Software Development Life Cycle (SDLC) Policy

Approved by	Signature	Effective Date
Security Team	Official Security Team Approved	03/11/2025

1. Purpose

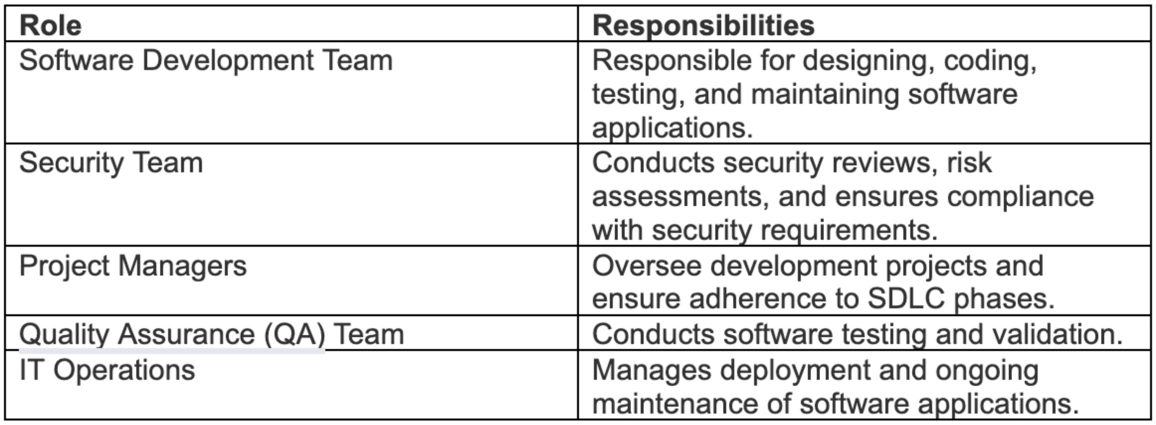
This policy establishes a structured framework for software development at DanfeCorp to ensure security, quality, and compliance throughout the Software Development Life Cycle (SDLC). It provides guidelines for secure coding practices, risk management, and adherence to regulatory and industry standards.

2. Scope

This policy applies to all software development projects, including internally developed applications, third-party software integrations, and outsourced development efforts. It is applicable to all developers, IT personnel, and project managers involved in software development at DanfeCorp.

The Software Development Life Cycle (SDLC) is a structured process that enables the creation of high-quality, cost-effective software in a predictable timeframe. By following a standardized approach to software development, DanfeCorp aims to minimize security vulnerabilities, ensure consistent quality, and maintain compliance with industry regulations. This policy outlines the key phases, responsibilities, and security requirements that must be integrated throughout the SDLC process.

3. Roles and Responsibilities



Role	Responsibilities
Software Development Team	Responsible for designing, coding, testing, and maintaining software applications.
Security Team	Conducts security reviews, risk assessments, and ensures compliance with security requirements.
Project Managers	Oversee development projects and ensure adherence to SDLC phases.
Quality Assurance (QA) Team	Conducts software testing and validation.
IT Operations	Manages deployment and ongoing maintenance of software applications.

4. SDLC Phases

DanfeCorp follows a structured SDLC framework comprising the following phases:

Planning
Requirements Gathering

Design
Architecture & Specifications

Development
Coding & Implementation

Testing
Quality Assurance

Deployment
Release & Maintenance

4.1 Planning and Requirements Gathering

- Define business objectives and project scope.
- Identify security and compliance requirements.
- Conduct risk assessments and feasibility analyses.

4.2 Design

- Establish software architecture and design specifications.
- Implement security-by-design principles.
- Define data flows, encryption, and authentication mechanisms.

4.3 Development

- Follow secure coding best practices (e.g., OWASP Top 10).
- Implement access controls and logging mechanisms.

- Use version control systems for source code management.

4.4 Testing and Quality Assurance

- Conduct functional, unit, integration, and user acceptance testing.
- Perform security testing (e.g., static/dynamic analysis, penetration testing).
- Validate compliance with regulatory and security standards.

4.5 Deployment

- Ensure software passes all security and quality checks before deployment.
- Use automated deployment pipelines with rollback capabilities.
- Monitor system performance and conduct post-deployment reviews.

4.6 Maintenance and Support

- Apply software patches and updates in a timely manner.
- Monitor for vulnerabilities and security incidents.
- Conduct periodic software reviews and audits.

5. Secure Development Practices

Practice	Description	Implementation
Secure Coding Guidelines	Standards for writing secure code	All developers must follow OWASP secure coding practices and company-specific guidelines
Least Privilege Access	Limiting access rights to minimum necessary	Developers should only have access to resources required for their specific role
Dependency Management	Tracking and updating external libraries	Regular audits and updates of all dependencies to address known vulnerabilities
Security Training	Education on security best practices	Mandatory periodic security awareness training for all development teams

Implementing secure development practices is essential to minimize vulnerabilities in software applications. DanfeCorp emphasizes a "security-first" approach by integrating security considerations from the earliest stages of development through to deployment and maintenance. By following established frameworks and guidelines, developers can create robust applications that protect both company and customer data.

6. Change Management

- All code changes must be reviewed and approved before deployment.

- Maintain an audit trail for all modifications.
- Ensure rollback procedures are in place for critical systems.

7. Third-party and Open-Source Software Management

- Evaluate third-party libraries for security and compliance risks.
- Regularly monitor and update third-party dependencies.
- Ensure licensing agreements comply with DanfeCorp's legal requirements.

8. Policy Review and Updates

This policy shall be reviewed annually or upon significant changes to software development practices, regulatory requirements, or the security landscape. The security team must approve all updates.

This policy document supersedes all previous software development guidelines and becomes effective as of the date specified above. All departments must align their operations with these requirements within 30 days of the effective date.

For questions or clarifications regarding this policy, contact the Security Team via secure channels only.