CAYLENT

Idea to Impact. Faster.

Software Development Lifecycle Policy

Purpose

This policy defines the requirements of Caylent's software life-cycle development process for internal software used by Caylent to deliver services to its Managed Services customers and ensure conformity with Caylent's Information Security Program.

Software Under Policy

This policy applies to all internal Caylent software that, when deployed in production, has access to, processes, or stores customer data, or controls access to that data. Software within this scope is tagged with "soc2" in Caylent's GitLab code repository.

Standardized Phases

Note: These phases are required depending on the stage of development and may be revisited as required (e.g., feature requests, bug fixes, etc.).

1. Validate business need.
2. Define system requirements and acceptance criteria with stakeholders, including security requirements.
3. Design and architect.
4. Build.
5. Validate against acceptance criteria.
6. Deploy and operationalize.

Project Management Principles

Timely meetings will be held with stakeholders. User acceptance and acceptance criteria are used to determine system readiness and quality assurance. Proper documentation must be available from the technical (e.g., architecture diagrams) and user (e.g., user guides) perspectives.

Secure System Engineer Principles

Information security implications will be addressed and reviewed regularly, and responsibilities for information security will be defined and allocated to defined roles, in accordance with this policy.

Engineering principles for a secure system fall into the following categories:

1. **Data Layer**:
    a. Data will be stored according to Caylent's Information Security Policies.
    b. Data access will be restricted to least privilege at both application and infrastructure levels.
2. **Applications**:
    a. Data in transit will be encrypted according to Caylent's Information Security Policies.
    b. All logs for custom applications are sent to CloudWatch, S3, or Datadog.
3. **Technology**:
    a. Default to cloud-native AWS managed services.
    b. No OS layer in architecture unless absolutely necessary.

Security Control Guidelines

1. Automated tools are used to scan for vulnerabilities in code and dependency vulnerabilities in development, staging, and production environments, in accordance with Vulnerability Management Policy.
2. Findings are remediated in accordance with Vulnerability Management Policy.
3. Developers are trained on the following standards for secure coding
    a. OWASP Top Ten
    b. CIS AWS Foundation Benchmark standard
    c. NIST 800-53

Change Management

Caylent uses GitLab for the software development lifecycle. All of the below procedures are specific to GitLab and implemented with GitLab security features.

1. Access to code is governed by Caylent's Information Security Policies.
2. Changes are only accepted by authorized users.
3. Main branch will be a "protected" branch.
4. Merge Requests will need at least 1 approval from a Maintainer (in GitLab) other than the author.
5. Tags are used on the main branch for production and may only be carried out by authorized users. Tag patterns starting with "v" are "protected" to control production releases to authorized users.
6. Tags will be used for production, optionally including an additional push-button release approval (after the tag) and/or a manual change set execution.
7. Code is reviewed by a second member of the internal development team.
8. Only authorized users are allowed to tag branches to release to production.
9. Before changes go to production the Deployer's Responsibility Checklist will be completed.
10. Any emergency changes (hotfixes) will still use a limited version of the Deployer's Responsibility Checklist.
11. Enforcement of these requirements is audited using a custom-built script on a regular basis, but at least annually.

Roles and Responsibilities

1. Roles:
    a. Security Team Member
    b. Internal Development Manager
    c. Internal Development Team Member
    d. Developer
2. Responsibilities:

a. A Security Team Member will review with the Internal Development Manager the processes and procedures used by the Internal Development Team to ensure the security practices are aligned with Trek10's security best practices. The Security Team Member will also help inform the Internal Development team of security issues that need to be addressed and the level priority the issue(s) should be addressed with, pursuant to Vulnerability Management Policy.

b. Internal Development Manager will ensure security best practices are enforced when making changes to the software or the processes and procedures followed by the Internal Development team.

c. Developers across Trek10 will follow the processes and procedures defined in this policy while performing the changes to code and systems that this policy applies to.

Revision History

Original publication: December 2025