FORTIFY®
An HP Company

Mastering SSA: A Case Study of the US Air Force Software Assurance Center of Excellence

Shakeel Tufail

Federal Practice Manager

HP - Fortify Software

TRUST YOUR SOFTWARE™

Agenda

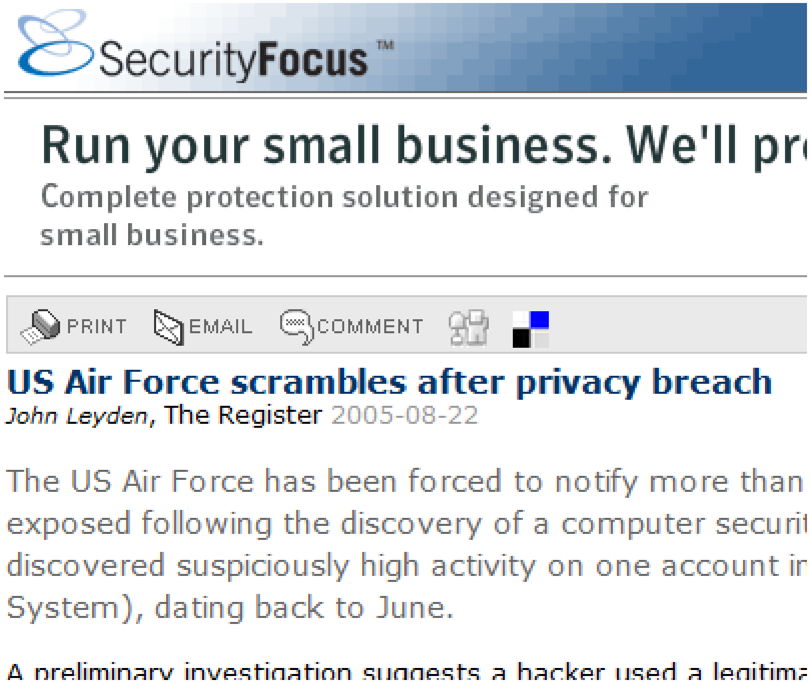
- History
- The ASACoE Process
- Challenges
- Best Practices
- Q&A

FORTIFY®
An HP Company

History

History

- August 2005 – Human Resource System Breached
- 33,000 Records Stolen
- Attack vector was software related



US Air Force scrambles after privacy breach

John Leyden, The Register 2005-08-22

The US Air Force has been forced to notify more than exposed following the discovery of a computer securit discovered suspiciously high activity on one account ir System), dating back to June.

A preliminary investigation suggests a hacker used a legitima

History

- Software Security Pilot Program
    - Lead by Maj. Bruce Jenkins
- Critical vulnerabilities were found in all pilot applications
- Decision was made to organize a group dedicated to software security – September 2006

Application
Software
Assurance
Center
of
Excellence

History

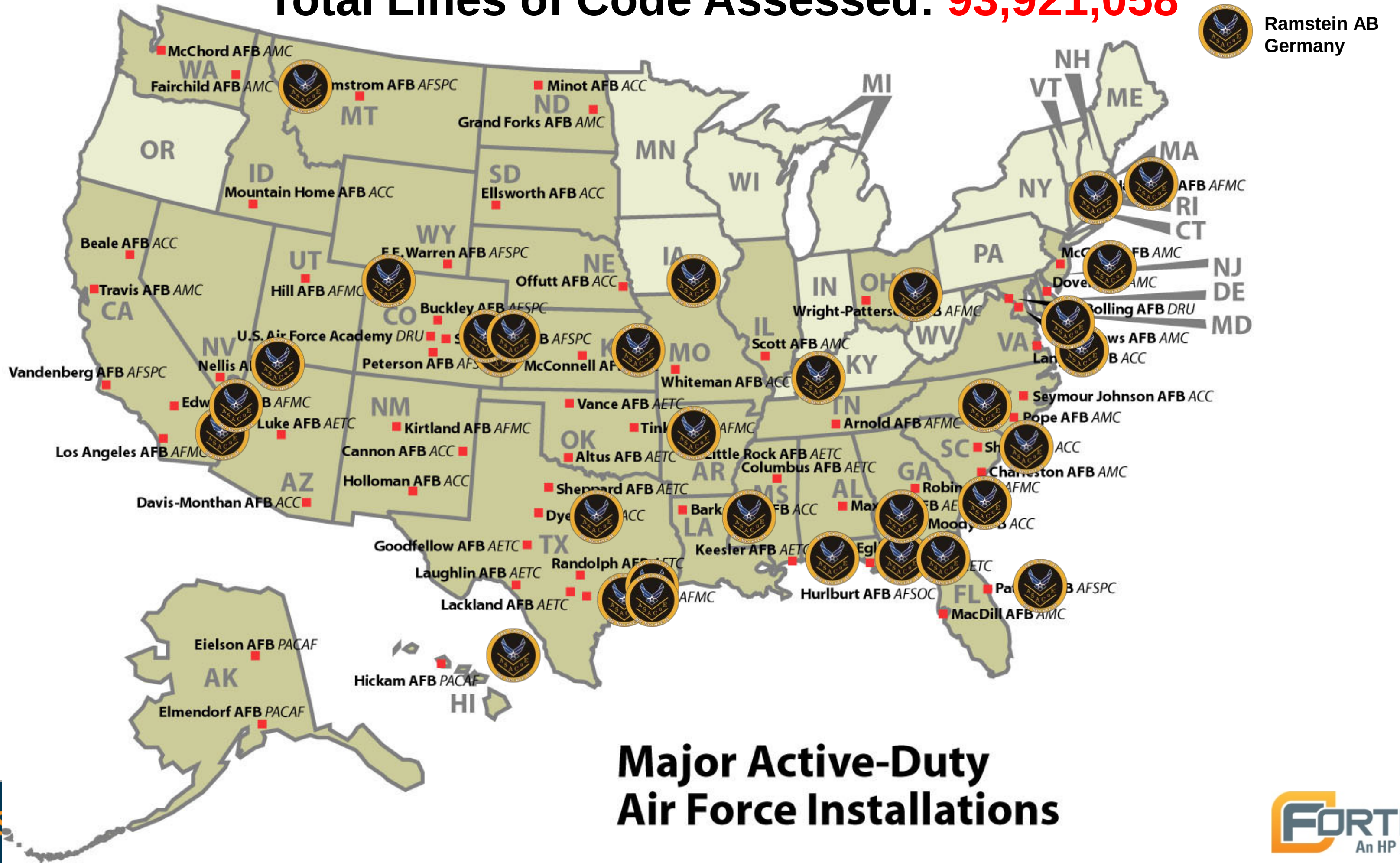
- **Contract competition to find best automated security software**
- **Focus on 3 areas:**
  - Static Analysis (Source Code Analysis)
  - Dynamic Analysis (Penetration Testing)
  - Data Tier Analysis (Database STIG Checking)

- **Software**
  - Fortify Software (SCA, 360 Server, & RTA)
  - IBM Rational AppScan
  - AppSecInc AppDetective

- **Services**
  - Prime Contractor – Telos
  - Subcontractors – Fortify and Cigital

Mastering SSA: ASACoE

Program Management Offices Visited: 96
Applications Assessed: 600+
Total Lines of Code Assessed: 93,921,058



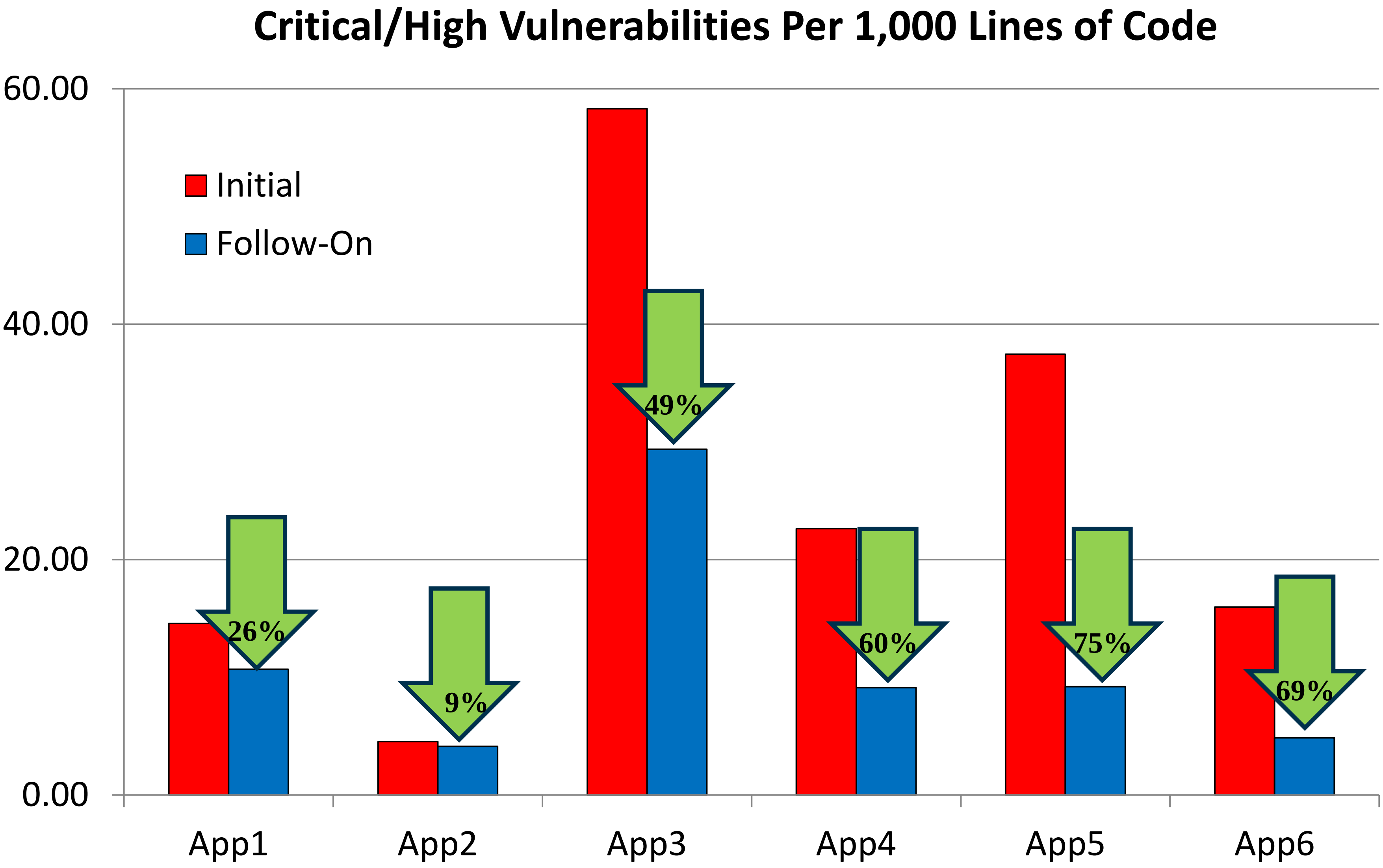
Major Active-Duty Air Force Installations

History

ASACoE Benefits

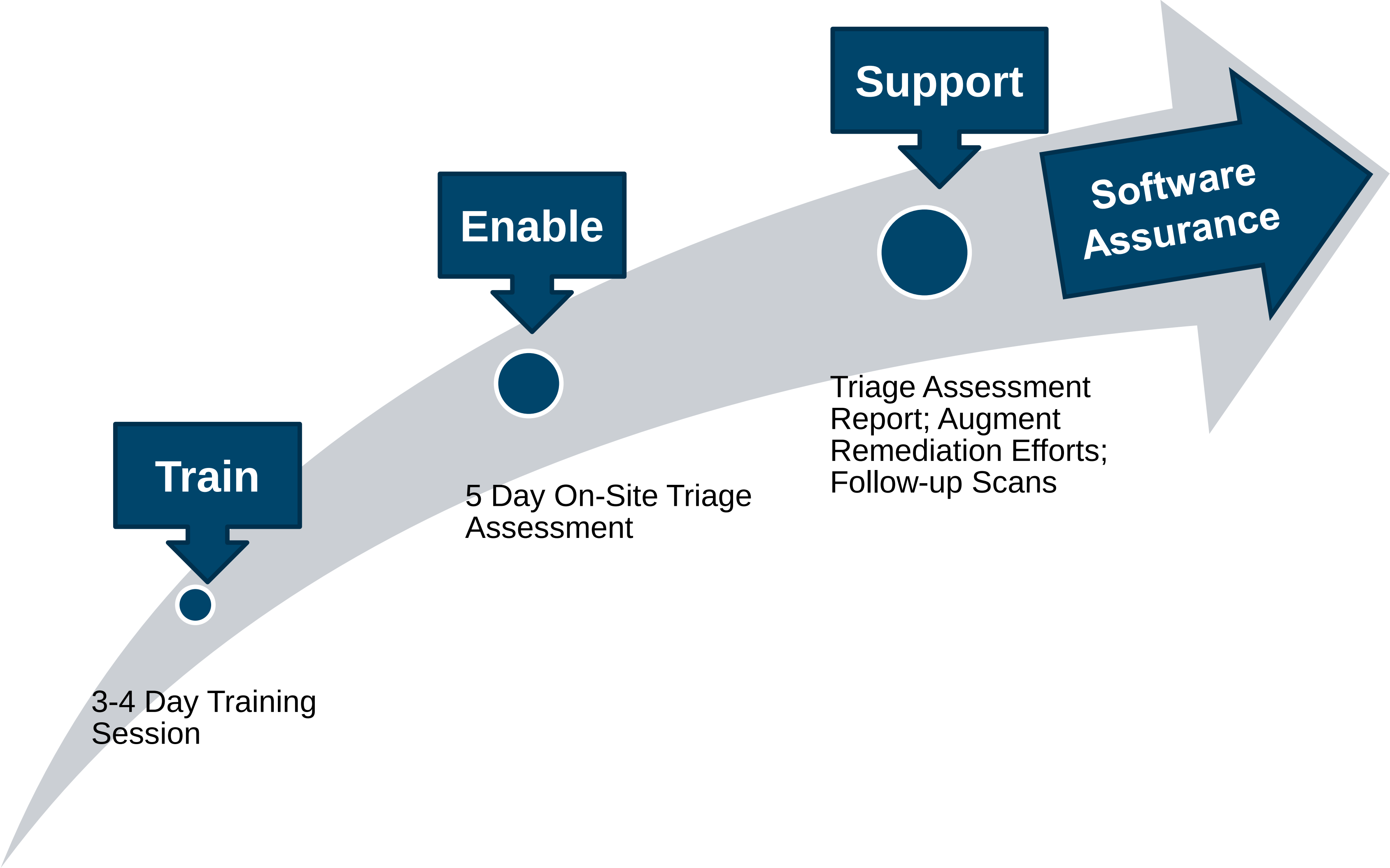
- Significant Risk Mitigation throughout the SDLC
- Cost and Time Savings for PMOs
- Certification & Accreditation Processing Time Reduced
- Real Time Protection for Fielded Operational Systems

History


Critical/High Vulnerabilities Per 1,000 Lines of Code
60.00
40.00
20.00
0.00
🟥 Initial
🟦 Follow-On
26%
9%
49%
60%
75%
69%
App1
App2
App3
App4
App5
App6

The ASACoE Process

The ASACoE Process



Train

3-4 Day Training Session

Enable

5 Day On-Site Triage Assessment

Support

Triage Assessment Report; Augment Remediation Efforts; Follow-up Scans

Software Assurance

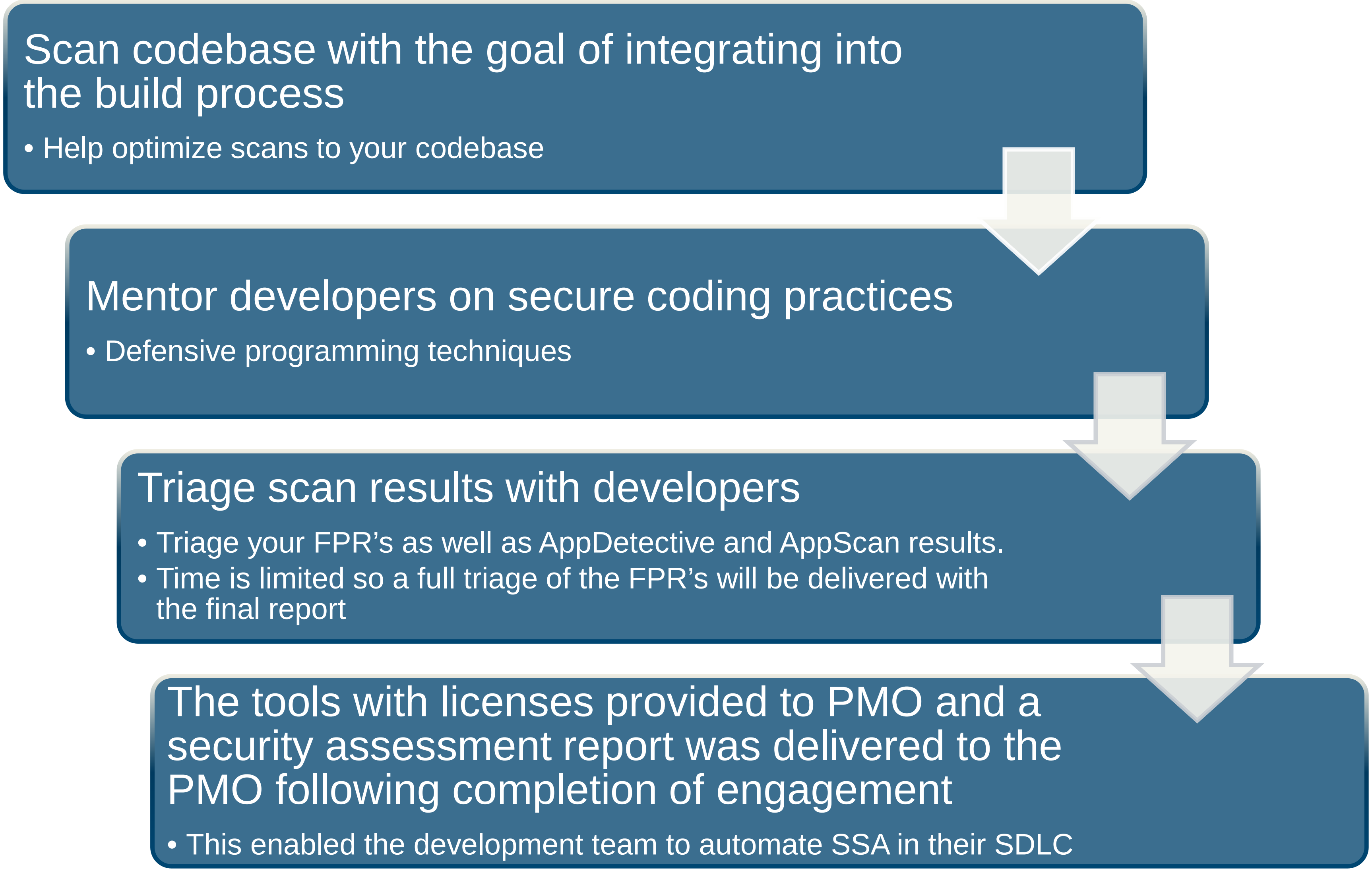
The ASACoE Process - Train

3 Day Training Session

- 1 Day Defensive Programming
  - Need for Software Assurance
  - Case Studies
  - Vulnerability Examples
- ½ Day AppDetective Training
- 1 Day Fortify SCA Training
- ½ Day Fortify RTA/PTA/360 Server
- <optional> 1 Day AppScan Training
- <optional> 1 Day Risk-Based Security Testing

- **Mixed audience: Managers, IA, QA, Developers**
- **Hosted US AFBs & contractor sites**

The ASACoE Process – On-Site



Scan codebase with the goal of integrating into the build process

- Help optimize scans to your codebase

Mentor developers on secure coding practices

- Defensive programming techniques

Triage scan results with developers

- Triage your FPR’s as well as AppDetective and AppScan results.
- Time is limited so a full triage of the FPR’s will be delivered with the final report

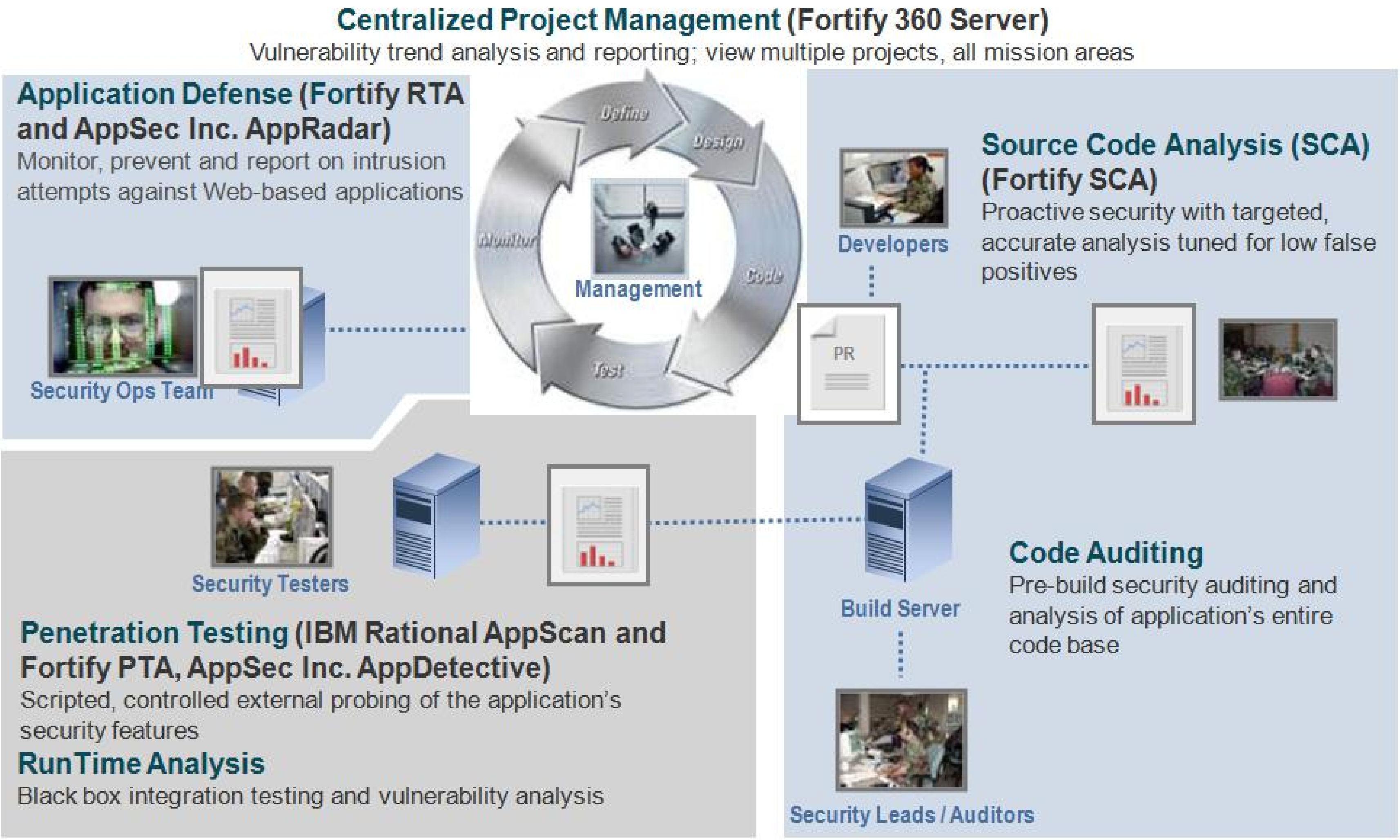
The tools with licenses provided to PMO and a security assessment report was delivered to the PMO following completion of engagement

- This enabled the development team to automate SSA in their SDLC

The ASACoE Process – On-Site

- **ASACoE Assessment Team (4 person team)**
  - 1 Organic (active military) and 3 Contractors
  - Contractors serve as Subject Matter Experts
  - Organics serve as Team Chiefs
- **All team members trained to use software suite**
- **Product specialization depending on background**
- **Periodic rotation of duties**

The ASACoE Process – On-Site



Centralized Project Management (Fortify 360 Server)
Vulnerability trend analysis and reporting; view multiple projects, all mission areas

Application Defense (Fortify RTA and AppSec Inc. AppRadar)
Monitor, prevent and report on intrusion attempts against Web-based applications

Source Code Analysis (SCA) (Fortify SCA)
Proactive security with targeted, accurate analysis tuned for low false positives

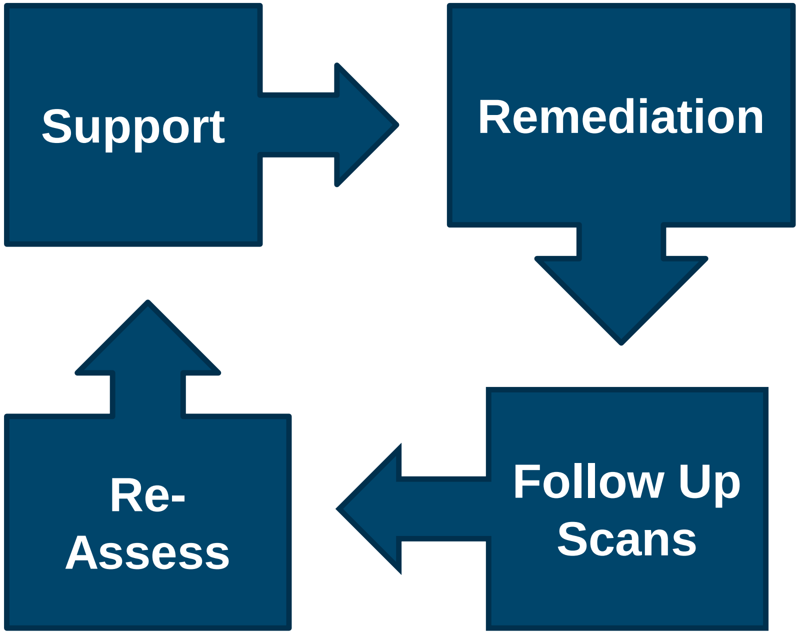
Code Auditing
Pre-build security auditing and analysis of application's entire code base

Penetration Testing (IBM Rational AppScan and Fortify PTA, AppSec Inc. AppDetective)
Scripted, controlled external probing of the application's security features

RunTime Analysis
Black box integration testing and vulnerability analysis

The ASACoE Process - Support

- 1st Tier Support
- Link to Vendors



Support

Remediation

Follow Up Scans

Re-Assess

- 3rd Party Resources
- Verification

- Further Analysis
- Custom Rules

- New Training
- New Assessment

Challenges

Challenge #1: NO MANDATE

- No clear vision for software assurance
- Currently working with proactive groups
- Large focus on new business
- No push for remediation
- Hard to market without mandate /policy

Challenge #2: Moderate Adoption

- Many re-assessments reveal moderate to low adoption of software assurance
- Focus on scanning leaves little time for process development and automation
- Need alternate training methods
- Staff churn / contract change

Challenge #3: Awareness and Education

- Complex problem with complex solution
- All leadership levels need to be made aware of the risks associated with software vulnerabilities
- Getting the word out
  - SAF/A6 and AFSPC – Provide policy recommendations and best practices
  - AF Institute of Technology, Academy, and Cyber Technical Schools
  - Aid US Navy, Army & Canadian Army to Stand Up Similar Centers

Next Steps

- The ASACoE process was designed to assess the largest amount of applications possible – not the best fit for everyone

- If you like the ASACoE approach, we can help with implementing their model

- When considering establishing a Center of Excellence, first consult industry standards (Open SAMM) www.opensamm.org

SAMM - Understanding the model

SAMM Business Functions

- Start with the core activities tied to any organization performing software development
- Named generically, but should resonate with any developer or manager

Governance

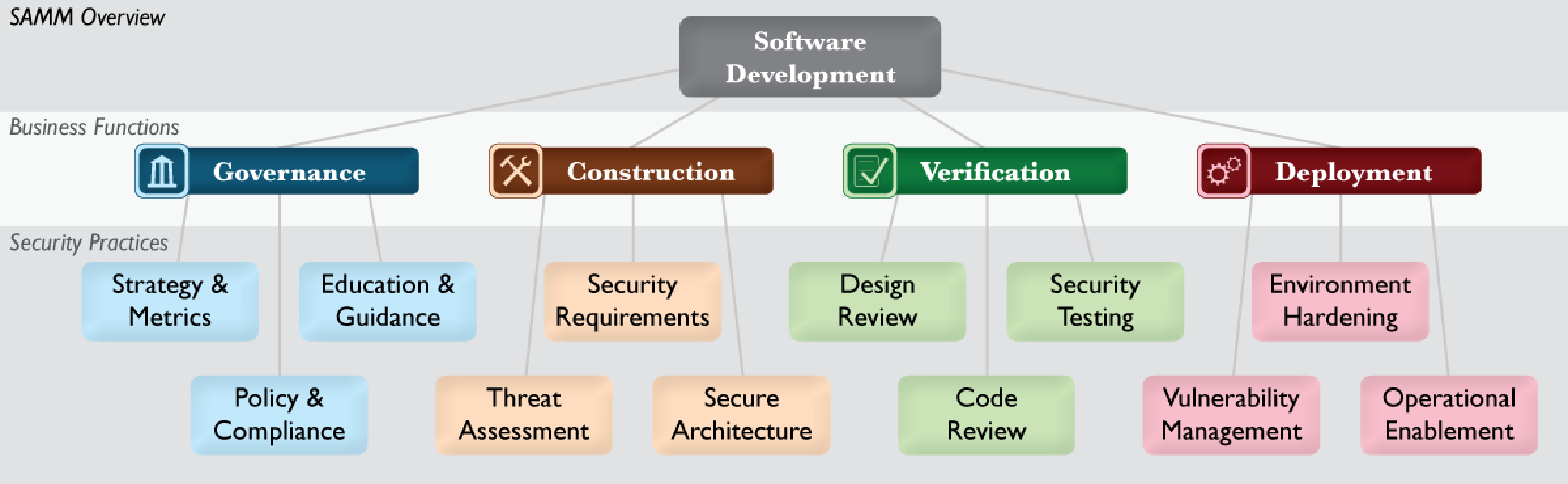
Construction

Verification

Deployment

SAMM Security Practices

- From each of the Business Functions, 3 Security Practices are defined
- The Security Practices cover all areas relevant to software security assurance
- Each one is a ‘silo’ for improvement

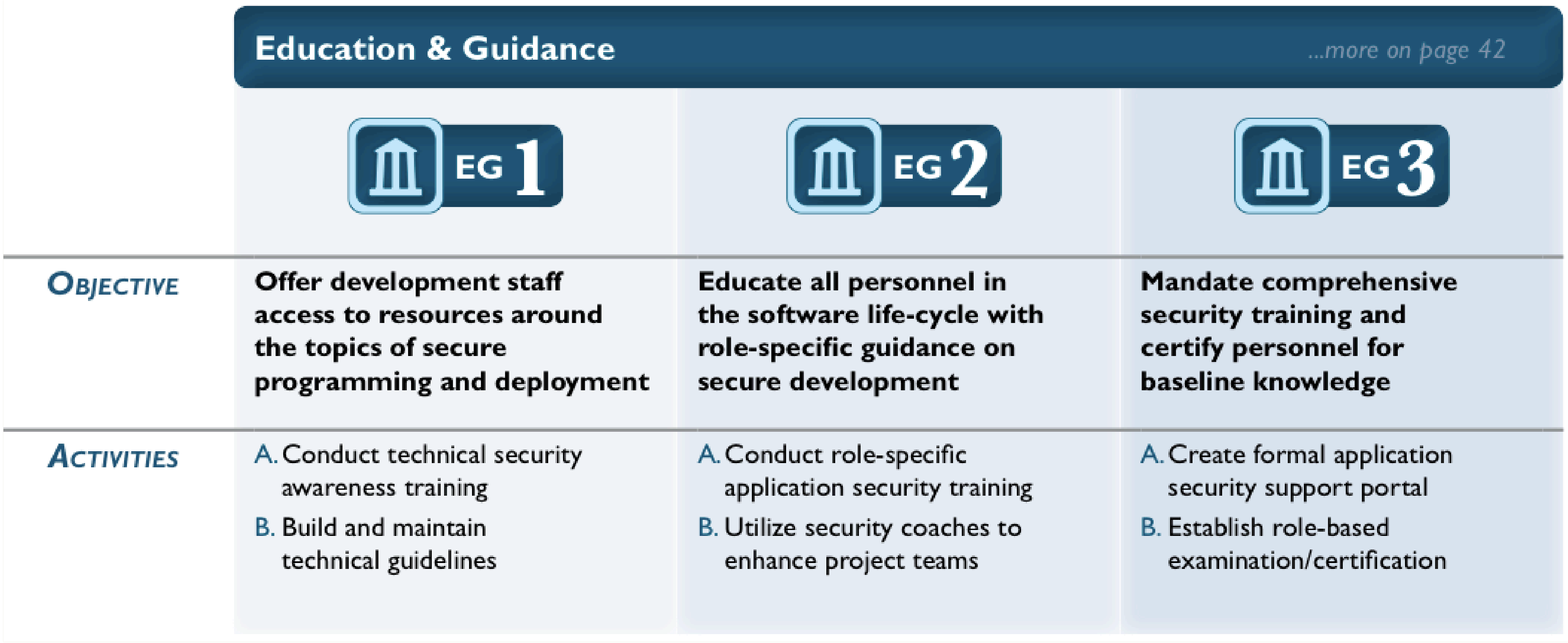


Under each Security Practice

- Three successive Objectives under each Practice define how it can be improved over time
  - This establishes a notion of a Level at which an organization fulfills a given Practice

- The three Levels for a Practice generally correspond to:
  - (0: Implicit starting point with the Practice unfulfilled)
  - 1: Initial understanding and ad hoc provision of the Practice
  - 2: Increase efficiency and/or effectiveness of the Practice
  - 3: Comprehensive mastery of the Practice at scale

Check out this one...

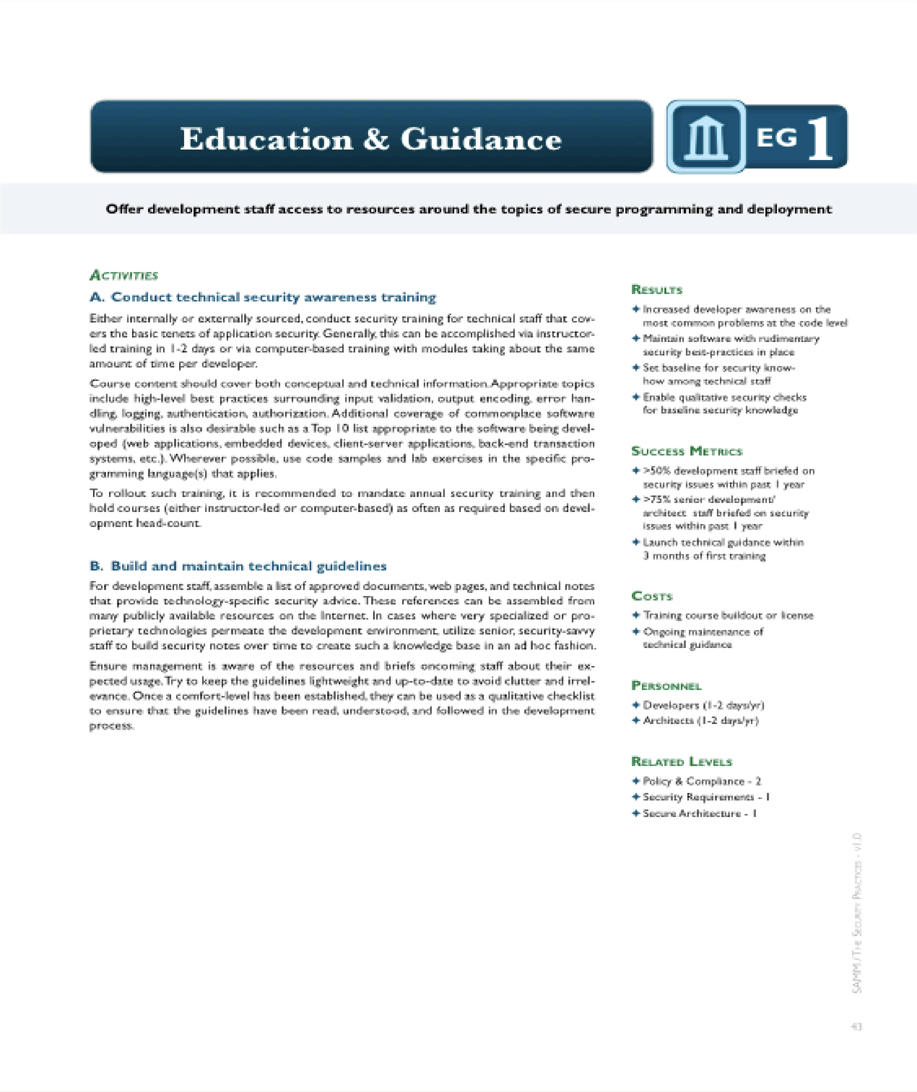


Education & Guidance *...more on page 42*

	EG 1	EG 2	EG 3
Objective	**Offer development staff access to resources around the topics of secure programming and deployment**	**Educate all personnel in the software life-cycle with role-specific guidance on secure development**	**Mandate comprehensive security training and certify personnel for baseline knowledge**
Activities	A. Conduct technical security awareness training B. Build and maintain technical guidelines	A. Conduct role-specific application security training B. Utilize security coaches to enhance project teams	A. Create formal application security support portal B. Establish role-based examination/certification

Per Level, SAMM defines...

- Objective
- Activities
- Results
- Success Metrics
- Costs
- Personnel



Education & Guidance

EG 1

Offer development staff access to resources around the topics of secure programming and deployment

Activities

A. Conduct technical security awareness training

Either internally or externally sourced, conduct security training for technical staff that covers the basic tenets of application security. Generally, this can be accomplished via instructor-led training in 1-2 days or via computer-based training with modules taking about the same amount of time per developer.

Course content should cover both conceptual and technical information. Appropriate topics include high-level best practices surrounding input validation, output encoding, error handling, logging, authentication, authorization. Additional coverage of commonplace software vulnerabilities is also desirable such as a Top 10 list appropriate to the software being developed (web applications, embedded devices, client-server applications, back-end transaction systems, etc.). Wherever possible, use code samples and lab exercises in the specific programming language(s) that applies.

To rollout such training, it is recommended to mandate annual security training and then hold courses (either instructor-led or computer-based) as often as required based on development head-count.

B. Build and maintain technical guidelines

For development staff, assemble a list of approved documents, web pages, and technical notes that provide technology-specific security advice. These references can be assembled from many publicly available resources on the Internet. In cases where very specialized or proprietary technologies permeate the development environment, utilize senior, security-savvy staff to build security notes over time to create such a knowledge base in an ad hoc fashion.

Ensure management is aware of the resources and briefs oncoming staff about their expected usage. Try to keep the guidelines lightweight and up-to-date to avoid clutter and irrelevance. Once a comfort-level has been established, they can be used as a qualitative checklist to ensure that the guidelines have been read, understood, and followed in the development process.

Results

- Increased developer awareness on the most common problems at the code level
- Maintain software with rudimentary security best-practices in place
- Set baseline for security know-how among technical staff
- Enable qualitative security checks for baseline security knowledge

Success Metrics

- >50% development staff briefed on security issues within past 1 year
- >75% senior development/ architect staff briefed on security issues within past 1 year
- Launch technical guidance within 3 months of first training

Costs

- Training course buildout or license
- Ongoing maintenance of technical guidance

Personnel

- Developers (1-2 days/yr)
- Architects (1-2 days/yr)

Related Levels

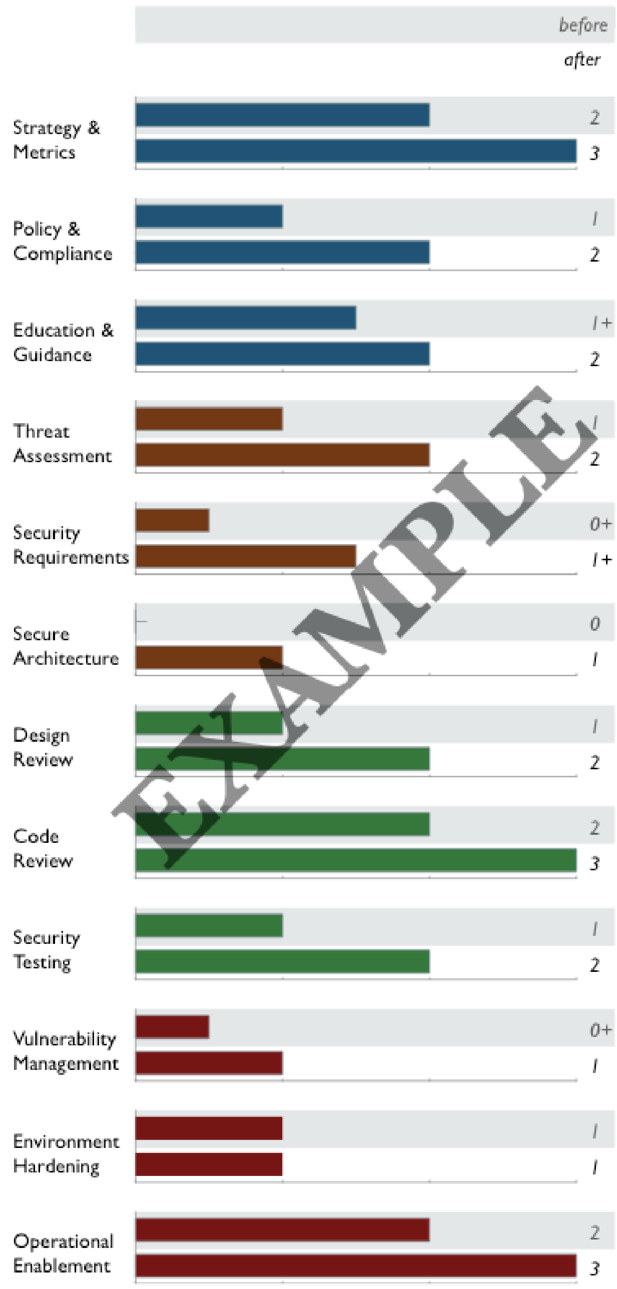
- Policy & Compliance - 2
- Security Requirements - 1
- Secure Architecture - 1

SAMM / The Security Practices - v1.0

43

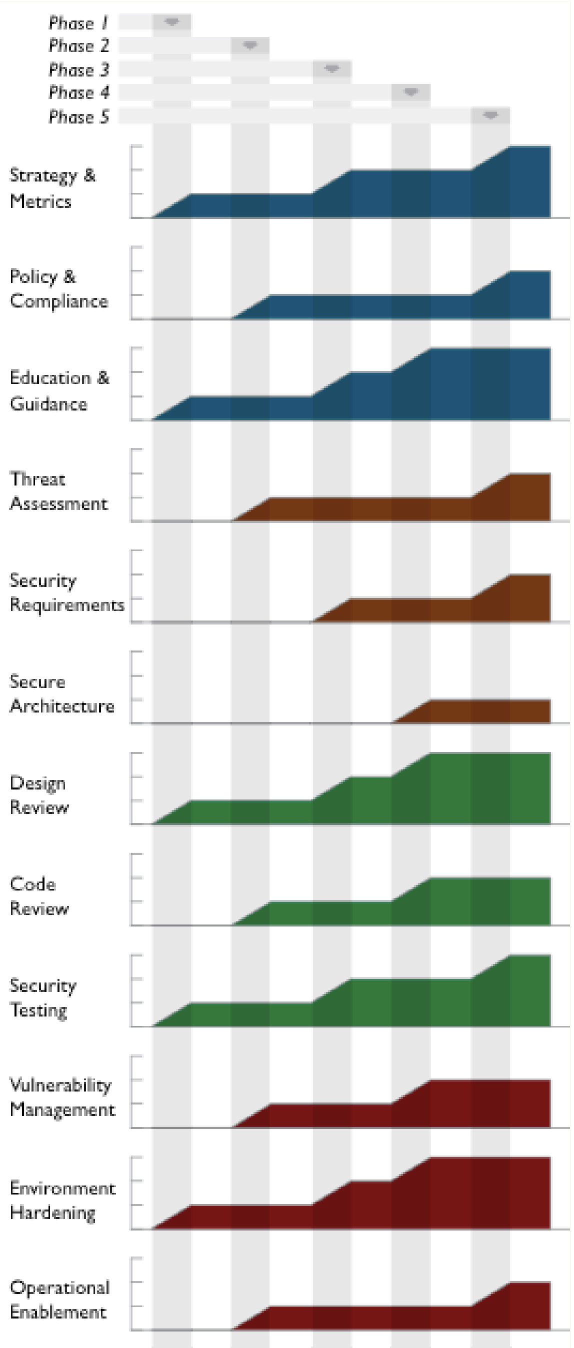
Creating Scorecards

- 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 measurement
  - Capturing scores over consistent time frames for an assurance program that is already in place



Roadmap templates

- To make the “building blocks” usable, SAMM defines Roadmaps templates for typical kinds of organizations
  - Independent Software Vendors
  - Online Service Providers
  - Financial Services Organizations
  - Government Organizations
- Organization types chosen because
  - They represent common use-cases
  - Each organization has variations in typical software-induced risk
  - Optimal creation of an assurance program is different for each



FORTIFY®
An HP Company

Thank you for your time.

Questions?