International Standard

ISO/IEC 26585

Information technology — Cybersecurity — Secure software development framework

First edition
2026-11

Reference number
ISO/IEC 26585:2026(en)

© ISO/IEC 2026

Contents

Page

Foreword v
Introduction vii
1 Scope 1
2 Normative references 1
3 Terms and definitions and abbreviations 1
4 Background 2
4.1 Approach 2
4.2 SSDF Use 3
4.3 The Secure Software Development Framework 3
5 Prepare the Organization 4
5.1 Define Security Requirements for Software Development 4
5.1.1 Identify and document security requirements for software development infrastructures and processes 4
5.1.2 Identify and document security requirements for developed software 5
5.1.3 Communicate requirements 5
5.2 Implement Roles and Responsibilities 5
5.2.1 Update role definitions 5
5.2.2 Provide training 5
5.2.3 Obtain and convey management commitment 5
5.3 Implement Supporting Toolchains 5
5.3.1 Specify tools 5
5.3.2 Secure tools and toolchains 5
5.3.3 Configure tools to generate artifacts 5
5.4 Define and Use Criteria for Software Security Checks 6
5.4.1 Define security check criteria 6
5.4.2 Gather and safeguard information for security checks 6
5.5 Implement and Maintain Secure Environments for Software Development 6
5.5.1 Separate and protect environments 6
5.5.2 Secure and harden endpoints 6
6 Protect the Software 6
6.1 Protect All Forms of Code from Unauthorized Access and Tampering 6
6.1.1 Securely store code 6
6.2 Provide a Mechanism for Verifying Software Release Integrity 6
6.2.1 Share software integrity verification information 7
6.3 Archive and Protect Each Software Release 7
6.3.1 Retain software releases 7
6.3.2 Track software provenance data 7
7 Produce Well-Secured Software 7
7.1 Design Software to Meet Security Requirements and Mitigate Security Risks 7
7.1.1 Model the security risk 7
7.1.2 Track design decisions 7
7.1.3 Use standardized security features and services 7
7.2 Review the Software Design to Verify Compliance with Security Requirements and Risk Information 8
7.2.1 Validate the software design 8
7.3 Reuse Existing, Well-Secured Software When Feasible Instead of Duplicating Functionality 8
7.3.1 Maintain well-secured third-party software components 8
7.3.2 Create and maintain well-secured in-house software components 8

© ISO/IEC 2026 – All rights reserved

7.3.3 Verify security of third-party software components 8
7.4 Create Source Code by Adhering to Secure Coding Practices 8
7.4.1 Follow appropriate secure coding practices 8
7.5 Configure the Compilation, Interpreter, and Build Processes to Improve Executable Security 8
7.5.1 Choose compiler, interpreter, and build tools 9
7.5.2 Configure compiler, interpreter, and build tools 9
7.6 Review and/or Analyze Human-Readable Code to Identify Vulnerabilities and Verify Compliance with Security Requirements 9
7.6.1 Determine when to review or analyze code 9
7.6.2 Review and analyze code 9
7.7 Test Executable Code to Identify Vulnerabilities and Verify Compliance with Security Requirements 9
7.7.1 Determine when and how to test code 9
7.7.2 Test code 9
7.8 Configure Software to Have Secure Settings by Default 9
7.8.1 Define a secure baseline 10
7.8.2 Implement and document the secure baseline 10

8 Respond to Vulnerabilities 10
8.1 Identify and Confirm Vulnerabilities on an Ongoing Basis 10
8.1.1 Investigate vulnerability reports 10
8.1.2 Evaluate software for vulnerabilities 10
8.1.3 Implement vulnerability disclosure and remediation 10
8.2 Assess, Prioritize, and Remediate Vulnerabilities 10
8.2.1 Analyze vulnerabilities 10
8.2.2 Implement risk responses 10
8.3 Analyze Vulnerabilities to Identify Their Root Causes 11
8.3.1 Determine root causes 11
8.3.2 Identify patterns with root causes 11
8.3.3 Eradicate similar vulnerabilities 11
8.3.4 Update the SDLC process 11

Annex A (informative) Implementation Examples 12

Bibliography 25

© ISO/IEC 2026 – All rights reserved

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by INCITS (as INCITS 585-2025) and drafted in accordance with its editorial rules. It was adopted, under the JTC 1 PAS procedure, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

© ISO/IEC 2026 – All rights reserved

Foreword (This foreword is not part of American National Standard INICTS 585-2025.)

The Secure Software Development Framework describes a core set of high-level secure software development practices that can be integrated into any software development life cycle (SDLC) implementation. Following such practices should help software producers reduce the number of vulnerabilities in released software, reduce the potential impact of the exploitation of undetected or unaddressed vulnerabilities, and address the root causes of vulnerabilities to prevent future recurrences. Because the framework provides a common vocabulary for secure software development, software acquirers can also use it to foster communications with suppliers in acquisition processes and other management activities. This standard was developed by Technical Committee Cybersecurity and Privacy of Accredited Standards Committee INCITS during 2023-2024. The standards approval process started in 2023.

Requests for interpretation, suggestions for improvements or addenda, or defect reports are welcome. They should be sent to the INCITS Secretariat, Information Technology Industry Council, 700 K St NW, Suite 600, Washington, DC 20001.

This standard was processed and approved for submittal to ANSI by the InterNational Committee for Information Technology Standards (INCITS).

© ISO/IEC 2026 – All rights reserved

Introduction

A software development life cycle (SDLC) is a formal or informal methodology for designing, creating, and maintaining software (including code built into hardware). There are many models for SDLCs, including waterfall, spiral, agile, and agile combined with software development and IT operations (DevOps) practices. Regardless of which SDLC model is used, secure software development practices should be integrated throughout it for three reasons: to reduce the number of vulnerabilities in released software, to reduce the potential impact of the exploitation of undetected or unaddressed vulnerabilities, and to address the root causes of vulnerabilities to prevent recurrences. Vulnerabilities include not just bugs caused by coding flaws, but also weaknesses caused by security configuration settings, incorrect trust assumptions, and outdated risk analysis.

Most aspects of security can be addressed multiple times within an SDLC, but in general, the earlier in the SDLC that security is addressed, the less effort and cost is ultimately required to achieve the same level of security. This principle, known as shifting left, is critically important regardless of the SDLC model. Shifting left minimizes any technical debt that would require remediating early security flaws late in development or after the software is in production. Shifting left can also result in software with stronger security and resiliency.

© ISO/IEC 2026 – All rights reserved

Information technology — Cybersecurity — Secure software development framework

1 Scope

This document provides a framework for secure software development practices.

2 Normative references

Bibliography

[1] NIST SP 800-218 Secure Software Development Framework (SSDF)

© ISO/IEC 2026 – All rights reserved