



# Developing Secure Applications with OWASP

Martin Knobloch  
[martin.knobloch@owasp.org](mailto:martin.knobloch@owasp.org)

OWASP NL Chapter Board  
OWASP Global Education Committee Chair

**OWASP**

Copyright © The OWASP Foundation  
Permission is granted to copy, distribute and/or modify this document  
under the terms of the OWASP License.

**The OWASP Foundation**  
<http://www.owasp.org>

**Home**  
News  
Projects  
Downloads  
Local Chapters  
Conferences  
Presentations  
Video  
Papers  
Mailing Lists  
About OWASP  
Membership

Reference  
How To...  
Principles  
Threat Agents  
Attacks  
Vulnerabilities  
Countermeasures  
Activities  
Technologies  
Glossary  
Code Snippets  
.NET Project  
Java Project

Search  
Go Search  
Toolbox  
What links here

**Welcome to OWASP**  
the free and open application security community

About • Searching • Editing • New Article • OWASP Categories

Guide  
Top Ten  
WebGoat  
CLASP  
WebScarab  
Contracting  
Testing  
Code Review  
More...  
Statistics • Recent Changes

**OWASP Overview**

The Open Web Application Security Project (OWASP) is dedicated to finding and fighting the causes of insecure software. Everything here is free and open source. The OWASP Foundation is a 501c3 not-for-profit charitable organization that ensures the ongoing availability and support for our work. Participation in OWASP is free and open to all.

**Join webappsec!**  
The OWASP mail list...

**Get Started**  
Find out more...

**Contact OWASP**  
owasp@owasp.org

**Become a Member**  
Support our efforts...

**Featured Story**

**Announcing the OWASP Sprajax Project - the first AJAX Security Scanner**

OWASP thanks Denim Group for the donation of Sprajax, an open source security scanner for AJAX-enabled applications. Sprajax, a Microsoft .Net-based application is the first web security scanner developed specifically to scan AJAX web applications for security vulnerabilities.

"Denim Group is committed to furthering the field of application security," said Dan Cornell, principal of Denim Group, "and by donating Sprajax to OWASP, we intend to generate more discussion around security

**OWASP Conferences**

Register for OWASP AppSec Conference in Seattle Oct. 16-18

The Open Web Application Security Project

AppSec Seattle Conference

Join us for our 5th AppSec Conference October 16-18 in Seattle. Microsoft's Michael Howard will be giving the keynote and you'll hear presentations on topics like Web Services Security, PCI status, Securing AJAX, the Microsoft Secure Development Lifecycle, all the new OWASP projects, and much more. Check the full [agenda](#) website.

OWASP is a not-for-profit, and the OWASP AppSec Conference is an incredible bargain (\$450, \$400 for OWASP members, and \$250 for students). You can attend one of 3 full-day training sessions on the 16th, and the main conference is two full days of presentations, poster sessions, and discussion on the 17th and 18th. You can read all the [details](#) then [register](#) online.

**OWASP Community (add)**



# OWASP Mission

- to make application security "visible," so that people and organizations can make informed decisions about application security risks



# OWASP Resources and Community

## Documentation (Wiki and Books)

- Code Review, Testing, Building, Legal, more ...

## Code Projects

- Defensive, Offensive (Test tools), Education, Process, more ...

## Chapters

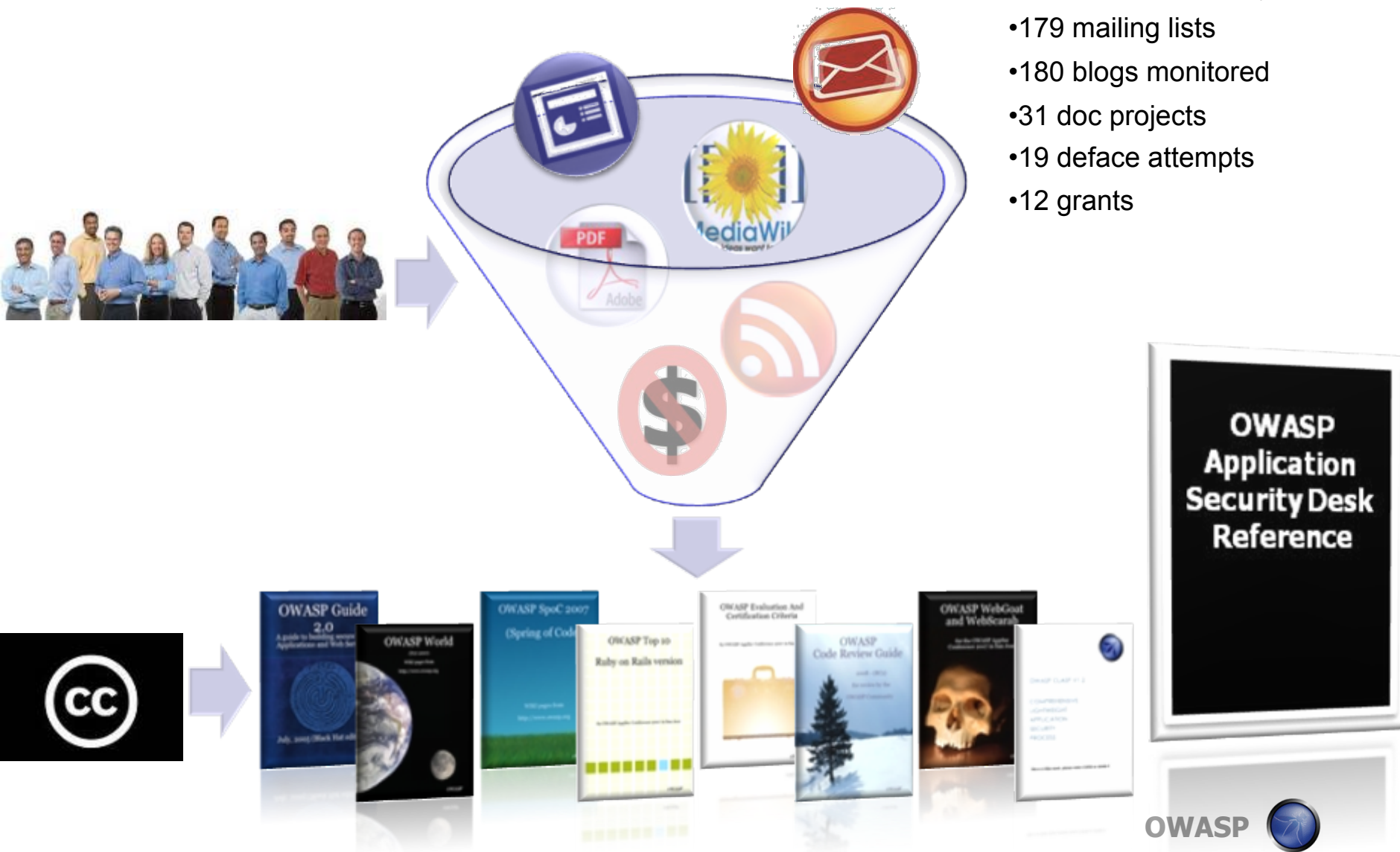
- Over 100 and growing

## Conferences

- Major and minor events all around the world

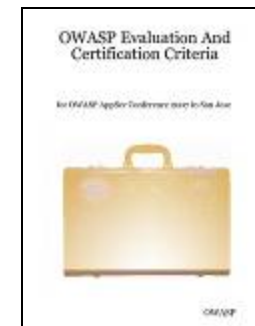
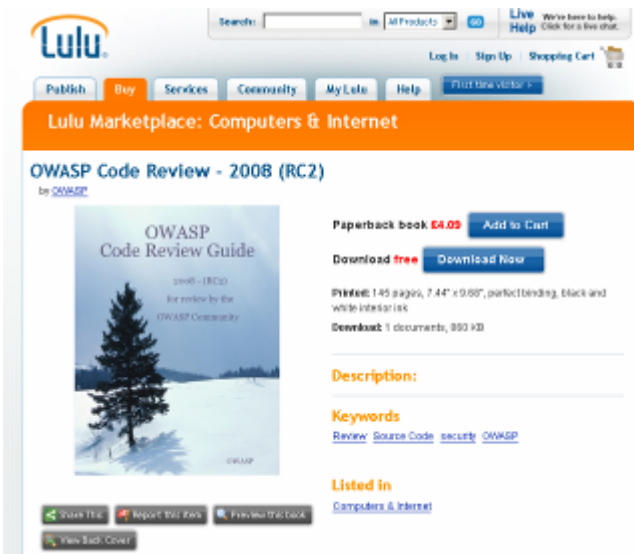
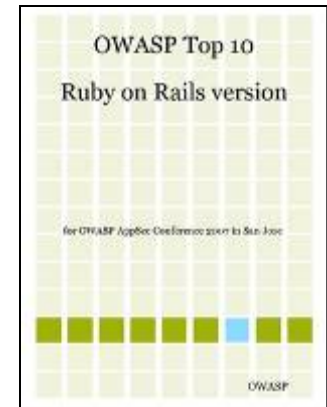
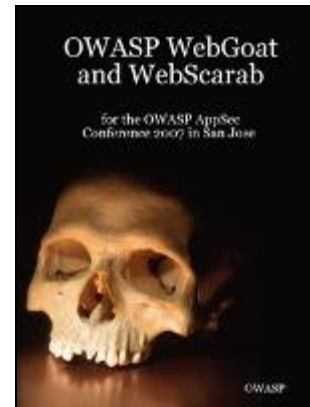
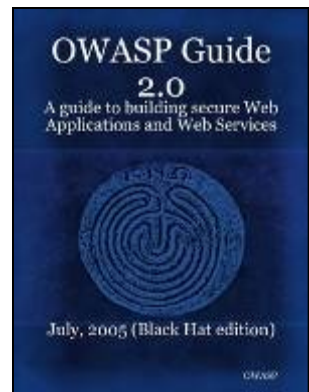
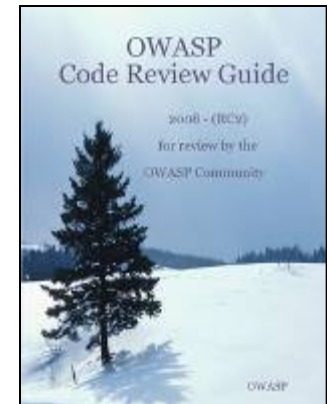
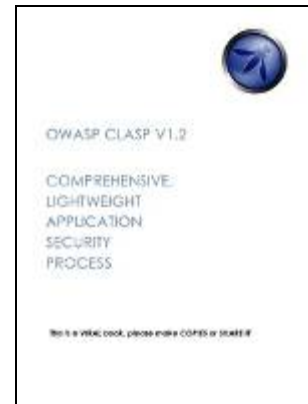
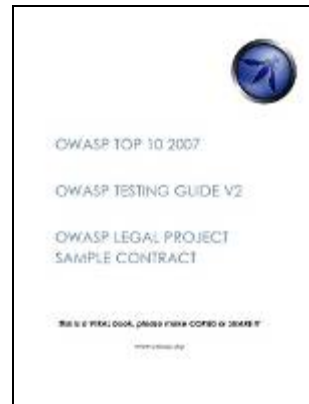
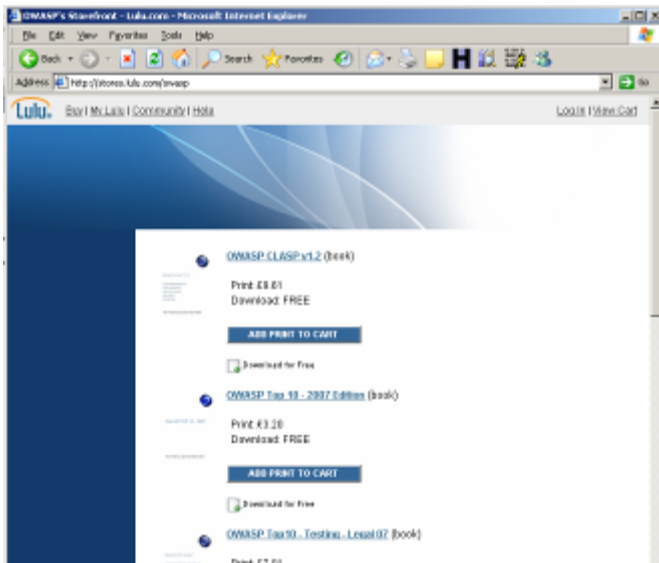
# OWASP KnowledgeBase

- 3,913 total articles
- 427 presentations
- 200 updates per day
- 179 mailing lists
- 180 blogs monitored
- 31 doc projects
- 19 deface attempts
- 12 grants





# OWASP Books (<http://stores.lulu.com/owasp>)



OWASP



# Part of the 'Big 4'

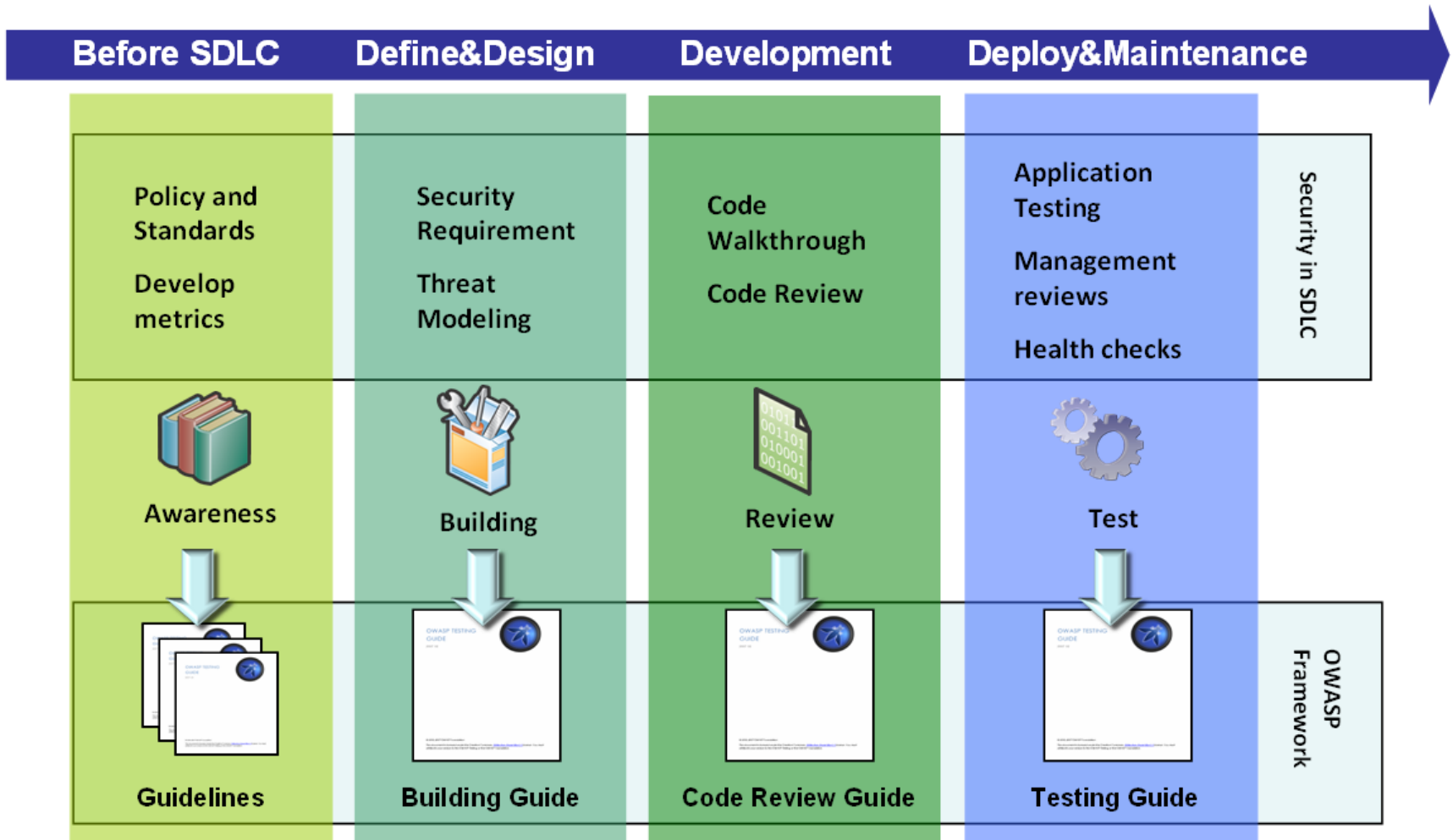
Building  
Guide

Code  
Review  
Guide

Testing  
Guide

Application Security Desk Reference (ASDR)

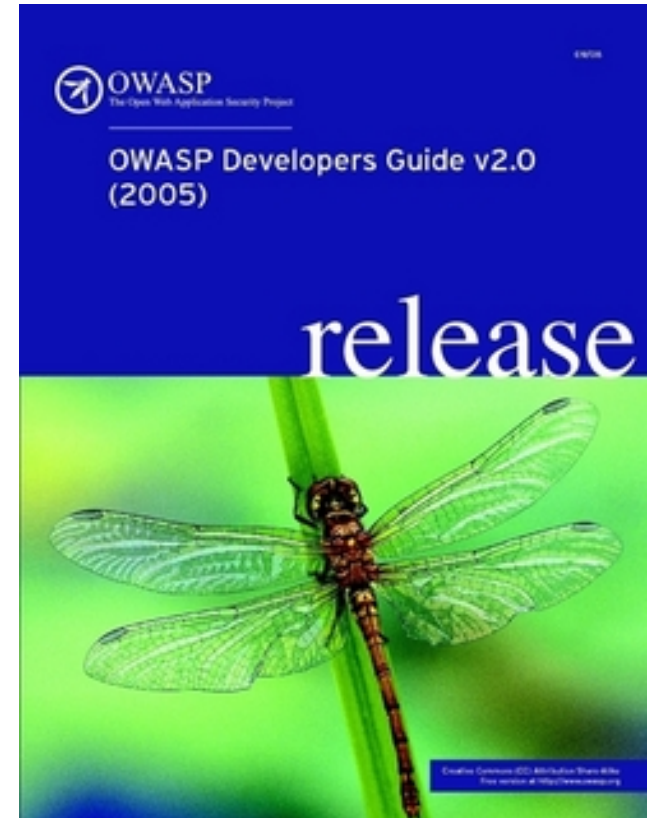
# SDLC & OWASP Guidelines





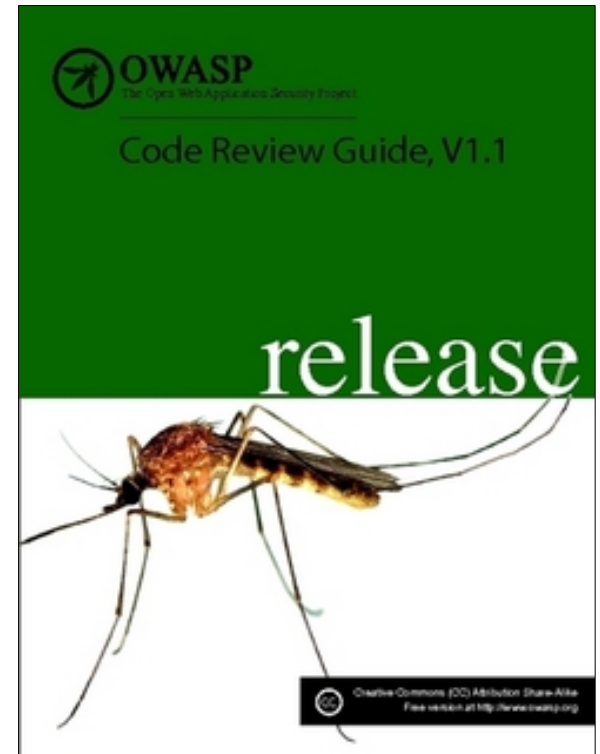
# The Guide v2.0

- Free and open source
  - ▶ Gnu Free Doc License
- Most platforms
  - ▶ Examples are J2EE, ASP.NET, and PHP
- Comprehensive
  - ▶ Thread Modeling
  - ▶ Advise & Best Practices
  - ▶ Web Services
  - ▶ Key AppSec Area's:
    - Authorization/Authentication
    - Session Management
    - Data Validation



# Code Review Guide v1.1

- Introduction
- Preparation
- Security Code Review in the SDLC
- Security Code Review Coverage
- Application Threat Modeling
- Code Review Metrics
- Crawling code
- Searching for code in..
- Code review and PCI DSS..
- Reviewing by technical control:
- Reviewing Code for...
- Additional security considerations:
- How to write an application code review finding



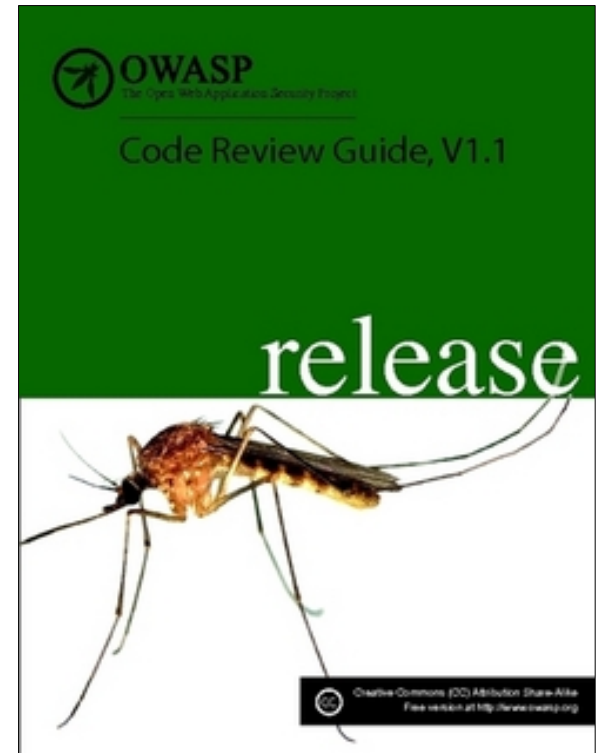
# Code Review Guide v1.1

## ■ Reviewing by technical control:

- ▶ Authentication
- ▶ Authorization
- ▶ Session Management
- ▶ Input Validation
- ▶ Error Handling
- ▶ Secure application deployment
- ▶ Cryptographic controls

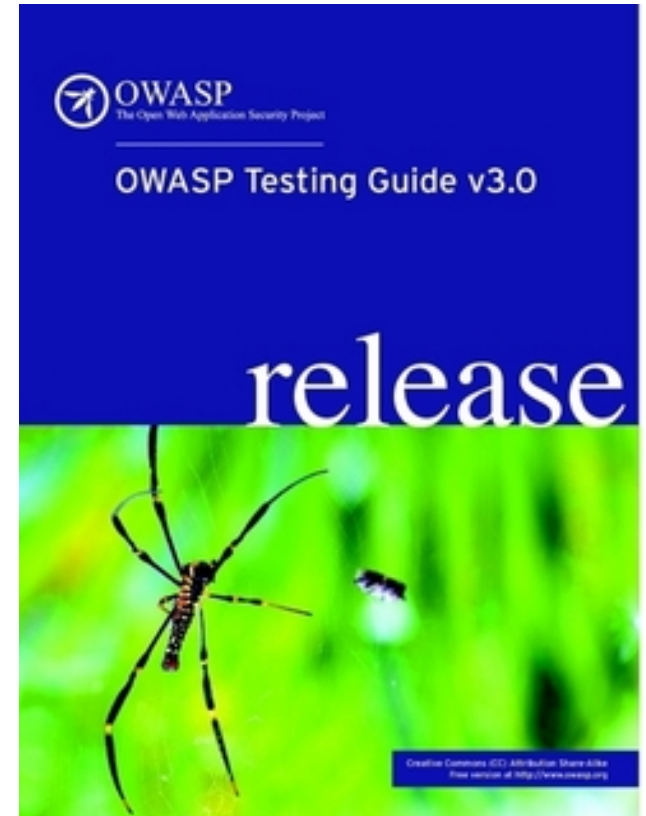
## ■ Reviewing Code for:

- ▶ Buffer Overruns and Overflows
- ▶ OS Injection
- ▶ SQL Injection
- ▶ Data Validation
- ▶ Cross-site scripting
- ▶ Cross-Site Request Forgery issues
- ▶ Logging Issues
- ▶ Session Integrity issues
- ▶ Race Conditions



# Testing Guide v3: Index

1. Frontispiece
  2. Introduction
  3. The OWASP Testing Framework
  4. Web Application Penetration Testing
  5. Writing Reports: value the real risk
- Appendix A: Testing Tools
- Appendix B: Suggested Reading
- Appendix C: Fuzz Vectors
- Appendix D: Encoded Injection



# What's new?

- V2 → 8 sub-categories (for a total amount of 48 controls)
- V3 → 10 sub-categories (for a total amount of 66 controls)
- 36 new articles!

- Information Gathering
- Business Logic Testing
- Authentication Testing
- Session Management Testing
- Data Validation Testing
- Denial of Service Testing
- Web Services Testing
- Ajax Testing

- Information Gathering
- **Config. Management Testing**
- Business Logic Testing
- **Authentication Testing**
- **Authorization Testing**
- **Session Management Testing**
- **Data Validation Testing**
- Denial of Service Testing
- Web Services Testing
- Ajax Testing
- **Encoded Appendix**

# OWASP Tools and Technology

- **Vulnerability Scanners**
- **Static Analysis Tools**
- **Fuzzing**

**Automated Security Verification**



- **Penetration Testing Tools**
- **Code Review Tools**

**Manual Security Verification**



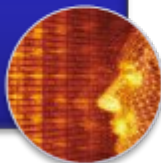
- **ESAPI**

**Security Architecture**



- **AppSec Libraries**
- **ESAPI Reference Implementation**
- **Guards and Filters**

**Secure Coding**



- **Reporting Tools**

**AppSec Management**



- **Flawed Apps**
- **Learning Environments**
- **Live CD**
- **SiteGenerator**

**AppSec Education**





# OWASP Projects

- OWASP .NET Project
- OWASP ASDR Project
- OWASP AntiSamy Project
- OWASP AppSec FAQ Project
- OWASP Application Security Assessment Standards Project
- OWASP Application Security Metrics Project
- OWASP Application Security Requirements Project
- OWASP CAL9000 Project
- OWASP CLASP Project
- OWASP CSRFGuard Project
- OWASP CSRFTester Project
- OWASP Career Development Project
- OWASP Certification Criteria Project
- OWASP Certification Project
- OWASP Code Review Project
- OWASP Communications Project
- OWASP DirBuster Project
- OWASP Education Project
- OWASP Encoding Project
- OWASP Enterprise Security API
- OWASP Flash Security Project
- OWASP Guide Project
- OWASP Honeycomb Project
- OWASP Insecure Web App Project
- OWASP Interceptor Project
- OWASP JBroFuzz
- OWASP Java Project
- OWASP LAPSE Project
- OWASP Legal Project
- OWASP Live CD Project
- OWASP Logging Project
- OWASP Orizon Project
- OWASP PHP Project
- OWASP Pantera Web Assessment Studio Project
- OWASP SASAP Project
- OWASP SQLiX Project
- OWASP SWAAT Project
- OWASP Sprajax Project
- OWASP Testing Project
- OWASP Tools Project
- OWASP Top Ten Project
- OWASP Validation Project
- OWASP WASS Project
- OWASP WSFuzzer Project
- OWASP Web Services Security Project
- OWASP WebGoat Project
- OWASP WebScarab Project
- OWASP XML Security Gateway Evaluation Criteria Project
- OWASP on the Move Project

# Part of the 'Big 4 + 1'

ASVS

Building  
Guide

Code  
Review  
Guide

Testing  
Guide

Application Security Desk Reference (ASDR)

# **\* Key Application Security Areas \***

# Authentication

- Major re-write
- Covers a long list of areas:
  - ▶ Strong authentication, federated authentication, client-side, positive authentication, referer checks, remember my password, default accounts, password strength, encryption/hashing, automated reset, brute force, timeout, logout, self-registration, CAPTCHA

# Authorization

## ■ Topics include:

- ▶ ACL's, centralization, authorization matrix, client-side authorization tokens, access to functions, access to static resources

# Session Management

- Complete re-write

- Topics Include:

- ▶ Permissive session generation, exposed session variables, page and form tokens, weak session ids, session encryption, session forging, timeout, logout, hijacking, session brute forcing, session fixation, HTTP split session attacks, HTTP request smuggling



# Data Validation

- Complete re-write

- ▶ Considerably shorter!

- State of the art validation strategies

- ▶ “Sanitize” is no longer an acceptable first choice
  - ▶ Practical advice for several platforms

- Topics:

- ▶ Integrity checks, validation, business rule validation, parameter tampering, hidden fields, ASP.NET viewstate, URL encoding, HTML entity encoding, special characters

# Interpreter Injection

- Shows how injection really works
  - ▶ For any interpreter
- Covers many different interpreters
  - ▶ User agent injection
  - ▶ SQL Injection
  - ▶ ORM Injection
  - ▶ OS Command Injection
  - ▶ Code Injection
  - ▶ LDAP Injection
  - ▶ XML Injection (XPath / XSLT)

# Canonicalization

- The process of making Unicode and other encodings “real” to the underlying application
  - ▶ One of the last bastions of unexplored vulnerability
- Difficult to protect against
  - ▶ Unicode
  - ▶ Locale
  - ▶ Multiple encoding

# Error Handling, Logging, Auditing

- Complete re-write

- Topics Include:

- ▶ Traceability - aims for SOX compliance
- ▶ Error messages, error handling
- ▶ Don't log noise, destruction, audit trails

# File System

- Goal: Minimize dangers from file based operations
- Topics:
  - ▶ File referencing, defacement, insecure permissions, insecure indexing, unmapped files, temp files, old files, second order injection

# Buffer overflows

- New(ish) section for one of the oldest security problems
  - ▶ Heap, Stack, Buffer overflows
  - ▶ Integer and array overflows
  - ▶ Unicode overflows
  - ▶ String format overflows
- Not really an issue for Java, .NET, PHP
  - ▶ **Unless** you're invoking native libraries or exec'ing operating system commands



# Administrative Interfaces

- Must have segregation of duties
- Administrators are not users
  - ▶ To be effective, ensure that admin application uses completely different RDBMS users
  - ▶ Prefer separate servers and access control lists
- Security through obscurity not good enough
- Strong authentication

# Cryptography

- Revamped section
- Future proofing (SHA1 / MD5 anyone?)
- How to select algorithms
- Poor secret storage
- Stream ciphers

# Privacy

- Objective is to ensure that the tracks left by an application are minimalist and safe (enough)
- Completely revamped
- Major controls:
  - ▶ Laws in effect
  - ▶ Look for browser droppings (cookies, history, logs, etc)
  - ▶ The (in)-effectiveness of cache control
  - ▶ GET vs POST
  - ▶ What SSL really hides
- Various EU, AU, and US laws compared
- Information disclosure
- “Front page of the paper” test

# Configuration

- New Section
- Objective is to ensure that an application is safe out of the box
- Code Access Security Policies
- Default passwords (NO!)
- Clear text passwords in config files
- Connecting to RDBMs and middleware

# Maintenance

## ■ Topics include:

- ▶ Security incident response, rescues and fixes, update notifications, permission checking

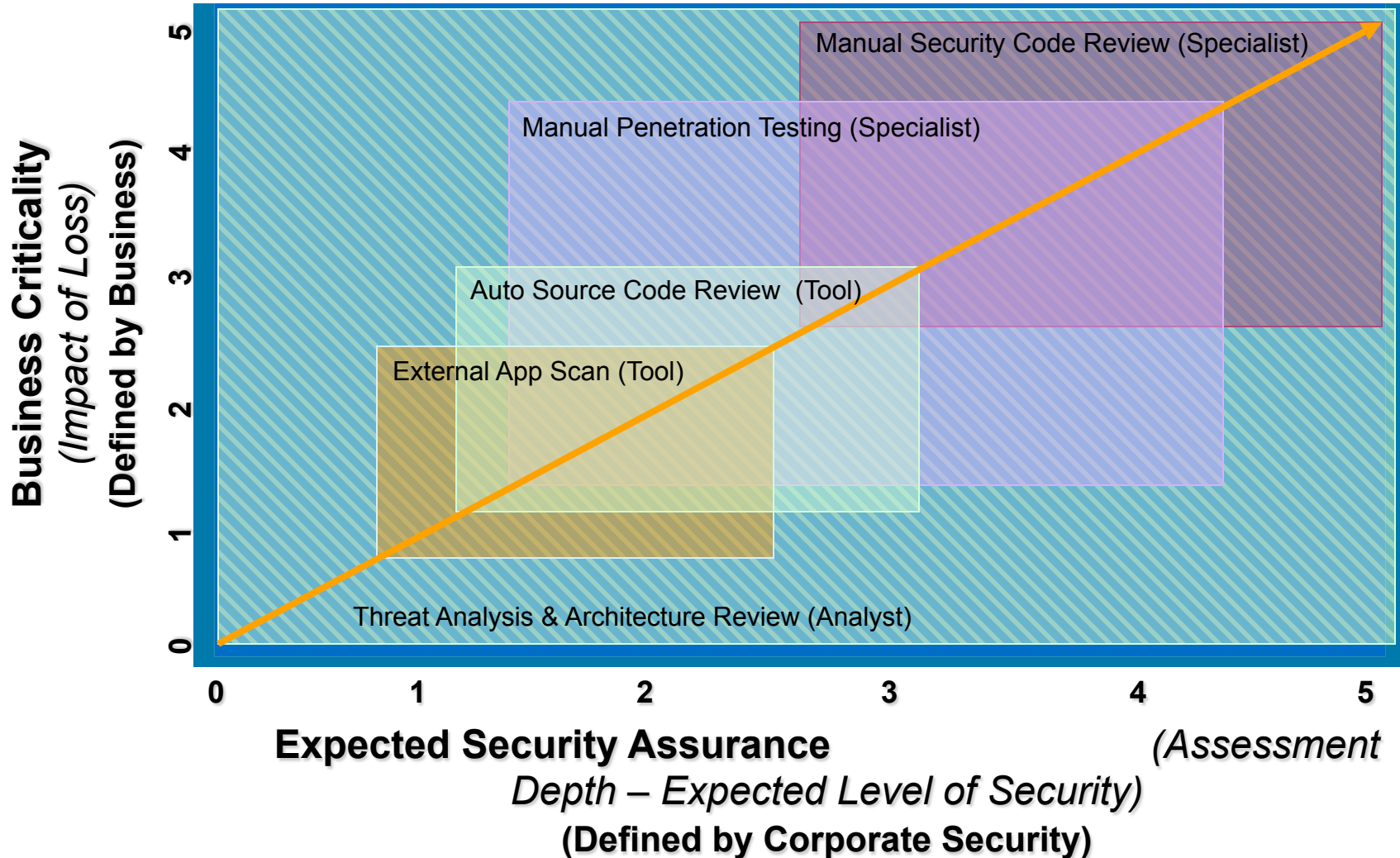
# Denial of Service Attacks

## ■ Topics include:

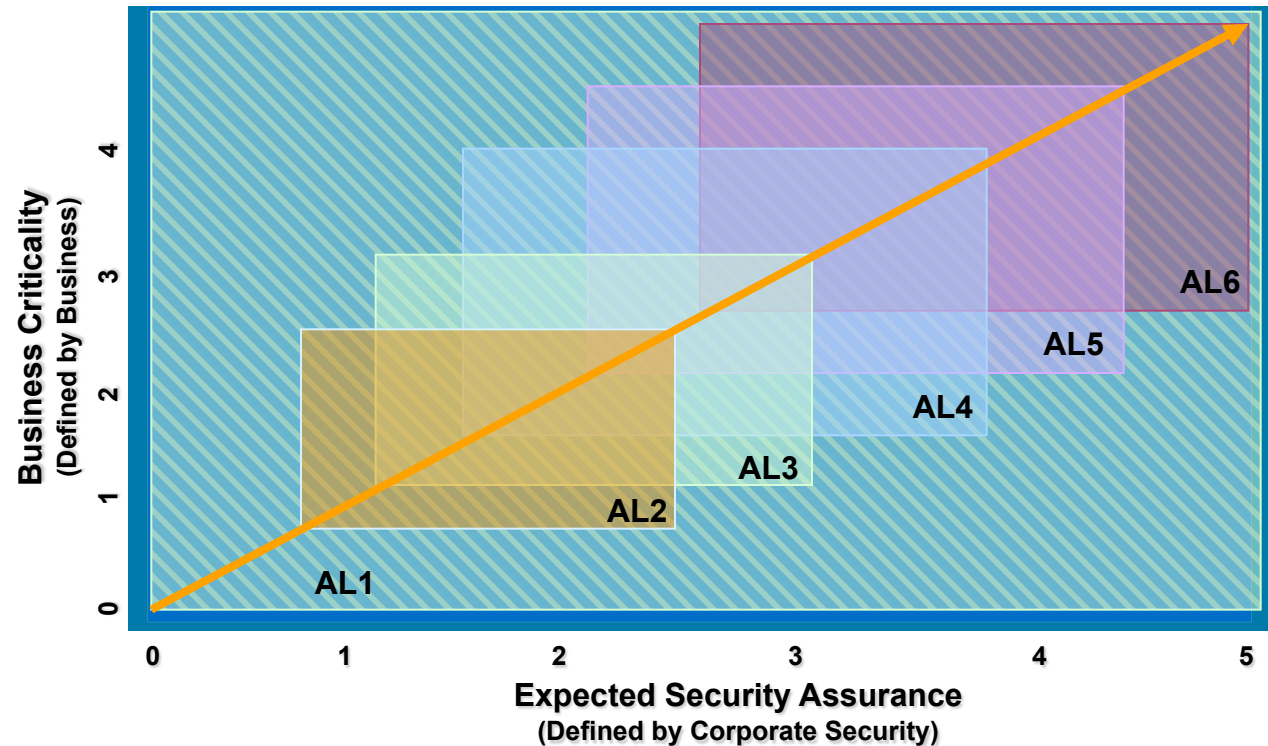
- ▶ Excessive consumption of resources
  - Disk I/O
  - CPU
  - Network I/O
- ▶ User Account Lockout



# Application Security Verification Standard

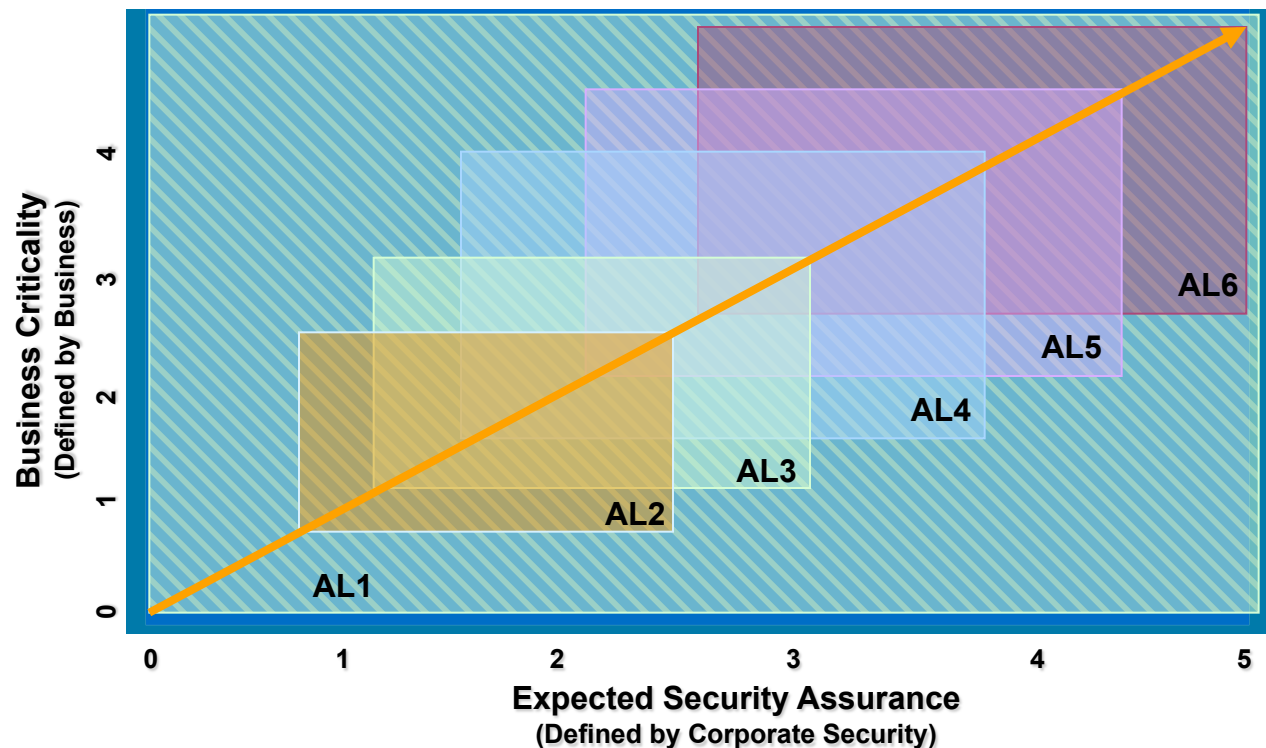


# Appliaction Security Verification Standaard



- **AL1: Architecture Review/Threat Analysis** - Design level review to identify critical assets, sensitive data stores and business critical interconnections. In addition to architecture reviews is threat analysis to determine potential attack vectors, which could be used in testing.
- **AL2: Quick Hit Application Security Check** - Automated scans (either external vulnerability scan or code scan or both) with minimal interpretation and verification.
- **AL3: Basic Application Security Check** – AL2 + verification and validation of scan results. Security areas not scanned (encryption, access control, etc.) must be lightly tested or code reviewed.

# Application Security Verification Standard



- **AL4: Standard Application Security Verification** – AL3 + verification of common security mechanisms and common vulnerabilities using either manual penetration testing or code review or both. Not all instances of problems found - Sampling allowed.
- **AL5: Enhanced Application Security Verification** – AL1 + AL3 + verification of all security mechanisms and vulnerabilities based on threat analysis model using either manual penetration testing or code review or both.
- **AL6: Comprehensive Application Security Verification** – AL1 + AL4 + search for malicious code. All code must be manually reviewed against a standard and all security mechanisms tested.

# CLASP

- Comprehensive, Lightweight Application Security Process
  - ▶ Centered around 7 AppSec Best Practices
  - ▶ Cover the entire software lifecycle (not just development)
- Adaptable to any development process
  - ▶ Defines roles across the SDLC
  - ▶ 24 role-based process components
  - ▶ Start small and dial-in to your needs



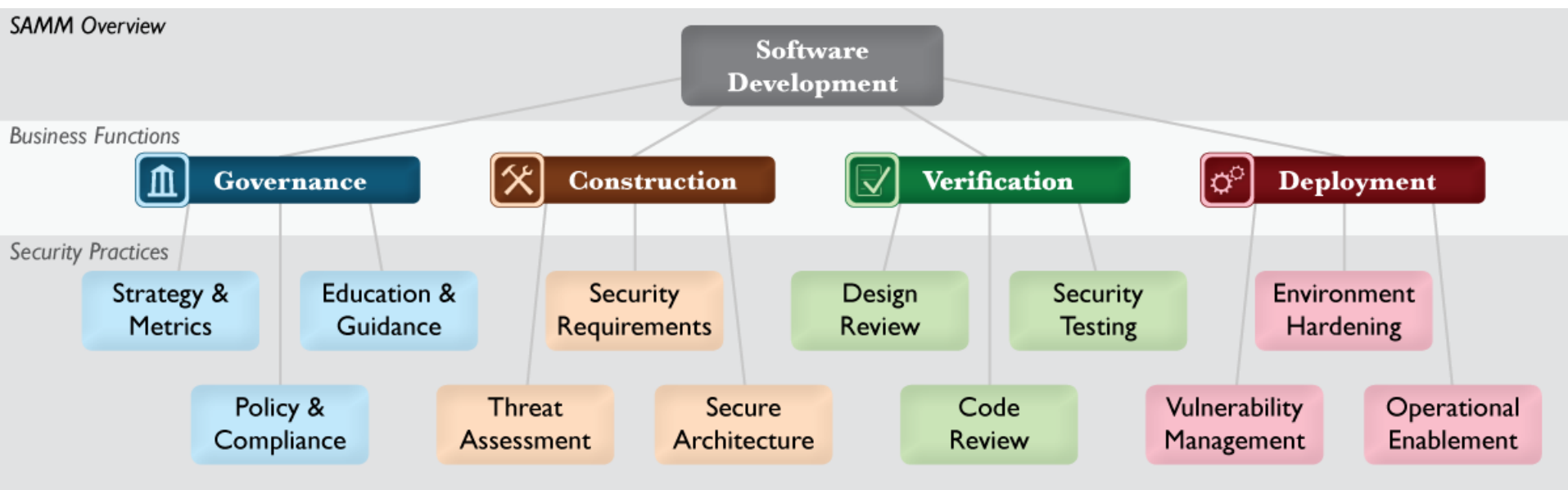
# SAMM Business Functions

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



# 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



# Subscribe to Chapter mailing list

- Post your (Web)AppSec questions
- Keep up to date!
- Get monthly news letters
- Contribute to discussions!



# That's it...

- Any Questions?



<http://www.owasp.org>

<http://www.owasp.org/index.php/Portuguese>

[martin.knobloch@owasp.org](mailto:martin.knobloch@owasp.org)

## Thank you!