scalefocus

Sofia | 2025

Secure Software Development

Overview and practical examples

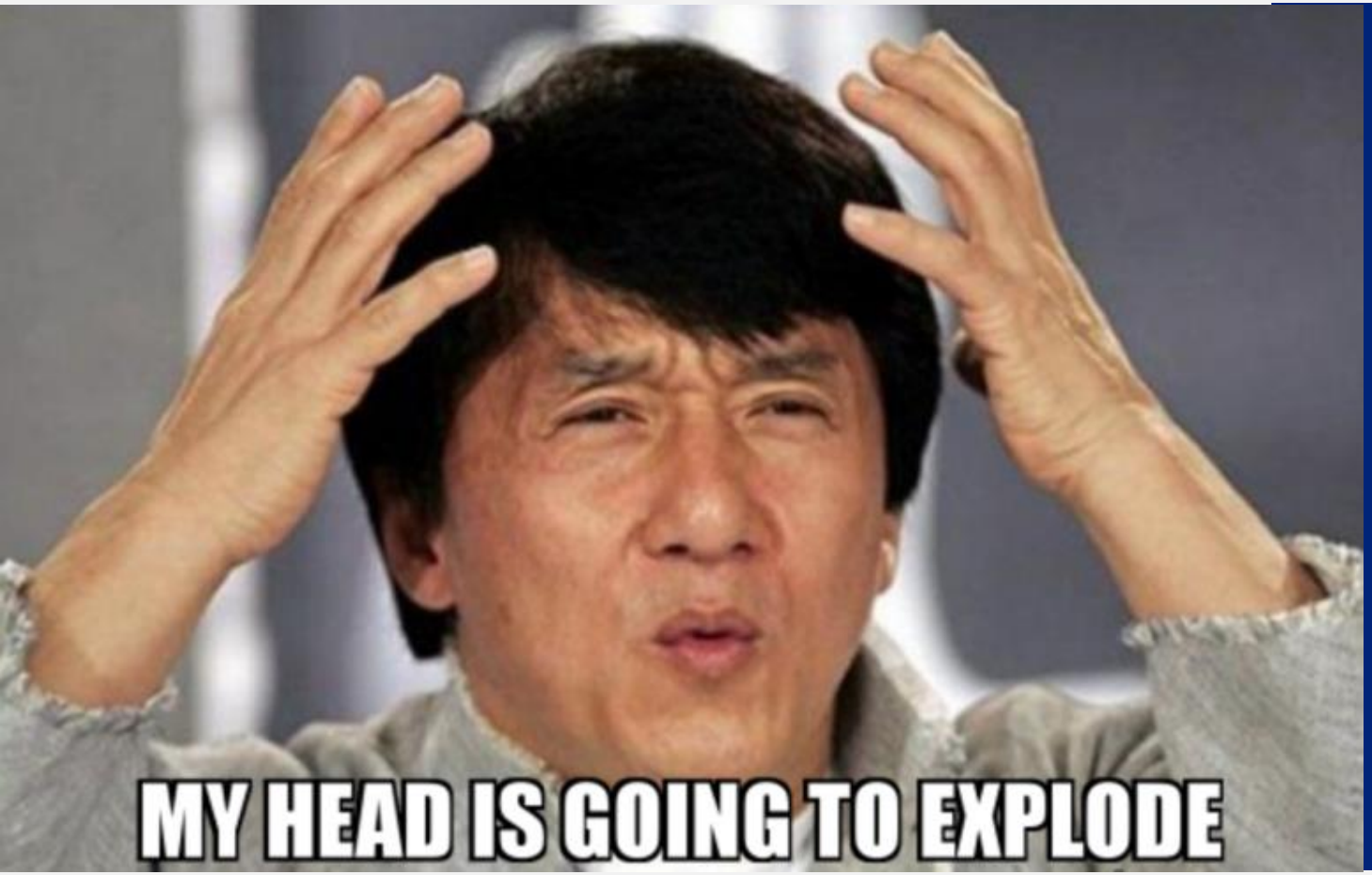
About me

Technology Team Lead & Manager Security Business in Scalefocus.

Mostly interested in SSDLC & Application Security, GRC, Security Automation & Offensive Security.

Social:
radostina.kondakova@protonmail.com
LinkedIn

Security Frameworks & Guides



NIST CSF 2.0,
NIST SSDF,
OWASP SAMM,
BSIMM,
ISO 27001,
COBIT,
MITRE ATT&CK,
CIS Controls,
CSA CCM,
Cloud Control Matrix,
SAFECode,
STIGs, XCCDF, SCAP
... and many many more.

How do we choose the right one for us?

- Business objectives and risk appetite.
- Regulatory compliance requirements.
- Industry standards and best practices.
- Existing security controls and infrastructure.
- Budget and resources.
- Internal expertise and capabilities.

Avoid

- Lack of executive sponsorship.
- Lack of stakeholder buy-in.
- Overlooking key requirements.
- Poor communication and collaboration.
- Failure to adapt to changing threats and risks.


(SSDLC) Secure Software Development
Life Cycle (==) Application Security OR (!=)?

1. Secure Software Development Lifecycle (SSDLC)

Process framework that embeds security practices and controls at every stage of software development to ensure secure code and systems.

Examples: OWASP SAMM, Microsoft SDL, SAFECode.

2. Training & Support

Structured approach to managing and improving the security of applications throughout their lifecycle by integrating security practices into the SDLC.

Examples: NIST SSDF, OWASP SAMM, BSIMM.

3. Cybersecurity

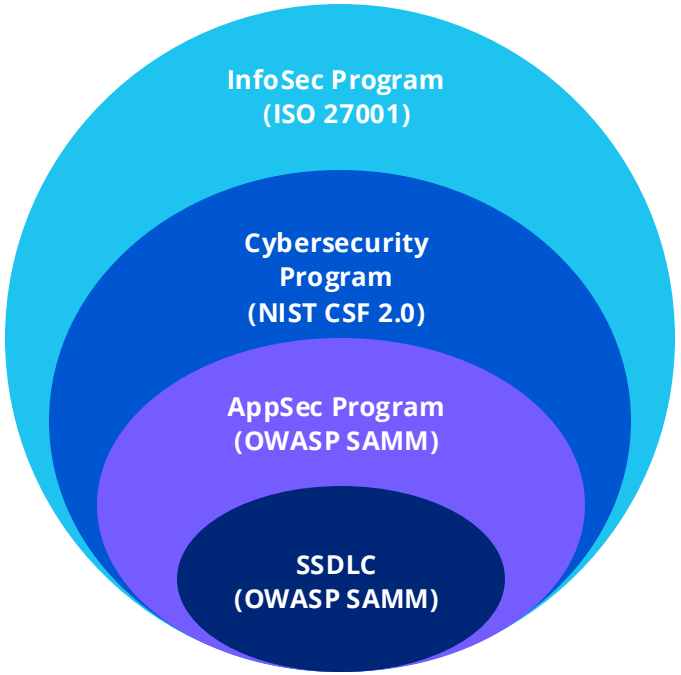
Comprehensive strategy for managing and mitigating security risks across an organization's digital infrastructure, including networks, systems, and applications.

Examples: NIST CSF, Cybersecurity Fundamentals.

4. Information Security

Broad initiative to protect an organization's information assets—both digital and physical—through policies, controls, and risk management practices.

Examples: ISO 27001, COBIT.



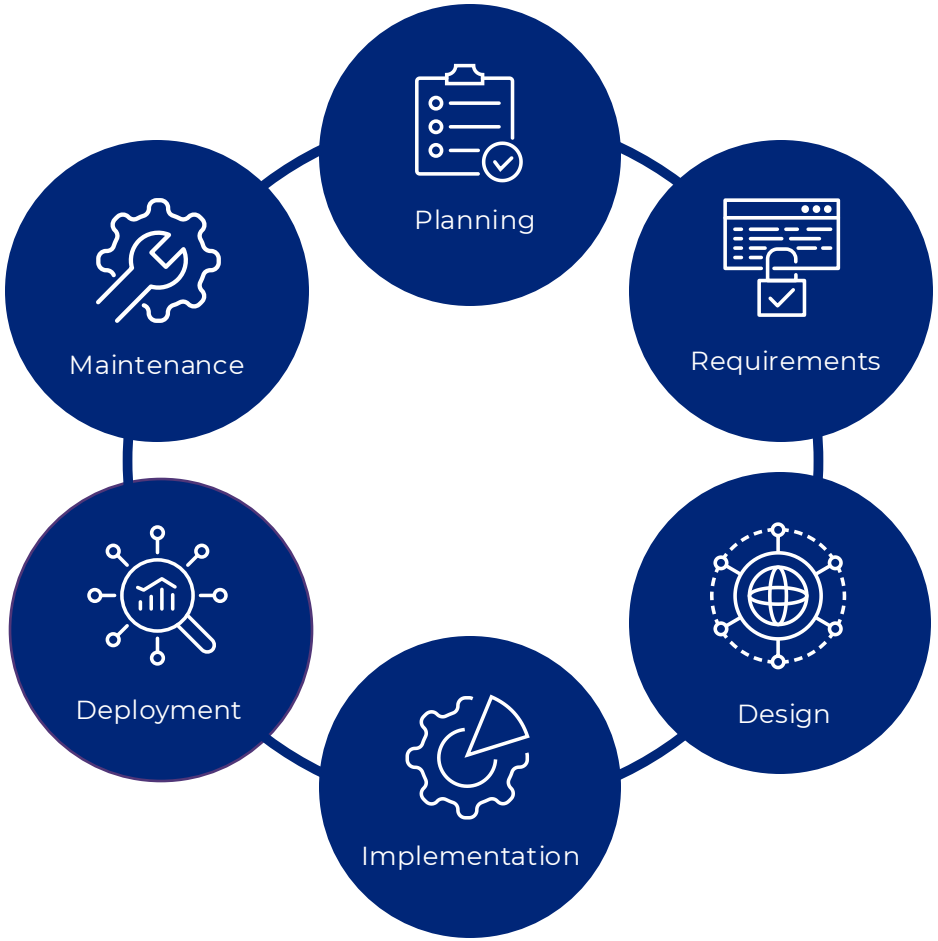
Secure **framework** ontology

Secure Software Development Process example

- Patch Management
- Policy & Document Maintenance
- Continues Assessment & Monitoring

- Vulnerability Assessment
- Penetration testing
- Runtime Security

- Secure Code Practices & Reviews
- Security Hardening
- Secrets Detection
- **DevSecOps**
- SAST & DAST
- **SCA**



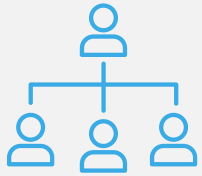
- Security Requirements
- Compliance & business objectives
- Budget
- **Standards & Frameworks**

- Technical Security Requirements
- Map Security & Privacy Requirements
- Risk analysis

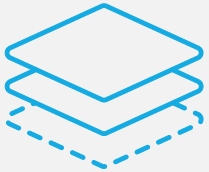
- Secure Design Principles
- Security controls/gates
- **Threat Modelling**
- **Policies & Procedures Enablement**
- Access Control Lists
- Security Coding Guide
- Secret Management

Application Security Programs: A Focused List

BSIMM (Building Security In Maturity Model):
Observational framework based on practices used in large organizations.



OWASP SAMM (Software Assurance Maturity Model):
Maturity model for proactively improving application security across teams.



NIST SSDF (Secure Software Development Framework):
Government-backed framework providing guidelines for integrating security into the software development lifecycle.

BSIMM

DOMAINS			
GOVERNANCE	**INTELLIGENCE**	**SSDL TOUCHPOINTS**	**DEPLOYMENT**
Practices that help organize, manage, and measure a software security initiative. Staff development is also a central governance practice.	Practices that result in collections of corporate knowledge used in carrying out software security activities throughout the organization. Collections include both proactive security guidance and organizational threat modeling.	Practices associated with analysis and assurance of particular software development artifacts and processes. All software security methodologies include these practices.	Practices that interface with traditional network security and software maintenance organizations. Software configuration, maintenance, and other environment issues have direct impact on software security.

PRACTICES			
GOVERNANCE	**INTELLIGENCE**	**SSDL TOUCHPOINTS**	**DEPLOYMENT**
1. Strategy & Metrics (SM) 2. Compliance & Policy (CP) 3. Training (T)	4. Attack Models (AM) 5. Security Features & Design (SFD) 6. Standards & Requirements (SR)	7. Architecture Analysis (AA) 8. Code Review (CR) 9. Security Testing (ST)	10. Penetration Testing (PT) 11. Software Environment (SE) 12. Configuration Management & Vulnerability Management (CMVM)

OWASP SAMM

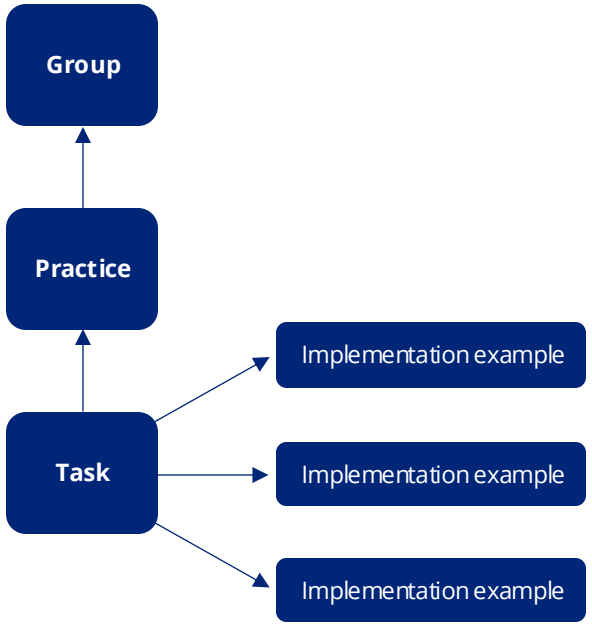
Governance		Design		Implementation		Verification		Operations	
Strategy and Metrics		Threat Assessment		Secure Build		Architecture Assessment		Incident Management	
Create and Promote	Measure and Improve	Application Risk Profile	Threat Modeling	Build Process	Software Dependencies	Architecture Validation	Architecture Mitigation	Incident Detection	Incident Response
Policy and Comliance		Security Requirements		Secure Deployment		Requirements-Driven Testing		Environment Management	
Policy and Standards	Compliance Management	Software Requirements	Supplier Security	Deployment	Secret Management	Control Verification	Misuse/Abuse Testing	Configuration Hardening	Patch and Update
Education and Guidance		Secure Architecture		Defect Management		Security Testing		Operational Management	
Training and Awareness	Organization and Culture	Architecture Design	Technology Management	Defect Tracking	Metrics & Feedback	Scalable Baseline	Deep Understanding	Data Protection	Legacy Management

NIST SSDF

SSDF organizes security practices into 4 groups

Practice are the central entities in SSDF

Each practice defines one or more tasks to fulfill it

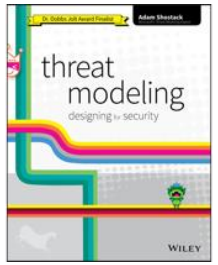


Main differences between BSIMM, OWASP SAMM and NIST SSDF

Aspect	BSIMM	OWASPSAMM	NIST SSDF
Model Type	Descriptive Observational Model	Prescriptive Framework	Prescriptive High-level Framework
Licensing	Commercial	Open Source	Public Domain
Approach	Descriptive: Documents observed practices	Prescriptive: Provides actionable steps	Prescriptive: Recommends practices and outcomes
Focus Area	Software Security Initiatives (SSI)	Software Security Assurance	Secure Software Development Lifecycle (SDLC)
Target Audience	Mature organizations	Organizations at all maturity levels	Universal (all sizes and industries)
Cost	Paid	Free	Free
Community Involvement	Low (Proprietary)	High (Community-driven)	Moderate (Government-backed)

Threat Modelling

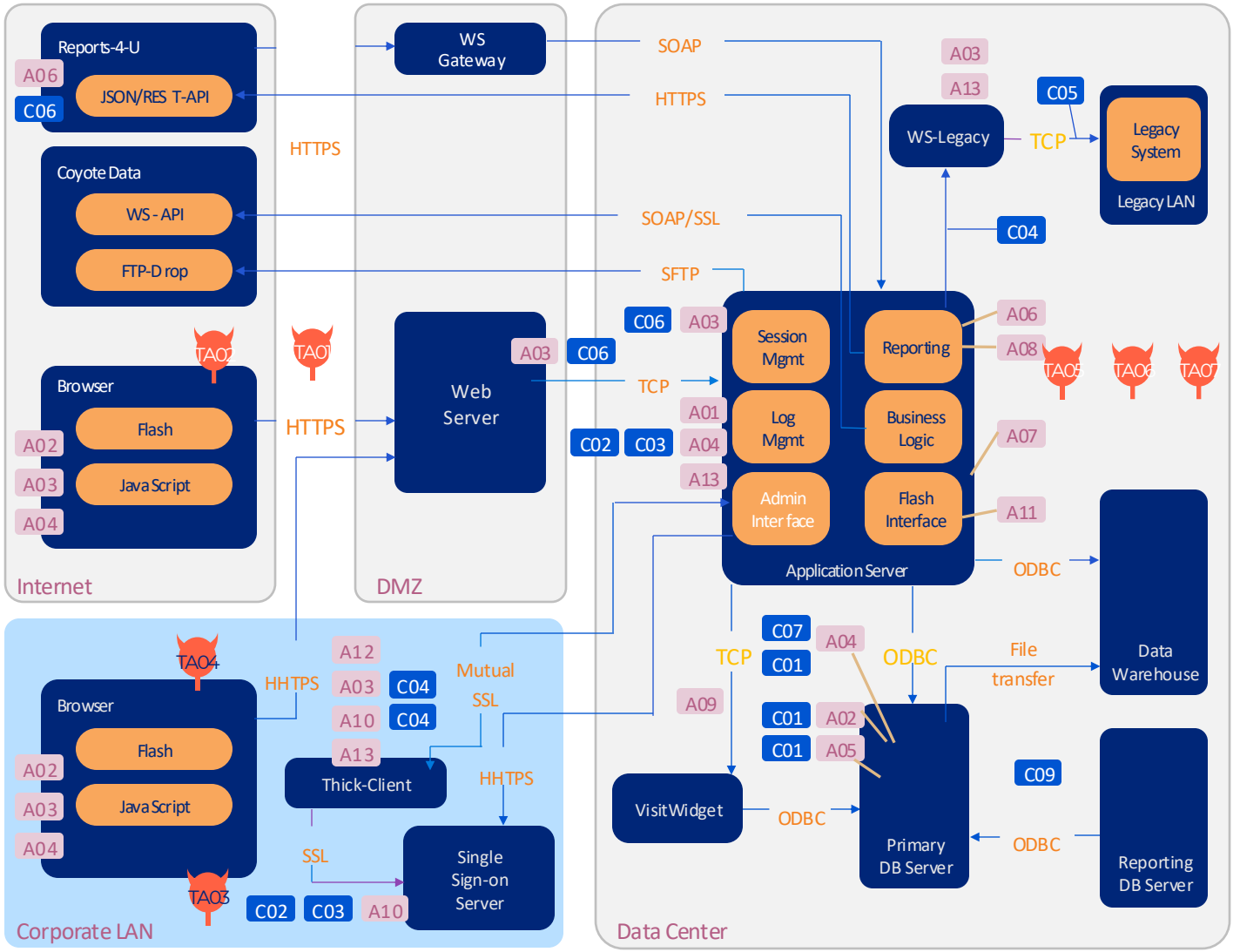
The process of using hypothetical scenarios, system diagrams, and testing to help secure systems and data. By identifying vulnerabilities, helping with risk assessment, and suggesting corrective action, threat modelling helps improve cybersecurity and trust in key business systems.



Book

Game

Tool



Policies and procedures enablement

And if this does not work.
Go back to **Principle 1**

- Agree on the processes which will be followed during development
- Educate the team on best coding practices, tools, frameworks & processes
- Cultivate a growth mindset
- Create a software security initiative (SSI)
- Think of ways to promote security:
  - Security Champions
  - Bug Bounty Programs
  - Capture the Flag
  - Security Games

SSDLC – Implementation

SCA

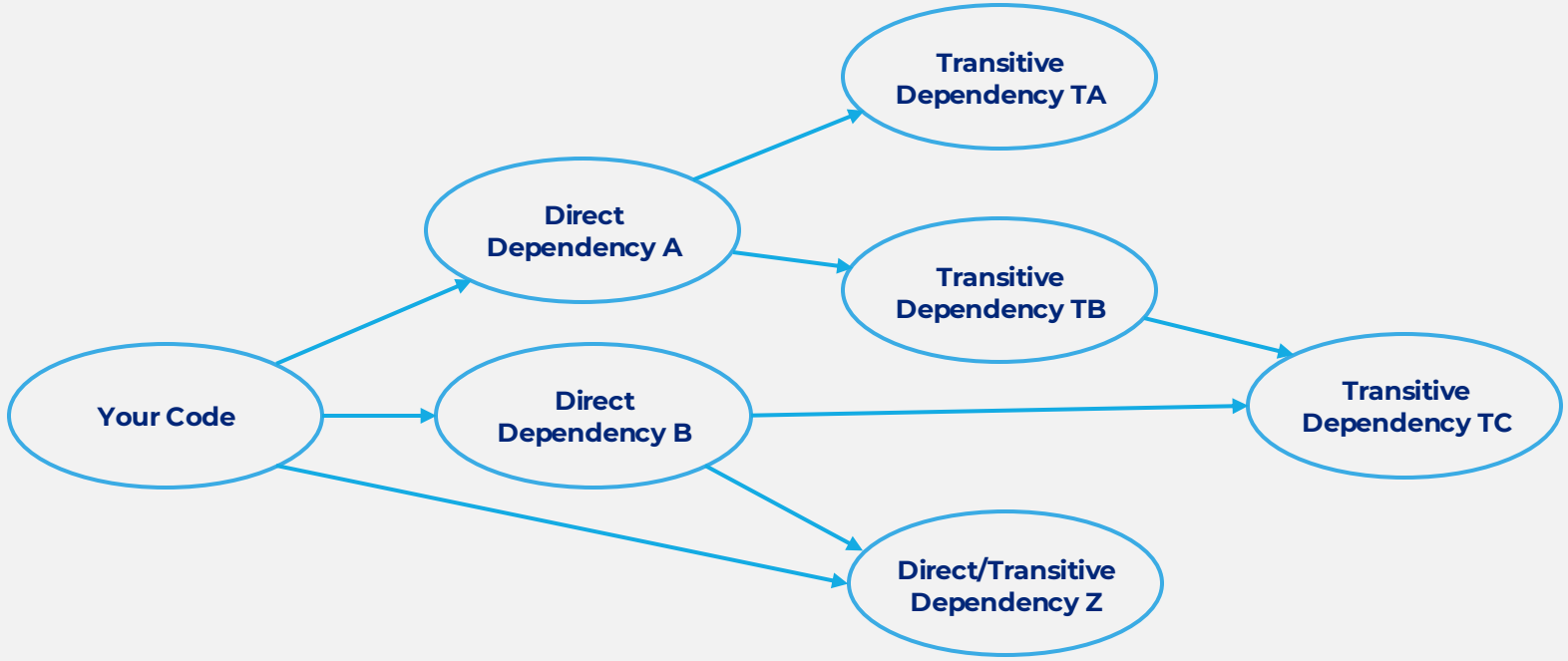
Software Composition Analysis

The process of using hypothetical scenarios, system diagrams, and testing to help secure systems and data. By identifying vulnerabilities, helping with risk assessment, and suggesting corrective action, threat modelling helps improve cybersecurity and trust in key business systems.

SBOM

Software Bill Of Materials: a list of software artifact components and metadata. This information can also include licensing information, persistent references, and other auxiliary information

CycloneDX- by OWASP
SPDX: *The Software Package Data Exchange*
SWID: *Software Identification Tags*



https://cyclonedx.org/tool-center/

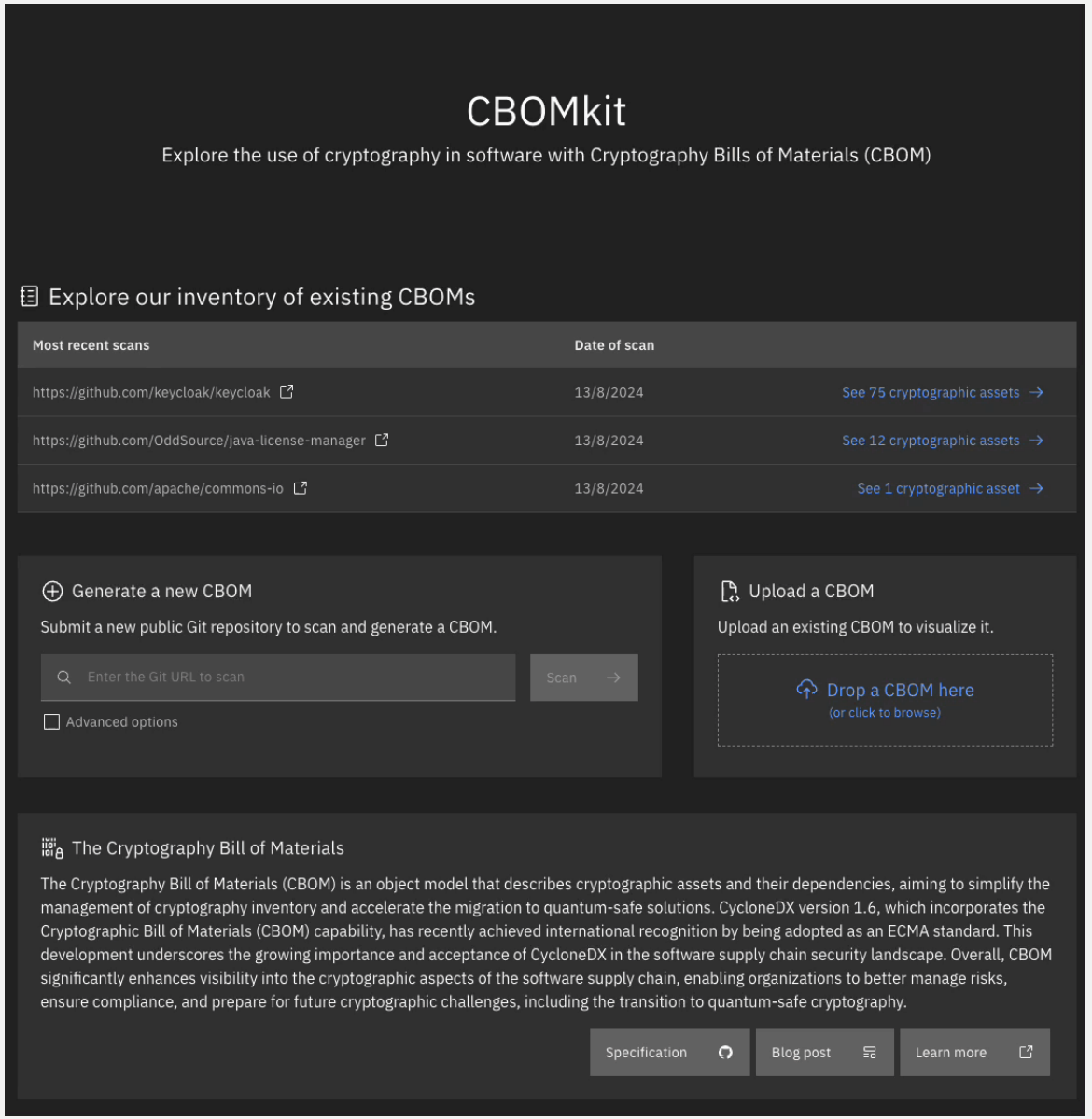
SSDLC – Implementation: SCA

CBOM

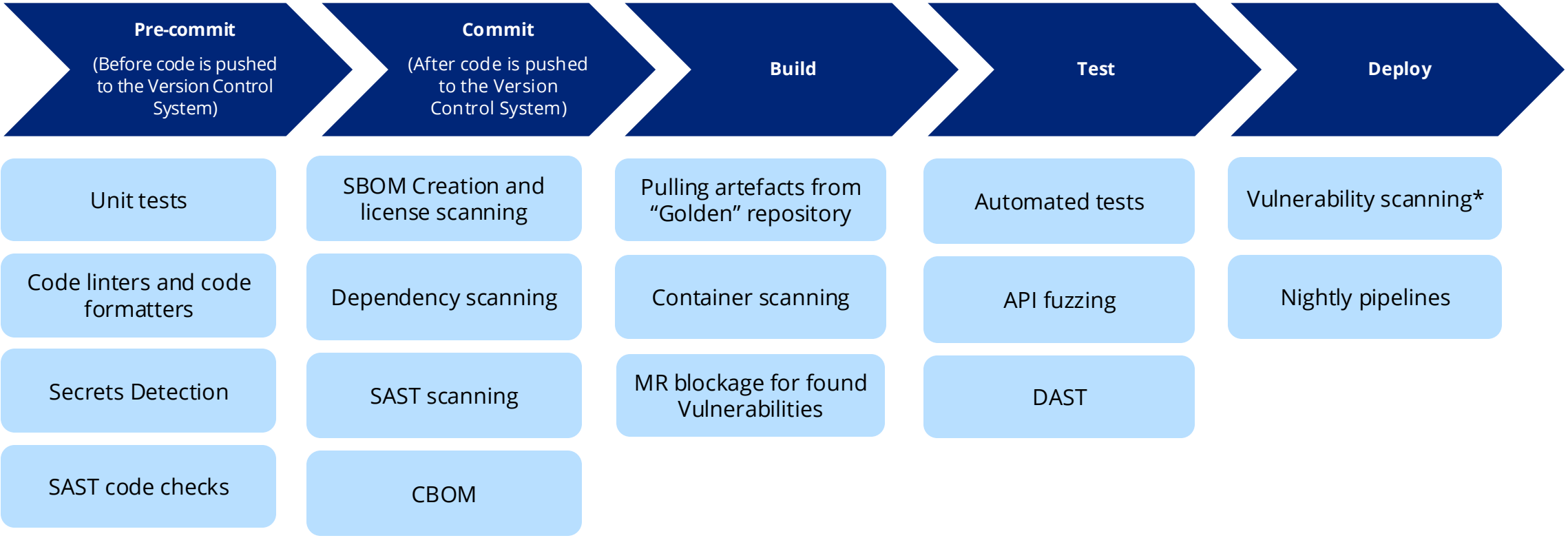
Cryptography Bill of Materials enables detailed representation of cryptographic assets within a system. This includes algorithms, keys, certificates, and their relationships to software components

CBOM kit by IBM

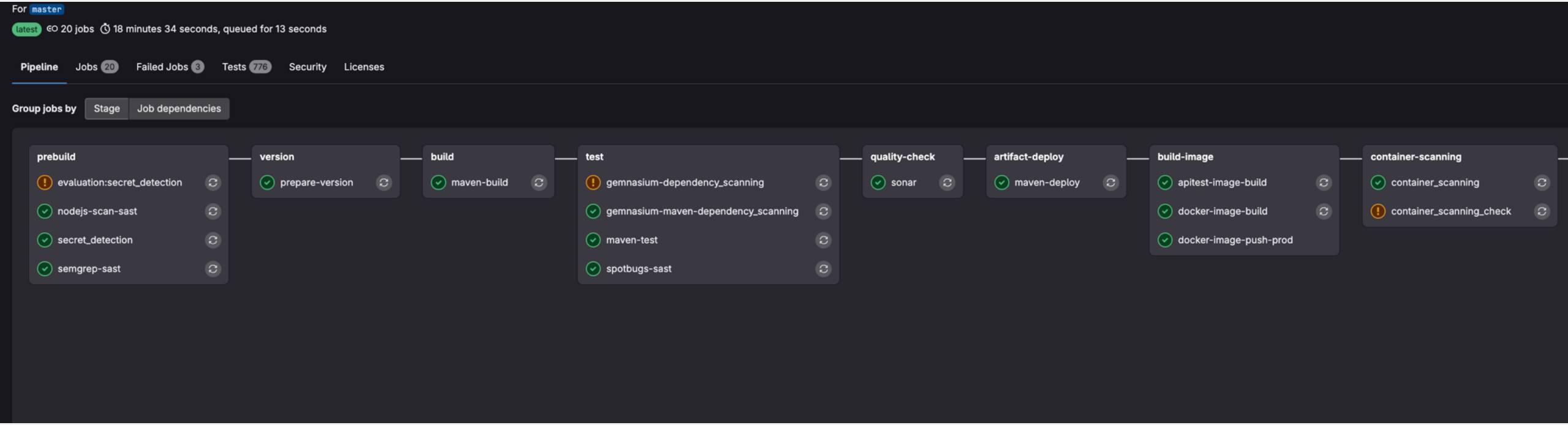
- SonarQube plugin
- Container plugin
- UI
- Compliance check
- Database



Example of a “shift left” pipeline



Example pipeline

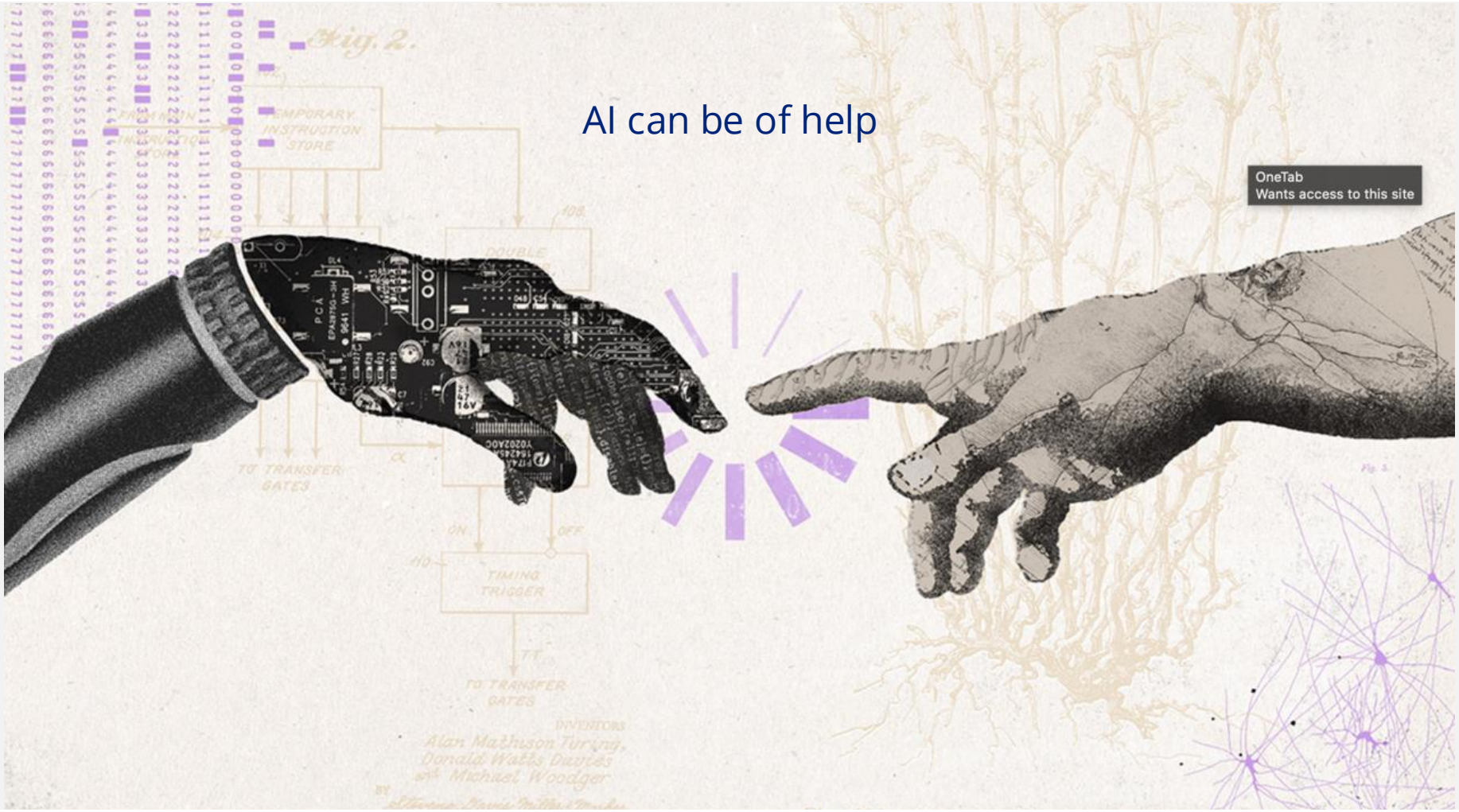


OS tools examples:

- Secret Detection - GitLeaks TruffleHog
  SAST – SonarQube, Semgrep, MobSF + OWASP List of Source Code tools
- SCA - DependencyTrack
- Container scanning - Trivy
- Vulnerability scanning – ZAP, Burp, Dastardly
- API fuzzing – restler-fuzzer oss-fuzz AFLplusplus jazzer

Complicated?

AI can be of help

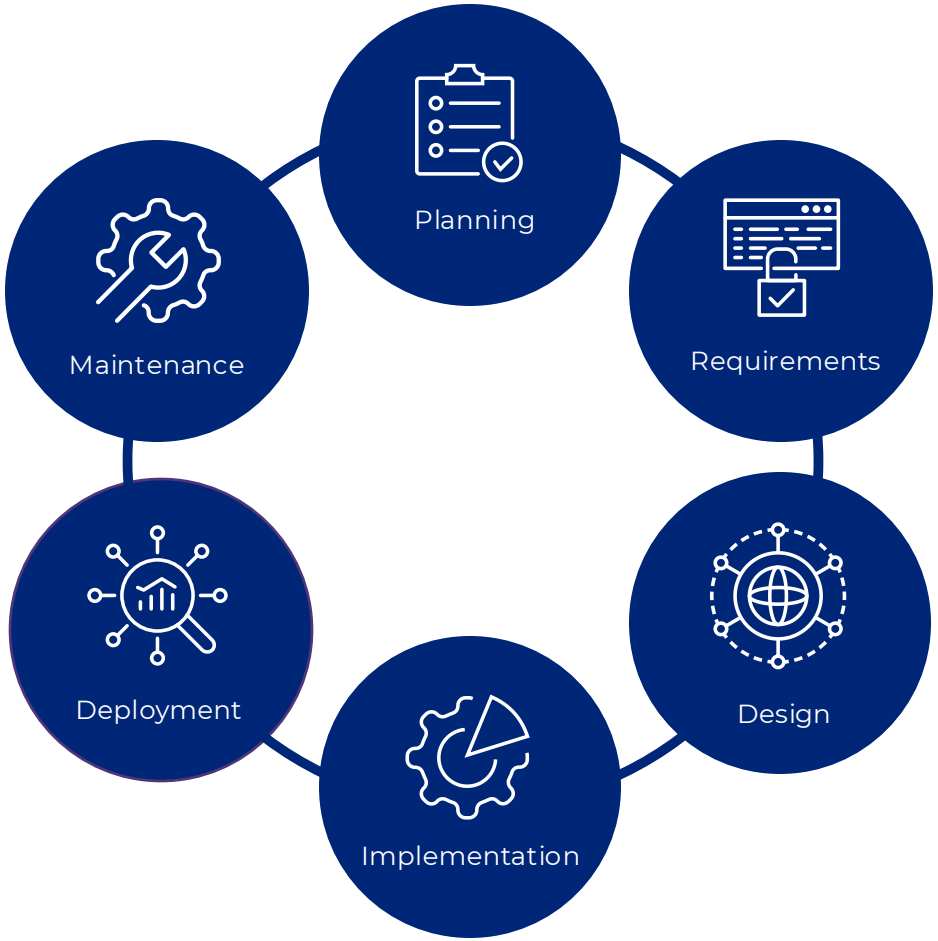


Secure Software Development Process with AI

- Patch Management
- Policy & Document Maintenance
- Continues Assessment & Monitoring
- AI enhancing runtime security
- Automatic respond to threats (predictive and behaviors analysis of a user)

- Vulnerability Assessment
- Penetration testing
- Runtime Security
- AI Assisted AI Red Teaming
- Pen testing

- Secure Code Practices & Reviews
- Security Hardening
- Secrets Detection
- **DevSecOps**
- SAST & DAST
- **SCA**
- AI Vulnerability Triaging
- AI Secure code reviews



- Security Requirements
- Compliance & business objectives
- Budget
- **Standards & Frameworks**
- AI Methodologies
- EU AI Act

- Technical Security Requirements
- Map Security & Privacy Requirements
- Risk analysis
- AI Mapping
- AI Risk Assessment

- Secure Design Principles
- Security controls/gates
- **Threat Modelling**
- **Policies & Procedures Enablement**
- Access Control Lists
- Security Coding Guide
- Secret Management
- Automated validation of security requirements
- AI Threat Modelling

Q&A

Be proactive and take measures.
Don't wait for the breach to teach.

Thank you!

www.scalefocus.com