SDLC Procedure

Version 1 - Approved by Youssef Ouyhya

Contents

1. Objective

This procedure discusses the objectives, the roles, the process flow, and the artifacts required in a typical software development life cycle. It provides an overview of how the process is adopted throughout various stages of a product and provides guidelines for secure development activities within CNTXT.

2. Scope

This document focuses on the development process for the CNTXT products. This document is to be followed by all CNTXT staff members, sub-contractors, and partners involved in software product development.

3. Engineering Team Responsibilities

- The Engineering Team shall be responsible for:
  - Following the guidelines in this document.
  - Tracking bugs/weaknesses/tools (as applicable).
  - Assisting third-party vendors in conducting Vulnerability Assessments/Penetration Testing.
- The Engineering Head shall be responsible for:
  - Analyzing new technology for security risks and known attack patterns.
  - Ensuring all members of the Engineering Team read and understand the process flow and follow the guidelines.

4. Process Description

4.1 Security Requirements for Information Systems

4.1.1 Information Security Requirements Analysis & Specification

The main objective of defining a standard process for product development is to ensure cost-effective and timely development, delivery, and deployment of a high-quality, differentiated product that brings tangible and sustained value for CNTXT customers. The following objectives are measured to ensure the effectiveness of the process:

- High Quality:
  - Software produced with less defect count in all development, testing, and delivery cycles.
  - Metrics: Measure and utilize information on defects/module, defects by severity, defects by priority, and defects by different phases of development.
- On-time delivery:
  - Ability to meet the specified delivery dates in every release.
  - Metrics: Measure the percentage of progress against planned task hours spent in every iterative phase of a product release.

4.2 Software Development Life Cycle

The Software development life cycle shall include all of the listed stages - Requirement Gathering, Design, Development, Testing, Implementation, Operations, and Maintenance.

4.2.1 Requirement Gathering Stage

- For each new feature being planned to any software or for changes to existing features, the high-level requirements should be documented by the Product team in consultation with business users. The output of the requirement gathering will form inputs for the Design team.
- Where required, it is the responsibility of the Product team to ensure the high-level requirements are detailed into low-level/functional requirements to better facilitate the subsequent tasks of design and development.

4.2.2 Design Stage

- The Design team is responsible to detail the UI and UX flows for the requirements shared by the Product team.
- These details are important inputs to the Engineering team to plan out their development activities.

4.2.3 Development Stage

- The inputs from the Product team and Design team for the basis for planning of the development stage.
- Team leads in the engineering team are responsible to break down the requirements into tasks and ensure all engineers are aware of the tasks assigned to them.
- The engineers are responsible to follow securing coding practices for development. They should also ensure to use the latest code libraries and fix any code-level vulnerabilities.
- All development activities should be facilitated by appropriate version control systems to ensure the latest version of code is used for development.
- It is the responsibility of the Engineering Head and Infosec officer to ensure the development environment is segregated from the staging and production environment

4.2.4 Testing Stage

- All features shall be tested after the development is done.
- The testing should be performed by any user apart from the developer to ensure segregation of duties.
- It is recommended to have the testing to ensure the software is working as intended.
- The Engineering Head and Information Security Officer must ensure that testing happens in a staging environment which is different from the development environment and production environment.
- Only after adequate testing is completed, the deployment process should commence.

4.2.5 Deployment Stage

- Any deployment should follow the change management process. It is essential that all deployments are tracked via a version control system or ticketing tool.
- All deployments/changes must be approved before merging with production code by an independent engineer who is different from the author to ensure segregation of duties.

4.2.6 Maintenance Stage

- The engineering team must facilitate the reporting of any errors or bugs identified by the end users.
- All such errors/bugs must be tracked.
- Any changes related to bug fixes must follow the change management process.

4.3 Security in Development & Support Processes

4.3.1 Implementing Change Management & Vulnerability Management Procedure

- Change management procedure shall be followed for changes to existing systems or the introduction of new systems. Any change made to the system is tested before implementing it. A list of changes made is recorded. Records of testing, updates, and results are documented.
- New releases/patches pertaining to the production server shall be tested before being implemented in the production environment to ensure that there is no adverse impact on operation, application controls, or security. In case of any exceptions due to technical limitations, approval shall be taken from the Engineering Head or respective team leads.
- Where feasible, automated scanners are used to identify code-level or network-level vulnerabilities, and fixing such vulnerabilities must follow the vulnerability management procedure.
- Pull requests/ change requests need to be reviewed by a peer or managers prior to merging the pull requests.
- The application functionalities shall be reviewed to ensure that they have not been compromised by the platform changes (as applicable).
- Previous version(s) of the software shall be retained as a contingency measure in case a rollback is required.

4.3.2 Restrictions on Changes to Software Packages

Changes or modifications for vendor-supplied software packages shall be adequately controlled and limited to personnel involved in the implementation of the change/modification based on peer approvals.

4.4 Test Data

It is the responsibility of product, design, and engineering teams to avoid using any PII data to perform testing. Testing must always be performed using dummy data.

5. Document Security Classification

Company Internal (please refer to the Data Classification policy for details).

6. Non-Compliance

Compliance with this policy shall be verified through various methods, including but not limited to automated reporting, audits, and feedback to the policy owner. Any staff member found to be in violation of this policy may be subject to disciplinary action, up to and including termination of employment or contractual agreement. The disciplinary action shall depend on the extent, intent, and repercussions of the specific violation.

7. Responsibilities

The Information Security Officer is responsible for approving and reviewing policy and related procedures. Supporting functions, departments, and staff members shall be responsible for implementing the relevant sections of the policy in their area of operation.

8. Schedule

This document shall be reviewed annually and whenever significant changes occur in the organization.

End of Software Development Lifecycle Procedure. For version history, please see the next page.

Version history

Version	Log	Date
1 Current	Policy version approved by Youssef Ouyhya	26 Oct, 2024
1	New Policy version Created	26 Oct, 2024