Software Security Initiatives
in the Real World

Claudio Merloni
Software Security Consultant
Fortify Software
cmerloni@fortify.com

OWASP
May 2009

The OWASP Foundation
http://www.owasp.org/

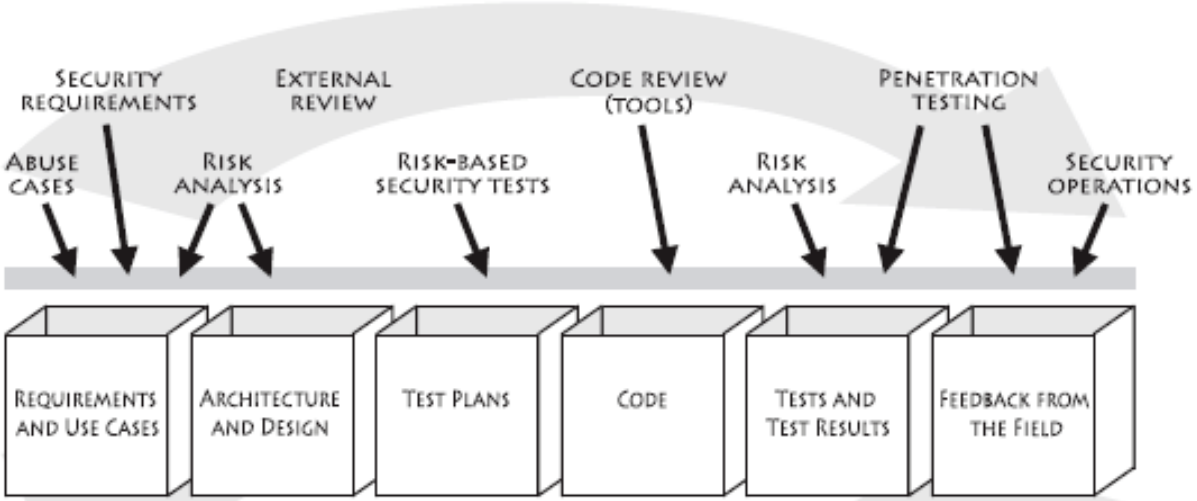
Agenda

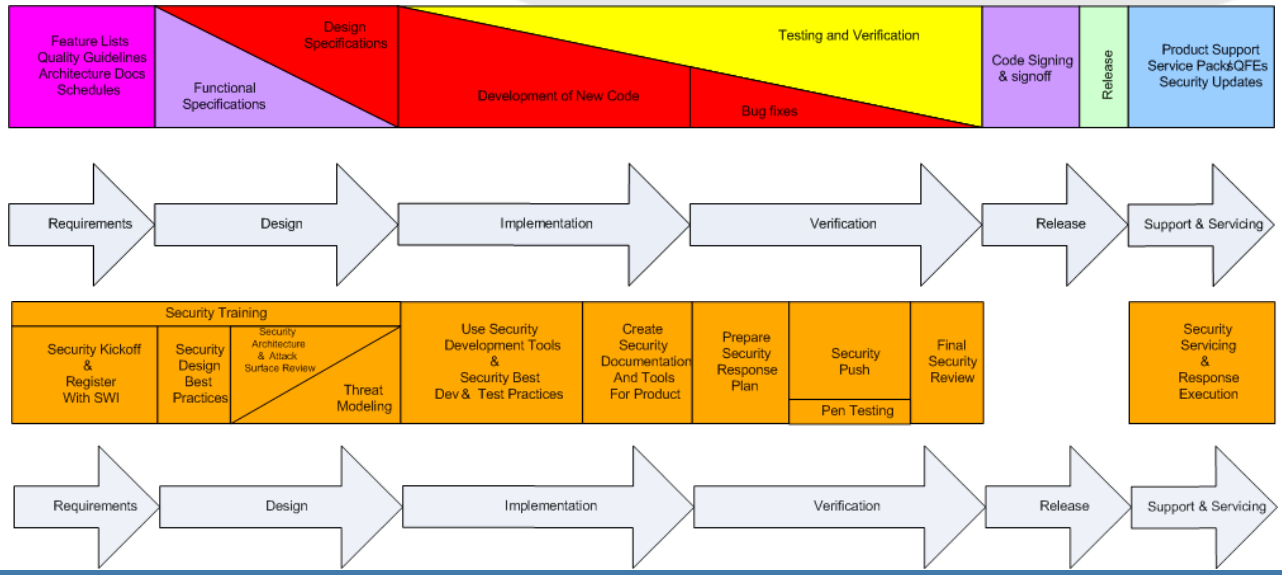
- Approaches to Security in the SDLC
- The Software Assurance Maturity Model
  - Business Functions
  - Security Practices
  - Objectives
- Using the Maturity Model in the organization
- What are companies doing?

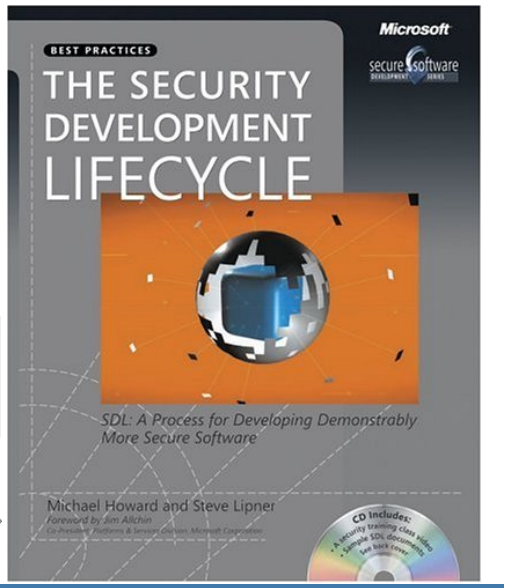
Applications are the New Frontier

- Applications are the New Frontier
  - Increasingly Accessible
  - Results are Profitable
- Application Vulnerabilities are the New Entry Point
  - Easy to exploit
  - Opportunities are plentiful
- Network-Based Security Solutions are ineffective for today's threat

Approaches to Security in the SDLC







Lessons learned

- Microsoft SDL
  - Heavyweight, appropriate to large ISVs selling boxed software
- Touchpoints
  - High-level map without enough details to execute against
- CLASP
  - Large collection of activities, but no priority ordering
  - Good for experts to use as a guide, but hard for non-security folks to use off the shelf

Motivation for a maturity model approach

- Changing an organization is hard

  Simple, well-defined, measurable
  always trumps
  complex, nuanced, ethereal

- Software security is a result of many activities
  - Combination of people, process, and automation
- There is no single formula for all organizations
  - Business risk from software depends on what the business does
- An assurance program must be built over time
  - Organizations can't change overnight. Use a phased approach.

The Software Assurance Maturity Model (SAMM)

Goals and Purpose

- To define building blocks for an assurance program
  - Delineate all functions within an organization that *could* be improved over time
- To allow organizations to create customized roadmaps
  - Each organization can choose the order and extent they improve each function
- To provide sample roadmaps for common types of organizations
  - Each roadmap is a baseline that can be tweaked based on the specific concerns of a given organization
- OpenSAMM – http://www.opensamm.org

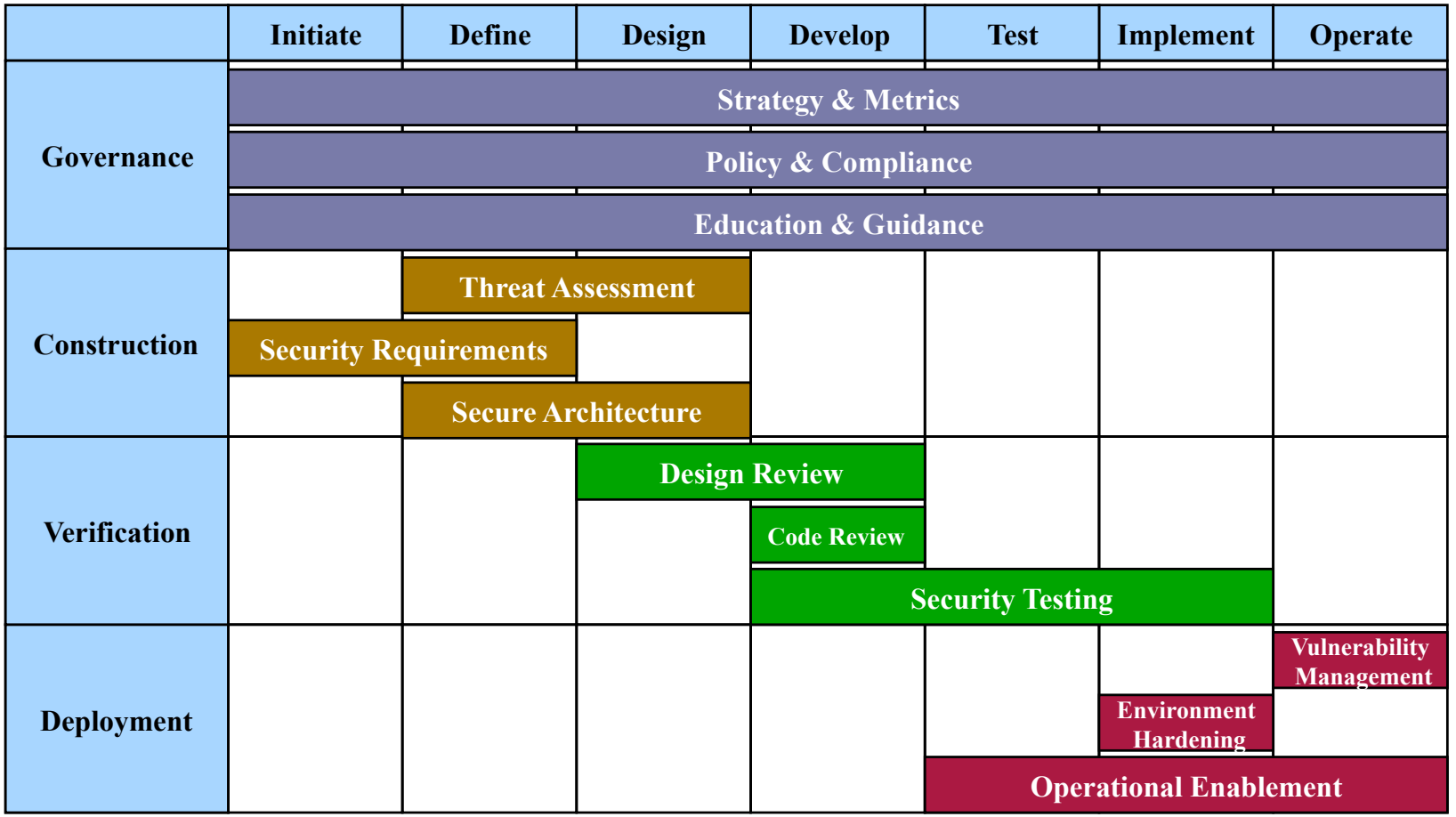
How does SAMM work?

Four high-level Business Functions

- All security-related activities mapped under 4 ***Business Functions***

Governance	Construction	Verification	Deployment
Activities related to security program management and cross-cutting organizational concerns	Activities related to the product conception and software design processes	Activities related to reviewing, testing, and validating software	Activities related to knowledge transfer and maintenance of running software

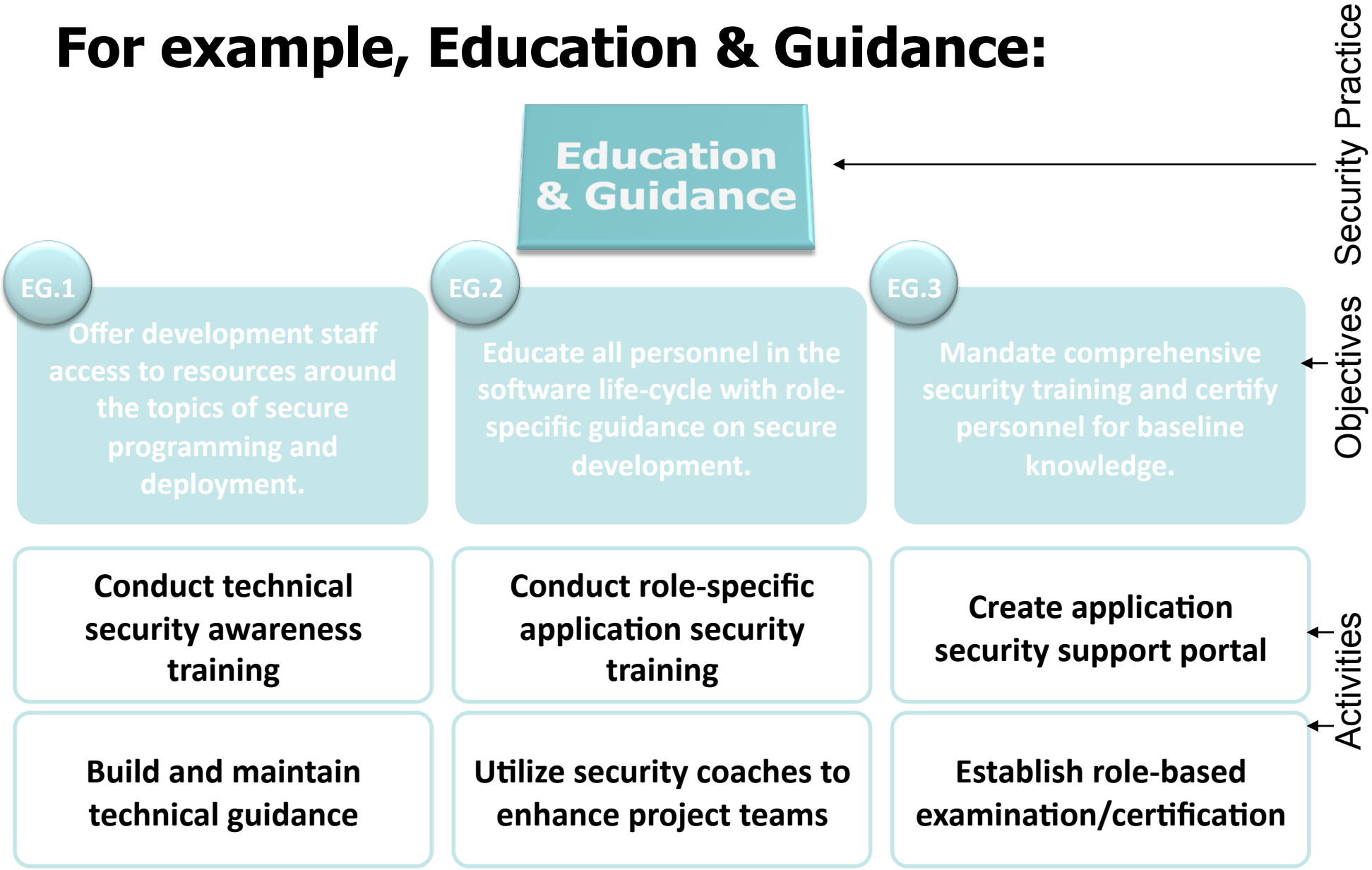
SSA - Secure Development Life-Cycle



What's under each Security Practice?

- Three successive Objectives under each Security Practice define how that Security Practice can be improved over time
  - This establishes a notion of a "level" at which an organization fulfills a given Security Practice
- The three Objectives for a Security Practice generally correspond to:
  - *0: Implicit starting point with the Security Practice unfulfilled
  - 1: Initial understanding and ad hoc provision of the Security Practice
  - 2: Increase efficiency and/or effectiveness of the Security Practice
  - 3: Comprehensive mastery of the Security Practice at scale
- Each Objective defines:
  - Activities that must be performed
  - Success metrics
  - Required personnel
  - Associated costs
  - Benefits for the organization

For example, Education & Guidance:



Objectives - Governance

- Strategy & Metrics
  1. Establish Business Assurance Program
  2. Application assurance prioritised based on risk
  3. Assurance program benchmarked against industry
- Policy & Compliance
  1. Build compliance check-list
  2. Perform base-line measurement
  3. Implement compliance gates
- Education & Guidance
  1. Security awareness training
  2. Role specific training for all personnel
  3. Mandatory training and certification

Objectives – Construction

- Threat Assessment
  1. Identify high-level threats to organization
  2. Identify application level threats and compensating controls
  3. Extend to external components
- Security Requirements
  1. Security considered during requirements phase
  2. Security requirements generated from abuse scenarios
  3. Mandated security requirements for all projects
- Secure Architecture
  1. Provide initial Secure Development Standards
  2. Identify secure design patterns and secure coding libraries
  3. Build reference platforms

Objectives - Verification

- Design Review
  1. Perform reviews against known risks
  2. Provide design review service against best practice
  3. Establish release gate for architecture review
- Code Review
  1. Source code review of high-risk code
  2. Automated source code analysis integrated into build process
  3. Establish release gates appropriate to development methodology
- Security Testing
  1. Explicitly test against security test-cases
  2. Utilise automated security testing tools integrated in QA process
  3. Establish release gates for security testing

Objectives - Deployment

- Vulnerability Management
  1. High-level plan for responding to vulnerability reports or incidents
  2. Detailed incident response and disclosure process
  3. Conduct root cause analysis to drive continuous improvement
- Environment Hardening
  1. Maintain operational environment to latest security patch level
  2. Identify and deploy relevant protection and monitoring tools
  3. Perform regular audits of operational infrastructure
- Operational Enablement
  1. Ensure communication between development and operational teams
  2. Maintain formal operational security guidelines
  3. Mandate security guideline and validate for completeness

Approach to phased improvement

- Since the twelve Security Practices are each a maturity area, the successive Objectives represent the "building blocks" for any assurance program
- Simply put, improve an assurance program in phases by:

1. Select Security Practices to be improved in next phase of assurance program
2. Achieve the next Objective in each Security Practice by performing the corresponding activities at the specified success metrics

Key roles

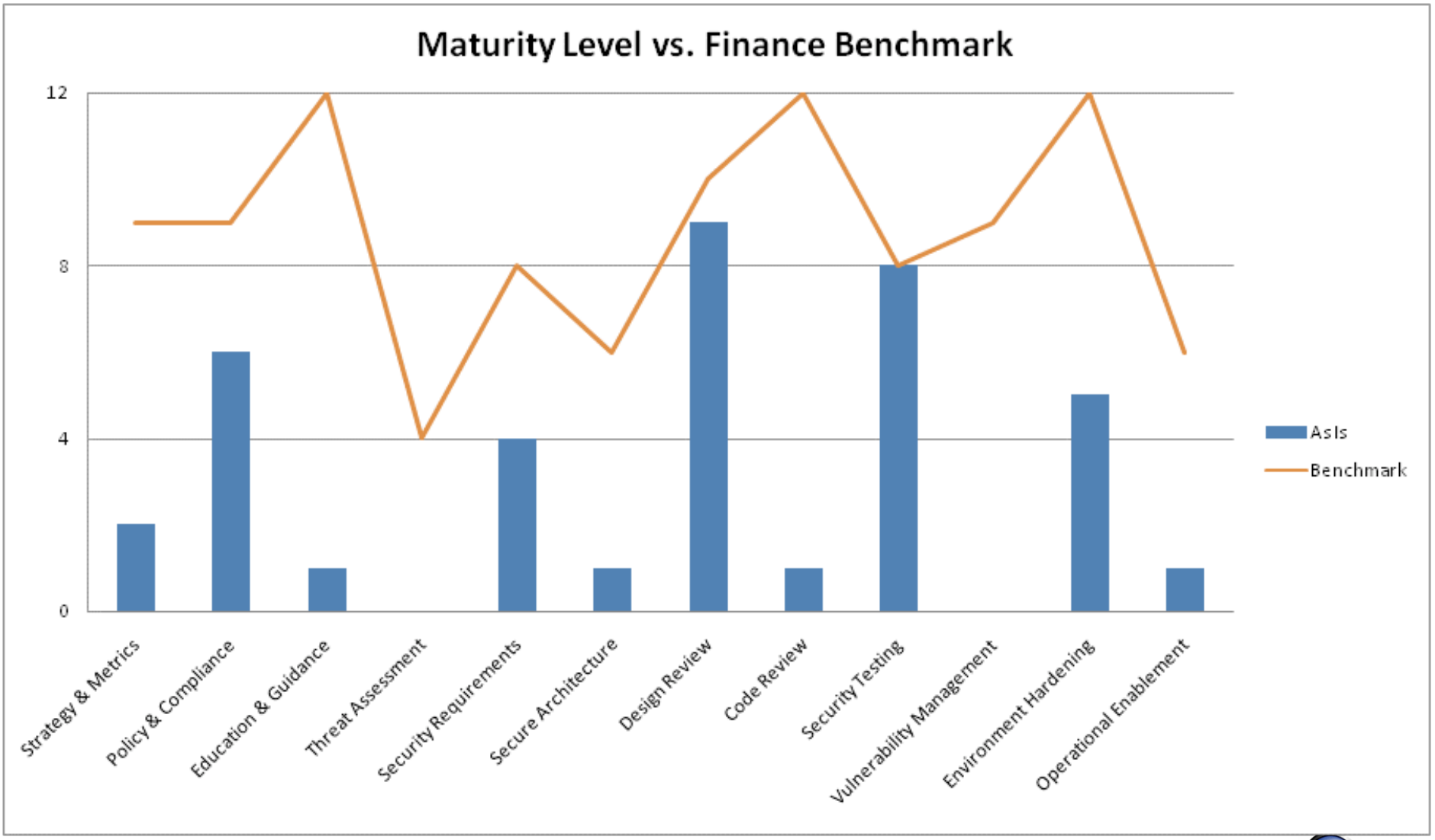
- Executive Leadership
  - managing operations, garnering resources, and providing political cover for a software security initiative.
- The Software Security Group (SSG)
  - The second most important role in a software security initiative is that of the Software Security Group.
- Everybody else
- Satellite Group

What can you do with SAMM?

Recommended roadmaps

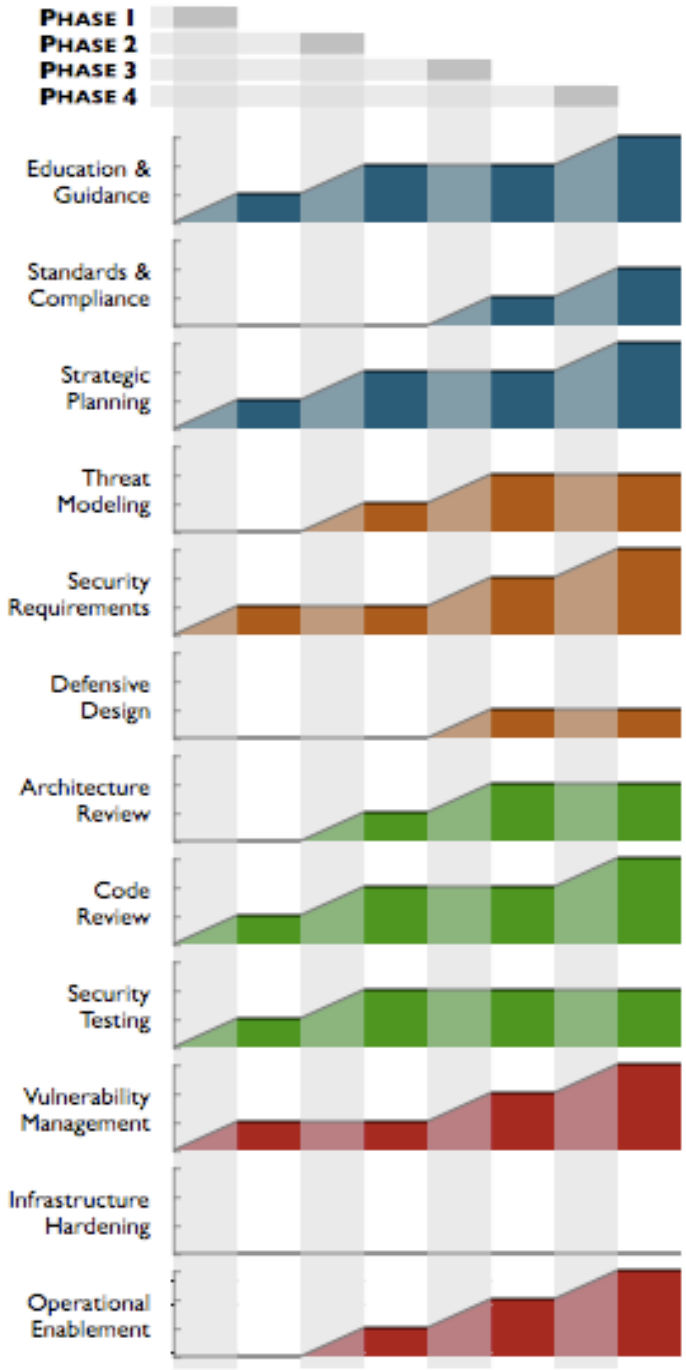
- To make the "building blocks" usable, SAMM defines sample Roadmaps for typical kinds of organizations
  - Independent Software Vendors (ISVs)
  - Online Service Providers (OSPs)
  - Financial Services Organizations (FSOs)
  - Government Organizations (GOs)
- Organization types chosen because
  - They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

Organization benchmark



A sample roadmap

- A roadmap is a phased plan for achieving Objectives for each Security Practice
- The sample on the right is a generic plan for ISVs
  - In Phase 1, several Functions are moved from 0 to 1
  - In Phase 2, some are further advanced from 1 to 2, some remain at 1, and others are moved from 0 to 1
  - Etc…
- SAMM includes case studies with specific details on implementation



SAMM Inventory today

- Introduction and role definition
- Definition of the maturity model
  - Details on each Objective in each Security Practice under each Discipline
- Recommended roadmaps
  - For ISVs and OSPs, planned additions for FSOs, GOs
- Case Studies
  - Example organizations and how they would employ SAMM
  - Medium ISV complete, planned additions for online retailer, etc.
- Mappings to standards and regulations
  - PCI partially complete, planned additions for COBIT, ISO, etc.