



Software nachhaltig sicher entwickeln

Ein Praxisbericht

Clemens Hübner - inovex GmbH





Clemens Hübner

Tech Lead Software Security

inovex GmbH

Developer, Consultant, Speaker, Trainer



@ClemensHuebner



clemens.huebner@inovex.de



@clemens@infosec.exchange



/clemens-huebner



@inovexlife

blog.inovex.de

AppSec Consultant

- Processes & Tools, Governance

Security Architecture

- Threat Modeling, Sec Audits, Trainings

M.Sc. Software Engineering & Security

Agile Software
Development

Security Developer

- AuthN, SAST, Pentest

Security Engineer

- DevSecOps, Testing, SCA, Trainings



Wir warten ständig auf
Freigaben von Security!

*Müssen wir das wirklich für
jedes Team neu erfinden?*

Die Scanner liefern uns mehr
Findings als wir bearbeiten können!

AppSec in Organisationen

*Features, Performance, Usability,
Maintainability, ... - jetzt auch noch Security?*

Security, das machen doch die
anderen im Team!

Haben wir eigentlich an
alles gedacht?

*Wir kommen einfach nicht aus
dem Feuerlöschen raus!*



AGENDA

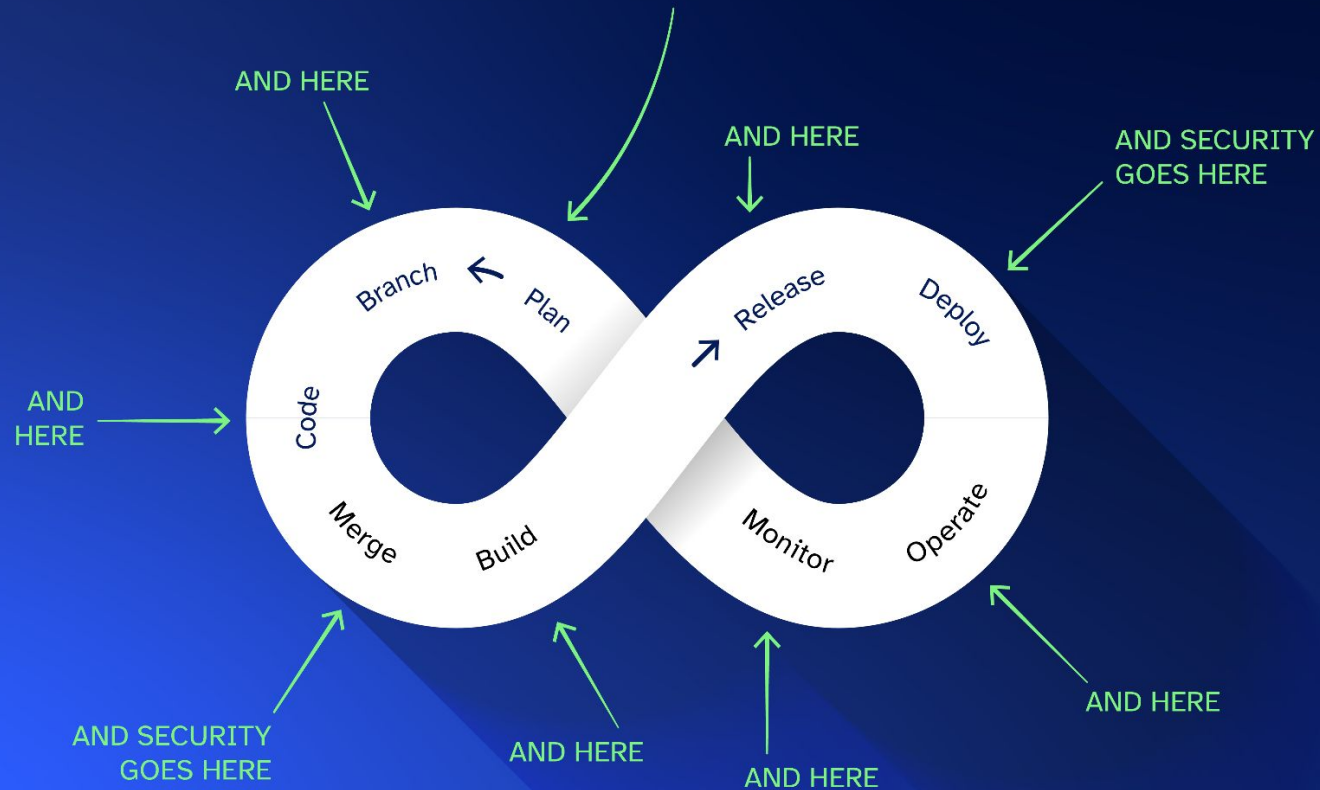
Software nachhaltig sicher entwickeln

- Warum?
- Was?
- Wer?
- Wie?



1. Was?

SECURITY GOES HERE



OWASP SAMM

Software Assurance Maturity Model



Reifegrad-Modell für AppSec

- offenes, flexibles Framework
- Fokus auf Einschätzung und Verbesserung
- anpassbar auf spezifische Risiken und Bedürfnisse

Bestandteile:

- Modell selbst
- Guidance
- offizielle Dokumentation und Community-Ressourcen

Level 0	keine Umsetzung
Level 1	erste, informelle Umsetzung mit reaktivem Fokus
Level 2	standardisierte Umsetzung mit proaktivem Fokus
Level 3	integrierte, kontinuierlich verbesserte Umsetzung

Die 3 Reifegrade von OWASP SAMM

Die fünf *business functions* von OWASP SAMM

Governance

Strategie,
Richtlinien und die
übergreifende
Verwaltung der
Softwaresicherheit

Design

Sichere
Architektur,
Erstellen von
Anforderungen
und Auswahl von
Technologien

Implementation

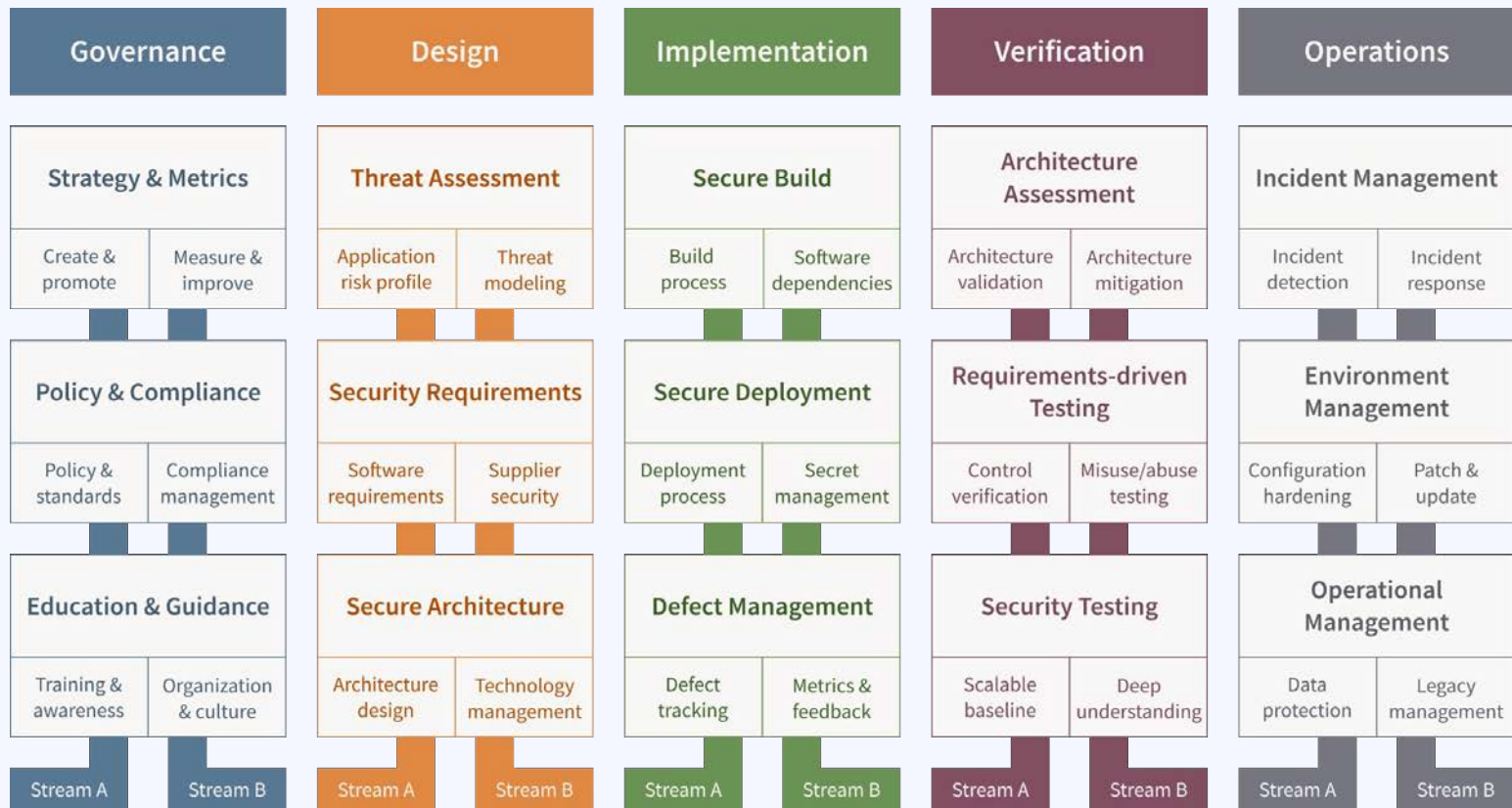
Coding, Build und
Deployment inkl.
Dependencies und
Defect
Management

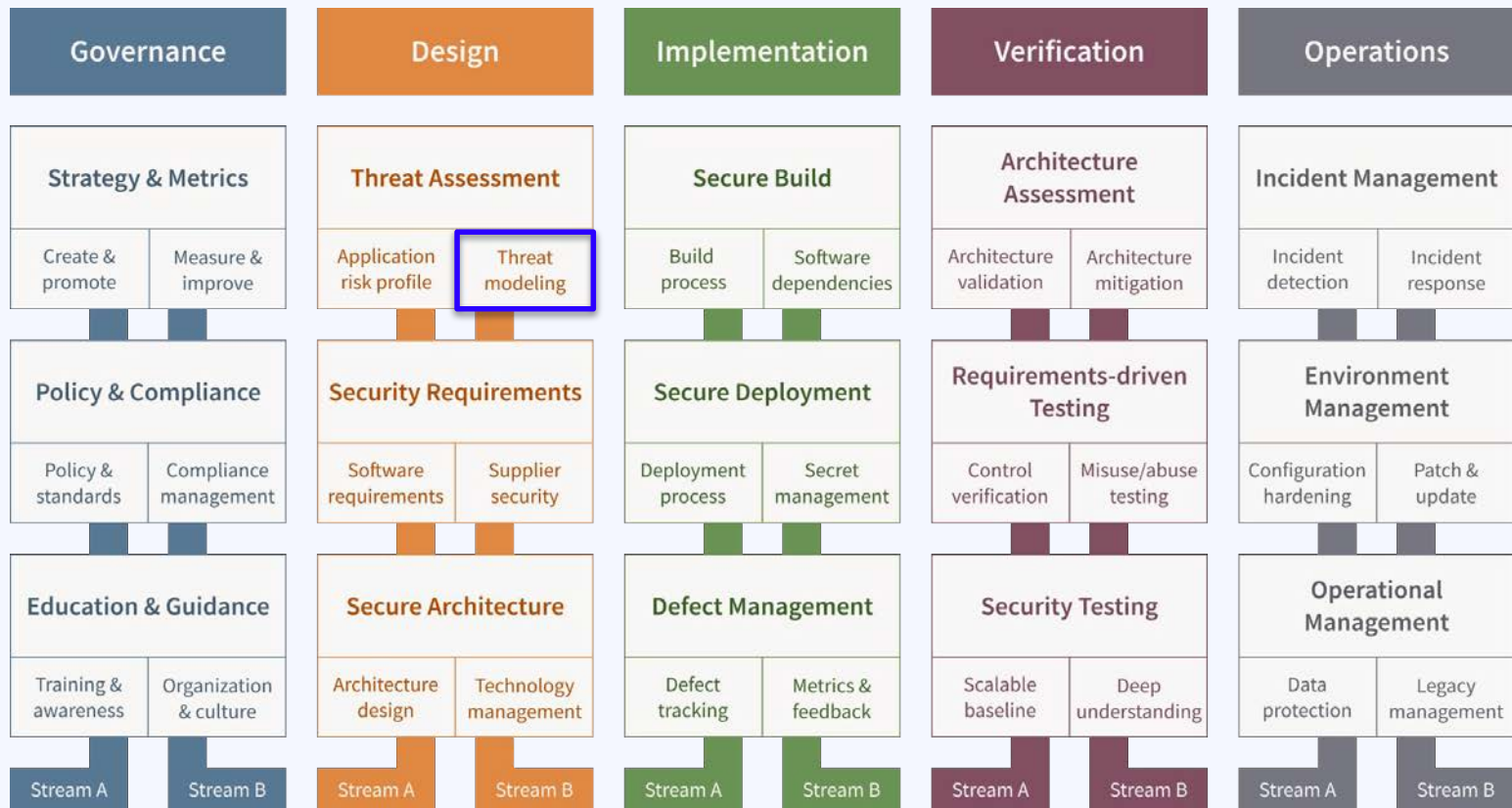
Verification

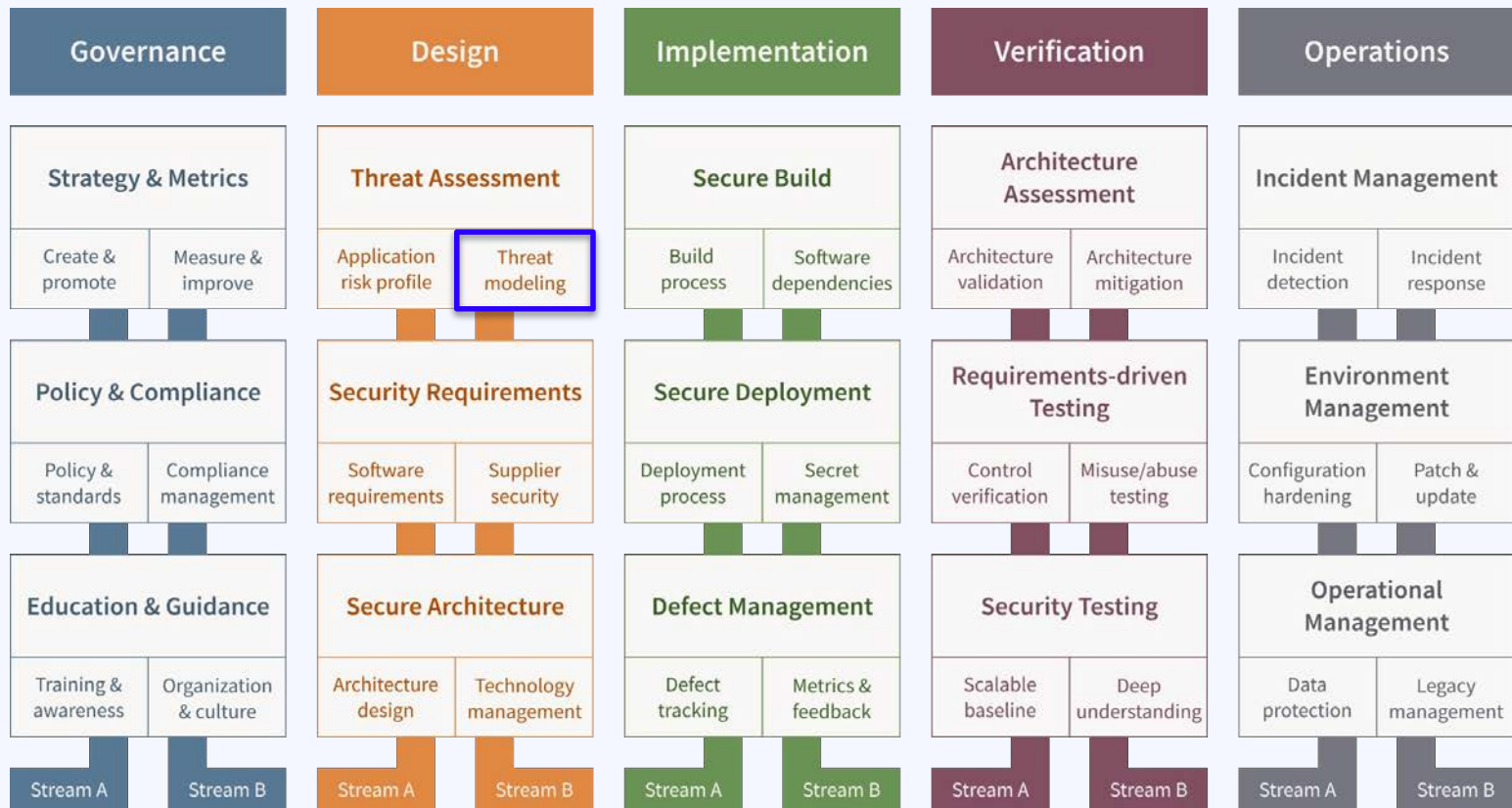
Assessments und
Testing: statisch
und dynamisch,
automatisch und
manuell

Operations

Wartung und
Betrieb inkl.
Incident
Management

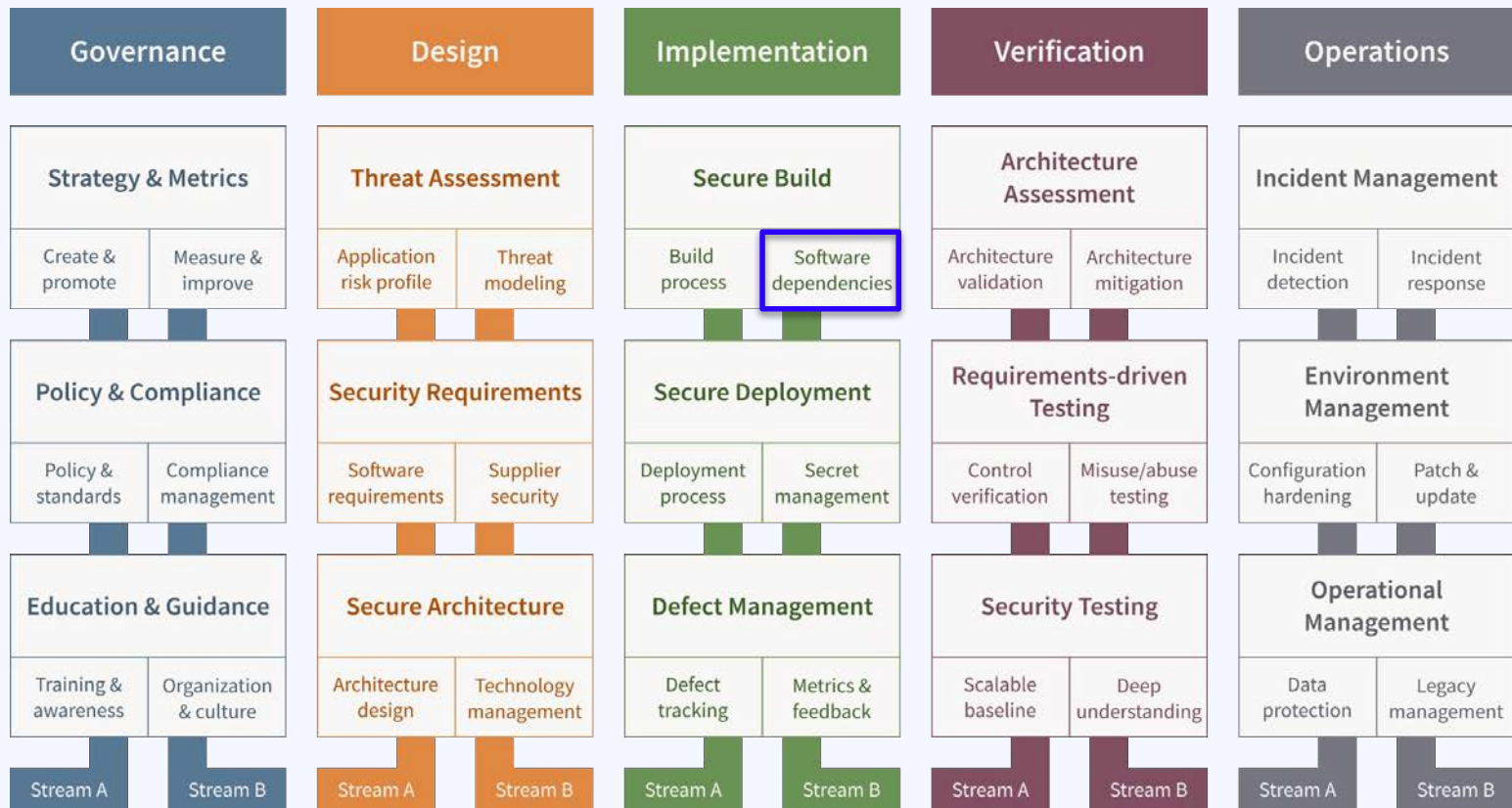






Design: Threat Assessment

Maternity level	Stream A Application Risk Profile	Stream B Threat Modeling	
1	A basic assessment of the application risk is performed to understand likelihood and impact of an attack.	Perform best-effort, risk-based threat modeling using brainstorming and existing diagrams with simple threat checklists.	Erste Umsetzung: ad-hoc, informell, priorisiert
2	Understand the risk for all applications in the organization by centralizing the risk profile inventory for stakeholders.	Standardize threat modeling training, processes, and tools to scale across the organization.	Vereinheitlichung: Standardisierte, regelmäßige Durchführung
3	Periodically review application risk profiles at regular intervals to ensure accuracy and reflect current state.	Continuously optimization and automation of your threat modeling methodology.	Kontinuierlich optimierend und voll-integriert



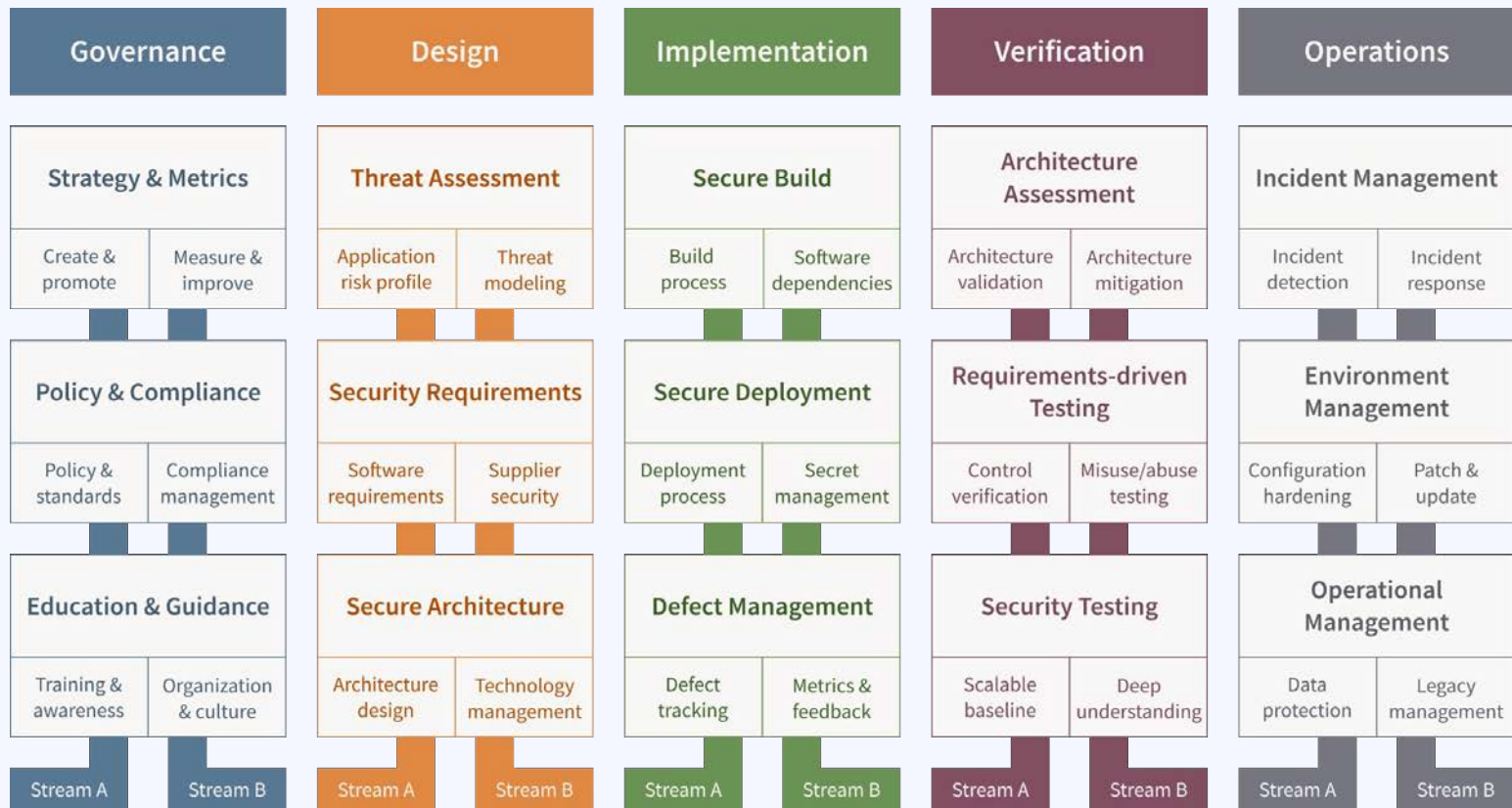
Implementation: Secure Build

Maturity level	Stream A Build Process	Stream B Software Dependencies
1	Create a formal definition of the build process so that it becomes consistent and repeatable.	Create records with Bill of Materials of your applications and opportunistically analyze these.
2	Automate your build pipeline and secure the used tooling. Add security checks in the build pipeline.	Evaluate used dependencies and ensure timely reaction to situations posing risk to your applications.
3	Define mandatory security checks in the build process and ensure that building non-compliant artifacts fails.	Analyze used dependencies for security issues in a comparable way to your own code.

Erste Umsetzung:
ad-hoc, informell,
priorisiert

Vereinheitlichung:
Standardisierte,
regelmäßige
Durchführung

Kontinuierlich
optimierend und
voll-integriert

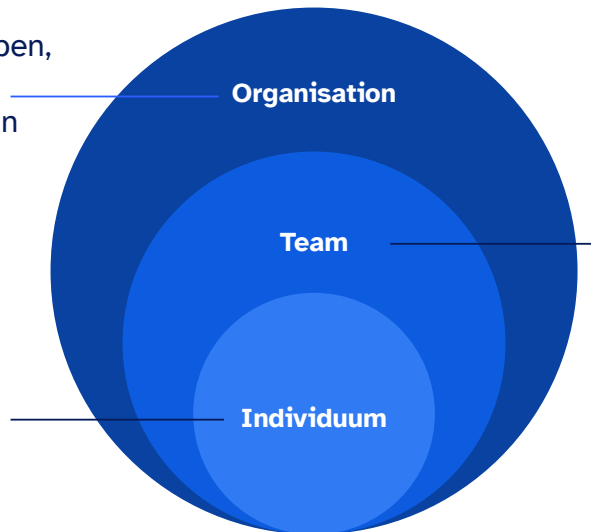


2. Wer?

Die drei Einflusskreise von AppSec

- Governance: zentrale Vorgaben, klares Management
- Bereitstellung von Prioritäten und Ressourcen

- individuelle Skills und Verantwortung
- limitierte Cognitive Load

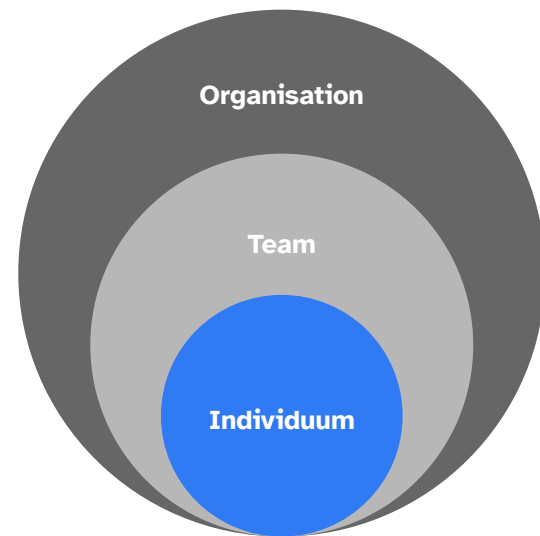


- Team Security Ownership
- Unterstützung durch Shared Services (Consulting und Tools)

Das Individuum und OWASP SAMM

Ziel: Persönliches Enablement & Reduktion der cognitive Load

- Passende Schulungen
- klare Aufgabenteilung
- so viel Unterstützung durch Team und Organisation wie möglich



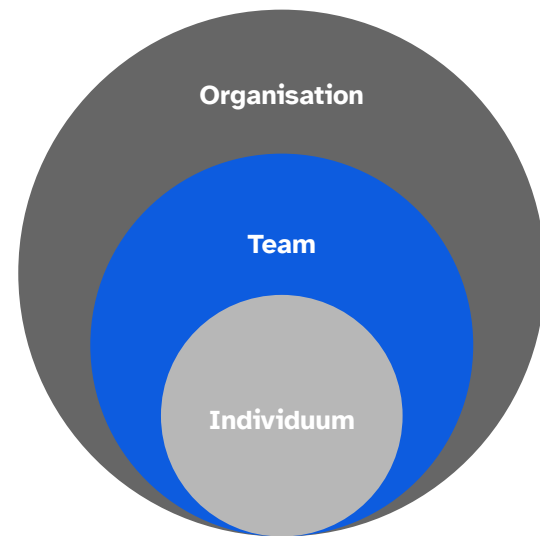
Das Entwicklungsteam und OWASP SAMM

Team Security Ownership: letztendliche Verantwortung für Sicherheit liegt beim Entwicklungsteam

- ermöglicht effiziente Umsetzung von Security-Vorgaben
- benötigt klare Vorgaben und Leitlinien
- unterstützt durch Consulting, Coaching und Shared Services

Security Champions

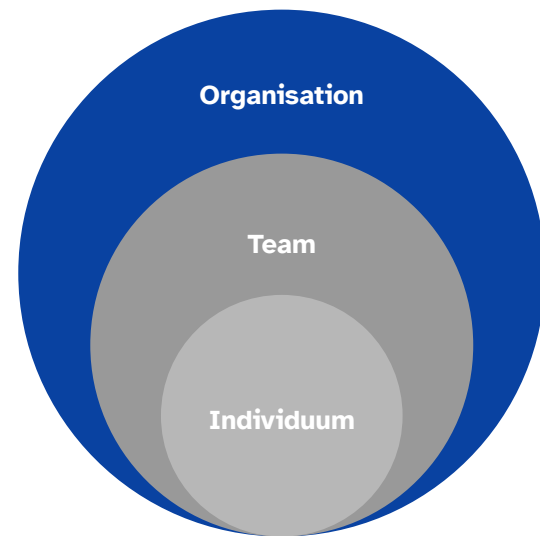
- als Bindeglied und Kommunikationsinstrument
- nicht als SPoF oder Feigenblatt



Die Organisation und OWASP SAMM

Zentrale Vorgaben reduzieren Cognitive Load und ermöglichen einheitliche Security-Aktivitäten

- OWASP SAMM als Strategieinstrument
- Zentrales Risikomanagement und Leitlinien
- Messbarkeit: Passende, schlanke Metriken von Anfang an



3. Wie?

OWASP SAMM im Unternehmen einführen



Case A: Regionaler Energieversorger

HINTERGRUND

- 6 Teams mit ~40 Devs für verschiedene Systeme und Anwendungen
- Abteilung und ihre Anwendungen selbst nicht KRITIS, aber erhöhte Visibilität und steigende Angriffsaktivität

HERAUSFORDERUNGEN

- erste unkoordinierte Aktivitäten an verschiedenen Stellen im Entwicklungsprozess (SAST-Scanner, Pentests)
- fehlende ganzheitliche Betrachtung, fehlende Strukturierung
- gleichzeitige agile Transformation

LÖSUNGSANSATZ

- risikobasierter Ansatz nach OWASP SAMM
- schrittweise Erhöhung der Security

Case A: Regionaler Energieversorger

ERSTE MASSNAHMEN

- Schulungen für alle Beteiligten
- Aufbau eines Application Inventory mit einheitlichen Schutzbedarfen
- Aufbau von Security Champions

SECURITY ORGANISATION

- zentrales Security-Team für Vorgaben und Unterstützung
- Security Champions in jedem Team bilden Brückenkopf
- Shared Services werden von Security Champions / Teams gemeinsam selbst umgesetzt

LESSONS LEARNED

- Rollen und Verantwortlichkeiten müssen explizit erarbeitet und festgehalten werden
- Prozesse wollen nicht nur konzipiert und eingeführt, sondern auch monitored und verbessert werden

Case B: AI-Insurance Start-Up

HINTERGRUND

- ~15 Devs in fluiden Teams
- Entwicklung von Whitelabel-KI-Agenten für Versicherungen
- hohe Velocity, moderne Architektur, starker Fokus auf Features

HERAUSFORDERUNGEN

- hohe Geschwindigkeit musste mit Sicherheitsbewusstsein der Kunden in Einklang gebracht werden
- erste Wachstumsschmerzen / Skalierungsschwierigkeiten

LÖSUNGSANSATZ

- OWASP SAMM zu starr für Unternehmenskultur
 - nützlich als Anhaltspunkt
 - Beschränkung auf Level 1
 - Ergänzung mit DSOMM
- Erhöhung der Automatisierung
- Vereinheitlichung von Tooling und Abläufen

Case B: AI-Insurance Start-Up

ERSTE MASSNAHMEN

- Fixen von blinden Flecken bei Design & Assessment
- Leichtgewichtiges, kontinuierliches Threat Modeling

SECURITY ORGANISATION

- CISO macht strategische Vorgaben
- Selbstverantwortung und -organisation der Teams bezüglich der Umsetzung
- Pläne für zentrales Plattformteam mit Security-Anteilen

LESSONS LEARNED

- SAMM ist nicht für jede Organisation das perfekte Umsetzungsinstrument

Case C: Telko-Produkt Mittelständler

HINTERGRUND

- Mittelständler mit ~80 Entwickler:innen
- eigenes Telko-Produkt
- Kernanwendung ist Monolith, über 20 Jahre gewachsen

HERAUSFORDERUNGEN

- Veraltete Codebasis erschwerte die Einführung moderner DevSecOps-Praktiken

LÖSUNGSANSATZ

- zweigeteilt:
Risiko-Management für das Altsystem, Aufbau von modernen Praktiken für Neuentwicklung mittels SAMM
- Erfahrungen der Vergangenheit fließen in Gestaltung der Abläufe für die Zukunft ein

Case C: Telko-Produkt Mittelständler

ERSTE MAßNAHMEN

- ausführliches Threat Modeling ergänzt bestehende Pentests
- Konsequente Umsetzung in neuen Projekten

SECURITY ORGANISATION

- Zentrales AppSec-Team kümmert sich um Governance, stellt Services zur Verfügung
- Security Champions noch in Planung

LESSONS LEARNED

- Devs haben nicht so viel Abneigung wie Vorgaben von oben wie oft gedacht
- SAMM ist nicht primär für Entwickler:innen

OWASP SAMM - Stärken und Limitierung



- + Flexibilität und Anpassbarkeit
- + Klare Ziele ermöglichen Messbarkeit
- + Ganzheitlicher Ansatz
- + Risikoorientiert
- + Offene Dokumente und aktive Community

→ **Mächtiges, erprobtes Framework**

- Subjektivität in der Bewertung
- Ressourcenintensive Einführung
- Starker Fokus auf Prozesse

→ **Bewusster, sorgfältiger Einsatz nötig**

AppSec braucht eine explizite Roadmap

Die Organisationsstruktur muss den Arbeitsfluss der Security optimieren

Bei Umsetzung Mensch und Teams im Fokus halten



Vielen Dank!



 /clemens-huebner

 @ClemensHuebner

 clemens.huebner@inovex.de

 @clemens@infosec.exchange

 @inovexlife

blog.inovex.de



[Blogpost](#)