Software Development Life Cycle Policy

Doorway

SOC 2 Criteria: CC8.1
ISO 27001 Annex A: A.6.1.2, A.6.1.5, A.12.1.1, A.12.1.2, A.12.1.4, A.12.5.1, A.14.2.1, A.14.2.2, A.14.2.3, A.14.2.4, A.14.2.5, A.14.2.6, A.14.2.7, A.14.2.8

Keywords: Change management, OWASP, Software lifecycle

Purpose

This policy defines the high-level requirements for providing business program managers, business project managers, technical project managers, and other program and project stakeholders guidance to support the approval, planning, and life-cycle development of Doorway software systems aligned with the Information Security Program.

Roles and Responsibilities

Doorway's Security Officer is responsible for updating, reviewing, and maintaining this policy.

Policy

Doorway must establish and maintain processes for ensuring that its computer applications or systems follow an SDLC process which is consistent and repeatable, and maintains information security at every stage.

Software Development Phases and Approach Standard

A Software Development Project consists of a defined set of phases:

Determine System Need Phase

The Determine System Need phase is the period of time in which an information system need is identified and the decision is made whether to commit the necessary resources to address the need.

Define System Requirements Phase

The Define System Requirements phase is the period in which the User Requirements are broken down into more detailed requirements which can be used during designing and coding. Applicable security requirements and controls will be identified through an information security risk assessment.

Design System Component Phase

The Design System Components phase transforms requirements into specifications to guide the work of the Development phase. The decisions made in this phase address how the system will meet the functional, physical, interface, data, and security requirements. Design phase activities may be conducted in an iterative fashion, producing a system design that emphasises the functional features of the system and technical detail.

Build System Component Phase

The Build phase transforms the detailed system design into complete coded software units and eventually, into an integrated product for release. Each software unit and subsequent integrated units are tested thoroughly, to include tests for security vulnerabilities. System documents that support installation and operations are also developed in this phase.

Evaluate System Readiness Phase

The evaluation phase ensures that the system, as designed and built, satisfies the requirements of the user, as well as applicable security requirements. Whenever possible, independent testers measure the system's ability to perform the functions that are required by the customer and ensure an acceptable level of quality, performance, and security. Once the phase is complete, it will be evident whether or not the system is ready for operation or redevelopment.

System Deployment Phase

System Deployment phase is the final phase of the development life cycle, when the system is released initially to a pilot site, where any further security vulnerabilities can be identified, and then into the production environment. All necessary training for using the system is accomplished.

Project Management

The sequence of the development phases depends on the software development approach taken. The project management approaches include but are not limited to:

- Waterfall Development
- Agile Development
- Iterative Development

- Staged Delivery Development

Based on the approach for and the size of the software development, some of the phases can be combined. In Iterative Development there may be multiple Cycles (iterations) of the above phases before the final software is released.

Secure System Engineering Principles

Information security implications will be addressed and reviewed regularly, and responsibilities for information security will be defined and allocated to the roles defined in the project management methods.

Engineering principles for a secure system will fall under the following categories:

- Business layer
    - Establish a sound security policy as the 'foundation' for design.
    - Treat security as an integral part of the overall system design.
    - Clearly delineate the physical and logical security boundaries governed by associated security policies.
    - Ensure that developers are trained in how to develop secure software.
    - Assume that external systems are insecure.
    - Implement 'least privilege' to provide no more authorisations than necessary to perform required functions.
    - Strive for simplicity.
    - Develop and exercise contingency or DR procedures to ensure appropriate availability.
- Data Layer
    - Processed lawfully, fairly, and in a transparent manner in relation to individuals.
    - Use secure methods (SSL/TLS) for exchanging information over a network.
- Application
    - Authenticate users and processes to ensure appropriate access control.
    - Data sharing with a users' associated organisation will be limited wherever possible.
    - Minimise vulnerable surface areas.
- Technology
    - Only open-source and state-of-the-art hardware and network infrastructure provided by select vendors should be used.
    - Security measures at 3rd party vendors must form a key part of the vendor evaluation criteria, and must be reviewed periodically.

SDLC Security Control Guidelines

The SDLC process will adhere to the following information security controls:

- Adequate procedures should be established to provide separation of duties in the origination and approval of source documents. This shall include but not be limited to separation of duties between Personnel assigned to the development/test environment and those assigned to the production environment.

- Modification of code or an emergency release will follow the change control standard.
- Secure programming standards should be followed. Secure code training should be provided to Doorway's developers.
- Secure development environment will be created, based on:
    - sensitivity of data to be processed, stored and transmitted by the system;
    - applicable external and internal requirements, e.g. from regulations or policies;
    - security controls already implemented by the organisation that support system development;
    - trustworthiness of personnel working in the environment;
    - the degree of outsourcing associated with system development;
    - the need for segregation between different development environments;
    - control of access to the development environment;
    - monitoring of change to the environment and code stored therein;
    - backups are stored at secure offsite locations; and,
    - control over movement of data from and to the environment.
- All software deployed on Corporate or Hosted infrastructure must prevent security issues including but not limited to those covered by SAN and OWASP.
- Code changes are reviewed by individuals other than the originating code author and by individuals who are knowledgeable in code review techniques and secure coding practices.
- Overrides of edit checks, approvals, and changes to confirmed transactions should be appropriately authorised, documented, and reviewed.
- Application development activity should be separated from the production and test environments. The extent of separation, logical or physical, is recommended to be appropriate to the risk of the business application or be in line with customer contractual requirements. The level of separation that is necessary between production, development, and test environments should be evaluated and controls established to secure that separation.
- All changes to production environments should strictly follow change control procedures, including human approval of all changes, granted by an authorised owner of that environment. Automated updates should be disallowed without such approval.
- Active production environments should not be re-used as test environments. Inactive and/or decommissioned production environments should not be used as test environments unless all private data has been removed. Test environments should not be re-used as production environments without going through a decommissioning and recommissioning process that cleans all remnants of test data, tools, etc.
- Individuals who are responsible for supporting or writing code for an internet-facing application, or internal application that utilises web technology and handles customer information, should complete **annual** security training specific to secure coding practices. For individuals supporting or writing code for an internet-facing application, training should also include topics specific to internet threats. The individual should complete the training prior to writing or supporting the code. The training must include OWASP secure development principles as well as OWASP top 10 vulnerability awareness for the most recent year available.
- Custom accounts and user IDs and/or passwords should be removed from applications before applications become active or are released to customers.
- Production data should not be used in testing or development environments.

- Security controls that are in place for the production copy in the test system should be production quality (e.g. mirroring the production controls over the data).
- When conducting quality assurance (QA) testing prior to the release of a new feature requiring user input where constraints on user input may be reasonably understood, feature acceptance tests must include testing of edge and boundary cases.

For situations demonstrating that testing needs to use production data, the requirements are the following:

- The Information Resource Owner will provide approval before production data can be used for testing purposes.
- Wherever possible, the production data should be tokenised or anonymised instead of using production data.
- Testing and parallel runs should use a separate copy of production data and the test location or destination should be acceptable (e.g. loading confidential production data to a laptop for testing is not acceptable).
- The data should not be extracted, handled, or used by the test process in a manner that subjects the data to unauthorised disclosure.
- The data should be accessed on a need-to-know basis.
- Normal test activities should not use production data. In cases where test activity requires access to production data, access to production data should be restricted to only those individuals who have a documented business need. Only the information with the documented business need should be accessible by those users.
- Production data used for testing should be securely erased upon completion of testing.
- Test data and accounts will be removed before being placed into production.
- Restricted/Protected Information will be encrypted according to the Encryption Standard while at rest or in transit.
- Error messages must be handled securely and they must not leak sensitive information.

Change Management

Software Installation and Change on Operational Systems

- Operating system applications and software will only be implemented after extensive and successful testing. The tests will cover:
    - Usability
    - Security
    - Effects on other systems
    - User- friendliness
    - Tests will be conducted on separate systems (test environment), and all corresponding program source libraries will also be updated, as appropriate.
- The operational software, applications, and program libraries of Doorway will only be updated by trained administrators upon appropriate management authorisation.
- Company operational systems will only hold approved executable code, not development code or compilers.

- A configuration control system will be used to keep control of all implemented software as well as the system documentation.
    - Previous versions of software will be retained as a contingency measure.
    - Old versions of software will be archived, together with all required information and parameters, procedures, configuration details and supporting software for as long as the data are retained in archive.
- There will be a rollback strategy in place before changes are implemented.
- An audit log will be maintained of all updates to operational program libraries.
- All decisions to upgrade to a new version release must take into account:
    - Business requirements for the change
    - Security of the release, e.g. the introduction of new information security functionality or the number and severity of information security problems affecting this version.

Change Control Procedures

- A record of agreed authorisation levels will be maintained.
- Changes are only submitted by authorised users.
- Controls and integrity procedures will be reviewed to ensure that they will not be compromised by the changes.
- All software, information, database entities and hardware that require amendment will be identified.
- Security critical code to minimise the likelihood of known security weaknesses will be identified and checked.
- Formal approval must be obtained for detailed proposals before work begins.
- Authorised users must accept changes prior to implementation.
- Changes will be implemented at a time that is least intrusive to business processes involved.
- Vendor-supplied software will be used without modification; in the event that a modification is necessary, the following will be evaluated:
    - Risk of compromising built-in controls and integrity processes
    - Vendor consent
    - Getting the modifications from vendor as standard updates
    - Impact of owning the responsibility for maintaining the program
    - Compatibility with other software in use.
- A technical review of applications will be conducted after changes to operating platforms (operating systems, databases and middleware platforms). The review will include:
    - Application control and integrity procedures to ensure that they have not been compromised by the operating platform changes.
    - Timely notification of operating platform changes to allow appropriate tests and reviews to take place before implementation.
    - Appropriate changes are made to the business continuity plans.

Revision History

Version	Date	Editor	Description of Changes
1.0	17.03.2022	Henry Sinclair	Initial Creation
2.0	05.08.2022	Hugh Fraser	Roles and Responsibilities Change