Problem statement

- Risk
  - Fines
  - Reputation damage
  - Stock price (*)
- Systematic approach is necessary
  - Tools are only part of the story
  - People, processes, tools, knowledge

Certification-focused approaches

- Compliance frameworks (ISO27001, SOC2)
- Nice and shiny label, but
  - Compliance ≠ security
  - Protecting against auditor and not the attacker
  - Pseudo risk-driven
  - Not focused on application security
  - No real measurability (yes / no label)

Application Security Programs

- BSIMM
- **OWASP SAMM**

BSIMM vs SAMM

BSIMM (by Synopsys)	SAMM (by OWASP)
Descriptive	Prescriptive
Proprietary	Open source
No tooling	Excel Toolbox, SAMMY, SAMMWise
Too complex	Concise and clear, Measurements-oriented
Industry-based prioritization	Risk-based prioritization
Activity levels	Maturity levels

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

www.owaspsamm.org
owaspsamm.thinkific.com

OWASP SAMM

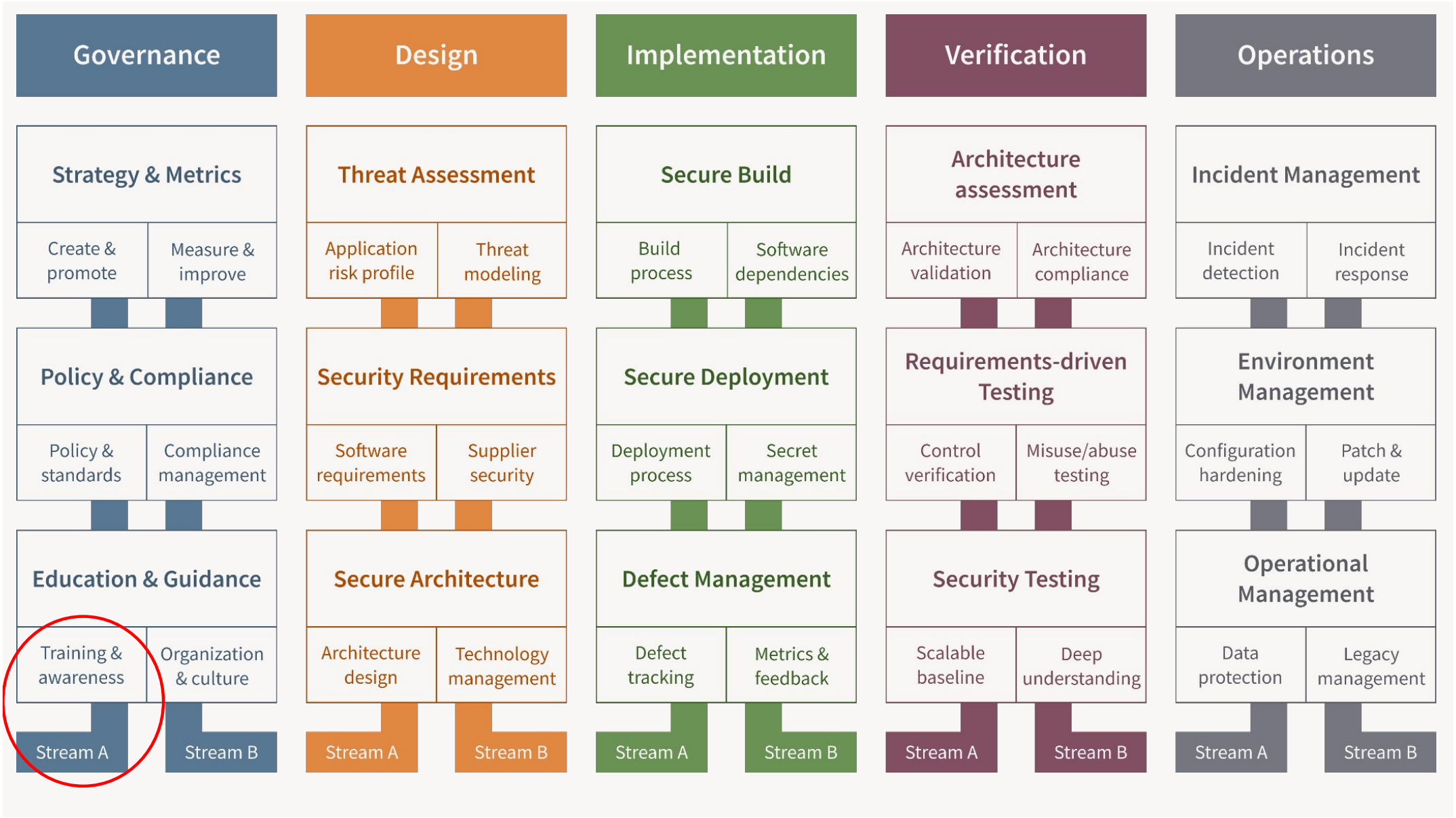
SAMM Use-cases

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


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

Education and Guidance Practice

Maturity Level	Stream A: Training and Awareness
1: Ad-hoc provisioning	Provide security awareness training for all personnel involved in SDLC.
2: Effectiveness and efficiency	Technology and role-specific guidance.
3: Comprehensive mastery	Standardized in-house guidance around the organization’s secure software development standards.

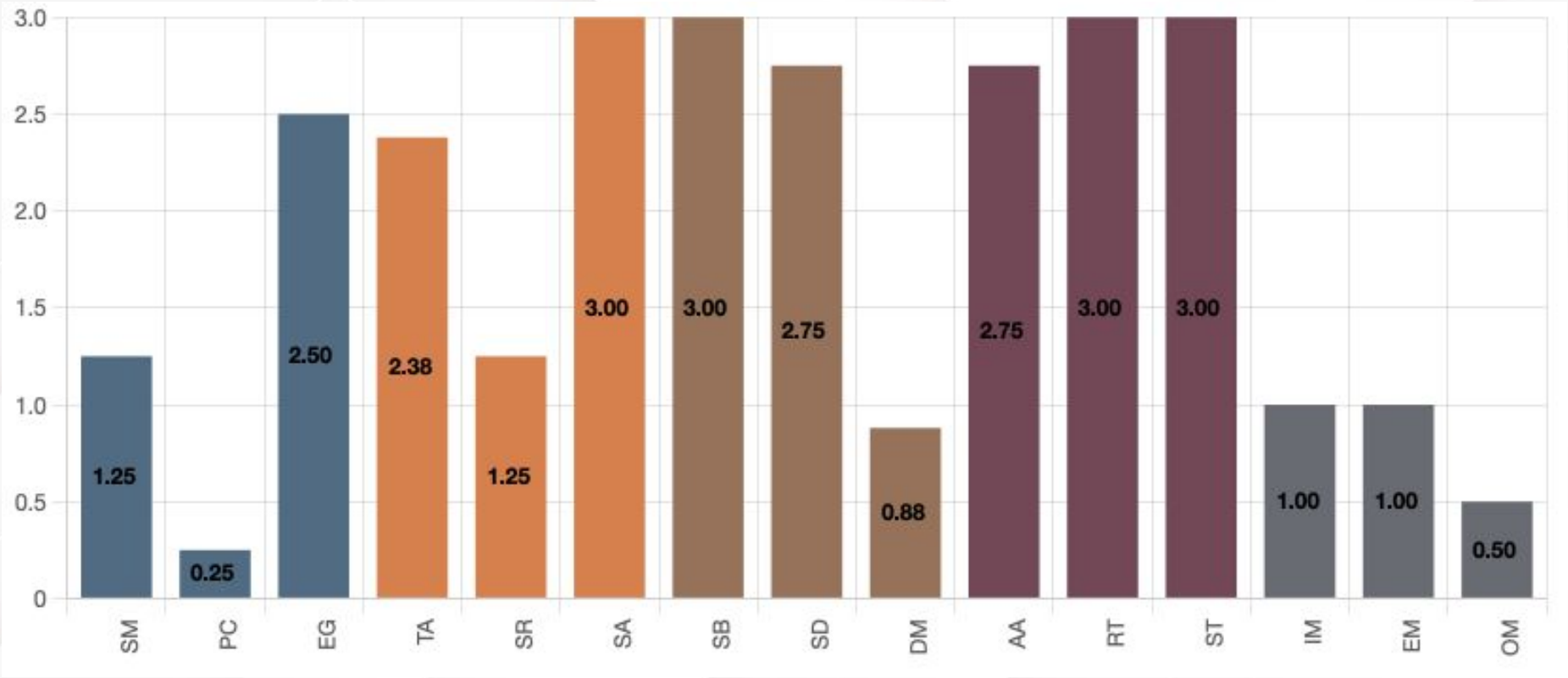
Training and Awareness Maturity Level 1

Do you require employees involved with application development to take SDLC training?

Answers
• No
• Yes, some of them
• Yes, at least half of them
• Yes, most or all of them

Quality criteria
• Training is repeatable, consistent, and available to anyone involved with software development lifecycle • Training includes at least OWASP Top 10, Security Design Principles • Training requires a sign-off or an acknowledgement from attendees • You have updated the training in the last 12 months • Training is required during employees' onboarding process

SAMM Assessment = 90 questions



Challenges

- “How is this different from other tools?”
- SAMM is open to interpretation
  - Self-assessment is a challenge
  - Lack of guidance for embedded teams
  - This is “not applicable” for my team
- Governance & Operations are shared themes

“Security Center of Excellence”

- Corporate-wide task-force in charge of application security
  - Processes & tools
  - Guidance
  - Best practices
- Governance / Operations
  - Strategy, policies, standards, compliance, training
  - Incident management, configuration hardening, patching & updating
- Bi-weekly meetings with all BU leads

SAMM Philosophy

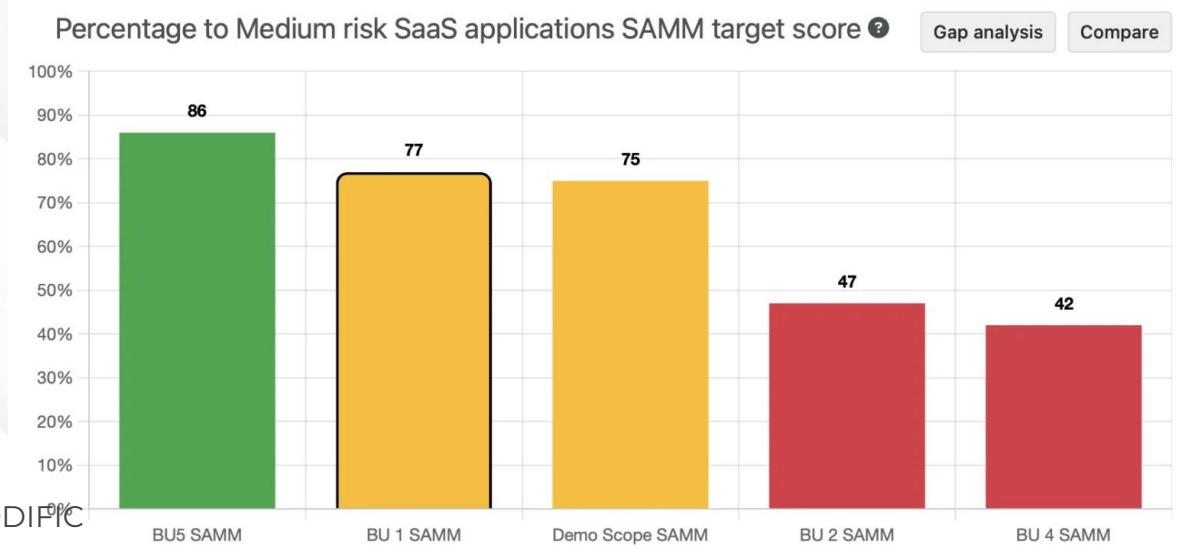
- No risk - no need for security
  - Risk tolerance should define your target score
- Getting to a max score is a waste of resources
- Problem 1: Full implementation of unnecessary activities
  - E.g., engaging legal to create contracts for subcontractors when you don’t have any
- Problem 2: Shallow implementation of necessary activities
  - E.g., creating a policy and standards document nobody will ever read

Path of least resistance

- Executive board needs a simple dashboard
- Teams would overachieve on simpler activities
  - Target score: 1.9
  - Overall score: 2.1

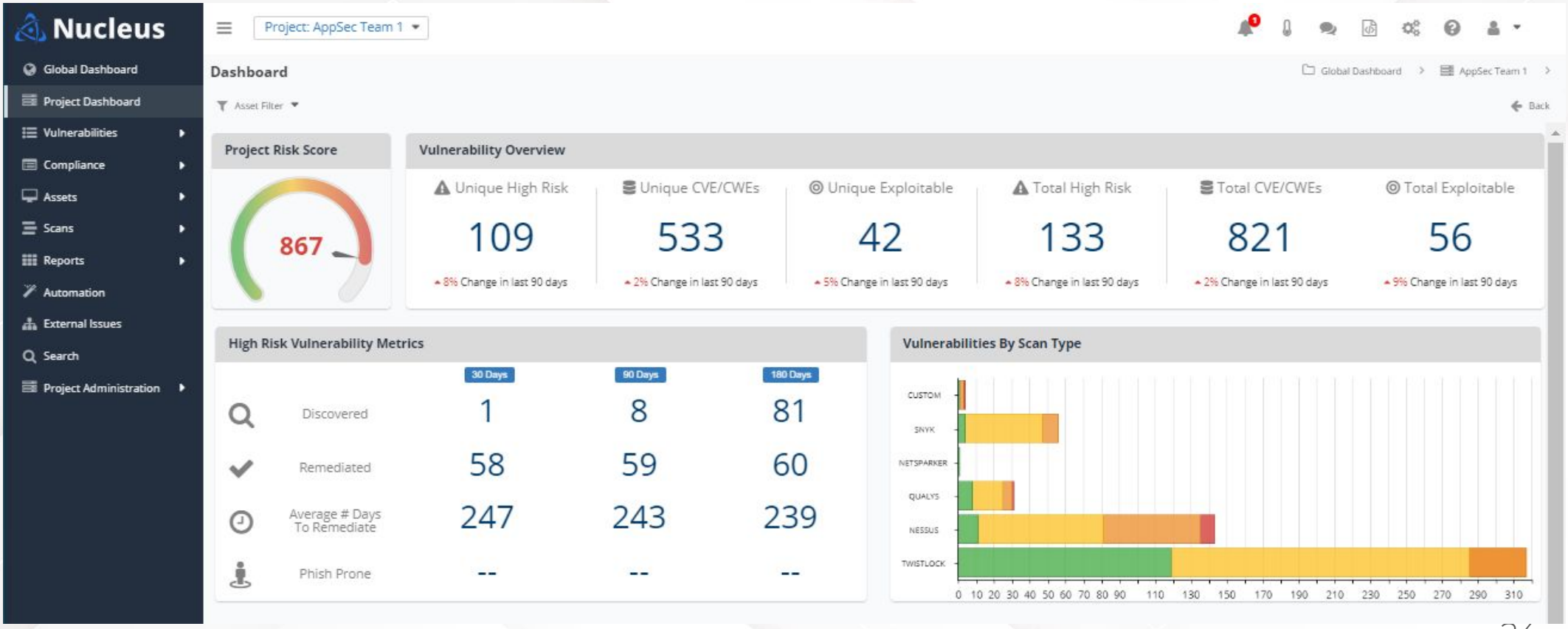
Percentage to target

- A score between 0 and 100%
- Penalty for underachieving
- No bonus for overachieving
- Fits with SAMM Core Team's Vision



THIS CHART IS BASED ON RANDOM DATA GENERATED BY CODIFIC

Application Security Posture Management (ASPM)



THIS CHART IS BASED ON RANDOM DATA GENERATED BY CODIFIC

SAMM Score correlation to Nucleus Risk score

- Inverse correlation for code repositories
  - Higher SAMM score = lower risk
- Direct correlation for infrastructure
  - Higher SAMM score = higher risk

	Infrastructure	Code	Infrastructure	Code
Risk correlation with SAMM Percentage To Target	**0.24**	**-0.48**	**0.38**	**-0.44**
Risk correlation with SAMM Absolute score	**0.4**	**-0.29**	**0.55**	**-0.28**

Remaining Challenges

- Defining target postures is a challenge
  - Each team has a unique risk profile / appetite
  - OWASP SAMM Benchmarking Project might help
- We need guidance for embedded / IoT devices
- Further refinements to the model
  - Architecture Assessment practice
  - Quality criteria consistency