OWASP
Open Web Application
Security Project

OWASP
SAMM

A Complete View of Application Security with OWASP SAMM

CODIFIC

Course contents

- The Application Security Challenge
- Software Development Lifecycle Overview
- OWASP SAMM
  - Vision, History, Structure
  - Assessment Tool
  - Deep Dive into Secure Build: Demo
  - Methodology
- Conclusion

OWASP SAMM Fundamentals

- Full OWASP SAMM Fundamentals course
  - https://owaspsamm.thinkific.com

Learning Objectives & Expectations

- Understand the application security challenge
- Get a clear view of the AppSec landscape
- Learn about SAMM (i.e., the solution)

Aram Hovsepyan

- CEO @ Codific
- PhD @ DistriNet, KULeuven
- OWASP SAMM core team member

https://www.linkedin.com/in/aramhovsep

https://www.linkedin.com/company/codific

Terms of reference

What is security?

C

CONFIDENTIALITY
unauthorized users cannot access assets

I

INTEGRITY
unauthorized users cannot modify assets

A

AVAILABILITY
assets are available on request

Terms of reference

- Application Security
  - Focus: application software engineering

- Cybersecurity
  - Broader focus: organization, software, network, etc.

SDLC vs Software Assurance

- Software Development Lifecycle (SDLC)
- Secure Software Development Lifecycle (Secure SDLC)
- Software Assurance Programme

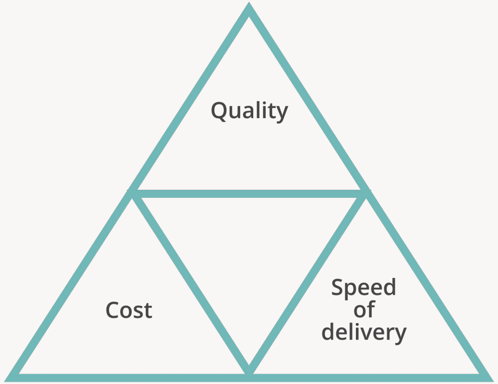
The application security problem

The application security problem

Complexity Availability Technology

Requirements Confidentiality

Cloud Microservices Integrity



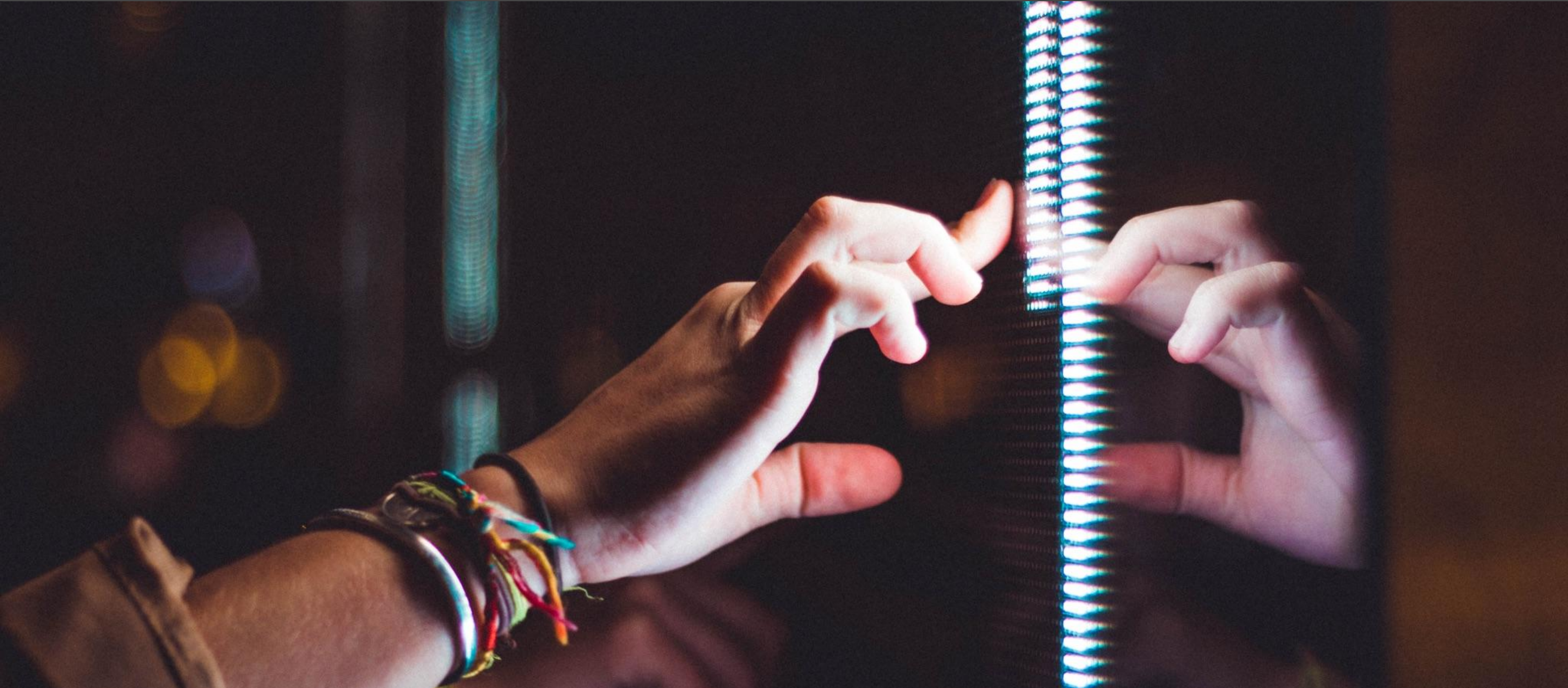
Quality

Cost

Speed of delivery

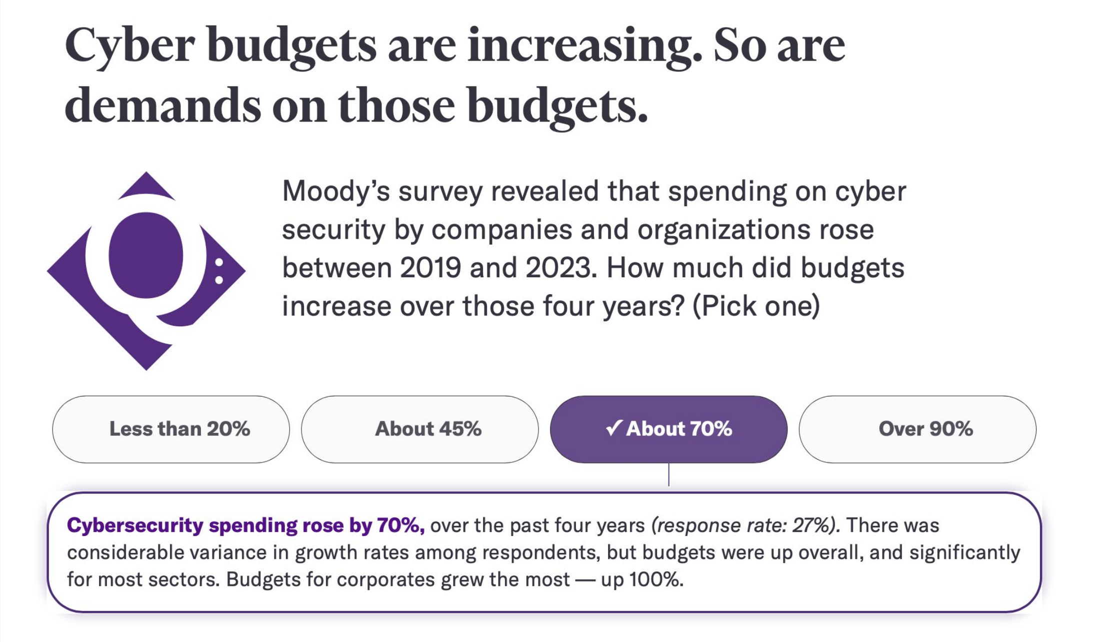
75% of vulnerabilities are application related

Security is intangible



When do we feel (in)security?

Investment in Software Security



Cyber budgets are increasing. So are demands on those budgets.

Moody's survey revealed that spending on cyber security by companies and organizations rose between 2019 and 2023. How much did budgets increase over those four years? (Pick one)

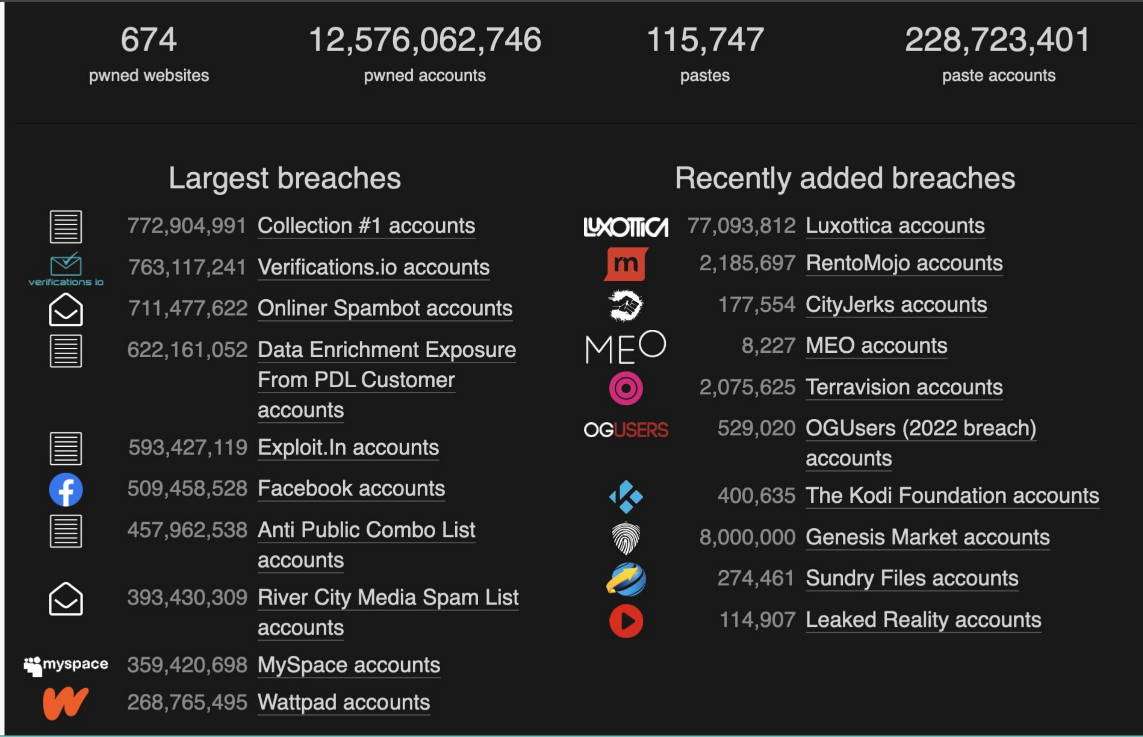
Less than 20% | About 45% | ✓ About 70% | Over 90%

Cybersecurity spending rose by 70%, over the past four years *(response rate: 27%)*. There was considerable variance in growth rates among respondents, but budgets were up overall, and significantly for most sectors. Budgets for corporates grew the most — up 100%.

https://www.moodys.com/web/en/us/about/insights/data-stories/2023-cyber-survey-highlights.html

Number of breaches is surging

https://haveibeenpwned.com



674	12,576,062,746	115,747	228,723,401
pwned websites	pwned accounts	pastes	paste accounts

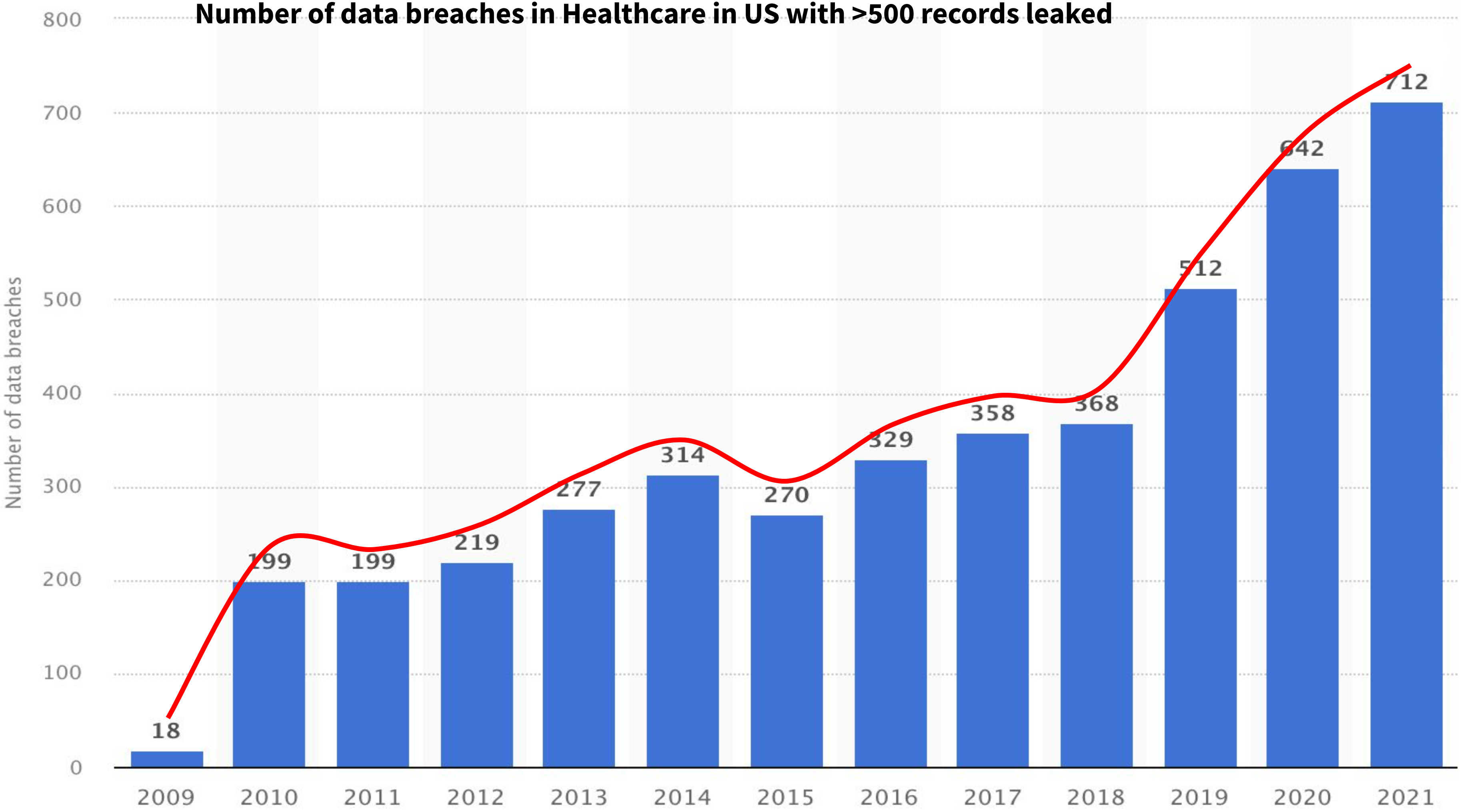
Largest breaches

772,904,991	Collection #1 accounts
763,117,241	Verifications.io accounts
711,477,622	Onliner Spambot accounts
622,161,052	Data Enrichment Exposure From PDL Customer accounts
593,427,119	Exploit.In accounts
509,458,528	Facebook accounts
457,962,538	Anti Public Combo List accounts
393,430,309	River City Media Spam List accounts
359,420,698	MySpace accounts
268,765,495	Wattpad accounts

Recently added breaches

77,093,812	Luxottica accounts
2,185,697	RentoMojo accounts
177,554	CityJerks accounts
8,227	MEO accounts
2,075,625	Terravision accounts
529,020	OGUsers (2022 breach) accounts
400,635	The Kodi Foundation accounts
8,000,000	Genesis Market accounts
274,461	Sundry Files accounts
114,907	Leaked Reality accounts

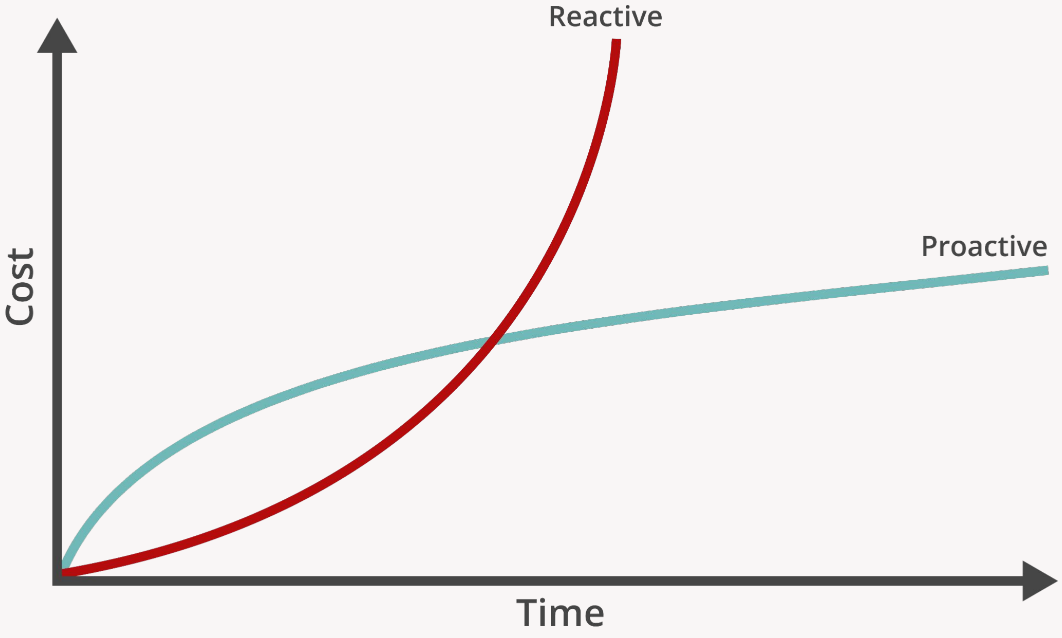
Number of data breaches in Healthcare in US with >500 records leaked



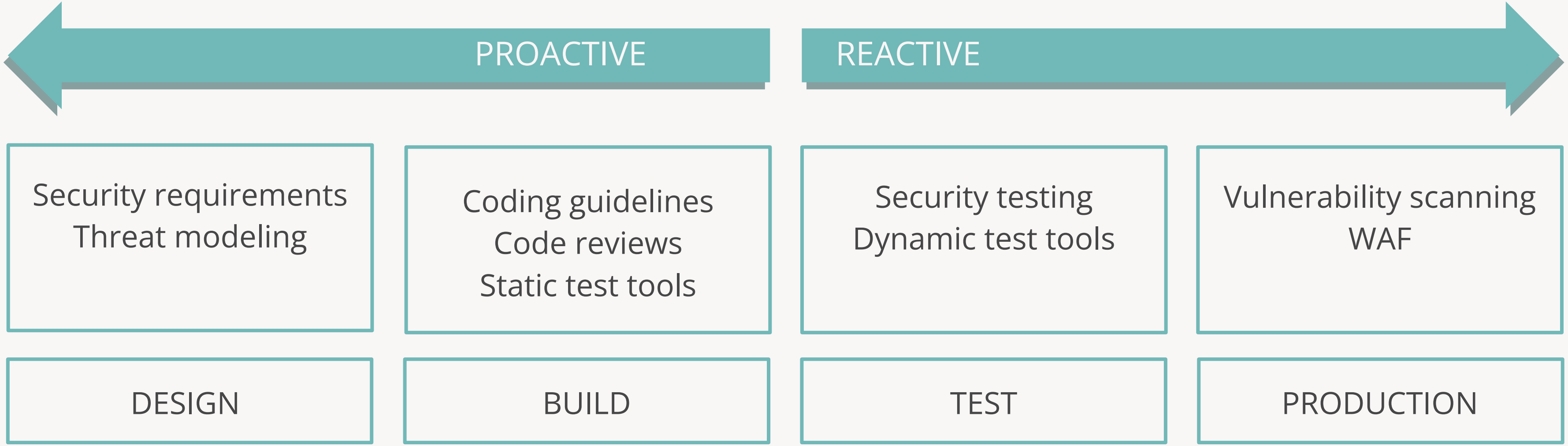
Data breach impact

- Fixing the issue
- Direct fines
- Loss of trust
- Reputational damage
- Stock price (*)
- Compensation requested by users

The cost of application security



Build in software assurance



PROACTIVE

REACTIVE

Security requirements Threat modeling	Coding guidelines Code reviews Static test tools	Security testing Dynamic test tools	Vulnerability scanning WAF
DESIGN	BUILD	TEST	PRODUCTION

Secure development lifecycle

The Startup Mindset

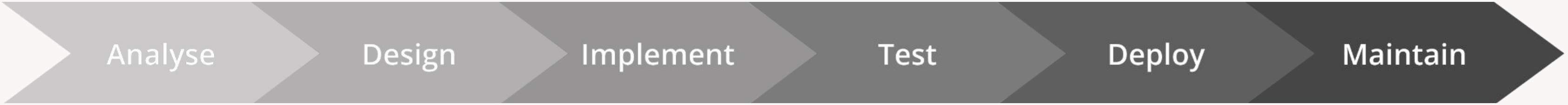
- Idea of a product
- Build the product
- Find paying customers
- Scale / grow

- Security is not on your top 10 todo list

Security in the SDLC

OWASP SAMM

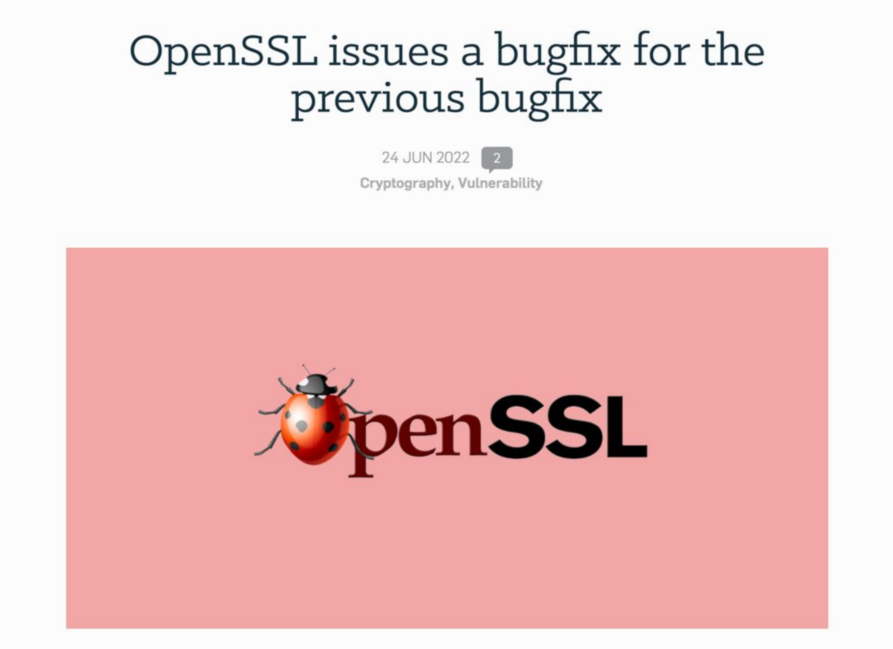
Security in a traditional SDLC



Why is this problematic?

- It’s not cost efficient
- There is no security assurance

Security in a traditional SDLC



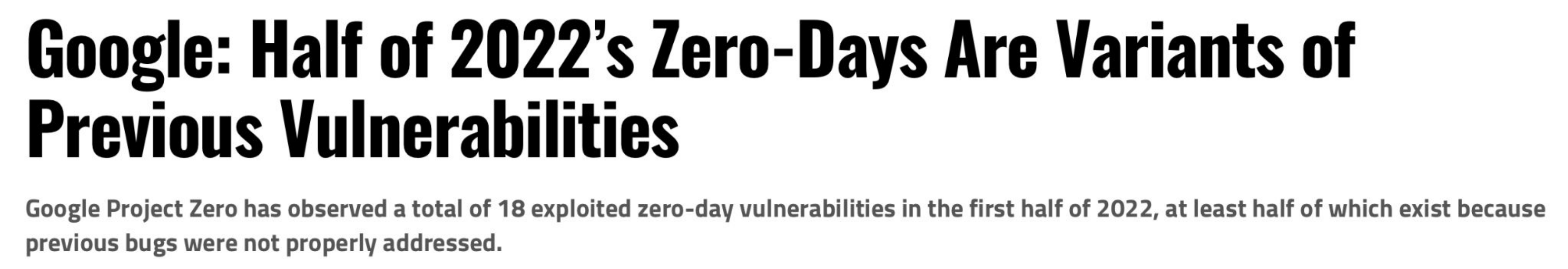
OpenSSL issues a bugfix for the previous bugfix

24 JUN 2022 2

Cryptography, Vulnerability

https://nakedsecurity.sophos.com/2022/06/24/openssl-issues-a-bugfix-for-the-previous-bugfix/

Security in a traditional SDLC

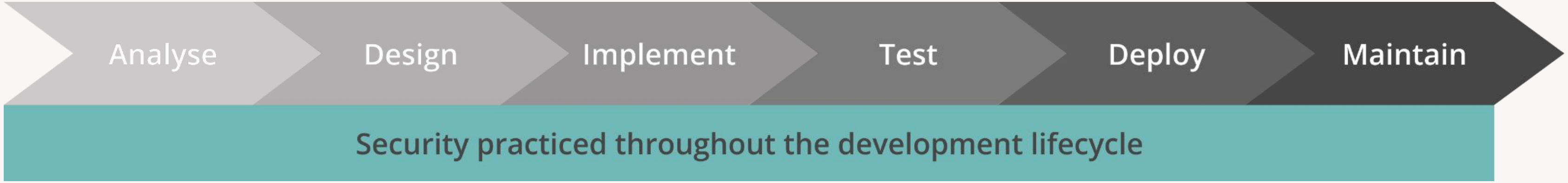


Google: Half of 2022’s Zero-Days Are Variants of Previous Vulnerabilities

Google Project Zero has observed a total of 18 exploited zero-day vulnerabilities in the first half of 2022, at least half of which exist because previous bugs were not properly addressed.

https://www.securityweek.com/google-half-2022s-zero-days-are-variants-previous-vulnerabilities/

Secure by Design



Analyse

Design

Implement

Test

Deploy

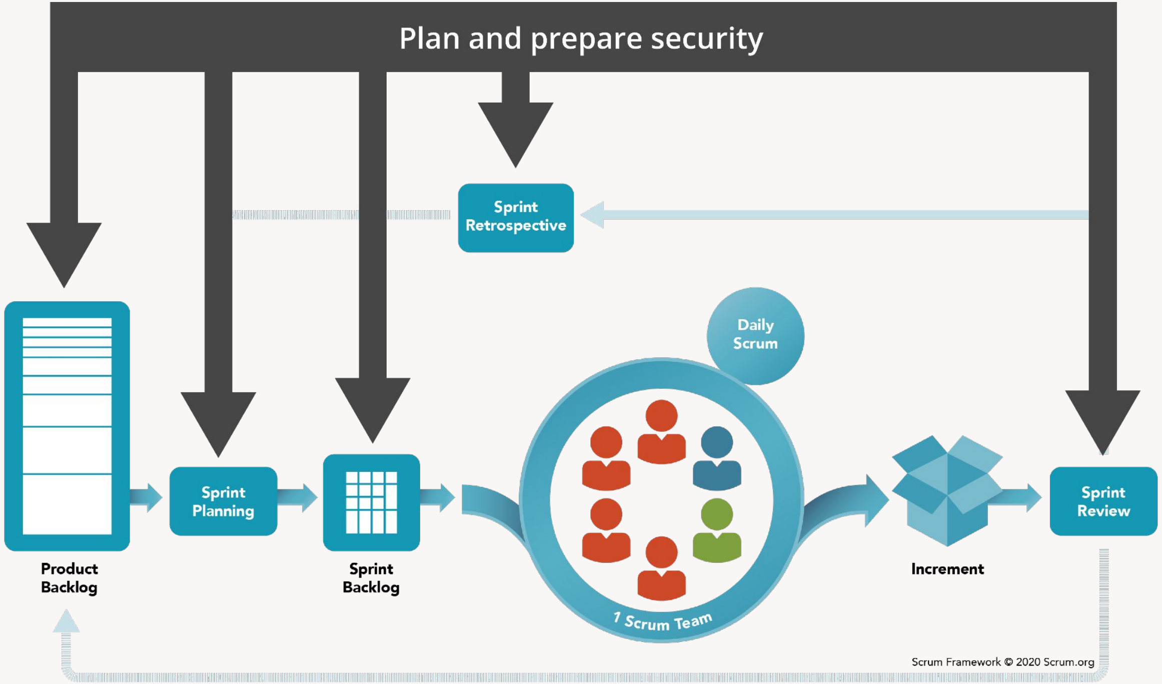
Maintain

Security practiced throughout the development lifecycle

Enterprise-wide software security improvement program

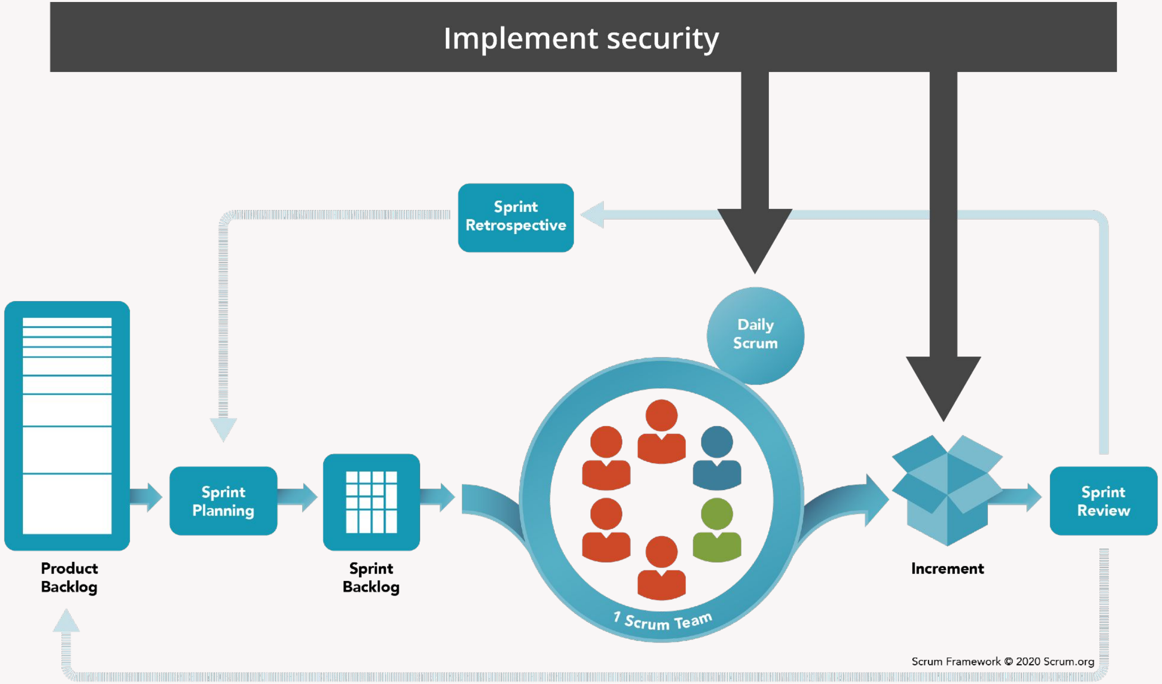
- Strategic approach to assure software quality
- Increase systematicity
- Focus on security functionality and security hygiene

Security in Scrum/Agile



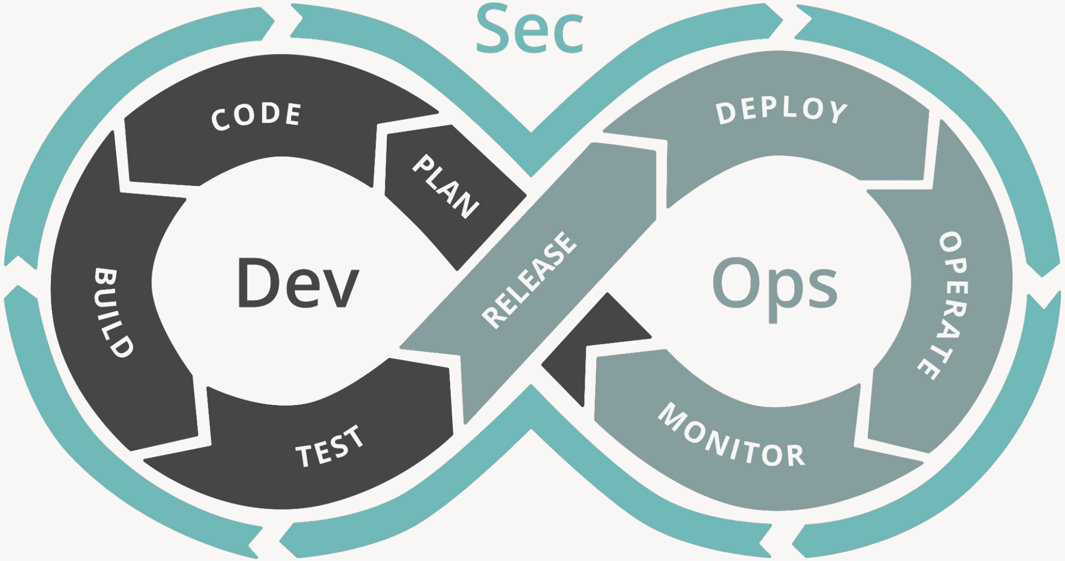
Scrum Framework © 2020 Scrum.org

Security in Scrum/Agile

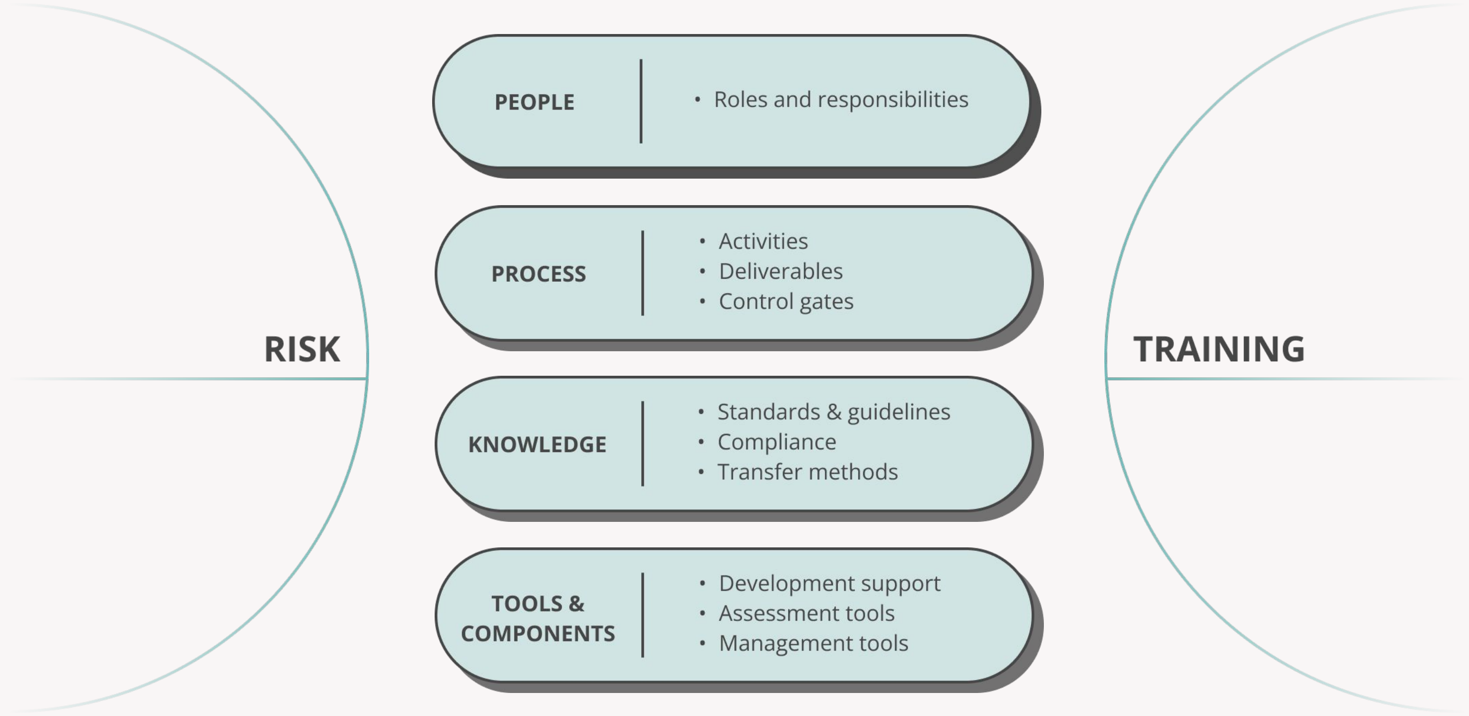


Scrum Framework © 2020 Scrum.org

Security in DevSecOps



SDLC Cornerstones



RISK

PEOPLE
- Roles and responsibilities

PROCESS
- Activities
- Deliverables
- Control gates

KNOWLEDGE
- Standards & guidelines
- Compliance
- Transfer methods

TOOLS & COMPONENTS
- Development support
- Assessment tools
- Management tools

TRAINING

SDLC initiatives

01010
0100000
010001
01000
0100001
SAFECode

NIST
NATIONAL INSTITUTE OF
STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

SAMM

BSIMM 13

Microsoft SDL

Software Engineering Institute
Carnegie Mellon

SDLC initiatives commonalities

- The security of deployed software is everyone’s concern
- Start from a clear understanding of what’s important to the organization
- Security requirements, implementation standards, security testing and feedback through metrics

Vision and history

OWASP SAMM

What is OWASP?

OWASP

Open Web Application Security Project

FLAGSHIP

mature projects

What is SAMM?

Software
Assurance
Maturity
Model

Measurable

Defined maturity levels across business practices

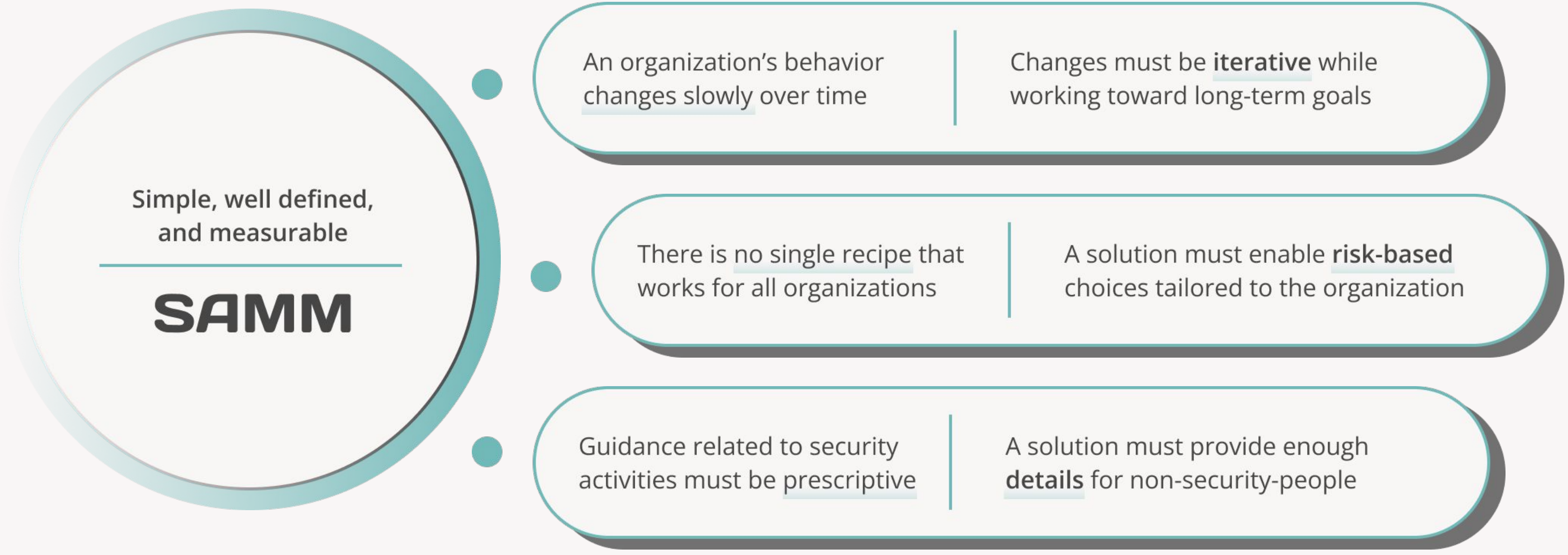
Actionable

Clear pathways for improving maturity levels

Versatile

Technology, process, and organization agnostic

Why a maturity model?



Simple, well defined, and measurable

SAMM

An organization's behavior changes slowly over time

Changes must be **iterative** while working toward long-term goals

There is no single recipe that works for all organizations

A solution must enable **risk-based** choices tailored to the organization

Guidance related to security activities must be prescriptive

A solution must provide enough **details** for non-security-people

SAMM Use-cases

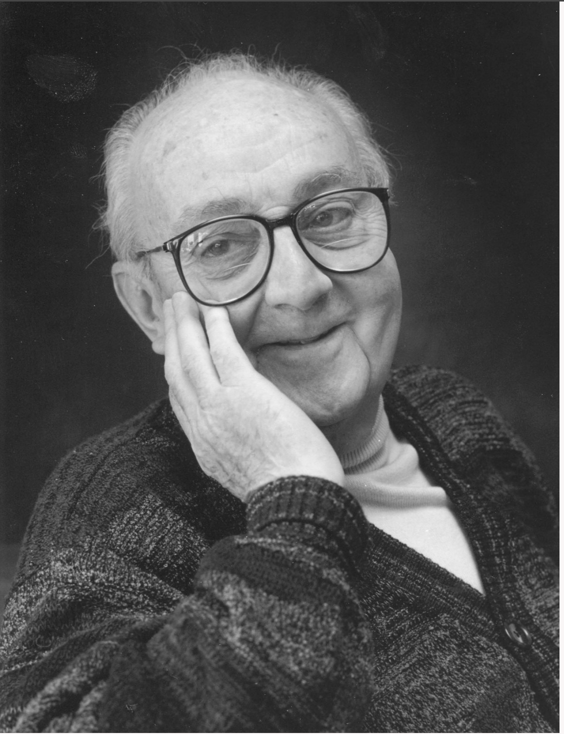
Evaluating an organization's existing software security practices

Building a balanced software security assurance program in defined iterations

Defining and **measuring** security-related activities throughout an organization

Demonstrating concrete improvements to a security assurance program

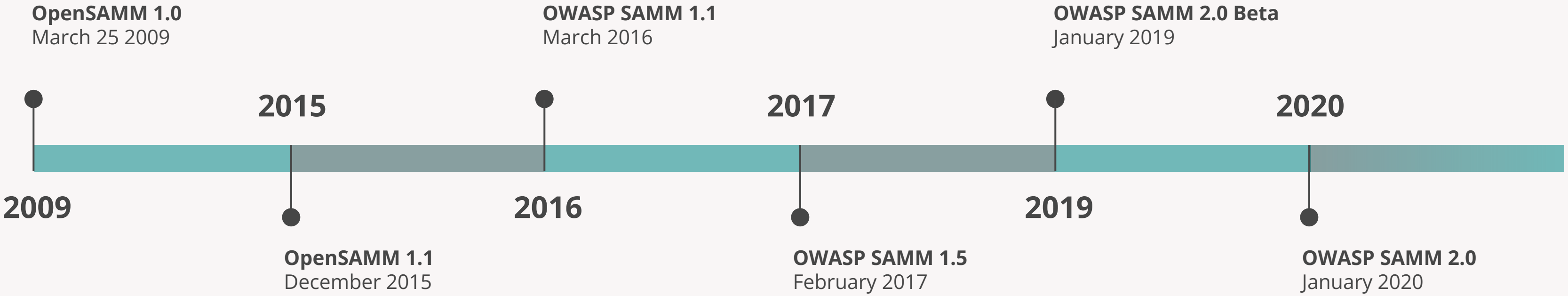
Why SAMM?



“All models are wrong, but some are useful”

-George Box

SAMM project history



2009

OpenSAMM 1.0
March 25 2009

2015

OpenSAMM 1.1
December 2015

2016

OWASP SAMM 1.1
March 2016

2017

OWASP SAMM 1.5
February 2017

2019

OWASP SAMM 2.0 Beta
January 2019

2020

OWASP SAMM 2.0
January 2020

Who is SAMM? Core team

🇧🇪
Sebastien Deleersnynder
Bart De Win
Maxim Baele
Aram Hovsepyan

🇫🇷
Romuald Szkudlarek

🇩🇪
Daniel Kefer

🇳🇿
John DiLeo

🇸🇪
John Kennedy

🇬🇧
Chris Cooper

🇺🇾
Patricia Duarte

🇺🇸
John Ellingsworth
Brian Glas

Sponsors

Checkmarx

CODIFIC

CONCORD

MICRO FOCUS Fortify

IMQ MINDED SECURITY

nccgroup

pwc

SECURITY INNOVATION

splunk>

TOREON

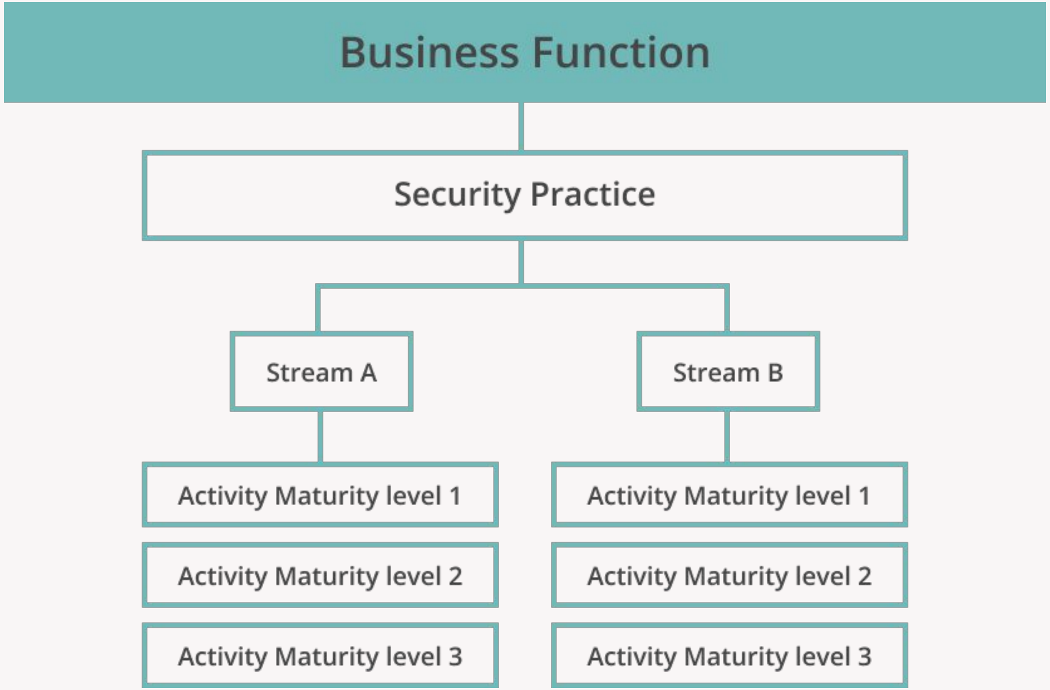
Indelible

owaspsamm.org/sponsors

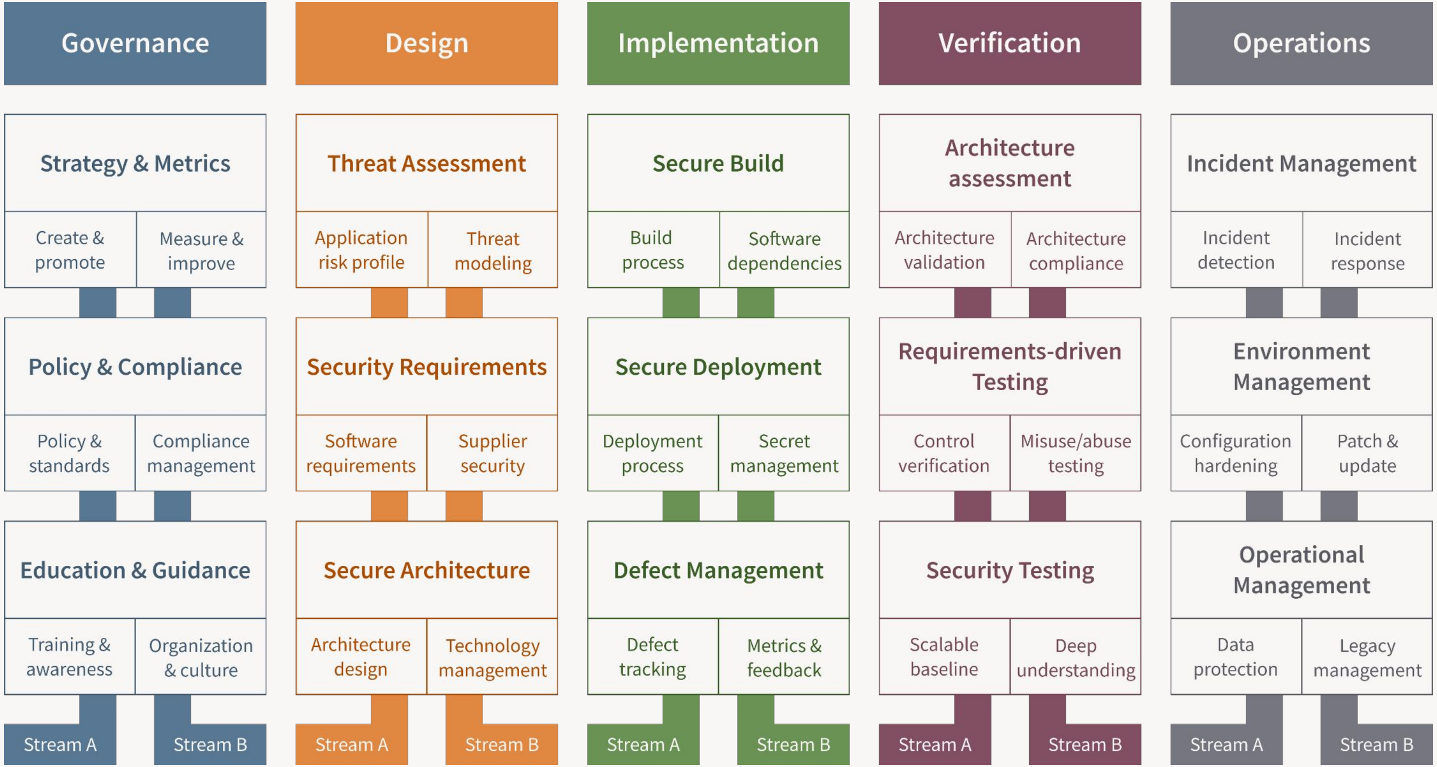
The structure of the model

OWASP SAMM

SAMM v2 Model structure


Business Function
Security Practice
Stream A
Stream B
Activity Maturity level 1
Activity Maturity level 2
Activity Maturity level 3
Activity Maturity level 1
Activity Maturity level 2
Activity Maturity level 3

SAMM v2 Model overview



Governance		Design		Implementation		Verification		Operations	
Strategy & Metrics		Threat Assessment		Secure Build		Architecture assessment		Incident Management	
Create & promote	Measure & improve	Application risk profile	Threat modeling	Build process	Software dependencies	Architecture validation	Architecture compliance	Incident detection	Incident response
Policy & Compliance		Security Requirements		Secure Deployment		Requirements-driven Testing		Environment Management	
Policy & standards	Compliance management	Software requirements	Supplier security	Deployment process	Secret management	Control verification	Misuse/abuse testing	Configuration hardening	Patch & update
Education & Guidance		Secure Architecture		Defect Management		Security Testing		Operational Management	
Training & awareness	Organization & culture	Architecture design	Technology management	Defect tracking	Metrics & feedback	Scalable baseline	Deep understanding	Data protection	Legacy management
Stream A	Stream B	Stream A	Stream B	Stream A	Stream B	Stream A	Stream B	Stream A	Stream B

SAMM v2 key changes

Version 1.5

- 4 Business Functions
- 12 Security Practices
- No prescriptive guidance for build and deploy domains
- Maturity level activities orphaned and sometimes unrelated to each other
- Maturity level activities not in order of increasing difficulty, cost of implementation
- Measurement only based on coverage

Version 2

- 5 Business Functions
- 15 Security Practices
- New Business Function: Implementation
- Maturity level activities aligned and linked per stream. Each stream has a clear objective
- Maturity level activities designed in order of increasing difficulty, implementation cost
- Measurement based on both coverage and quality

SAMM Security Practices

- 3 Security Practices for each Business Function
- They cover key areas relevant to software security assurance
- Each one is a silo for improvement

Governance	Strategy & Metrics Education & Guidance Policy & Compliance
Design	Threat Assessment Security Requirements Secure Architecture
Implementation	Secure Build Secure Deployment Defect Management
Verification	Architecture Assessment Requirements-driven Testing Security Testing
Operations	Incident Management Environment Management Operational Management

SAMM Maturity Levels

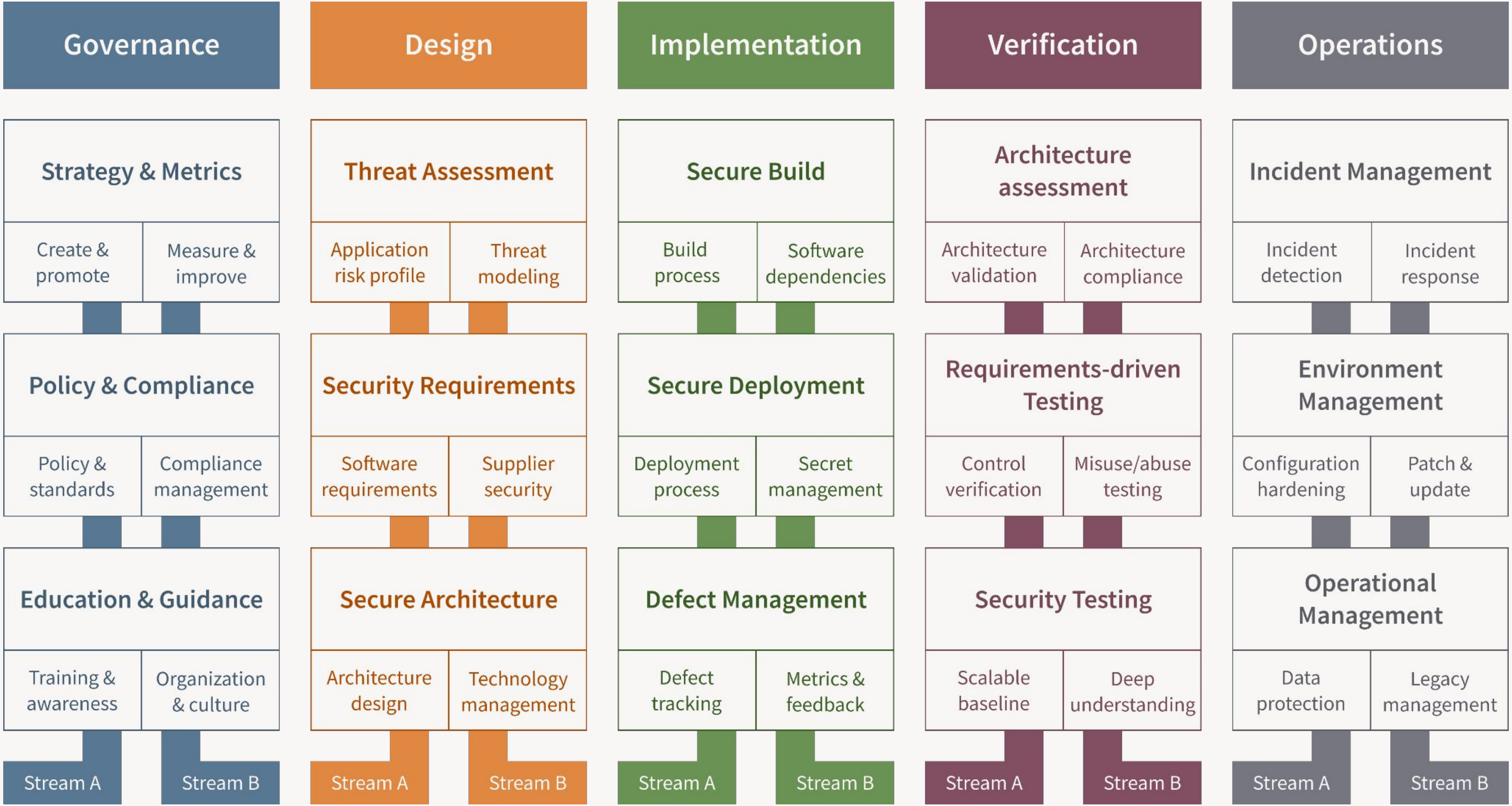
Fulfilling Practices and improving using 3 successive objectives

0 (Implicit starting point with the Practice unfulfilled)

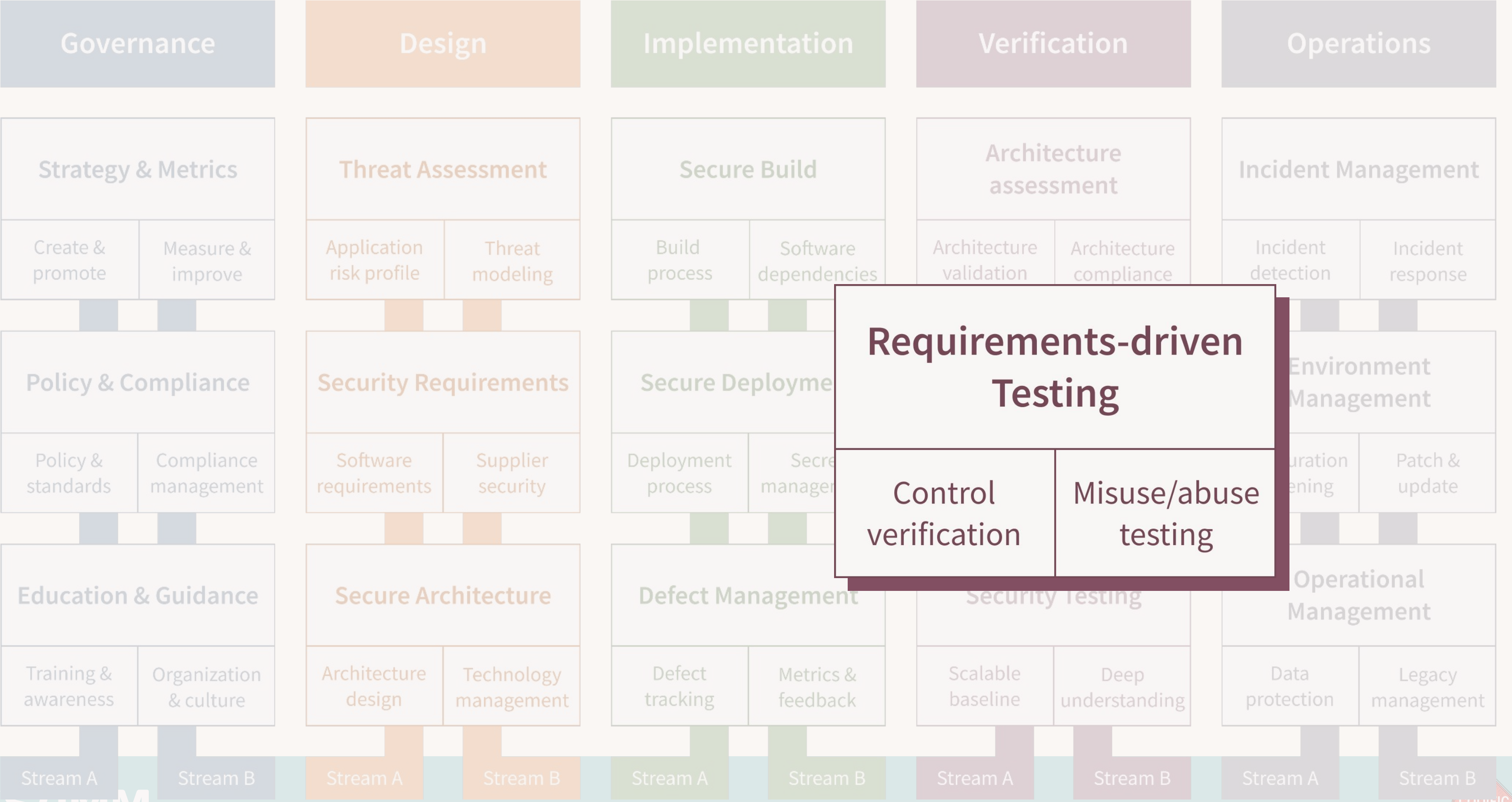
1 Initial understanding and ad hoc provision of the Practice

2 Increase efficiency or effectiveness of the Practice

3 Comprehensive mastery of the Practice at scale



Governance	Design	Implementation	Verification	Operations
Strategy & Metrics	Threat Assessment	Secure Build	Architecture assessment	Incident Management
Create & promote Measure & improve	Application risk profile Threat modeling	Build process Software dependencies	Architecture validation Architecture compliance	Incident detection Incident response
Policy & Compliance	Security Requirements	Secure Deployment	Requirements-driven Testing	Environment Management
Policy & standards Compliance management	Software requirements Supplier security	Deployment process Secret management	Control verification Misuse/abuse testing	Configuration hardening Patch & update
Education & Guidance	Secure Architecture	Defect Management	Security Testing	Operational Management
Training & awareness Organization & culture	Architecture design Technology management	Defect tracking Metrics & feedback	Scalable baseline Deep understanding	Data protection Legacy management
Stream A Stream B	Stream A Stream B	Stream A Stream B	Stream A Stream B	Stream A Stream B



Requirements-driven Testing

Control verification	Misuse/abuse testing

Governance	Design	Implementation	Verification	Operations
Strategy & Metrics	Threat Assessment	Secure Build	Architecture assessment	Incident Management
Create & promote Measure & improve	Application risk profile Threat modeling	Build process Software dependencies	Architecture validation Architecture compliance	Incident detection Incident response
Policy & Compliance	Security Requirements	Secure Deployme		Environment Management
Policy & standards Compliance management	Software requirements Supplier security	Deployment process Secre manager		uration ening Patch & update
Education & Guidance	Secure Architecture	Defect Management	Security Testing	Operational Management
Training & awareness Organization & culture	Architecture design Technology management	Defect tracking Metrics & feedback	Scalable baseline Deep understanding	Data protection Legacy management
Stream A Stream B	Stream A Stream B	Stream A Stream B	Stream A Stream B	Stream A Stream B

Security practice structure

Maturity Level		STREAM A **Control Verification**	STREAM B **Misuse/Abuse Testing**
1 ●○○	Opportunistically find basic vulnerabilities and other security issues	Test for software security controls	Perform security fuzzing testing
2 ●●○	Perform implementation review to discover application-specific risks against the security requirements	Derive test cases from known security requirements	Create and test abuse cases and business logic flaw test
3 ●●●	Maintain the application security level after bug fixes, changes or during maintenance	Perform regression testing (with security unit tests)	Denial of service and security stress testing

The model as an assessment tool

OWASP SAMM

Assessment process

One question per activity

Do you perform the activity in the organization*?

Two-dimensional assessment of activities

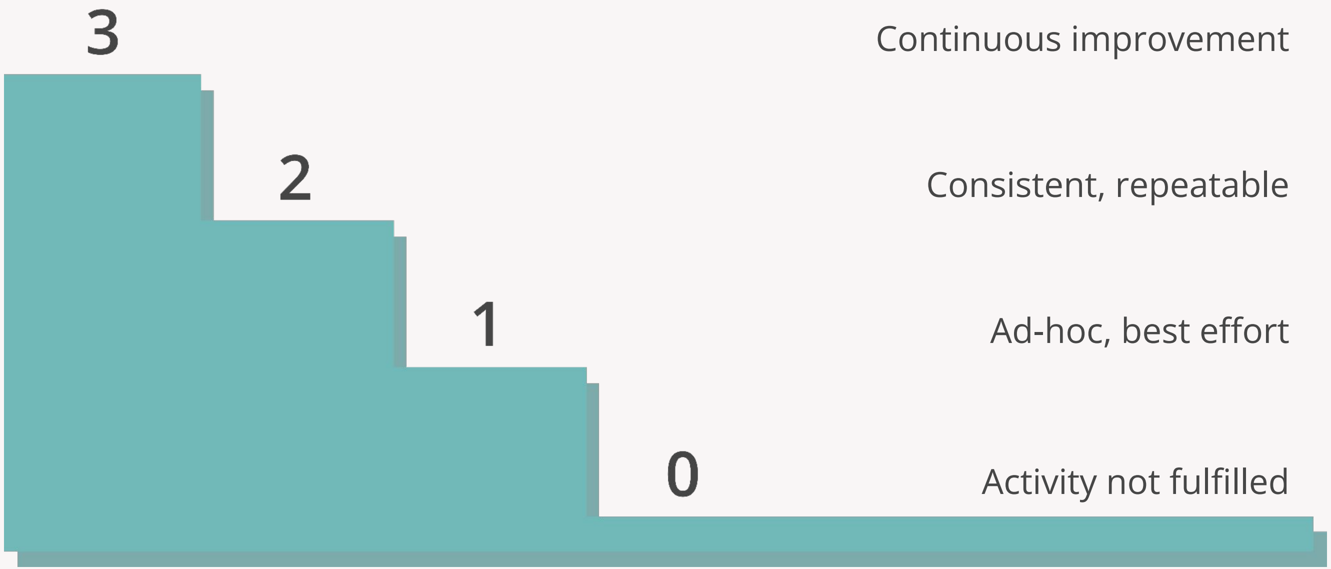
Coverage

Across what portion of the organization you perform the activity

Quality

Criteria you must meet before counting towards coverage

Assessment – maturity levels



Continuous improvement

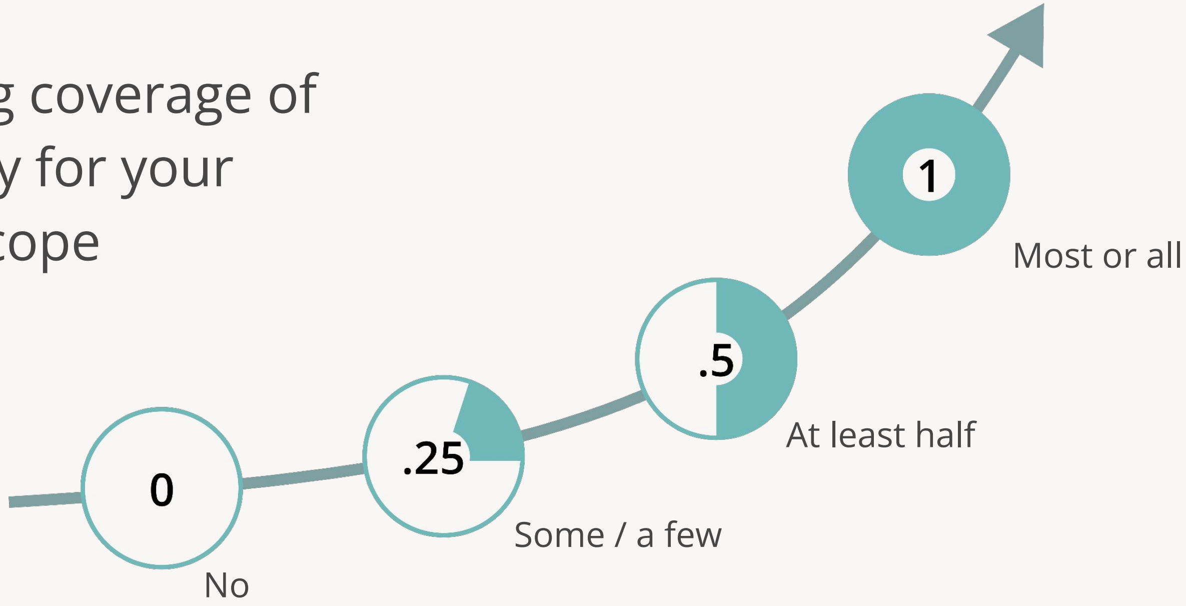
Consistent, repeatable

Ad-hoc, best effort

Activity not fulfilled

Assessment - scores

Measuring coverage of the activity for your defined scope



Example activity

Maturity Level		STREAM A Control Verification	STREAM B Misuse/Abuse Testing
1 ●○○	Opportunistically find basic vulnerabilities and other security issues	Test for software security controls	Perform security fuzzing testing
2 ●●○	Perform implementation review to discover application-specific risks against the security requirements	Derive test cases from known security requirements	Create and test abuse cases and business logic flaw test
3 ●●●	Maintain the application security level after bug fixes, changes or during maintenance	Perform regression testing (with security unit tests)	Denial of service and security stress testing

Example question

Do you test applications for the correct functioning of standard security controls?

Answers

- No
- Yes, some of them
- Yes, at least half of them
- Yes, most of them

Quality criteria

Security testing at least verifies the implementation of authentication, access control, input validation, encoding and escaping data, and encryption controls

Security testing executes whenever the application changes its use of the controls

Creating scorecards

Gap analysis

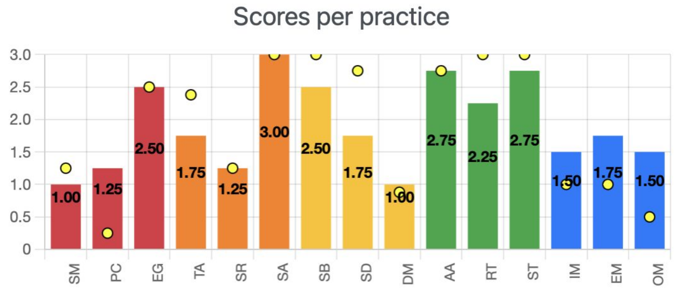
Capturing scores from detailed assessments versus expected performance levels

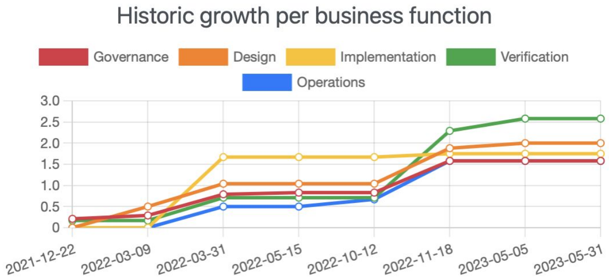
Demonstrating improvement

Capturing scores from before and after an iteration of assurance program build-out

Ongoing measurements

Capturing scores over consistent timeframe for an assurance program already in place

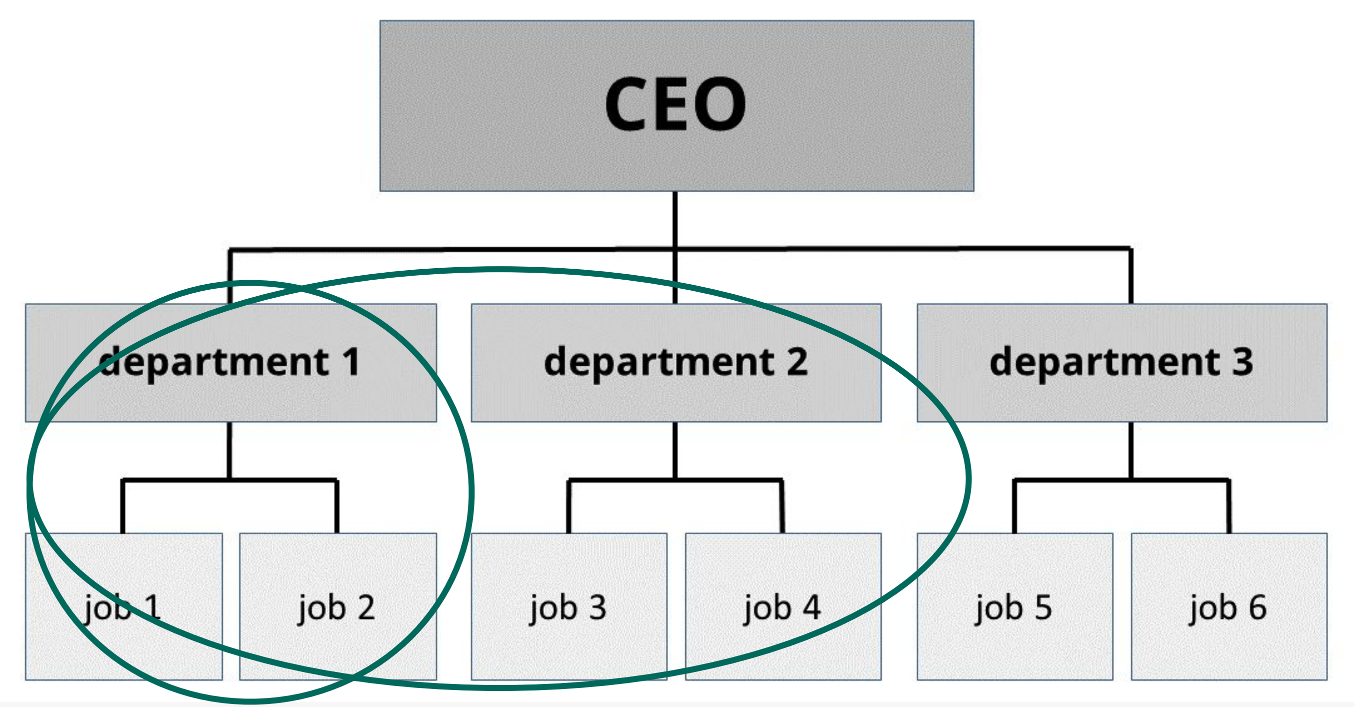




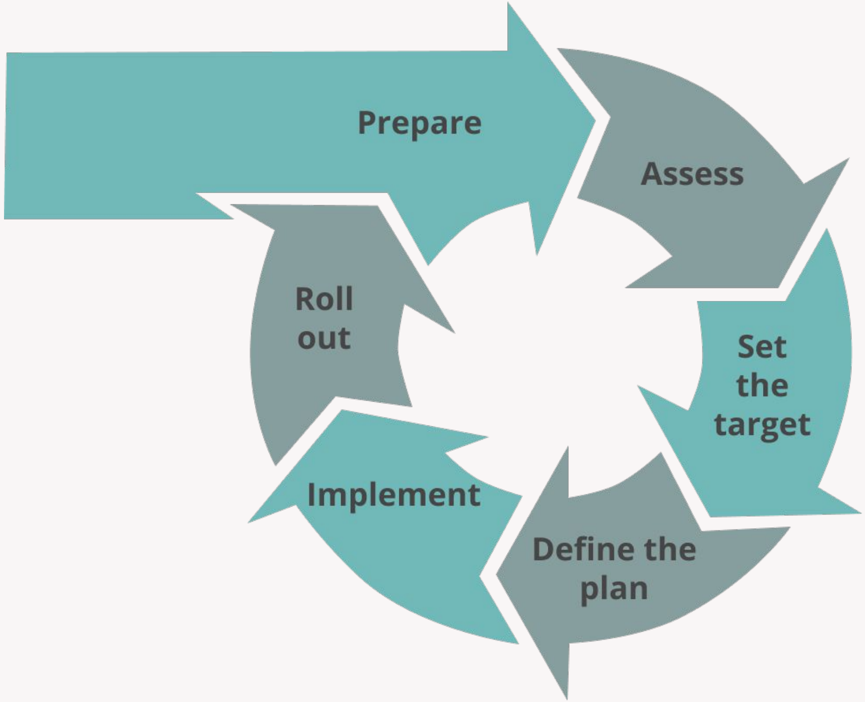
Methodology

for using the model

Set your scope



Methodology – steps



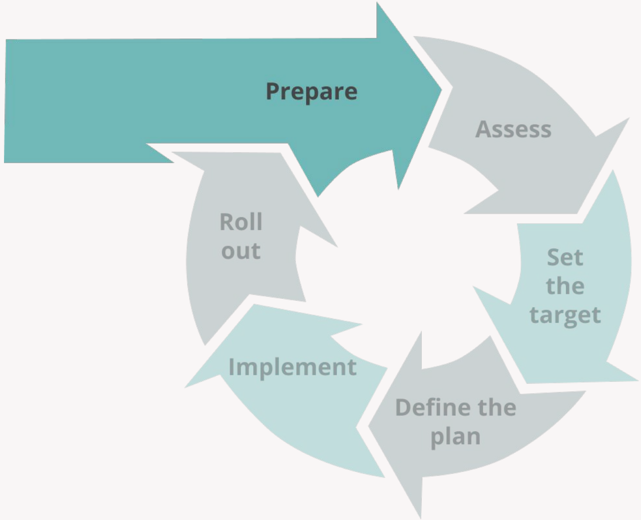
Prepare

Purpose

Ensure a proper start of the project

Activities

- Define the scope
- Identify stakeholders
- Socialize – spread the word!



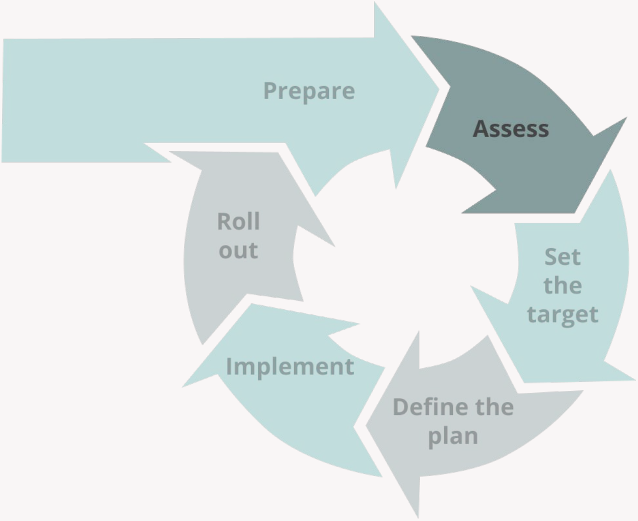
Assess

Purpose

Identify and understand the maturity in each of the 15 practices for the chosen scope

Activities

- Evaluate current practices
- Determine maturity level



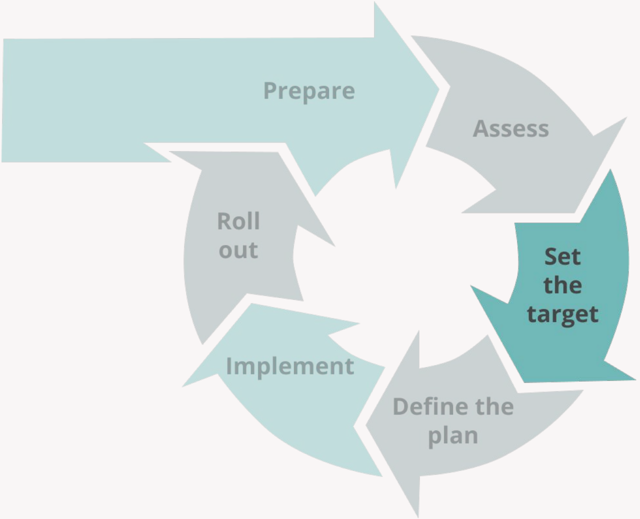
Set the target

Purpose

Develop a target score to guide you in future improvements

Activities

- Define the target
- Estimate overall impact



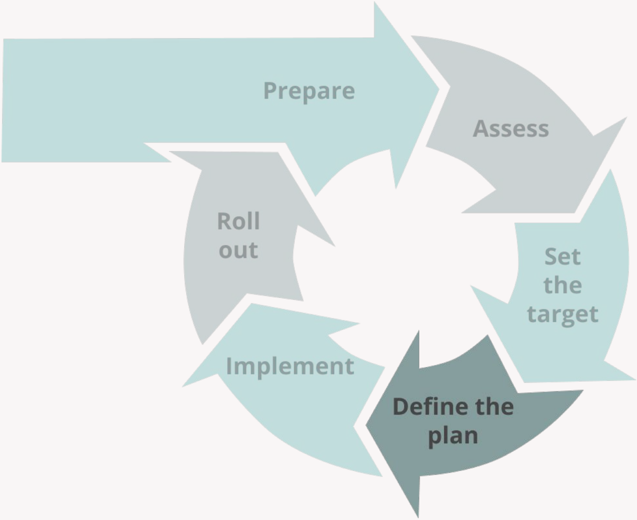
Define the plan

Purpose

Define or update the plan to take you to the next level

Activities

- Determine change schedule
- Develop/update the roadmap plan



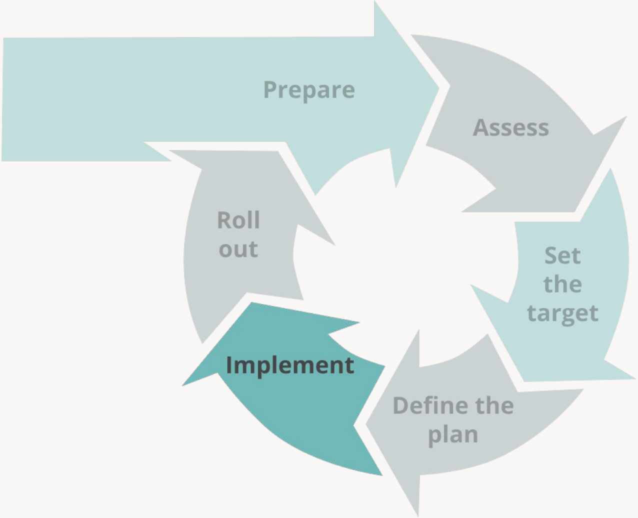
Implement

Purpose

Work the plan

Activities

- Implement activities



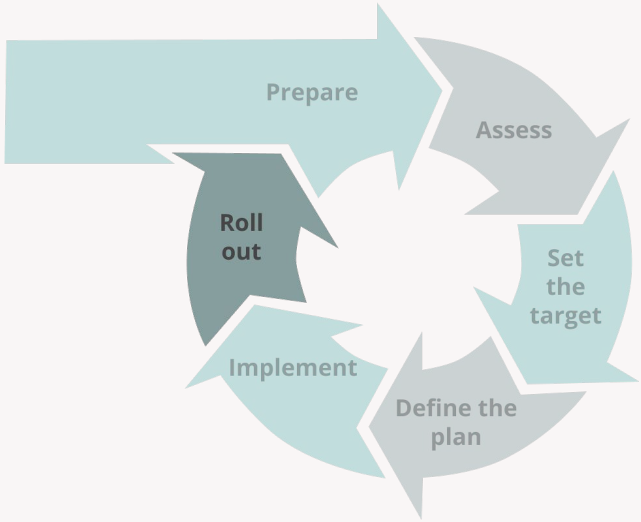
Roll out

Purpose

Ensure improvements are available and effectively used

Activities

- Evangelize improvements
- Measure effectiveness



Secure Build

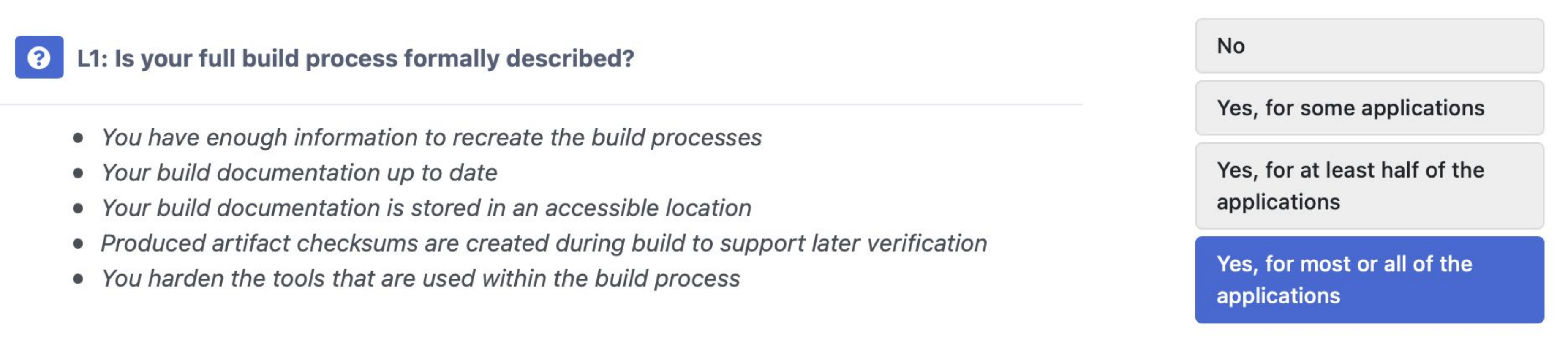
Demo

Secure Build

Stream A: Build Process

Stream B: Software Dependencies

Build Process



L1: Is your full build process formally described?

- *You have enough information to recreate the build processes*
- *Your build documentation up to date*
- *Your build documentation is stored in an accessible location*
- *Produced artifact checksums are created during build to support later verification*
- *You harden the tools that are used within the build process*

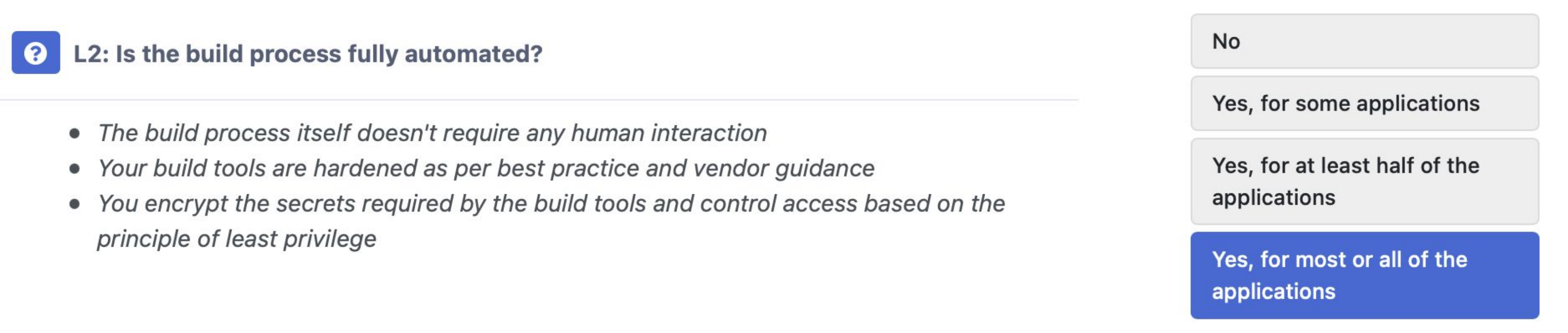
No

Yes, for some applications

Yes, for at least half of the applications

[✓ Yes, for most or all of the applications]

Build Process



L2: Is the build process fully automated?

- *The build process itself doesn't require any human interaction*
- *Your build tools are hardened as per best practice and vendor guidance*
- *You encrypt the secrets required by the build tools and control access based on the principle of least privilege*

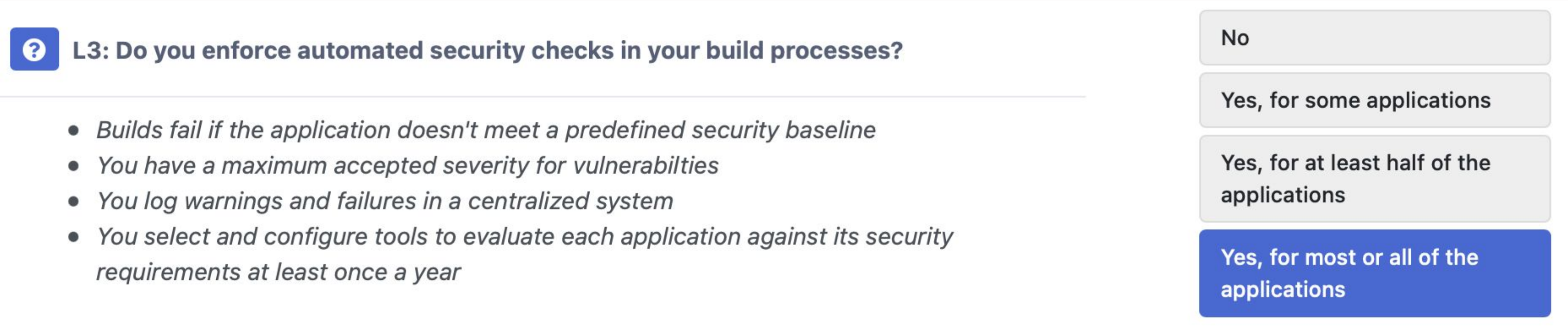
No

Yes, for some applications

Yes, for at least half of the applications

[✓ Yes, for most or all of the applications]

Build Process



L3: Do you enforce automated security checks in your build processes?

- *Builds fail if the application doesn't meet a predefined security baseline*
- *You have a maximum accepted severity for vulnerabilties*
- *You log warnings and failures in a centralized system*
- *You select and configure tools to evaluate each application against its security requirements at least once a year*

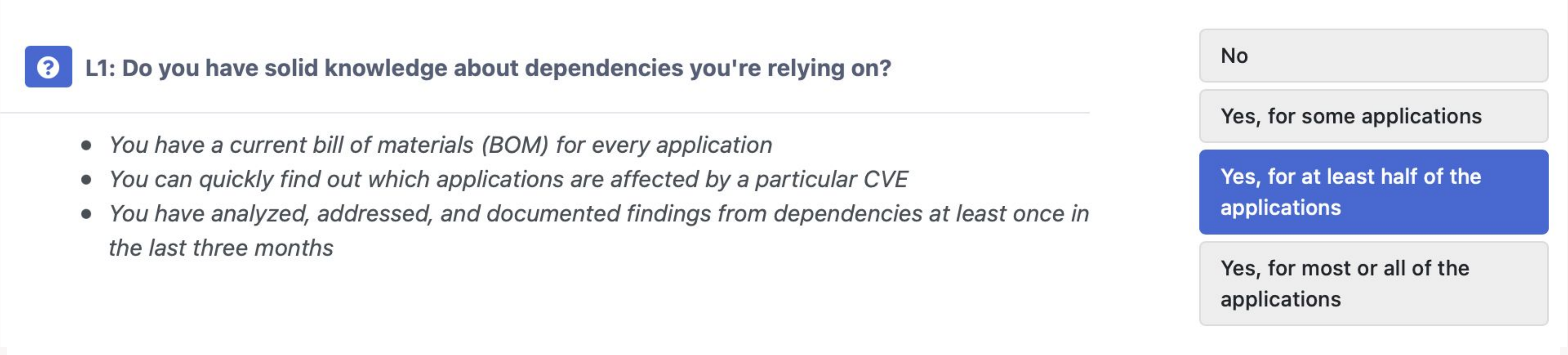
No

Yes, for some applications

Yes, for at least half of the applications

[✓ Yes, for most or all of the applications]

Software Dependencies



L1: Do you have solid knowledge about dependencies you're relying on?

- *You have a current bill of materials (BOM) for every application*
- *You can quickly find out which applications are affected by a particular CVE*
- *You have analyzed, addressed, and documented findings from dependencies at least once in the last three months*

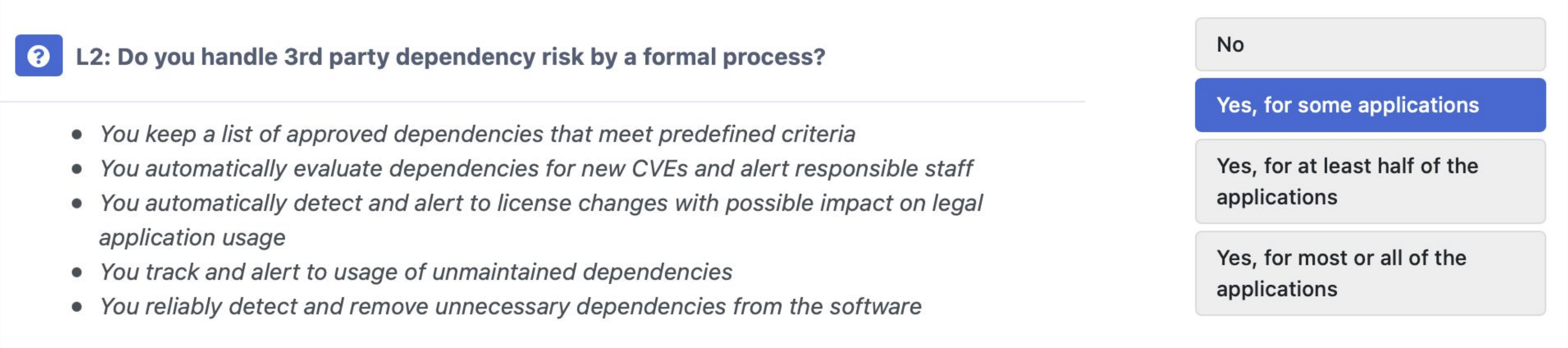
No

Yes, for some applications

[✓ Yes, for at least half of the applications]

Yes, for most or all of the applications

Software Dependencies



L2: Do you handle 3rd party dependency risk by a formal process?

- *You keep a list of approved dependencies that meet predefined criteria*
- *You automatically evaluate dependencies for new CVEs and alert responsible staff*
- *You automatically detect and alert to license changes with possible impact on legal application usage*
- *You track and alert to usage of unmaintained dependencies*
- *You reliably detect and remove unnecessary dependencies from the software*

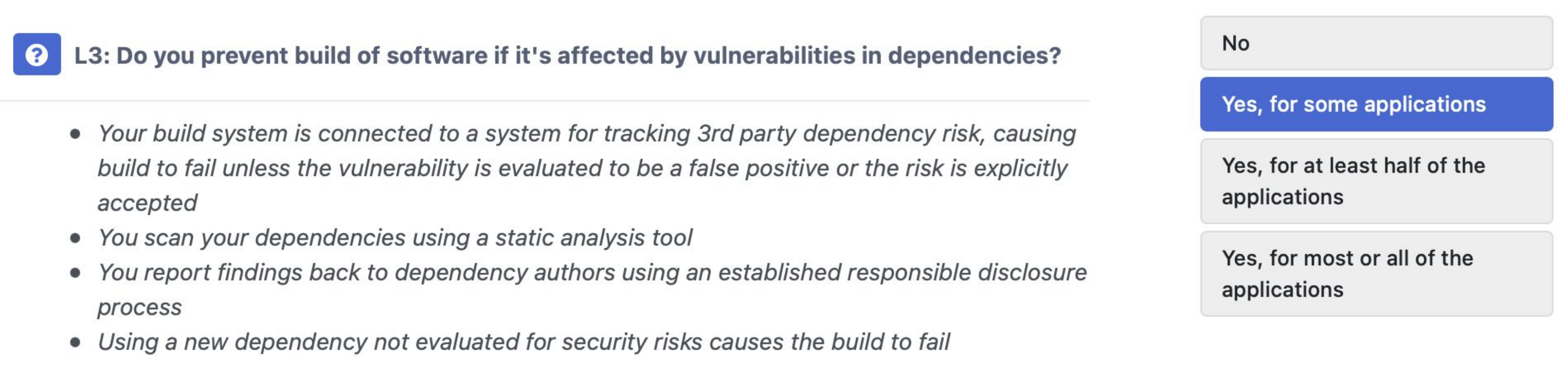
No

[✓ Yes, for some applications]

Yes, for at least half of the applications

Yes, for most or all of the applications

Software Dependencies



L3: Do you prevent build of software if it's affected by vulnerabilities in dependencies?

- *Your build system is connected to a system for tracking 3rd party dependency risk, causing build to fail unless the vulnerability is evaluated to be a false positive or the risk is explicitly accepted*
- *You scan your dependencies using a static analysis tool*
- *You report findings back to dependency authors using an established responsible disclosure process*
- *Using a new dependency not evaluated for security risks causes the build to fail*

No

[✓ Yes, for some applications]

Yes, for at least half of the applications

Yes, for most or all of the applications

Wrap-up

OWASP SAMM

Conclusion

- Application Security is a challenging problem
    - Complex
    - Broad
    - Evolving
- AppSec requires a continuous assurance programme
- SAMM is a simple, well-defined and measurable maturity model

OWASP
Open Web Application
Security Project

OWASP SAMM

Thank you!

https://www.linkedin.com/in/aramhovsep

https://www.linkedin.com/company/codific