against attacks (e.g. injection, access controls, information disclosure, buffer overflow, input validation, cross site scripting, etc.), error handling, and other related secure coding techniques.

For public-facing web applications, ABC Company shall address new threats and vulnerabilities on an on-going basis and ensure these applications are protected against known attacks by:

* Reviewing public-facing web applications via manual or automated application vulnerability security assessment tools or methods, at least annually and after any changes.
* Installing an automated technical solution that detects and prevents web-based attacks (for example, a web-application firewall) in front of public-facing web applications, to continually check all traffic.
* Periodic independent audits performed annually or after a major software upgrade/release. See Audit Policy for more information.

6. Quality assurance

Quality evaluation and acceptance criteria for information systems, upgrades, and new versions shall be established, documented and tests of the system(s) shall be carried out both during development and prior to acceptance to maintain security. The development team must ensure that software security features are properly coded and enabled.

7. Documentation

Documentation shall include procedures to restrict access to systems based upon prior authorization.

8. Systems testing and acceptance

Test data shall be carefully selected, protected, and controlled. Management shall have a clear oversight capacity in the quality testing process with the final product being certified as:

* Fit for purpose - the product should be suitable for the intended purpose, and
* Right first time - mistakes should be eliminated prior to release.

Processes shall test security functionality prior to software release with security testing performed under conditions as close to production conditions as possible.

Data Owners shall review test results and provide approval prior to moving new software or software updates into production. Data Owners shall also review and approve data migrations from one application or system to another.

9. Installation

There shall be a separation between the production, development, and test environments. This ensures that security is rigorously maintained for the production system, while the development, and test environments maximize software development productivity.

* Software development staff must not be permitted to have access to production systems and related data.
* Development systems must not contain sensitive or confidential information.
* All application-program-based access paths other than the formal user access paths must be deleted or disabled before software is moved into production.
* Software debugging code must be removed before programs are placed into production.

IT Department’s operations staff will move programs from development into production. Software developers shall not be permitted to move programs into the production environment.

The IT Department’s operations staff shall complete any required security activities (logging, anti-malware scanning, encryption security certificates, etc.). Additional actions required can include changes to backup procedures, user training, work flows, etc.

10. Maintenance through disposal

Access to program source code shall be restricted to authorized personnel on a need to know basis. Procedures shall be developed to control the risks related to systems development access to production systems during emergencies (program aborts, etc.).

Upon software end of life and final disposition of the application, the following procedures shall be followed:

* Data Owner gives permission to terminate the application.
* IT department identifies retention requirements for software and data and ensures appropriate backups are stored in a secure location.
* IT Department operations staff securely removes or destroys the application and related code.
* IT Department operations staff ensures secure disposal of hardware and software according to the Disposal Policy.
* IT Department operations staff ensures disposition of licensed software meets requirements of the software license or other relevant agreements.

# V. Enforcement

Any Staff found to have violated this policy may be subject to disciplinary action, up to and including termination.

# VI. Distribution

This policy is to be distributed to IT Staff that provide and manage software development and programming services.

**Policy History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Description | Approved By |
| 1.0 | 1/1/20XX | Initial policy release |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**References:**

COBIT APO01.03, APO01.11, APO05.05, APO12.07, APO13.07, BAI03.05, MEA02.11

GDPR Article 25, 32

HIPAA 164.308(a)(1)(i), 164.308(a)(1)(ii)(B), 164.308(a)(2)

ISO 27001:2013 5.2, 6.2, A.14

NIST SP 800-37 3.2, 3.4

NIST SP 800-53 PL-2, PL-7, PL-8, SA-3, SA-4, SA-15, SA-17

NIST Cybersecurity Framework ID.AM-6, ID.GV-3-4, ID.RA-6, PR.IP-2-3, PR.MA-1

PCI 6.1-7