OWASP
Open Web Application
Security Project

OWASP SAMM v2.0 and SAMM Benchmark

Thank You to Our Sponsors and Hosts!

OWASP
NEW
ZEALAND
owasp.org.nz

QUANTUM
SECURITY

AUT

CyberCX

DATACOM

snyk

Auth0

Checkmarx

HCL AppScan

kordia®

LATERAL
SECURITY

MICRO®
FOCUS

Pulse Security
www.pulsesecurity.co.nz

RedShield

Flux

SEQA™
Information Security

Cobalt

LACEWORK.

SecureFlag

Without them, OWASP New Zealand Day couldn't happen

INTRODUCTIONS – BRIAN GLAS

- Married father of four
- Assistant Prof of Comp Sci/CyberSec
- 21+ yrs of Development-Security Experience
- Contributing to:
  - OWASP SAMM and SAMM Benchmark
  - Top 10 2017 & 2021

@infosecdad
brian.glas@gmail.com

What is SAMM?

- The Software Assurance Maturity Model (SAMM) is an open framework to help organizations formulate and implement a strategy for software security that is tailored to the specific risks facing the organization.

- The resources provided by SAMM will aid in:
  - *Evaluating an organization’s existing software security practices.*
  - *Building a balanced software security assurance program in well-defined iterations.*
  - *Demonstrating concrete improvements to a security assurance program.*
  - *Defining and measuring security-related activities throughout an organization.*

Who is SAMM?

- Sebastien (Seba) Deleersnyder – Project Co-Leader, Belgium
- Bart De Win – Project Co-Leader, Belgium
- Brian Glas – United States
- Daniel Kefer – Germany
- Yan Kravchenko – United States
- Chris Cooper – United Kingdom
- John DiLeo – New Zealand
- Nessim Kisserli – Belgium
- Patricia Duarte – Uruguay
- John Kennedy – Sweden
- Hardik Parekh - United States
- John Ellingsworth - United States
- Sebastian Arriada - Argentina
- Brett Crawley – United Kingdom

CONCORD MICRO FOCUS Fortify nccgroup SECURITY INNOVATION splunk>

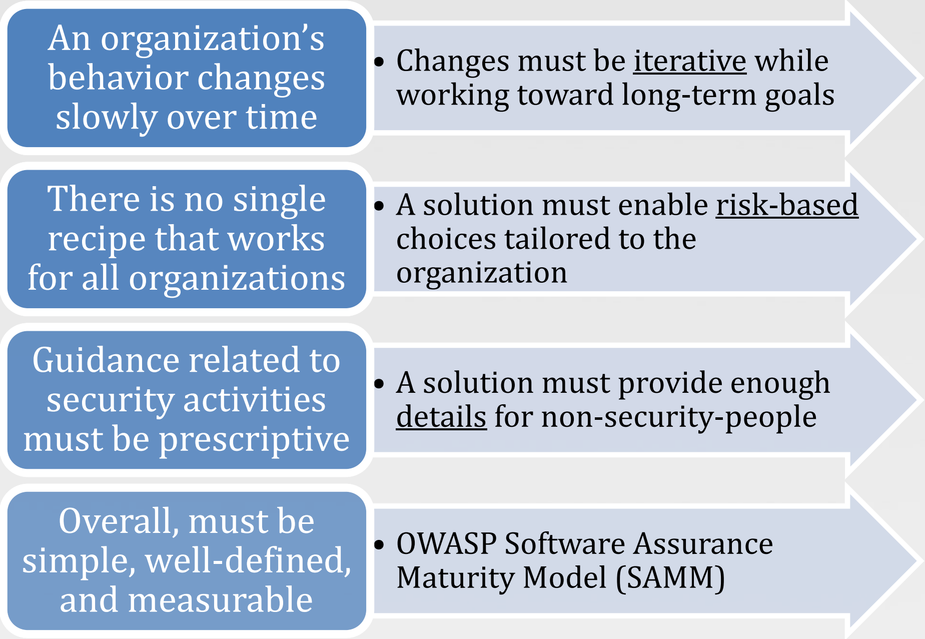
Sponsors:

Why SAMM?

”The most that can be expected from any model is that it can supply a useful approximation to reality: All models are wrong; some models are useful.”

– George E. P. Box

Core Principles of 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

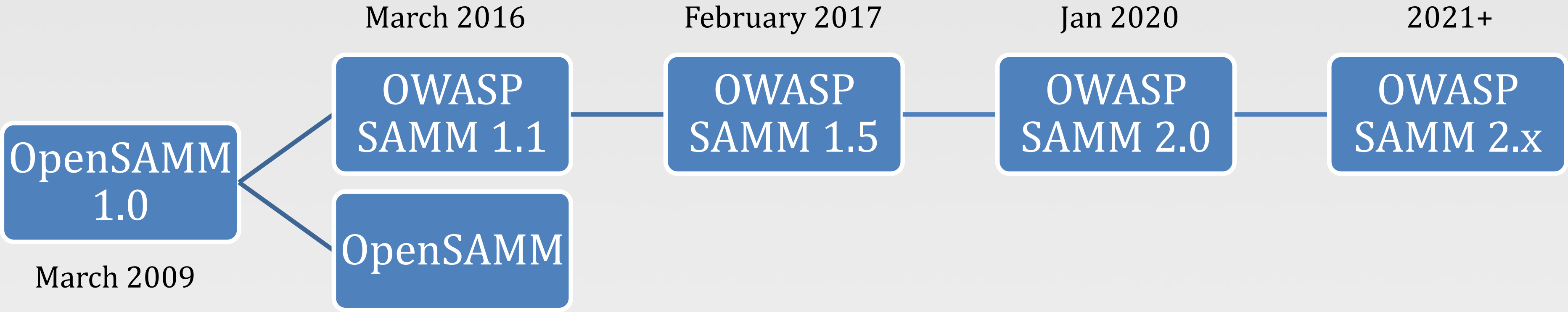
Guidance related to security activities must be prescriptive

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

Overall, must be simple, well-defined, and measurable

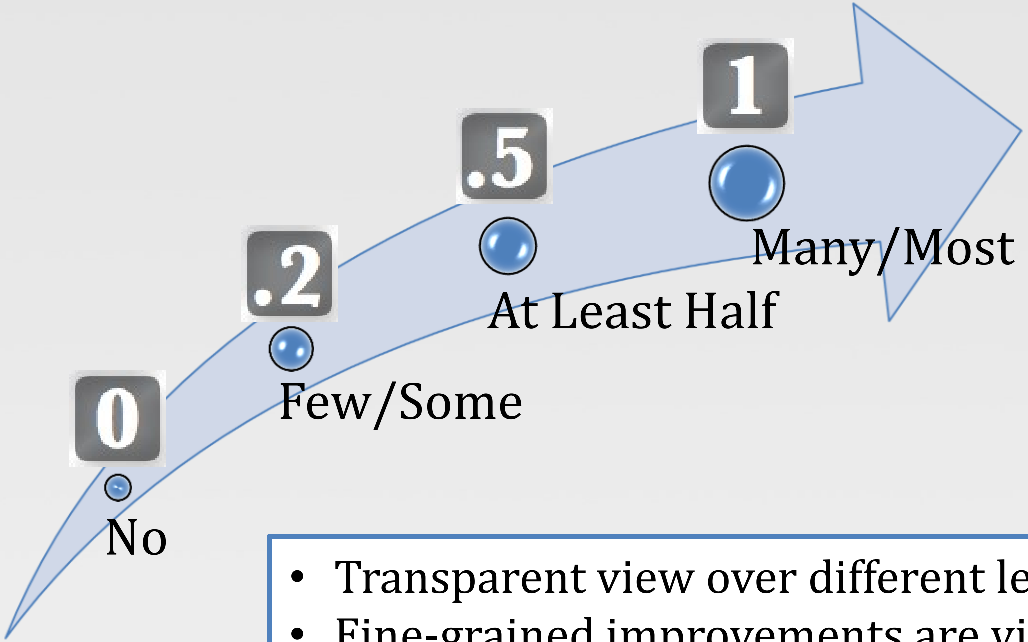
- OWASP Software Assurance Maturity Model (SAMM)

Project History



Maturity Levels & Assessment Scores

- 3 Comprehensive mastery at scale
- 2 Increased efficiency/effectiveness
- 1 Ad-hoc provision
- 0 Practice unfulfilled



0 No

.2 Few/Some

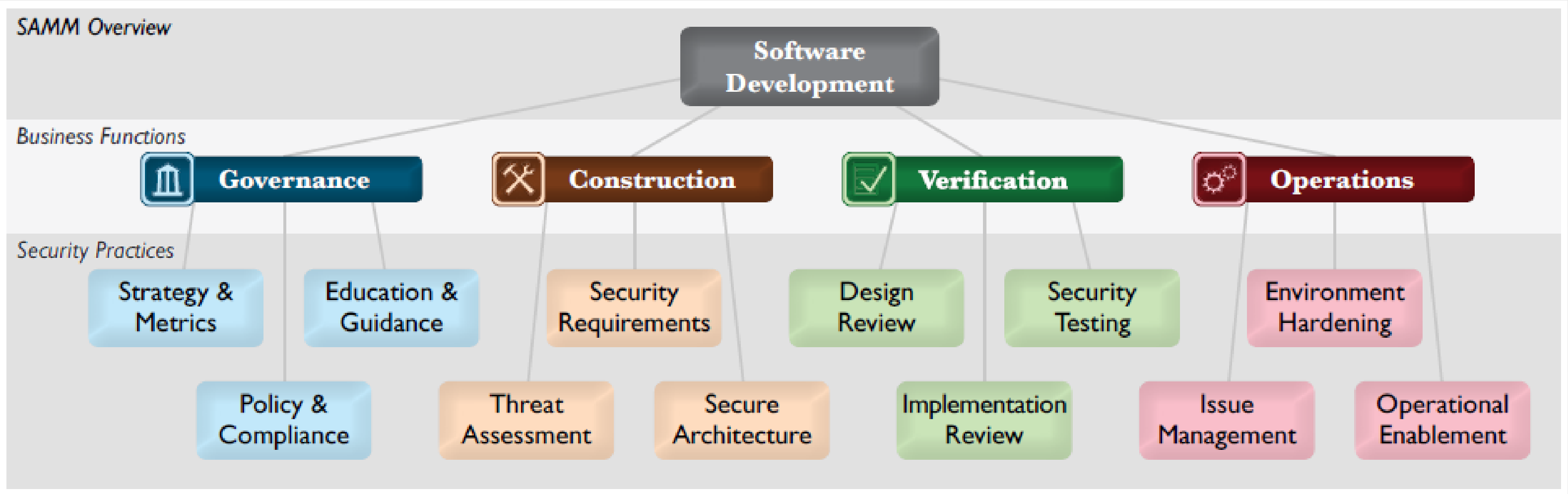
.5 At Least Half

1 Many/Most

- Transparent view over different levels
- Fine-grained improvements are visible

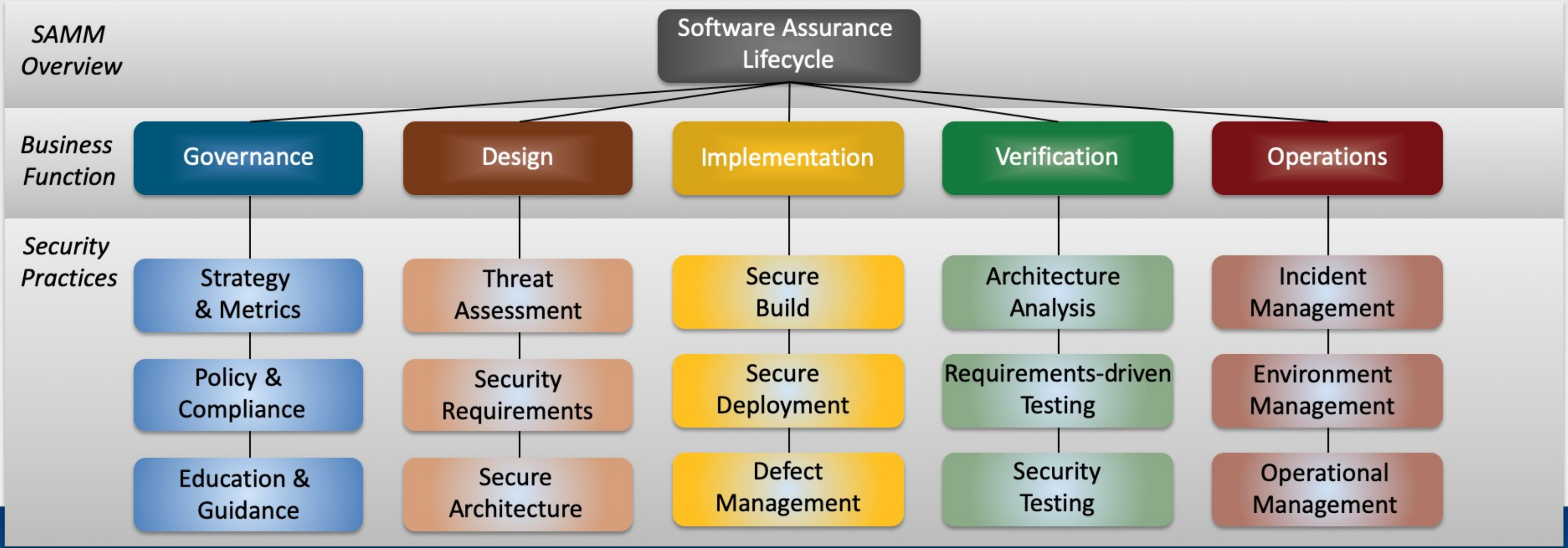
SAMM Framework v1.5

- For each of the four Business Functions, three Security Practices are defined
- The security practices cover areas relevant to software security assurance



SAMM Framework v2.0

- For each of the five Business Functions, three Security Practices are defined
- The security practices cover areas relevant to software security assurance



SAMM Overview

Software Assurance Lifecycle

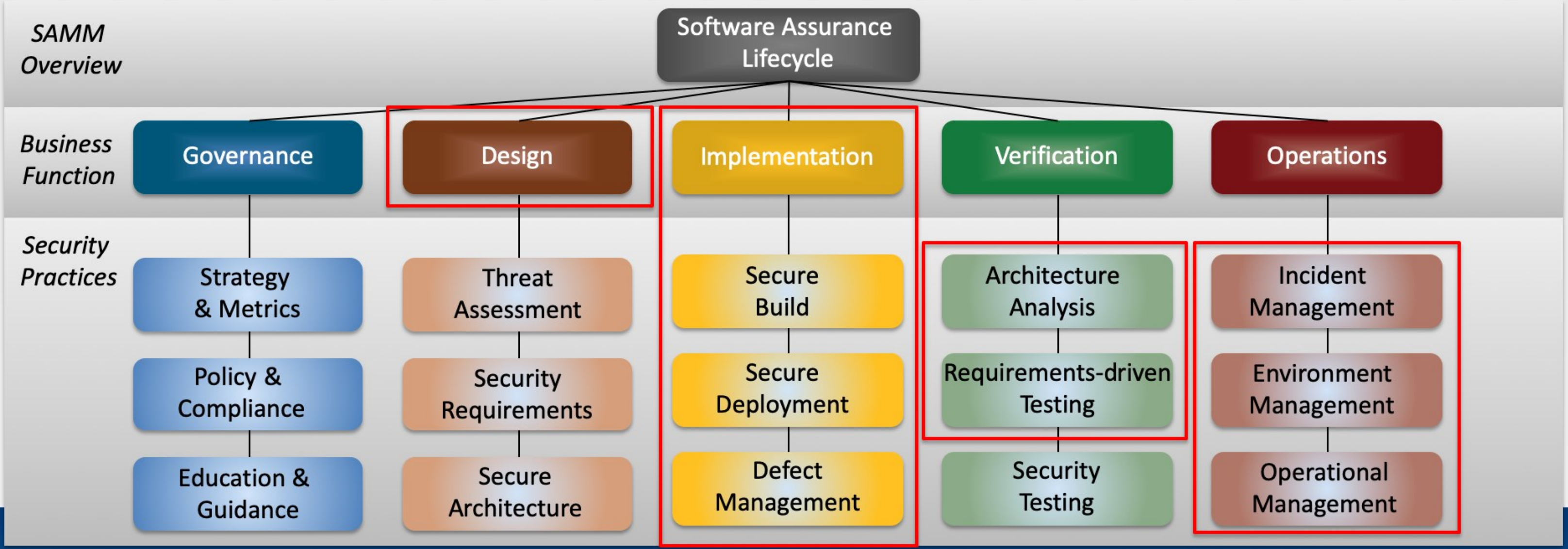
Business Function

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

Security Practices

SAMM Framework v2.0

- For each of the five Business Functions, three Security Practices are defined
- The security practices cover areas relevant to software security assurance

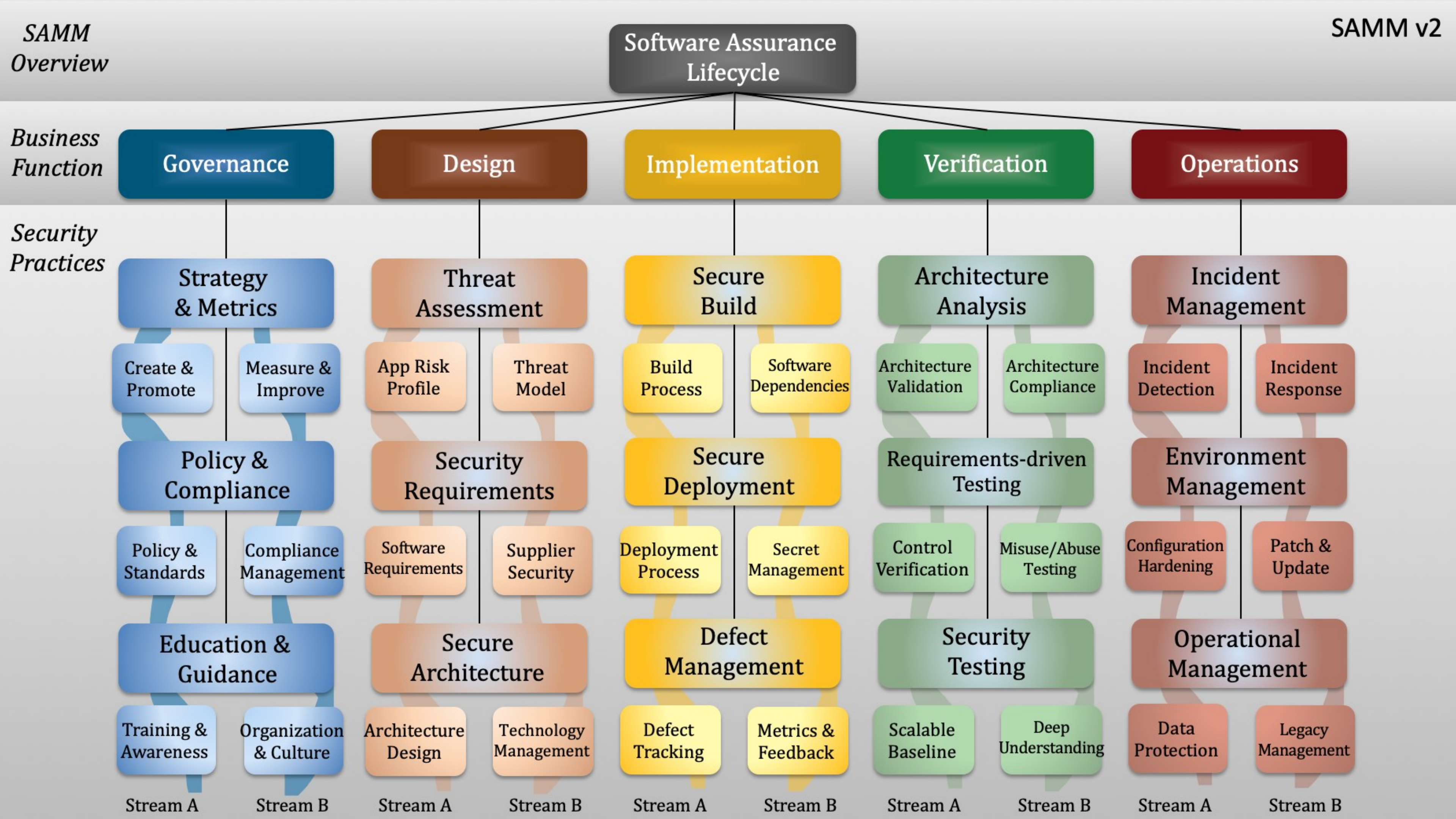


SAMM Overview: Software Assurance Lifecycle

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

SAMM Overview

SAMM v2



Software Assurance Lifecycle

Business Function

Security Practices

Governance

- Strategy & Metrics
  - Create & Promote (Stream A)
  - Measure & Improve (Stream B)
- Policy & Compliance
  - Policy & Standards (Stream A)
  - Compliance Management (Stream B)
- Education & Guidance
  - Training & Awareness (Stream A)
  - Organization & Culture (Stream B)

Design

- Threat Assessment
  - App Risk Profile (Stream A)
  - Threat Model (Stream B)
- Security Requirements
  - Software Requirements (Stream A)
  - Supplier Security (Stream B)
- Secure Architecture
  - Architecture Design (Stream A)
  - Technology Management (Stream B)

Implementation

- Secure Build
  - Build Process (Stream A)
  - Software Dependencies (Stream B)
- Secure Deployment
  - Deployment Process (Stream A)
  - Secret Management (Stream B)
- Defect Management
  - Defect Tracking (Stream A)
  - Metrics & Feedback (Stream B)

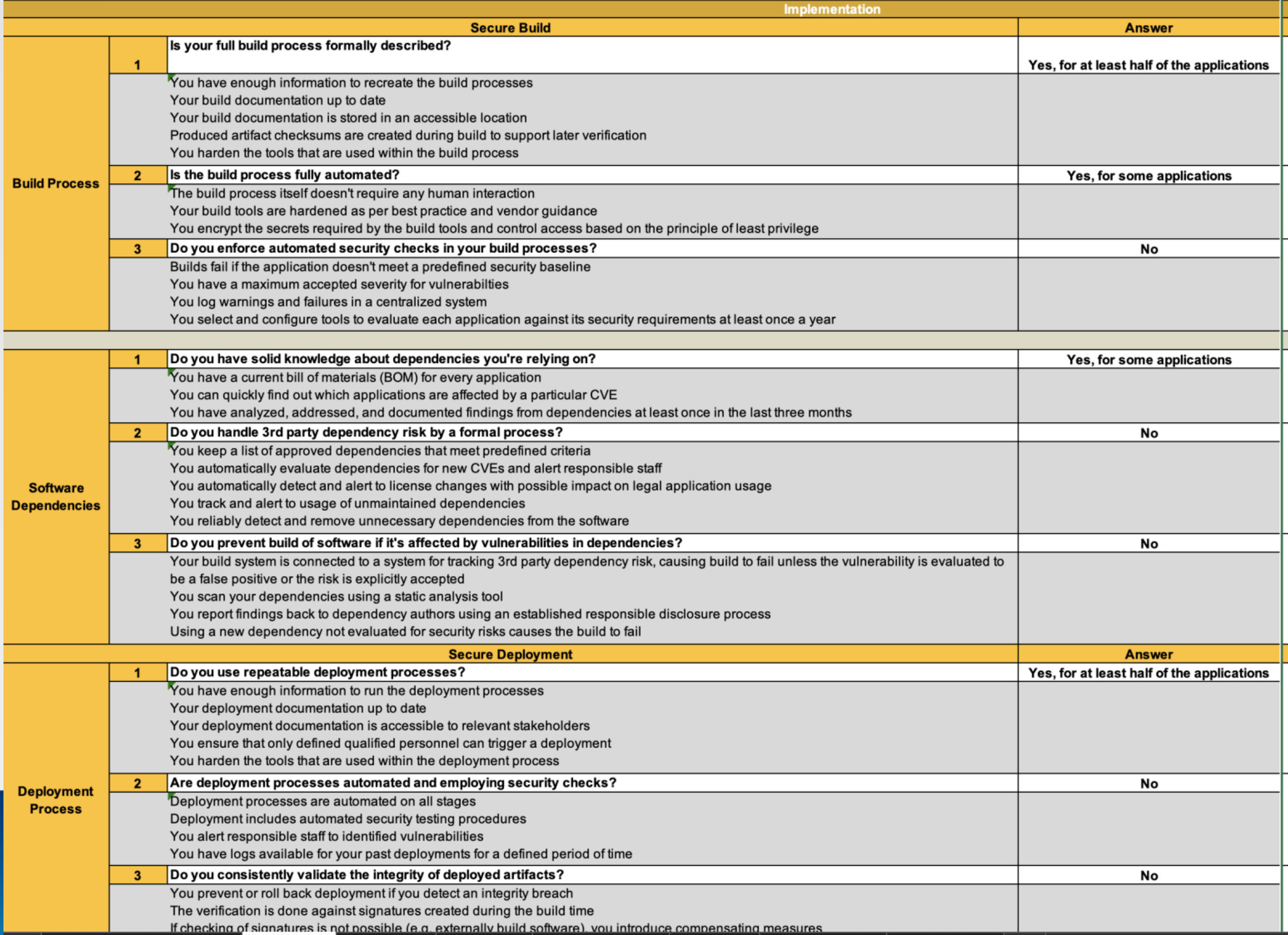
Verification

- Architecture Analysis
  - Architecture Validation (Stream A)
  - Architecture Compliance (Stream B)
- Requirements-driven Testing
  - Control Verification (Stream A)
  - Misuse/Abuse Testing (Stream B)
- Security Testing
  - Scalable Baseline (Stream A)
  - Deep Understanding (Stream B)

Operations

- Incident Management
  - Incident Detection (Stream A)
  - Incident Response (Stream B)
- Environment Management
  - Configuration Hardening (Stream A)
  - Patch & Update (Stream B)
- Operational Management
  - Data Protection (Stream A)
  - Legacy Management (Stream B)

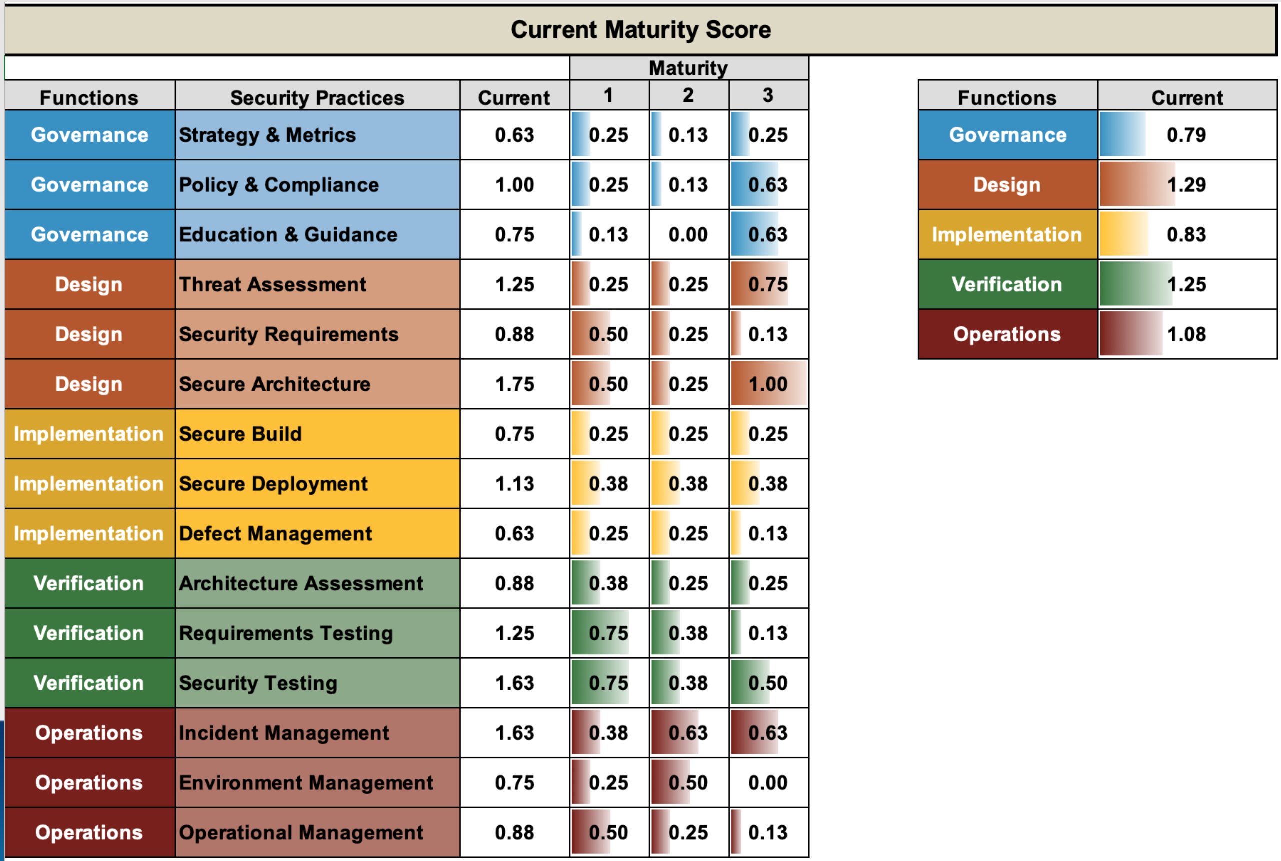
Assess via Toolbox



Implementation			
		Secure Build	**Answer**
Build Process	1	**Is your full build process formally described?**	Yes, for at least half of the applications
		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	
	2	**Is the build process fully automated?**	Yes, for some applications
		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	
	3	**Do you enforce automated security checks in your build processes?**	No
		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	
Software Dependencies	1	**Do you have solid knowledge about dependencies you're relying on?**	Yes, for some applications
		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	
	2	**Do you handle 3rd party dependency risk by a formal process?**	No
		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	
	3	**Do you prevent build of software if it's affected by vulnerabilities in dependencies?**	No
		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	
		Secure Deployment	**Answer**
Deployment Process	1	**Do you use repeatable deployment processes?**	Yes, for at least half of the applications
		You have enough information to run the deployment processes Your deployment documentation up to date Your deployment documentation is accessible to relevant stakeholders You ensure that only defined qualified personnel can trigger a deployment You harden the tools that are used within the deployment process	
	2	**Are deployment processes automated and employing security checks?**	No
		Deployment processes are automated on all stages Deployment includes automated security testing procedures You alert responsible staff to identified vulnerabilities You have logs available for your past deployments for a defined period of time	
	3	**Do you consistently validate the integrity of deployed artifacts?**	No
		You prevent or roll back deployment if you detect an integrity breach The verification is done against signatures created during the build time If checking of signatures is not possible (e.g. externally build software), you introduce compensating measures	

OWASP
Open Web Application
Security Project

Dashboards



Current Maturity Score

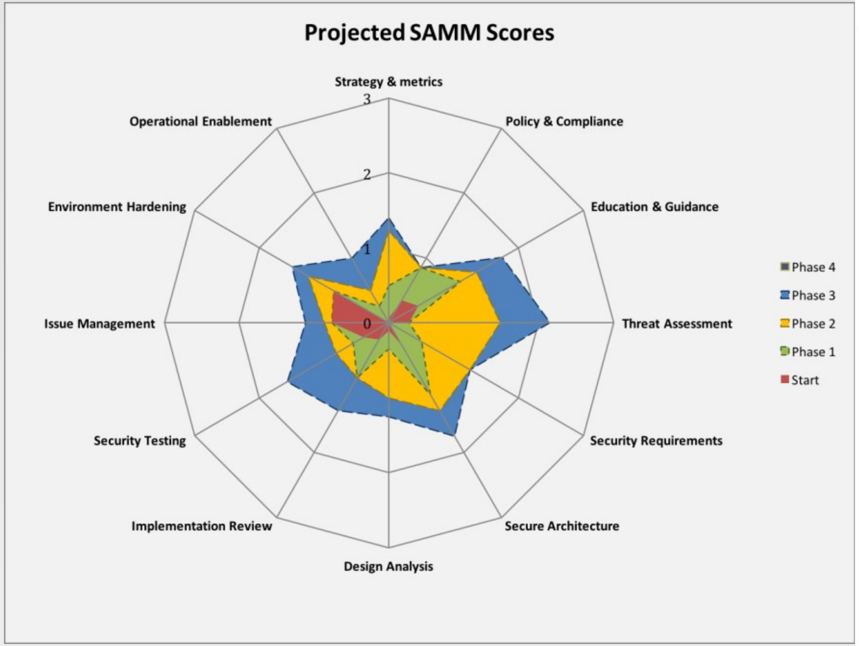
			Maturity		
Functions	Security Practices	Current	1	2	3
Governance	Strategy & Metrics	0.63	0.25	0.13	0.25
Governance	Policy & Compliance	1.00	0.25	0.13	0.63
Governance	Education & Guidance	0.75	0.13	0.00	0.63
Design	Threat Assessment	1.25	0.25	0.25	0.75
Design	Security Requirements	0.88	0.50	0.25	0.13
Design	Secure Architecture	1.75	0.50	0.25	1.00
Implementation	Secure Build	0.75	0.25	0.25	0.25
Implementation	Secure Deployment	1.13	0.38	0.38	0.38
Implementation	Defect Management	0.63	0.25	0.25	0.13
Verification	Architecture Assessment	0.88	0.38	0.25	0.25
Verification	Requirements Testing	1.25	0.75	0.38	0.13
Verification	Security Testing	1.63	0.75	0.38	0.50
Operations	Incident Management	1.63	0.38	0.63	0.63
Operations	Environment Management	0.75	0.25	0.50	0.00
Operations	Operational Management	0.88	0.50	0.25	0.13

Functions	Current
Governance	0.79
Design	1.29
Implementation	0.83
Verification	1.25
Operations	1.08

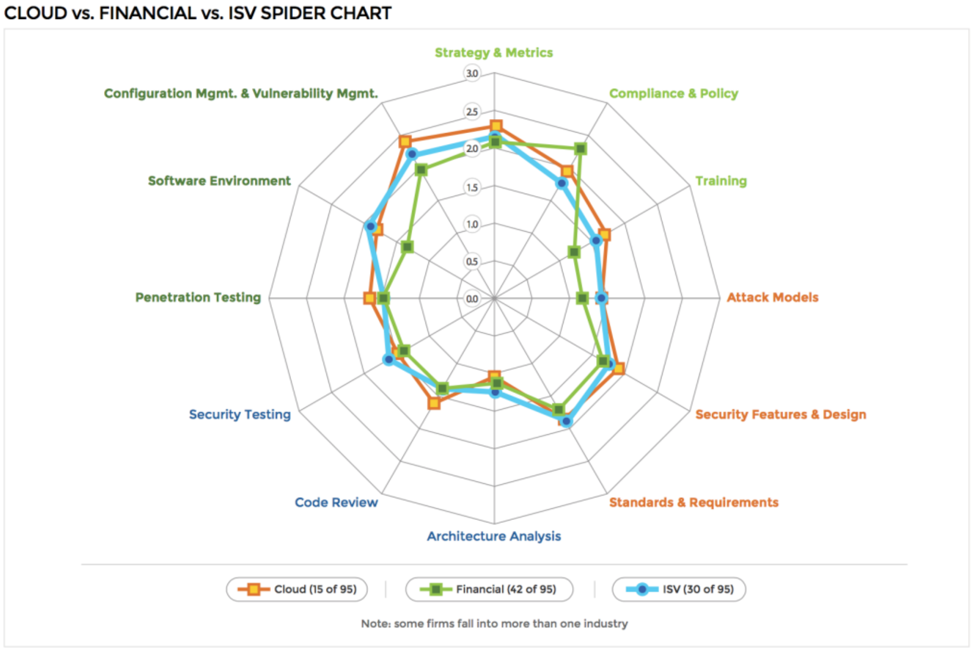
Critical Success Factors

- Get buy-in from stakeholders
- Adopt a risk-based approach
- Awareness & Education is the foundation
- Integrate & automate security in your development, acquisition, and deployment processes
- Measure: Provide Management Visibility

SAMM can(sorta) map to BSIMM



SAMM



BSIMM

SAMM vs BSIMM

- https://owaspsamm.org/blog/2020/10/29/comparing-bsimm-and-samm/

SAMM BENCHMARK

Time to answer the question…

How do I compare?

https://owaspsamm.org/benchmarking/

What is SAMM Benchmark

- The goal of this project is to collect the most comprehensive dataset related to organizational maturity of application or software security programs.
- This data should come from both self-assessing organizations and consultancies that perform third party assessments.

Contribution Infrastructure

- The plan is to leverage Azure Cloud Infrastructure to collect, analyze, and store the data contributed.
- There will be a minimal number of administrators that have access to manage the raw data.
- Dashboards and comparative analysis will be performed with data that is aggregated and/or separated from the submitting organization.

Data Contributions

Verified Data Contribution

- Scenario 1: The submitter is known and has agreed to be identified as a contributing party.
- Scenario 2: The submitter is known but would rather not be publicly identified.
- Scenario 3: The submitter is known but does not want it recorded in the dataset.

Unverified Data Contribution

- Scenario 4: The submitter is anonymous.

Contribution Process

There are a few ways that data can be contributed:

- Email a CSV/Excel/Doc file with the dataset(s) to brian.glas@owasp.org

- Upload a CSV/Excel/Txt file to a "contribution page" (future)
- Complete the web-based form (future)
- Upload the data from the SAMM Toolbox (future)

Data Structure

The following data elements are required* or optional:

- *Contributor Name (org or anon)
- Contributor Contact Email
- *Date assessment conducted (MM/YYYY)
- *Type of Assessment (Self or 3rd Party)
- *Answers to the SAMM Assessment Questions
- Geographic Region (Global, North America, EU, Asia, other)
- Primary Industry (Multiple, Financial, Industrial, Software, ??)
- Language (English, Spanish, etc.)
- SAMM Version (1.5, 2.0, 2.1, etc.)
- Scope of Assessment (Team, Department, Organization, Enterprise)
- Approximate number of developers (1-100, 101-1000, 1001-10000, 10000+)
- Approximate number of primary AppSec (1-5, 6-10, 11-20, 20+)
- Approximate number of secondary AppSec (0-20, 21-50, 51-100, 100+)
- Primary SDL Methodology (Waterfall, Agile, DevOps, Other)

SAMM -> US Election Technology

Center for Internet Security (CIS) and
Election Assistance Commission (EAC)

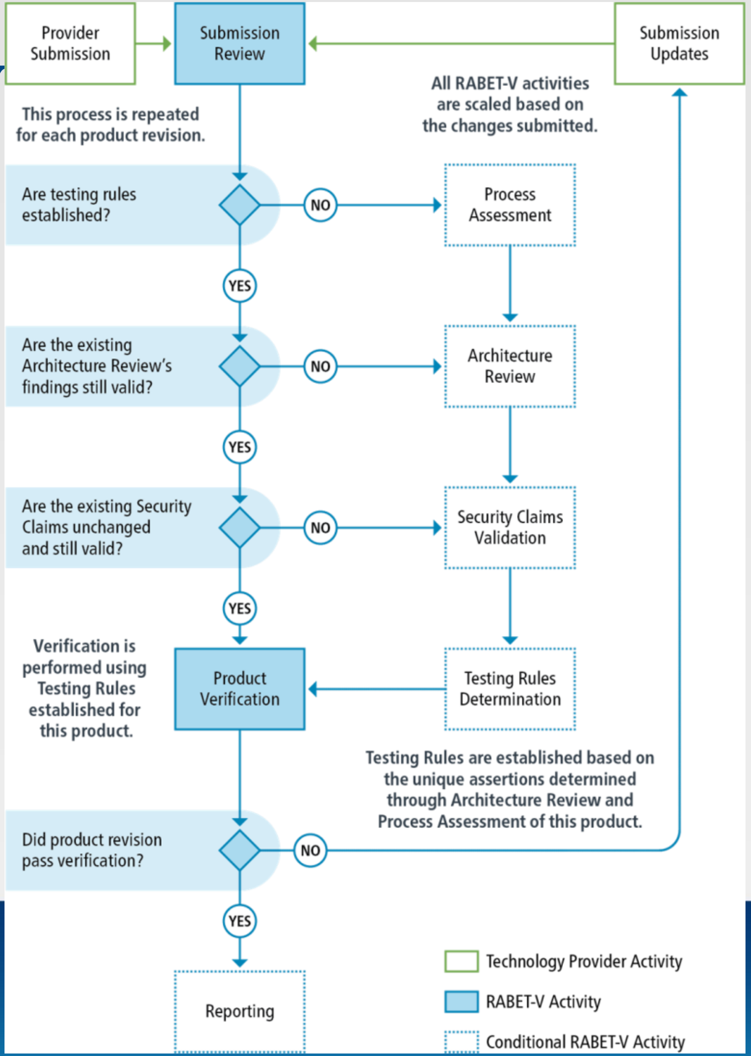
Rapid Architecture-Based Election Technology Verification (RABET-V)

RABET-V Goals

- Produce a flexible, rapid, and cost-efficient process for verifying non-voting election systems for
  - Initial product version
  - Subsequent product revisions
- Non-Voting Election Technology
  - Electronic Pollbooks
  - Election Night Reporting
  - Electronic Ballot Delivery
  - Other internet-connected election administration technology

RABET-V Process Flow

- RABET-V is a total of eight activities
  - Initial Submission – all activities
  - Product Revision Submission – varies
- The **Process Assessment**, **Architecture Review**, and **Security Claims Validation** activities provide assertions about the system's construction which inform the **Testing Rules Determination**
- **Product Verification** verifies security claims and basic product functionality



Provider Submission

Submission Review

Submission Updates

This process is repeated for each product revision.

All RABET-V activities are scaled based on the changes submitted.

Are testing rules established?

NO

Process Assessment

YES

Are the existing Architecture Review's findings still valid?

NO

Architecture Review

YES

Are the existing Security Claims unchanged and still valid?

NO

Security Claims Validation

YES

Verification is performed using Testing Rules established for this product.

Product Verification

Testing Rules Determination

Testing Rules are established based on the unique assertions determined through Architecture Review and Process Assessment of this product.

Did product revision pass verification?

NO

YES

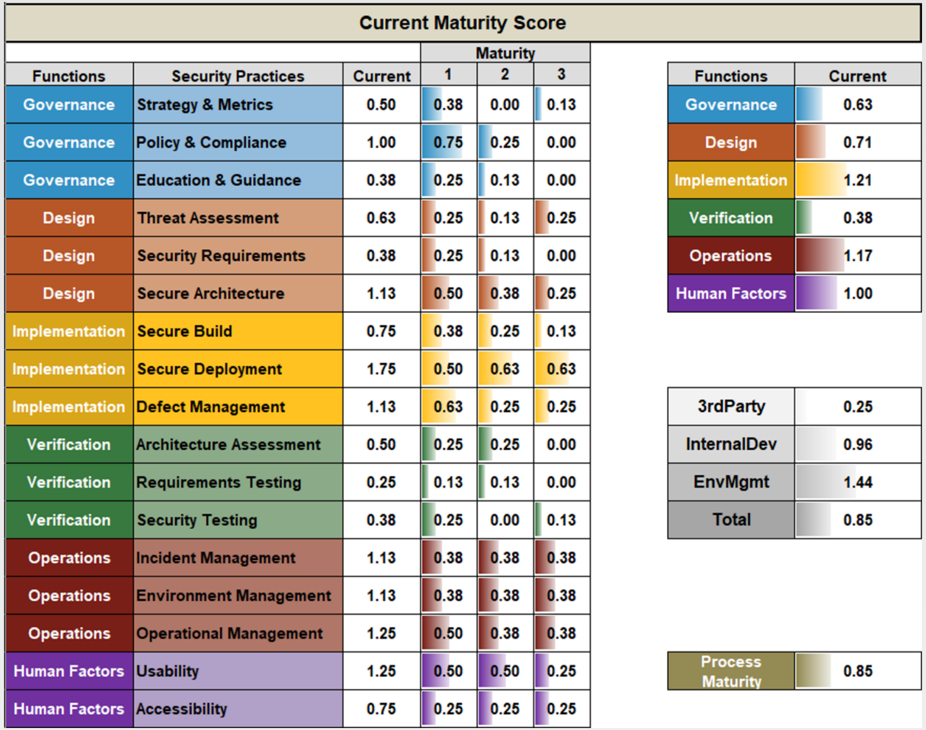
Reporting

□ Technology Provider Activity

■ RABET-V Activity

⬚ Conditional RABET-V Activity

Software Development Maturity (SDM)



Current Maturity Score

Functions	Security Practices	Current	Maturity 1	 2	 3
Governance	Strategy & Metrics	0.50	0.38	0.00	0.13
Governance	Policy & Compliance	1.00	0.75	0.25	0.00
Governance	Education & Guidance	0.38	0.25	0.13	0.00
Design	Threat Assessment	0.63	0.25	0.13	0.25
Design	Security Requirements	0.38	0.25	0.13	0.00
Design	Secure Architecture	1.13	0.50	0.38	0.25
Implementation	Secure Build	0.75	0.38	0.25	0.13
Implementation	Secure Deployment	1.75	0.50	0.63	0.63
Implementation	Defect Management	1.13	0.63	0.25	0.25
Verification	Architecture Assessment	0.50	0.25	0.25	0.00
Verification	Requirements Testing	0.25	0.13	0.13	0.00
Verification	Security Testing	0.38	0.25	0.00	0.13
Operations	Incident Management	1.13	0.38	0.38	0.38
Operations	Environment Management	1.13	0.38	0.38	0.38
Operations	Operational Management	1.25	0.50	0.38	0.38
Human Factors	Usability	1.25	0.50	0.50	0.25
Human Factors	Accessibility	0.75	0.25	0.25	0.25

Functions	Current
Governance	0.63
Design	0.71
Implementation	1.21
Verification	0.38
Operations	1.17
Human Factors	1.00

3rdParty	0.25
InternalDev	0.96
EnvMgmt	1.44
Total	0.85

Process Maturity	0.85

- Indicates the maturity of the providers software development process for security and usability
- Based on OWASP Software Assurance Maturity Model (SAMM), added usability/accessibility
- Scores only change when the Process Assessment activity is executed
- Used in the Testing Rules Determination

Get involved

- SAMM User Day: https://www.youtube.com/watch?v=BpNbWZg_pKY
- Website: https://owaspsamm.org
- Github: https://github.com/OWASP/samm/
- Slack: OWASP - #project-samm
- Use and donate (feed)back!
- Donate resources
- Sponsor SAMM

SAMM Newsletter

Thank you!

Questions?

brian.glas@gmail.com (or @owasp.org)

@infosecdad