

REQUIREMENTS · ARCHITECTURE · DEVELOPMENT · RELEASE · OPERATIONS

From requirement to controlled release.

How security is integrated throughout the software engineering lifecycle.

DOCUMENT ID	VERSION	CLASSIFICATION	DOCUMENT OWNER	REVIEW CYCLE	PUBLISHED
ITO-TRUST-SDL-001	1.0	PUBLIC	Engineering	Annual	October 2026

Secure software starts with understanding

Secure software is not created through security testing alone.

Security begins with understanding what a system is expected to do, what information it processes, who will use it, which systems it communicates with and what could happen if its confidentiality, integrity or availability is compromised.

ITO therefore integrates security considerations throughout the software engineering lifecycle. Our approach combines requirements analysis, architecture, secure development practices, review, testing, controlled deployment and continuous maintenance.

This document provides a high-level overview of that process.

SECURITY BEGINS BY ASKING

- What should the system do?
- What information does it process?
- Who will use it?
- Which systems does it talk to?
- What if C, I or A is compromised?



This document intentionally does not disclose sensitive internal procedures, security configurations or product-specific implementation details.

CONTENTS

01	Purpose	02
02	Our engineering lifecycle	03
03	Requirements & risk	04
04	Secure architecture	05
05	Secure development	06
06	Code review & change control	07
07	Security verification	07
08	Third-party dependencies	08
09	Controlled release & deployment	08
10	Security after release	09
11	Continuous improvement	10
12	Shared responsibility	10
13	Development assurance	11
14	Our engineering principle	11

DOCUMENT CONTROL

DOCUMENT ID	ITO-TRUST-SDL-001
VERSION	1.0
CLASSIFICATION	PUBLIC
OWNER	Engineering
REVIEW CYCLE	Annual
PUBLISHED	October 2026

Eight stages, one continuous loop



Security is not a final testing phase.
It is an **engineering consideration**
throughout the lifecycle.

Understanding before building

Secure development begins before source code is written.

ITO engineering teams first seek to understand the operational environment in which the software will function. Relevant considerations may include:

Users?

Who will access the system and with what responsibilities?

Information?

What information will be processed and how sensitive is it?

Integrations?

Which internal and external systems will communicate with the solution?

Availability?

What are the operational consequences of downtime?

Auditability?

Which activities must be traceable?

Infrastructure?

Where and how will the solution operate?

Governance?

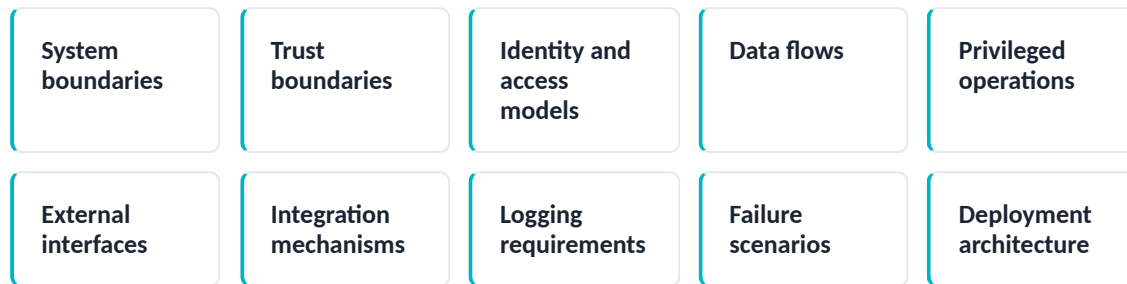
Which institutional, contractual and regulatory requirements affect the system?

Security requirements can then become part of the solution requirements rather than an additional layer introduced after implementation.

Security starts with architecture

Many security properties of a system are determined by architectural decisions made before implementation begins.

ITO SOLUTION ARCHITECTURE MAY THEREFORE CONSIDER



The objective is to **reduce unnecessary exposure** and establish clear boundaries between users, services, information and external systems.

Defence in depth

ITO favours layered security architecture rather than reliance on a single security mechanism. Depending on the solution, controls may exist across:



Failure or compromise of one layer should not automatically provide unrestricted access to the broader system.

Building with security in mind

During implementation, developers are expected to apply secure engineering practices appropriate to the technology and application being developed.

THESE MAY INCLUDE

01 Input and output validation

02 Appropriate authentication and authorisation

03 Secure handling of sensitive information

04 Controlled error handling

05 Protection against common application vulnerabilities

06 Secure configuration practices

07 Management of third-party dependencies

08 Separation of sensitive configuration from application code

Relevant OWASP guidance may be used as a reference for application-security risks and secure development practices.

Secrets and configuration

AVOIDED

Credentials, API keys, certificates and other sensitive configuration unnecessarily embedded within application source code.

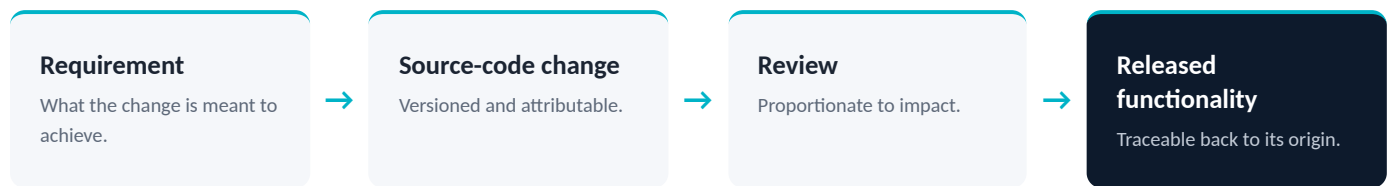
INSTEAD

Sensitive configuration should be handled according to the architecture and deployment model of the relevant solution.

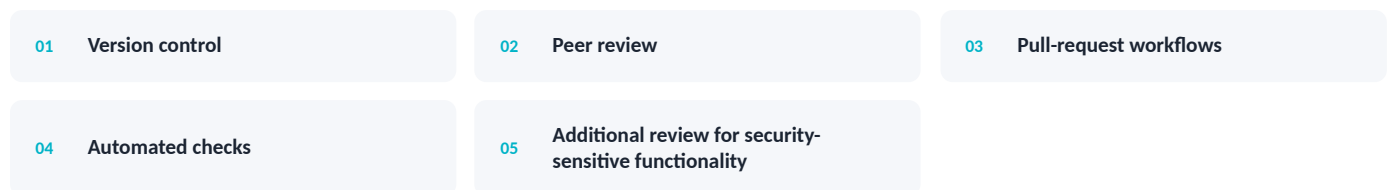
Changes should be reviewable and traceable

Software development requires controlled change.

ITO development workflows should provide appropriate traceability between software requirements, source-code changes and released functionality.



CODE CHANGES MAY BE SUBJECT TO

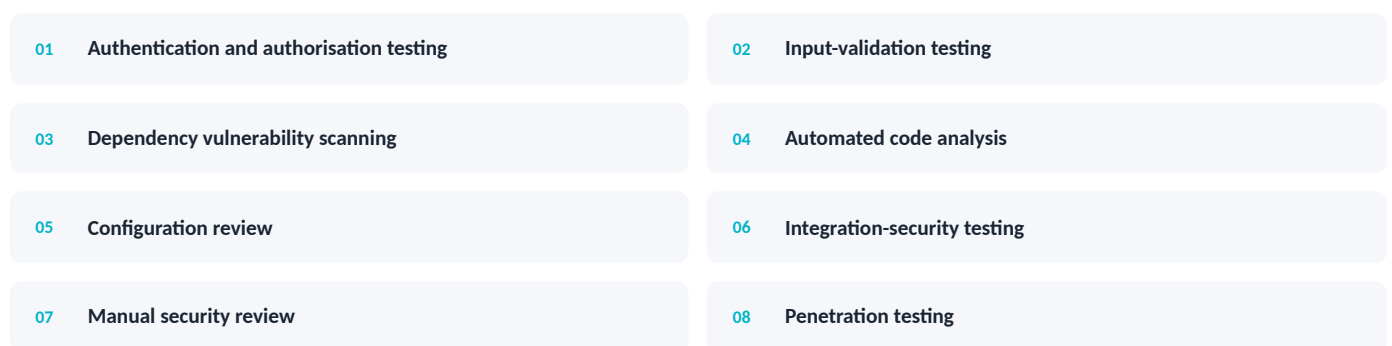


The level of review depends on the nature, complexity and potential impact of the change.

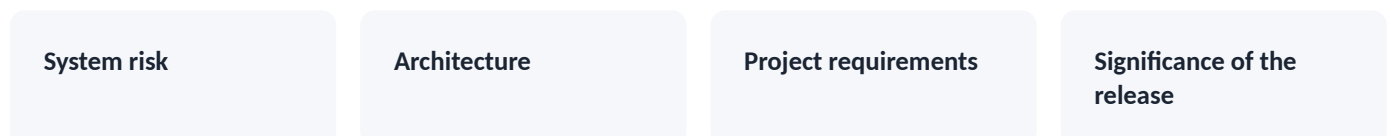
Verify before release

Security verification complements functional and quality testing.

Depending on the solution, release and risk profile, verification activities may include:



THE DEPTH AND COMBINATION OF ACTIVITIES ARE DETERMINED BY



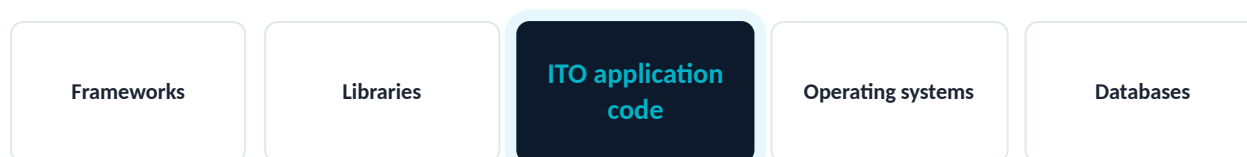
Modern software has a supply chain

Modern applications depend on frameworks, libraries, operating systems, databases and other third-party technologies.

These dependencies can introduce vulnerabilities even when ITO-developed application code has not changed.

Relevant third-party components should therefore be considered throughout development and maintenance.

Where security vulnerabilities affecting relevant dependencies are identified, their applicability and potential impact should be assessed and appropriate remediation considered.



From source code to production

Production software should be introduced through controlled release and deployment processes. Depending on the project model, controls may include:



Production environments should be treated differently from development environments because they may contain **operational information** and support **live institutional processes**.

Software security continues in production

A secure release can become vulnerable over time.

New vulnerabilities may be discovered in application frameworks, dependencies, infrastructure components or technologies on which the solution depends. ITO therefore treats maintenance as part of the security lifecycle. Activities may include:

01

Vulnerability assessment

Evaluating vulnerabilities relevant to supported software.

02

Dependency updates

Updating affected components where appropriate.

03

Security fixes

Developing corrective releases for confirmed vulnerabilities within ITO-maintained components.

04

Operational feedback

Using incidents, defects and customer feedback to improve future releases.

05

Lifecycle management

Considering support status and security implications of underlying technologies.

Engineering processes evolve

Secure development is not a static checklist. Engineering practices must evolve as technologies, threats, vulnerabilities, regulatory expectations and operational requirements change. ITO therefore uses lessons learned from development, testing, production operations, security assessments and customer environments to improve its engineering practices over time.

A secure development lifecycle is not defined by never finding vulnerabilities.

It is defined by the ability to **identify, understand, remediate and learn** from them.

12 • SHARED RESPONSIBILITY

ITO develops and maintains software within an ecosystem that may also include customer-managed infrastructure, external systems and third-party technologies. Secure operation therefore requires coordination between:

ITO

Application architecture, engineering, product security and maintenance within the agreed scope.

CUSTOMER INSTITUTION

Users, permissions, governance, operational procedures and customer-controlled infrastructure.

INFRASTRUCTURE PROVIDERS

Hosting, network, operating systems and infrastructure controls where separately managed.

TECHNOLOGY PARTNERS

Security and lifecycle of independently provided products and services.

Detailed responsibilities depend on the specific deployment and contractual arrangement.

Information for technical due diligence

Customers and partners may require additional information about ITO engineering processes during technical or supplier due diligence.

Where appropriate, ITO may provide additional documentation concerning secure development, security testing, architecture, vulnerability management and release processes.

Sensitive internal procedures and technical information may be provided through controlled assurance channels where justified.

FOR ADDITIONAL INFORMATION

[ITO Security Whitepaper](#)[ITO Responsible Vulnerability Disclosure Policy](#)[ITO Privacy & Data Governance](#)[ITO Responsible AI Principles](#)

14 • OUR ENGINEERING PRINCIPLE

Security is not something we add to finished software.

It influences how we understand requirements, design architecture, write code, verify changes, deploy systems and maintain them over time.

OUR OBJECTIVE

To build software that is not only functional at delivery, but **maintainable**, **auditable** and **capable of evolving securely** throughout its operational lifecycle.



ITO TRUST DOCUMENTS

Engineering Digital Trust.

RELATED DOCUMENTS

ITO-TRUST-SEC-001

ITO Security Whitepaper

ITO-TRUST-PRV-001

ITO Privacy & Data Governance

ITO-TRUST-AI-001

ITO Responsible AI Principles

DOCUMENT

ITO Secure Development Overview

ID · VERSION · CLASSIFICATION

ITO-TRUST-SDL-001 · 1.0 · PUBLIC

OWNER · REVIEW

Engineering · Annual · October 2026

CONTACT

security@ito.dev · ito.dev/en/trust-center

COMPANY

IT Odjel Mostar d.o.o., Mostar, Bosnia and Herzegovina