Software Development 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 solid security policies, blocking unnecessary access to source code, implementing testing procedures, and managing the transfer of software from development to production helps mitigate ABC Company’s software development risks.

# II. Purpose

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

# III. Scope

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

# IV. Policy

A. Software Development Life Cycle (SDLC)

The IT Director 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

The SDLC methodology ensures that the software will be adequately documented and tested before it is used for critical business processes and storing of information.

B. Access Controls

All production systems must have an access control system to restrict who can access the system as well as restrict the privileges available to these Users. A designated access control administrator (who is not a regular User on the system in question) must be assigned for all production systems.

Software protection mechanisms shall be developed to protect the confidentiality, integrity, and availability of critical assets and sensitive data.

There shall be a separation between development, testing, and operational environments to reduce the risks of unauthorized access or changes to the operational environment. 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.

Access to program source code shall be restricted to authorized personnel.

Prior to moving software into production:

* All application-program-based access paths other than the formal user access paths must be deleted or disabled.
* Software debugging code must be removed.
* Test user IDs and passwords must be removed.
* Custom code shall be reviewed prior to release to identify any potential coding vulnerability. Secure coding practices shall be appropriate to the programming language and development environment. Code changes shall be reviewed by individuals other than the originating code author and by individuals knowledgeable about code-review techniques and secure coding practices. Such review to ensure code is developed according to secure coding guidelines and appropriate corrections are implemented prior to release. Code review results are reviewed and approved by IT management prior to release. The requirement for code reviews applies to all custom code (both internal and public-facing), as part of the system development life cycle. Code reviews shall be conducted by knowledgeable internal personnel or third parties. Public-facing web applications are also subject to additional controls, to address on-going threats and vulnerabilities after implementation.

The 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.

ABC Company’s IT Department must perform periodic risk assessments of production systems to determine whether the controls employed are adequate. Procedures shall be developed controlling the risks related to systems development access during emergencies (program aborts, etc.). Refer to the Risk Assessment Policy for more information.

C. Data Owners

All production systems must have designated Owners and Custodians for the critical information they process.

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.

D. Quality assurance

The development of all software shall be supervised and monitored by the organization and must include security requirements, independent security review of the environment by a Certified Information Systems Auditor, certified security training for software developers, and code reviews.

Applications shall be securely designed, coded, and maintained per the Secure Software Development Lifecycle Policy. 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.

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.

Quality assurance procedures shall include systematic monitoring and evaluation of software developed, outsourced, or acquired by ABC Company. 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. Refer to the Quality Assurance Policy for more information.

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.

Static and dynamic analysis tools shall be used to verify that secure coding practices are being adhered to for internally developed software:

* Static testing - application data and control paths shall be modeled and then analyzed for security weaknesses. Static analysis shall test the internal structure of the application, rather than functional testing.
* Dynamic analysis of a program/application shall occur when it is in operation.

Procedures shall control the risks related to production software and hardware changes that may include applications, systems, databases and network devices requiring patches, service packs, and other updates and modifications. This includes:

* Separate development/test environments from production environments and enforced accesses controls.
* Separation of duties between development/test and production environments.
* Production data (e.g. live Primary Account Numbers, Names, etc.) is not used for testing or development.
* Where appropriate, processes exist for the removal of test data and accounts before production systems become active.
* Change control procedures for the implementation of security patches and software modifications include: Documentation of impact, change approval by authorized parties, functionality testing to verify that the change does not adversely impact the security of the system, and back-out procedures.

E. Software Security

The Chief Security Officer (CSO) shall ensure secure software requirements:

* Staff implement secure software lifecycle management practices.
* Processes exist to identify, assess, and manage threats and vulnerabilities to the software.
* Secure software releases and updates are provided in a timely manner.
* Appropriate personnel provide clear and thorough guidance on the secure implementation, configuration, and operation of the software.

The CSO shall ensure secure software lifecycle requirements:

* A formal software security governance program establishes responsibility and authority for the security of software products and services. Resources are allocated to execute the strategy and ensure that personnel are appropriately skilled. Staff define, maintain, and communicate a software security policy and a strategy for ensuring the secure design, development, and management of software products and services. Performance against the software security strategy shall be monitored and tracked.
* Staff shall continuously identify, assess, and manage risk to the software and services. Staff shall detect and mitigate vulnerabilities in the software and its components to ensure that software remains resistant to attacks throughout its entire lifetime.
* Staff shall identify and manage software changes throughout the software lifecycle. Staff shall protect the integrity of the software throughout the software lifecycle. The confidentiality, integrity, and availability of Sensitive Information shall be maintained.
* Staff shall provide clear and thorough guidance on the secure implementation, configuration, and operation of software applications. Staff shall maintain communication channels with Data Owners regarding potential security issues and mitigation options. Staff shall provide Data Owners with detailed explanations of all software changes.

# 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 the Chief Security Officer and all Staff that develop, implement, acquire, or manage software development.

**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, PCI Software Security Framework