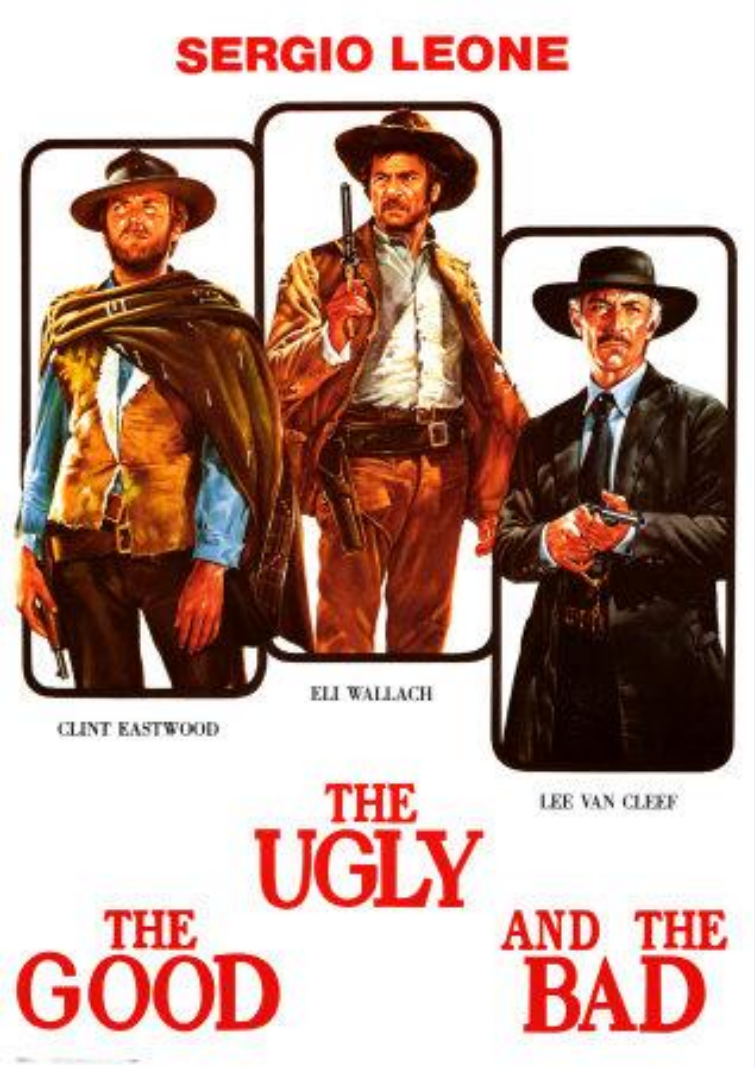
OWASP InfoSec Romania 2013

Secure Development Lifecycle, The good, the bad and the ugly!

OWASP
The Open Web Application Security Project

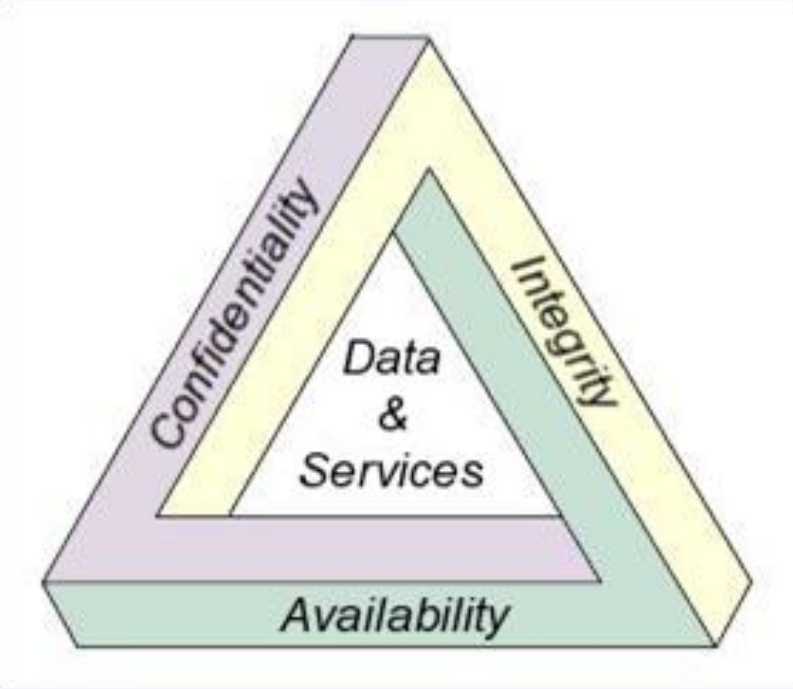
October 25th 2013

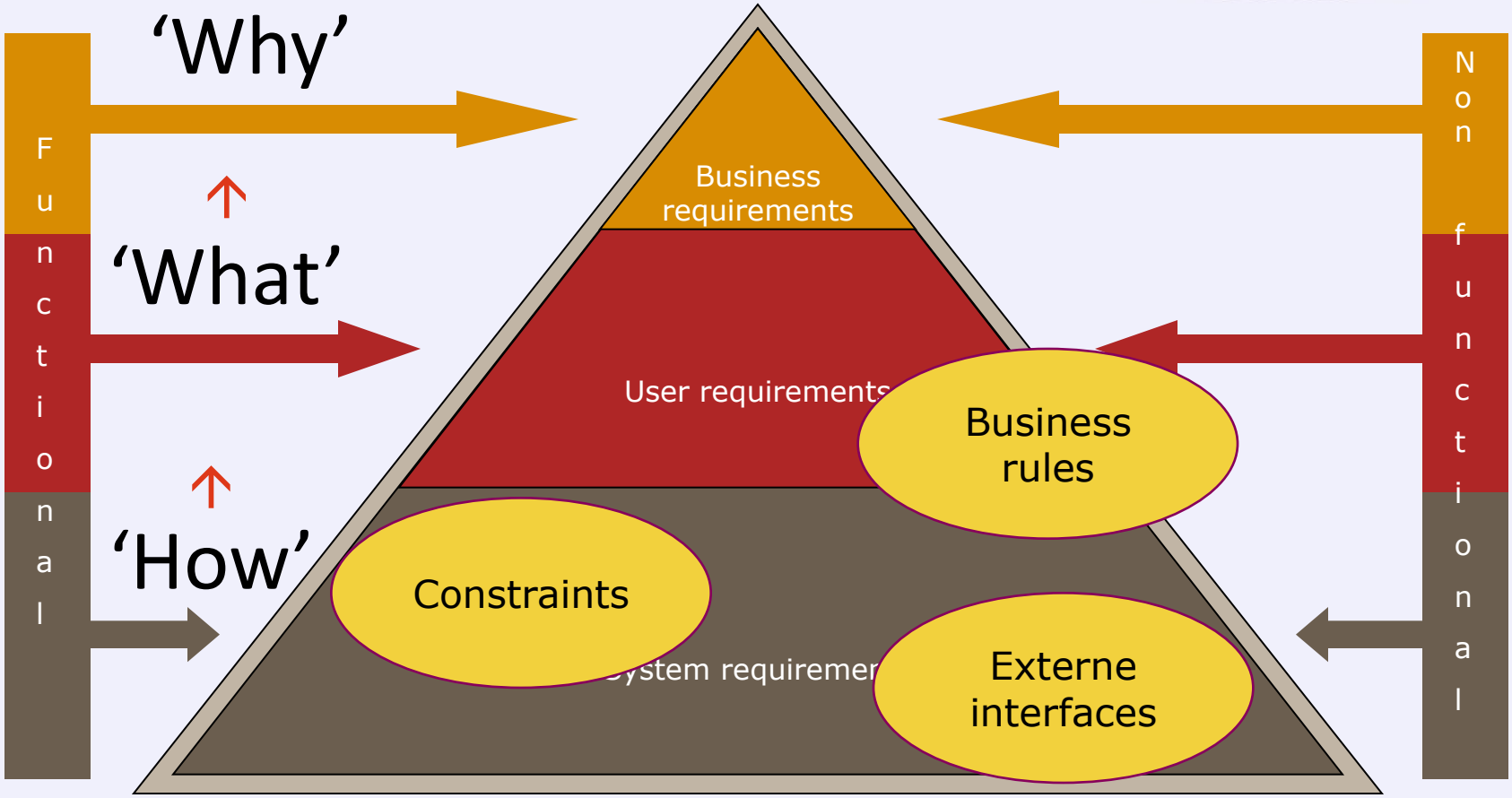
Martin Knobloch
OWASP Netherlands Chapter Leader

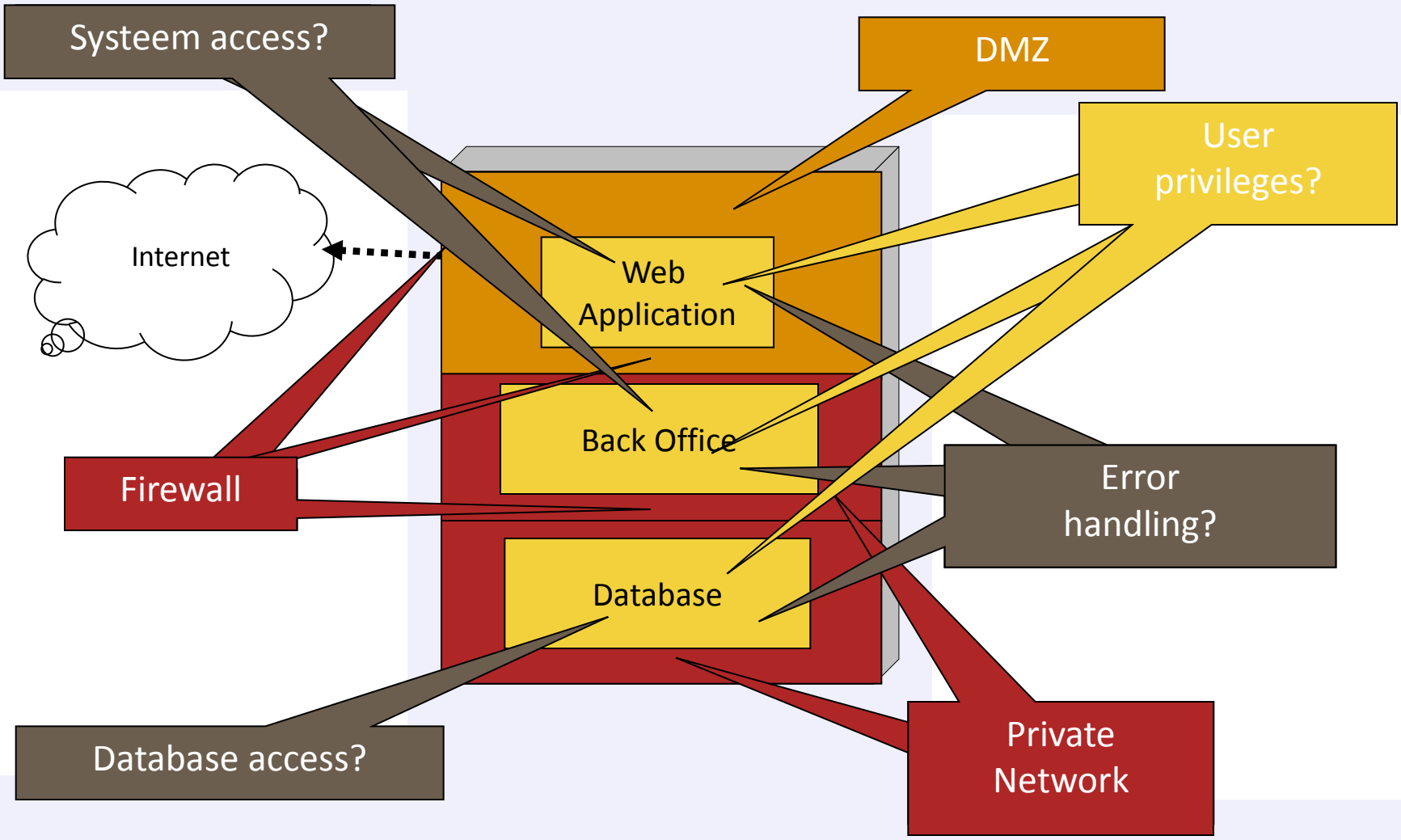

SERGIO LEONE
CLINT EASTWOOD
ELI WALLACH
LEE VAN CLEEF
THE GOOD
THE UGLY
AND THE BAD

Applications are about information!

- 3 pillars of Information Security:
  - Confidentiality
  - Integrity
  - Availability

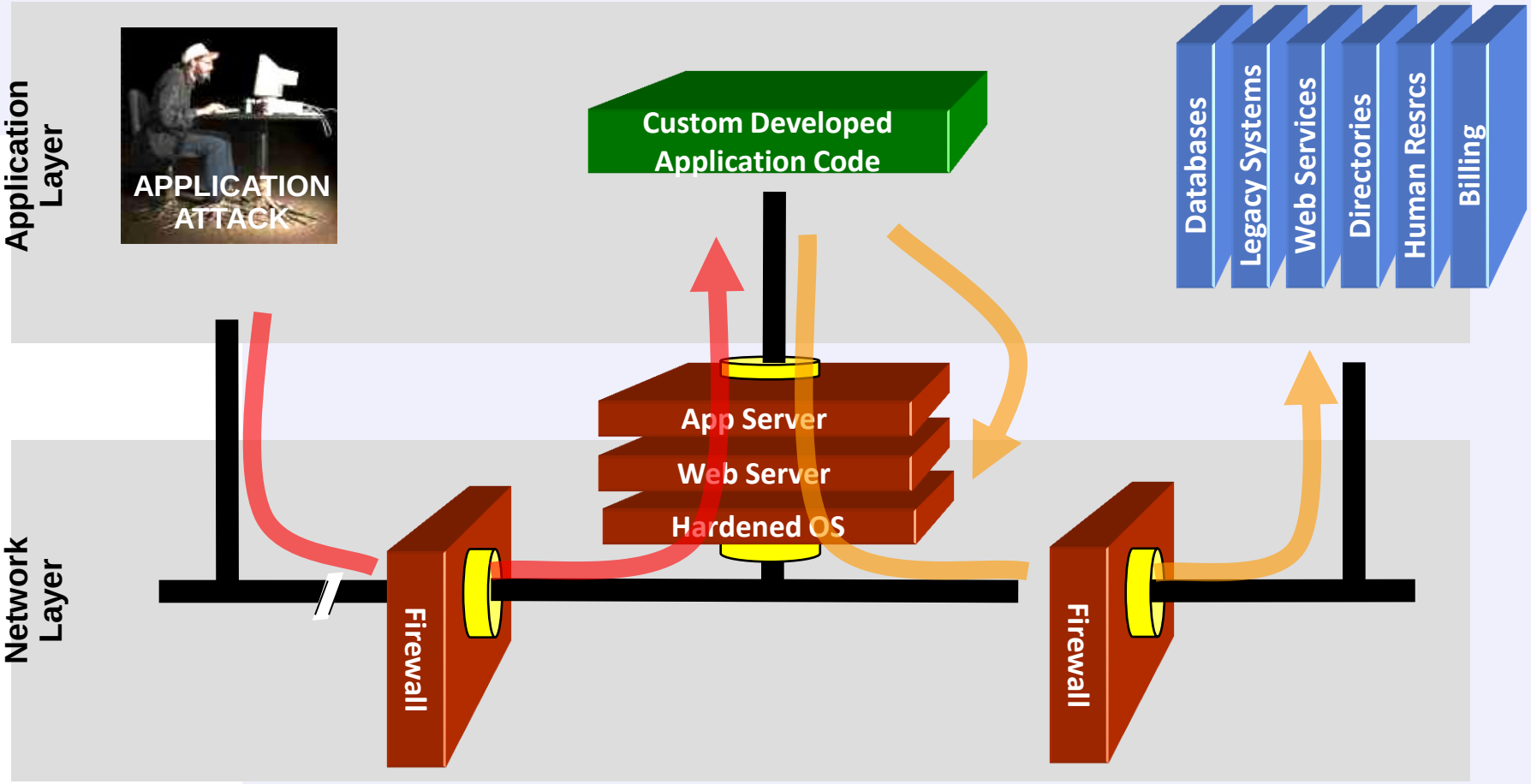



OWASP
The Open Web Application Security Project
‘Why’
‘What’
‘How’
Functional
Non functional
Business requirements
User requirements
System requiremer
Business rules
Constraints
Externe interfaces


Systeem access?
DMZ
User privileges?
Internet
Web Application
Back Office
Firewall
Error handling?
Database
Database access?
Private Network

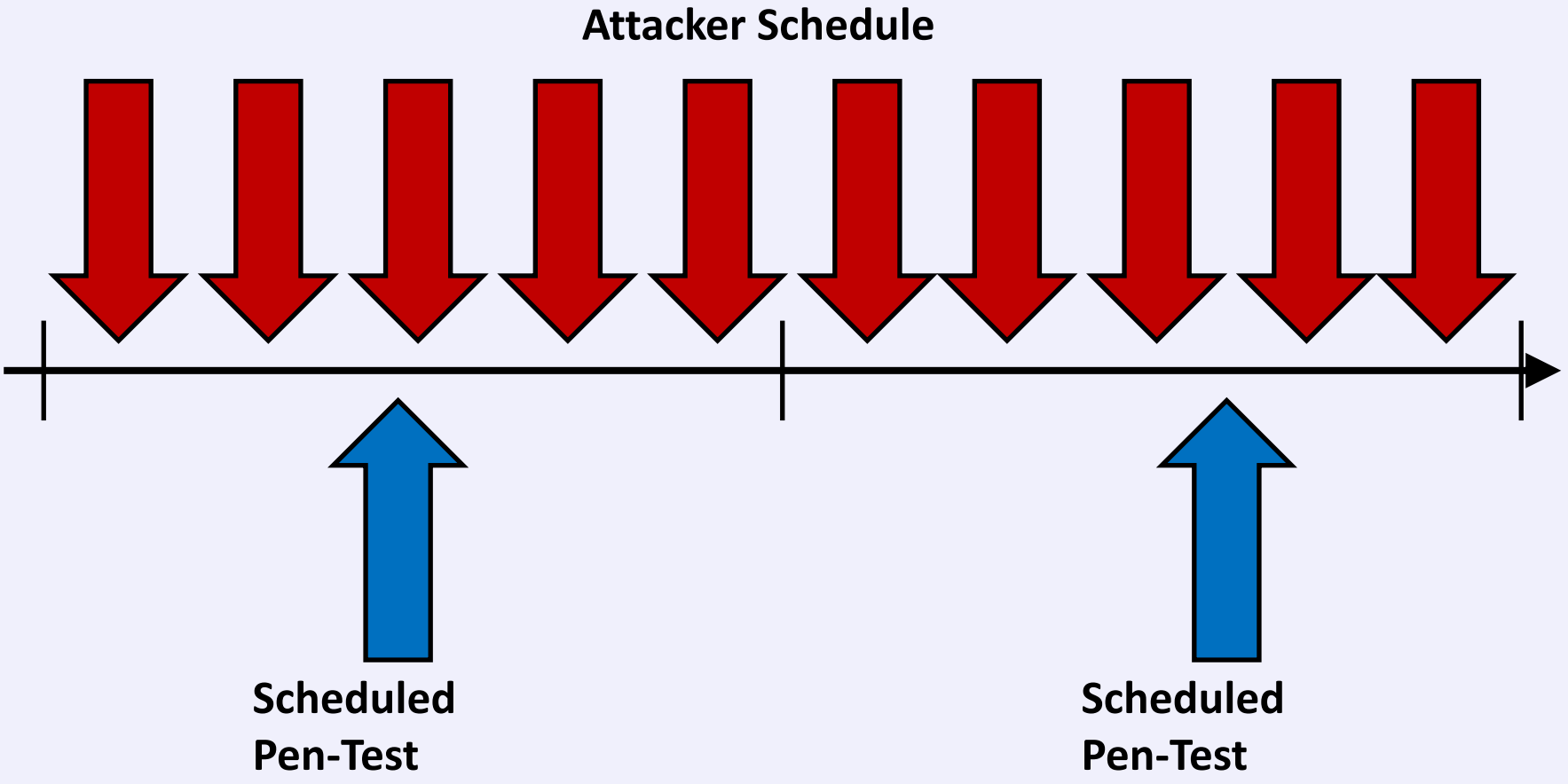
Your security "perimeter" has huge holes at the application layer

OWASP
The Open Web Application Security Project



You can't use network layer protection (firewall, SSL, IDS, hardening) to stop or detect application layer attacks

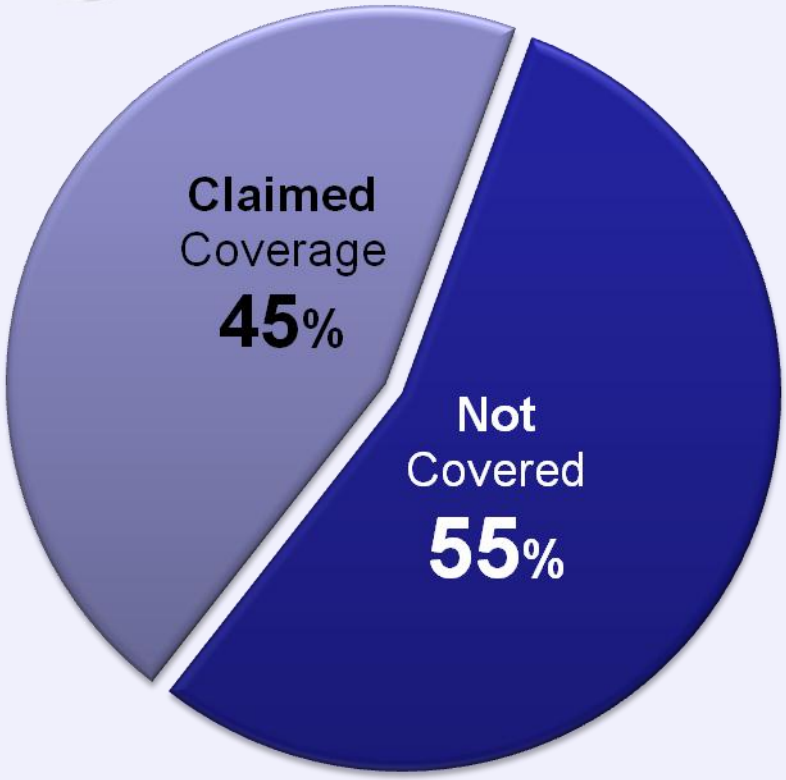
An Attacker has 24x7x365 to Attack



The Defender has 20 man days per year to detect and defend

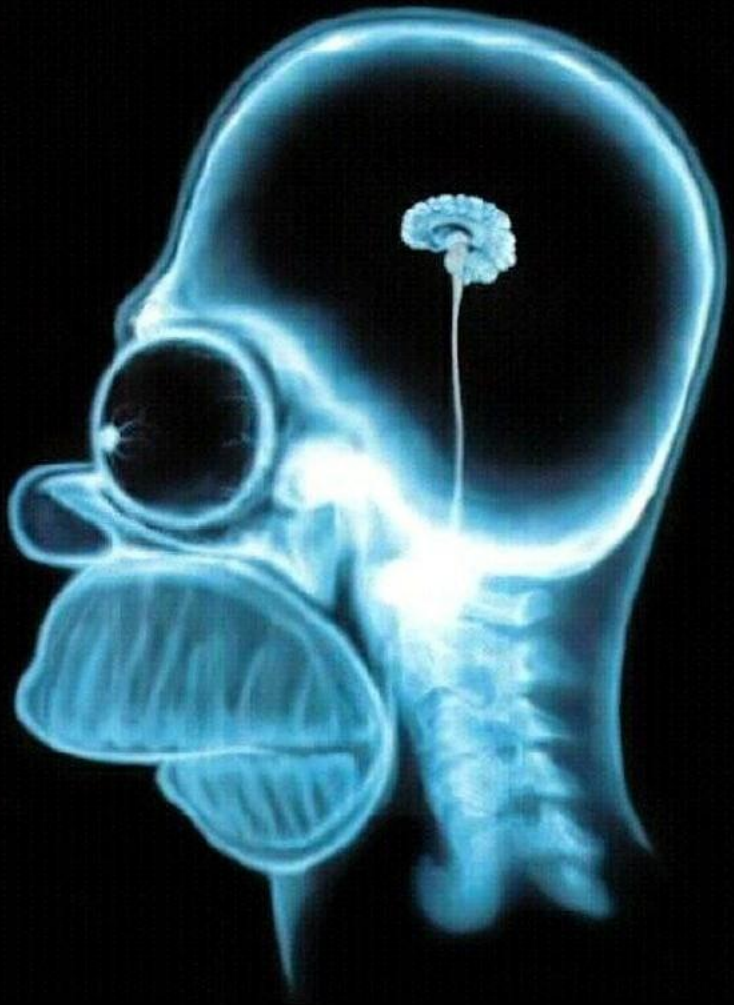
Tools – At Best 45%

OWASP
The Open Web Application Security Project



- MITRE found that all application security tool vendors' **claims** put together cover only 45% of the known vulnerability types (695)
- They found very little overlap between tools, so to get 45% you need them all (assuming their claims are true)

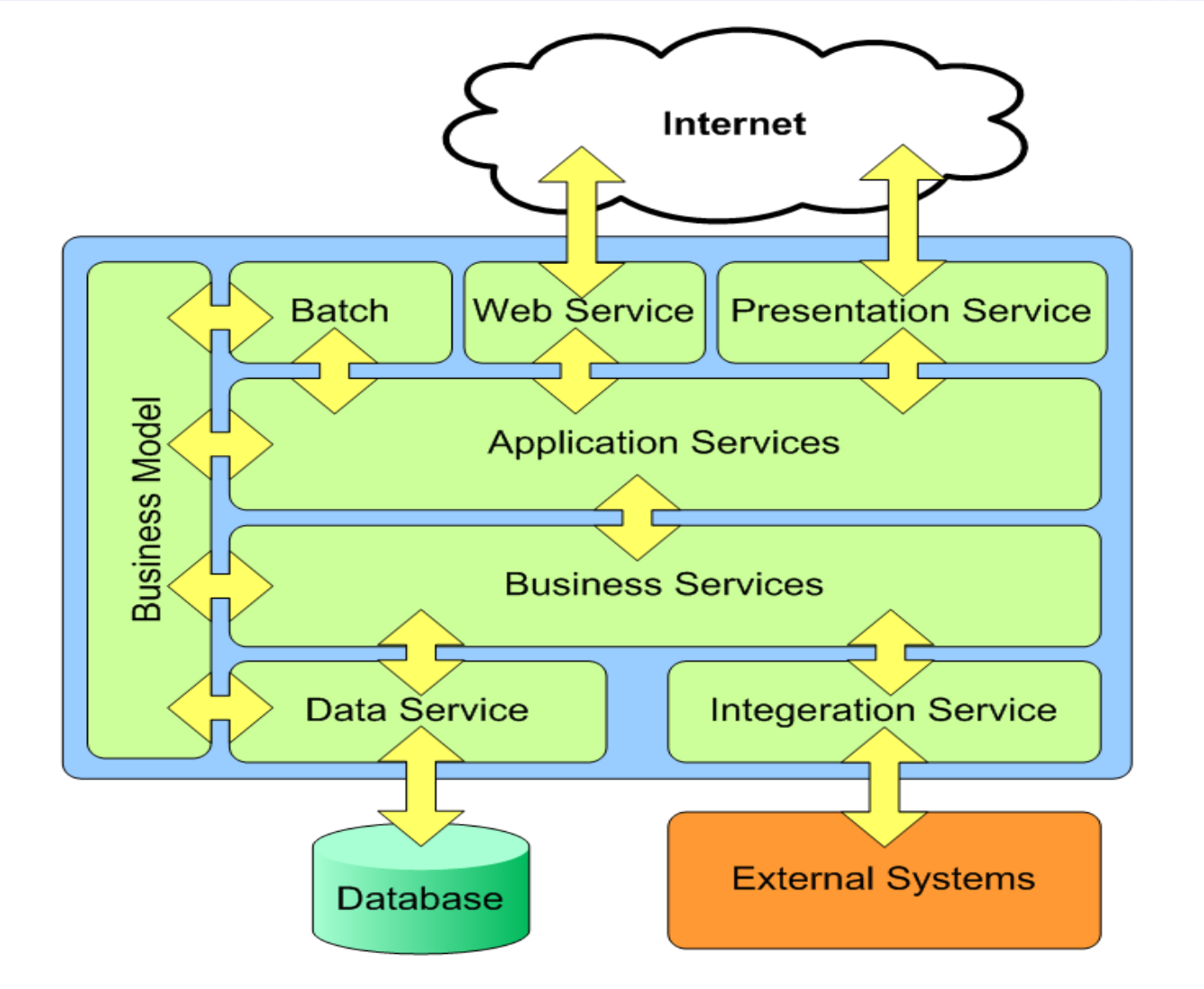
SECURITYINNOVATION · FORTIFY SOFTWARE · PALAMIDA · watchfire · Klocwork · Secure Software · AppSIC · CENZIC · CORE SECURITY TECHNOLOGIES · PARASOFT · Coverity · proServices · SofCheck · SPI DYNAMICS · VERACODE · OUNCE LABS

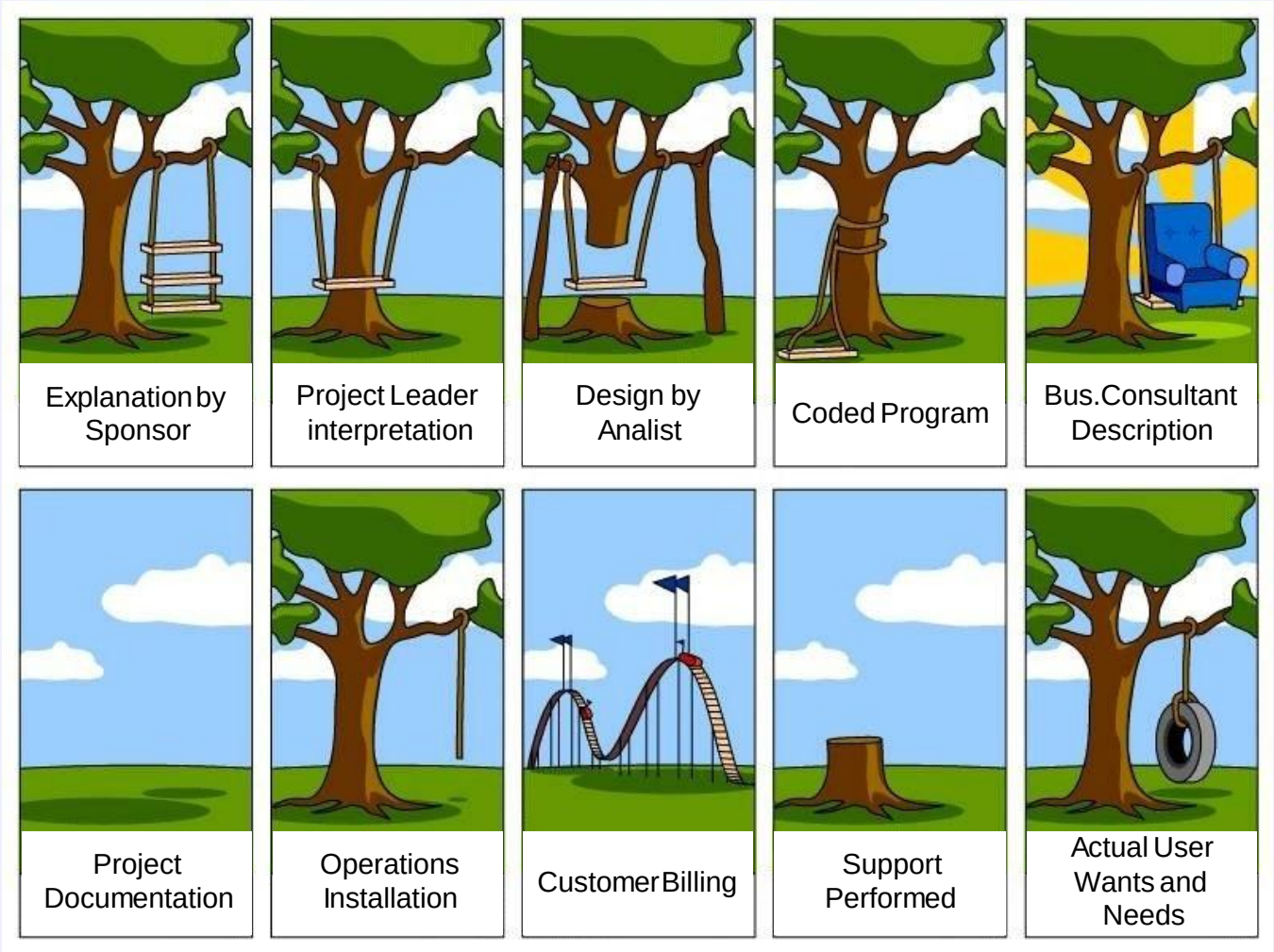


OWASP

The Open Web Application Security Project

OWASP Top 10 – 2010 (Previous)	OWASP Top 10 – 2013 (New)
A1 – Injection	A1 – Injection
A3 – Broken Authentication and Session Management	A2 – Broken Authentication and Session Management
A2 – Cross-Site Scripting (XSS)	A3 – Cross-Site Scripting (XSS)
A4 – Insecure Direct Object References	A4 – Insecure Direct Object References
A6 – Security Misconfiguration	A5 – Security Misconfiguration
A7 – Insecure Cryptographic Storage – Merged with A9 →	A6 – Sensitive Data Exposure
A8 – Failure to Restrict URL Access – Broadened into →	A7 – Missing Function Level Access Control
A5 – Cross-Site Request Forgery (CSRF)	A8 – Cross-Site Request Forgery (CSRF)
<buried in A6: Security Misconfiguration>	A9 – Using Known Vulnerable Components
A10 – Unvalidated Redirects and Forwards	A10 – Unvalidated Redirects and Forwards
A9 – Insufficient Transport Layer Protection	Merged with 2010-A7 into new 2013-A6


Internet
Batch
Web Service
Presentation Service
Business Model
Application Services
Business Services
Data Service
Integeration Service
Database
External Systems

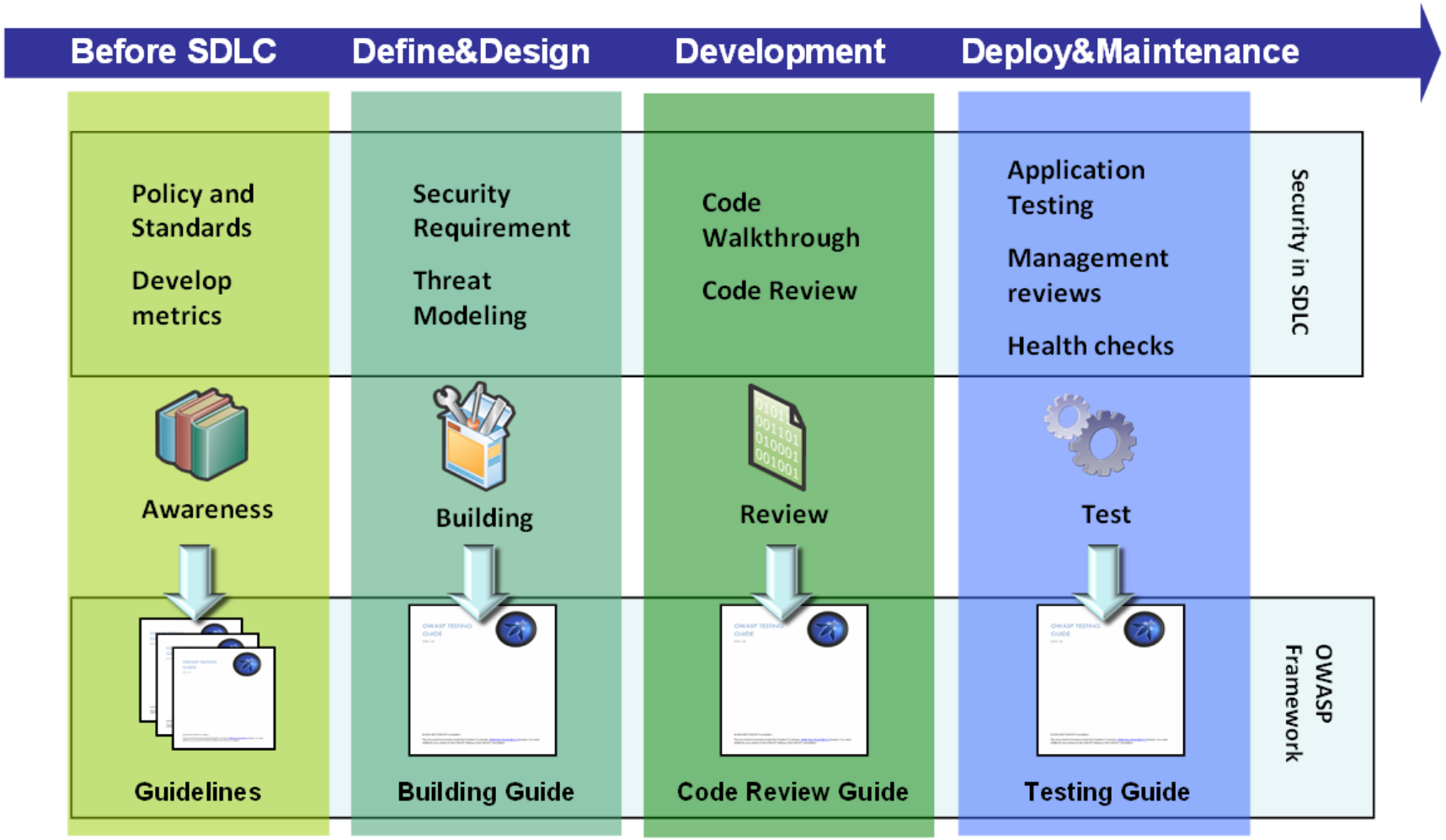

Explanation by Sponsor
Project Leader interpretation
Design by Analist
Coded Program
Bus.Consultant Description
Project Documentation
Operations Installation
Customer Billing
Support Performed
Actual User Wants and Needs


OWASP
The Open Web Application Security Project
40 km

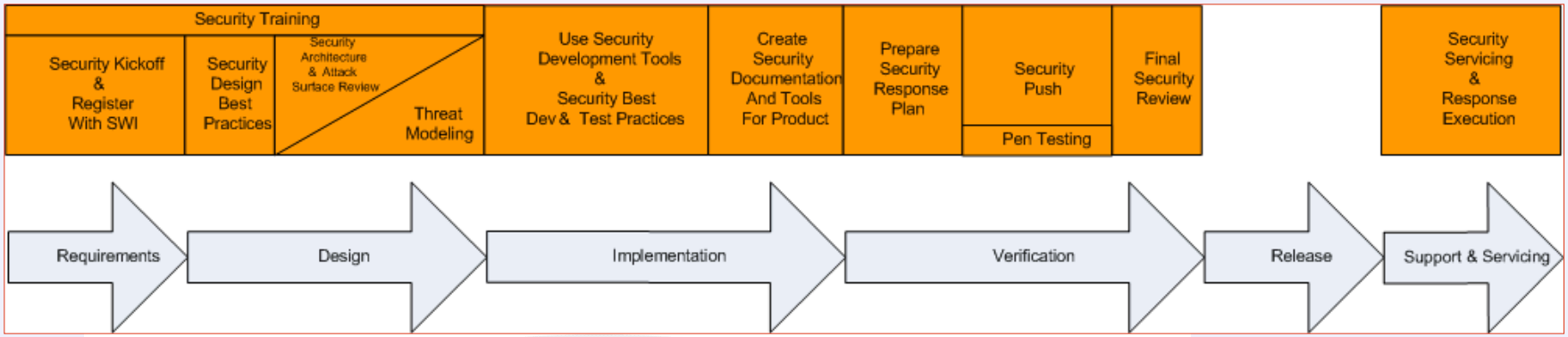

OUR GOAL IS TO WRITE BUG-FREE SOFTWARE. I'LL PAY A TEN-DOLLAR BONUS FOR EVERY BUG YOU FIND AND FIX.
S. Adams E-mail: SCOTTADAMS@AOL.COM
YAHOO!
WE'RE RICH
YES!!! YES!!! YES!!!
1/13 © 1995 United Feature Syndicate, Inc.(NYC)
I HOPE THIS DRIVES THE RIGHT BEHAVIOR.
I'M GONNA WRITE ME A NEW MINIVAN THIS AFTER-NOON!

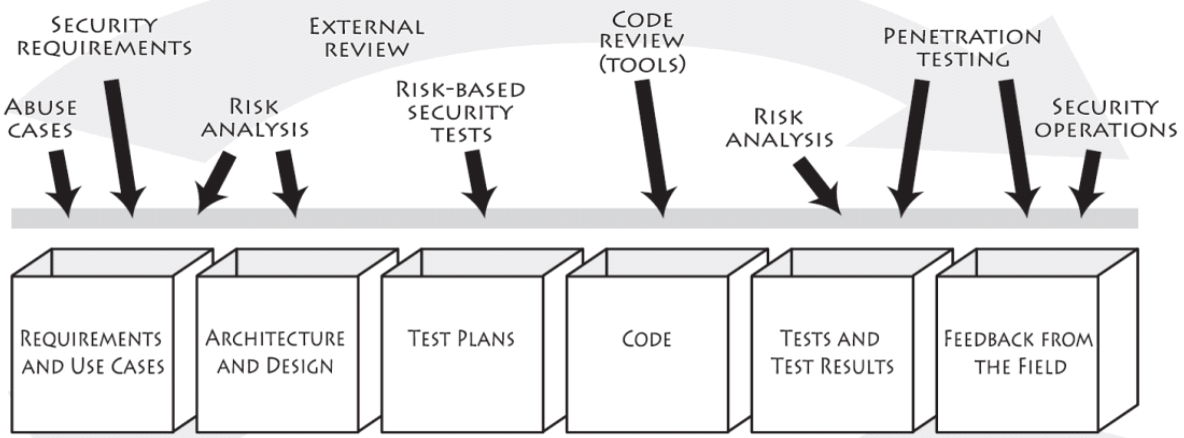
SDLC & OWASP Guidelines

OWASP
The Open Web Application Security Project



Microsoft SDL

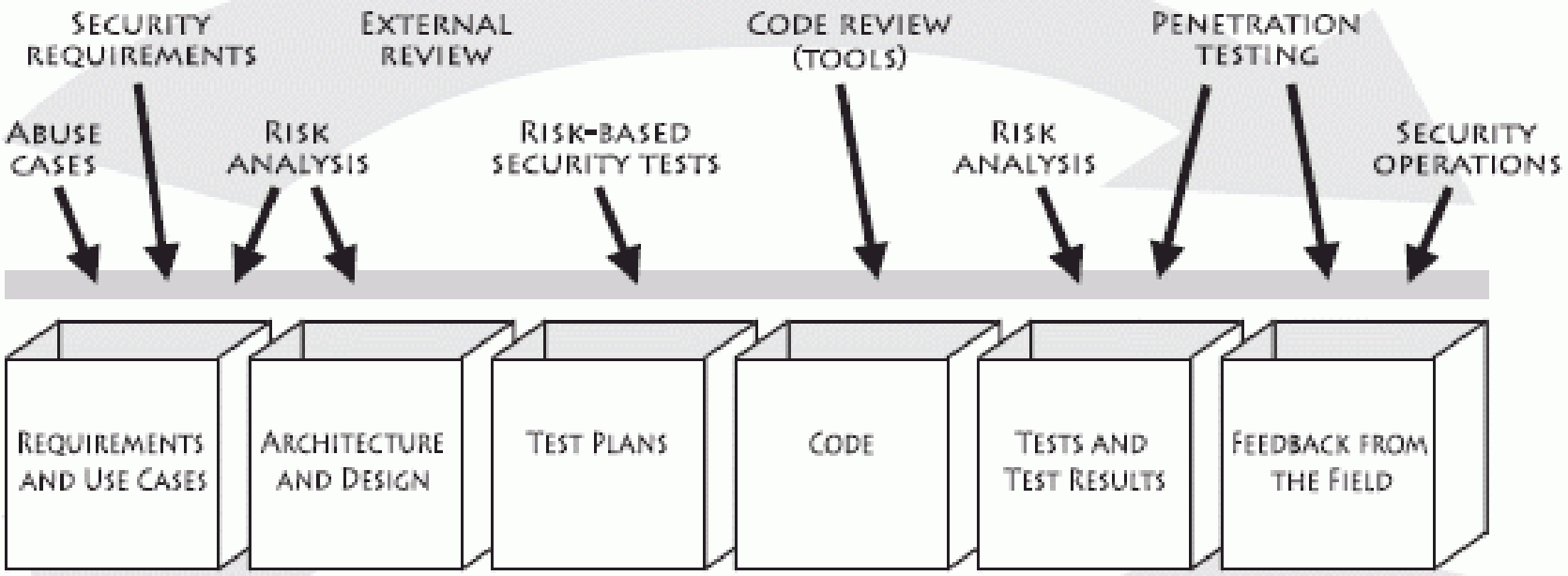




Touchpoints

CLASP

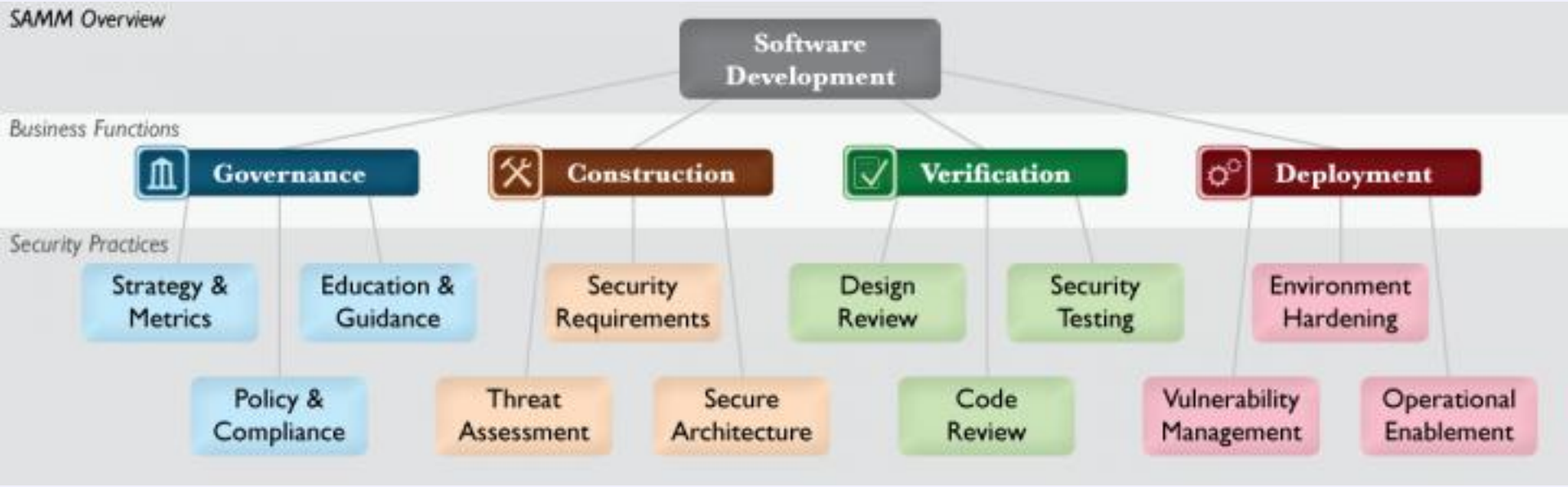
Training	Requirements	Design	Implementation	Verification	Release	Response
• Core training	• Define quality gates/bug bar • Analyze security and privacy risk	• Attack surface analysis • Threat modeling	• Specify tools • Enforce banned functions • Static analysis	• Dynamic/Fuzz testing • Verify threat models/attack surface	• Response plan • Final security review • Release archive	• Response execution


Security Requirements
External Review
Code Review (Tools)
Penetration Testing
Abuse Cases
Risk Analysis
Risk-based Security Tests
Risk Analysis
Security Operations
Requirements and Use Cases
Architecture and Design
Test Plans
Code
Tests and Test Results
Feedback from the Field

CLASP

OWASP
The Open Web Application Security Project

- 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 Overview
Software Development
Business Functions
Governance
Construction
Verification
Deployment
Security Practices
Strategy & Metrics
Education & Guidance
Security Requirements
Design Review
Security Testing
Environment Hardening
Policy & Compliance
Threat Assessment
Secure Architecture
Code Review
Vulnerability Management
Operational Enablement

Part of the ‘Big 4’

OWASP
The Open Web Application Security Project

Building Guide

Code Review Guide

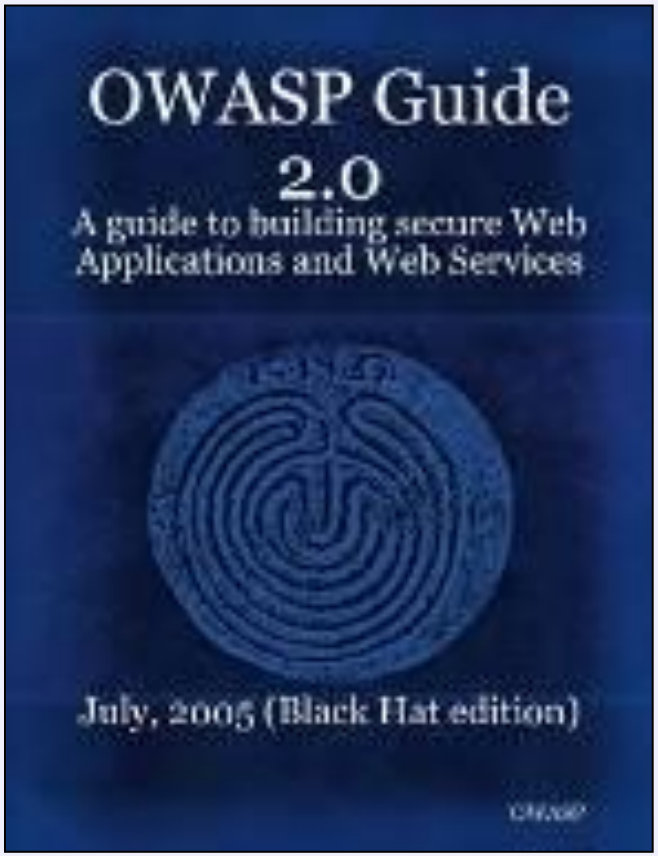
Testing Guide

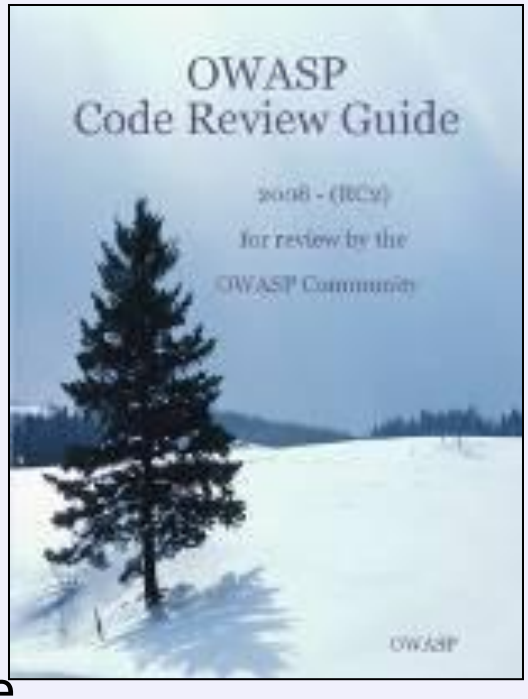
Application Security Desk Reference (ASDR)

OWASP

The Open Web Application Security Project

- 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

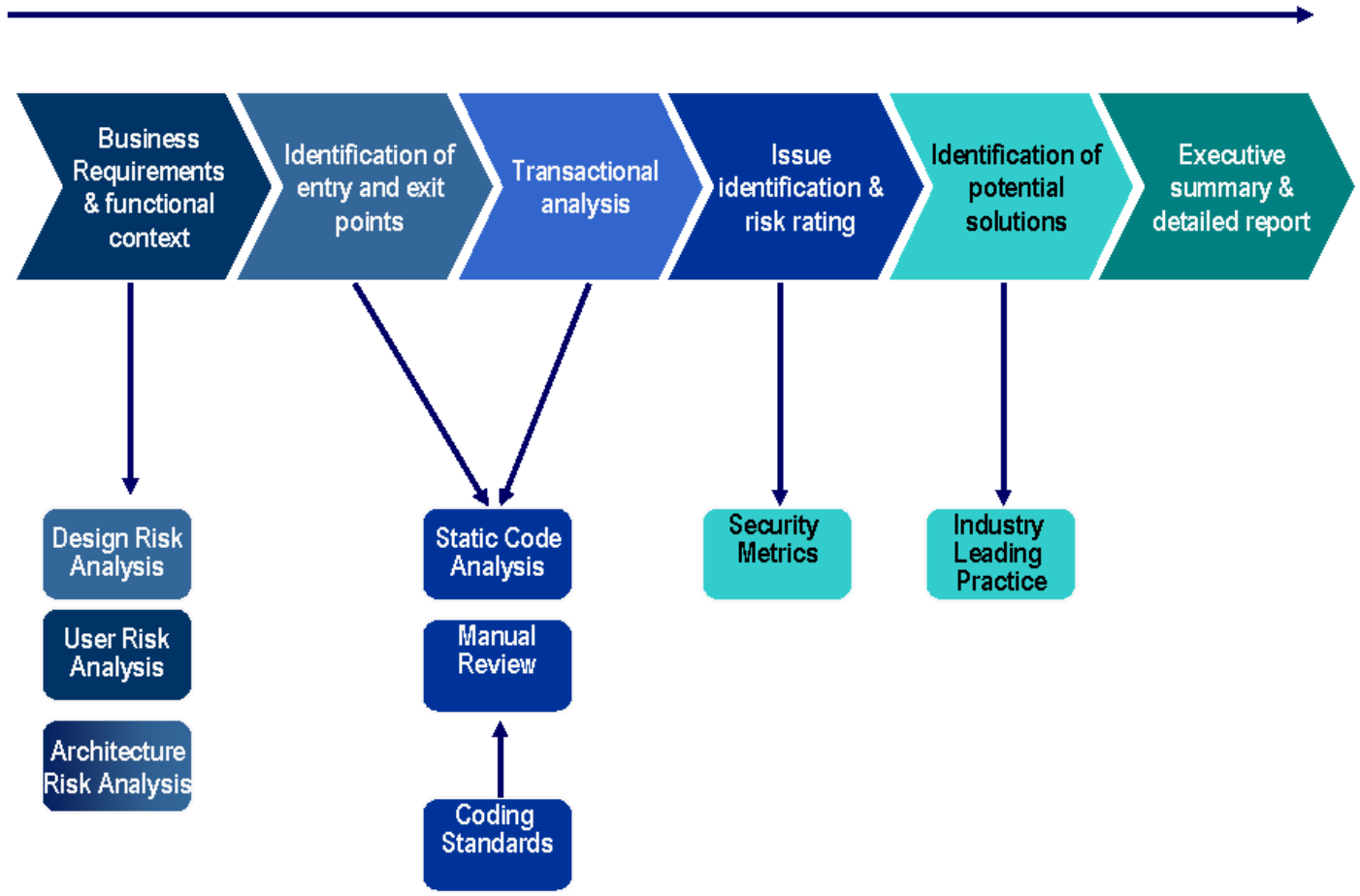




- What it is:
  - Examination of developed source code for quality.
  - Security = Quality
  - Robust & Stable code
  - More Expensive
  - Can be more Accurate
  - Requires unique skill set to do properly

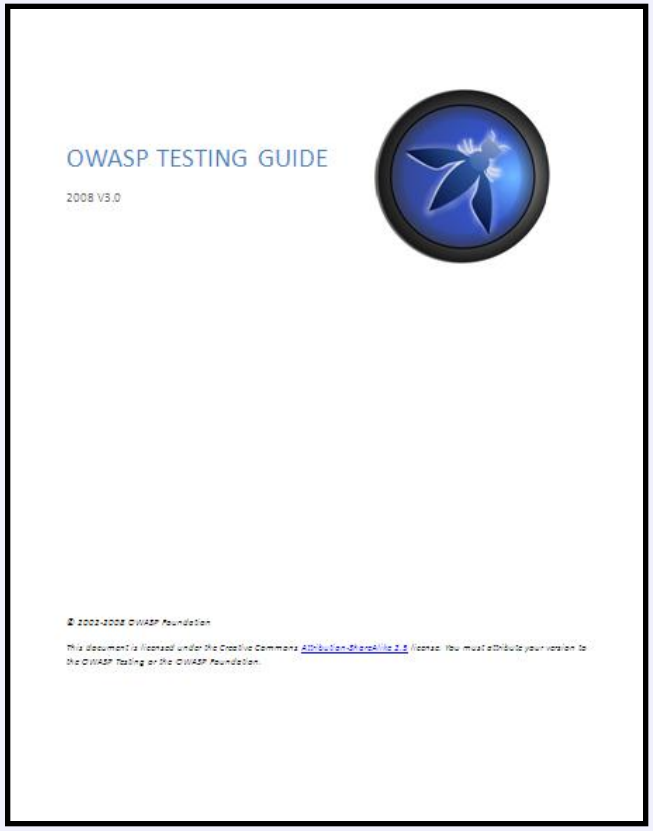
- What it isn't:
  - Silver Bullet
  - Replacement for other security controls
  - Replacement for poor application development
  - Easy
  - Cheap (Not Manual anyways)

Secure Code review process – Operational process



OWASP
The Open Web Application Security Project

- 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



OWASP

The Open Web Application Security Project

- Vulnerability Scanners
- Static Analysis Tools
- Fuzzing

Automated Security Verification

- Penetration Testing Tools
- Code Review Tools

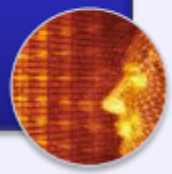
Manual Security Verification

- 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

Part of the ‘Big 4 +1’

OWASP
The Open Web Application Security Project

ASVS

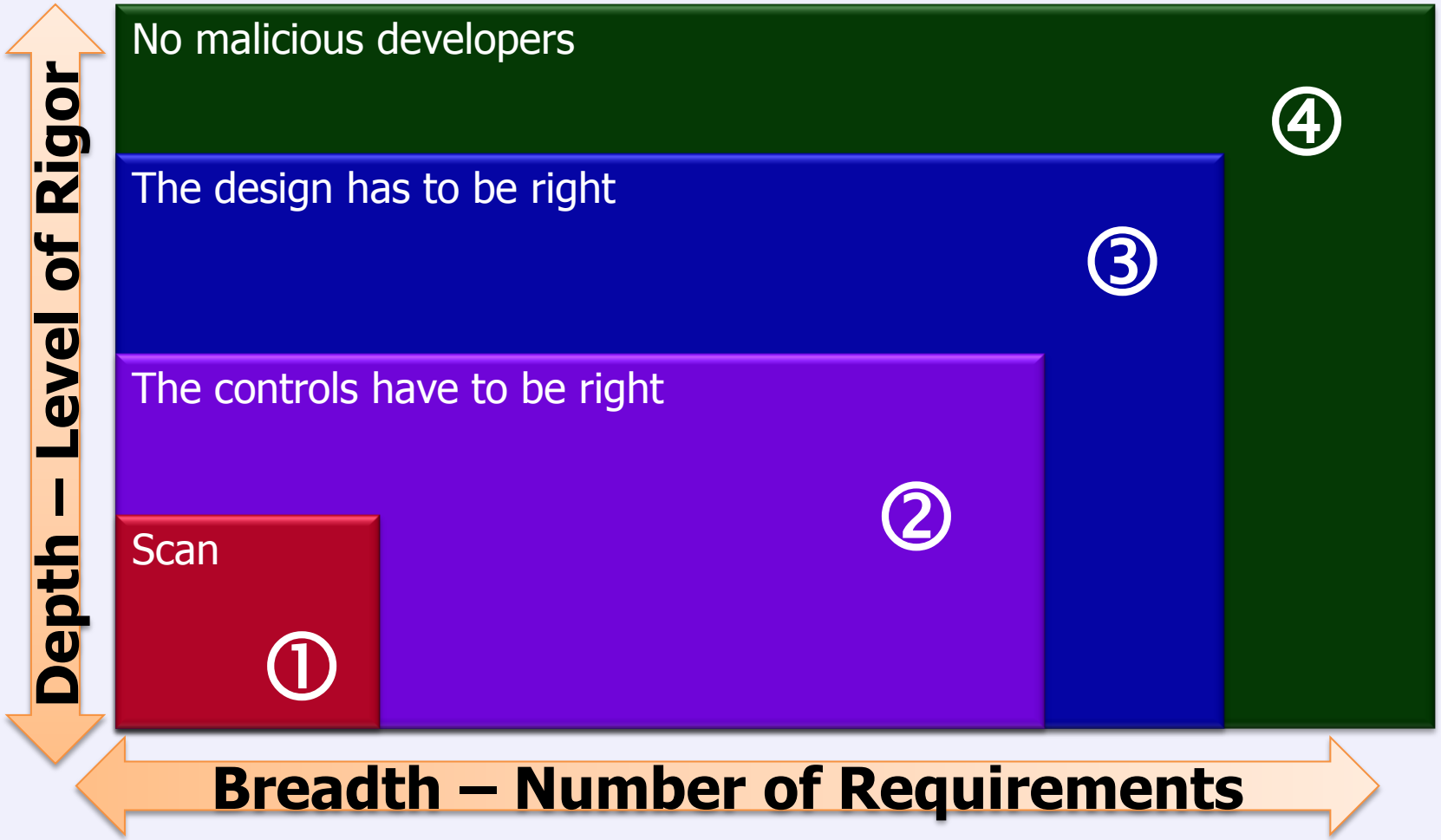
Building Guide

Code Review Guide

Testing Guide

Application Security Desk Reference (ASDR)

Coverage


No malicious developers
④
The design has to be right
③
The controls have to be right
②
Scan
①
Depth – Level of Rigor
Breadth – Number of Requirements

OWASP
The Open Web Application Security Project

Find Vulnerabilities
Using the Running Application

Find Vulnerabilities
Using the Source Code

Manual Application
Penetration Testing

Automated Application
Vulnerability Scanning

Manual Security
Code Review

Automated Static
Code Analysis

Part of the ‘Big 4 +2’

OWASP
The Open Web Application Security Project

ASVS

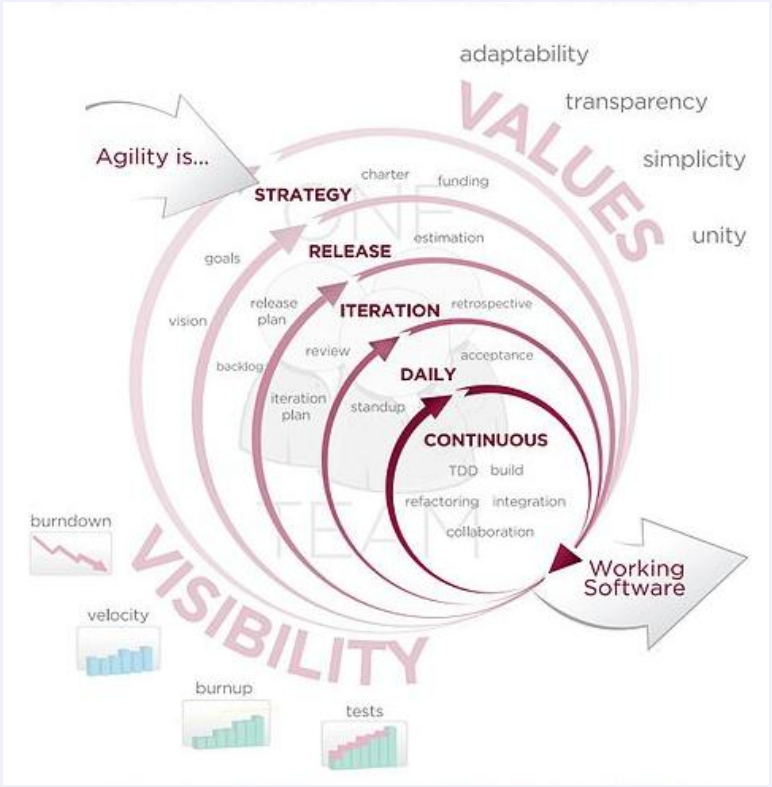
SAMM

Building Guide

Code Review Guide

Testing Guide

Application Security Desk Reference (ASDR)



Governance

Construction

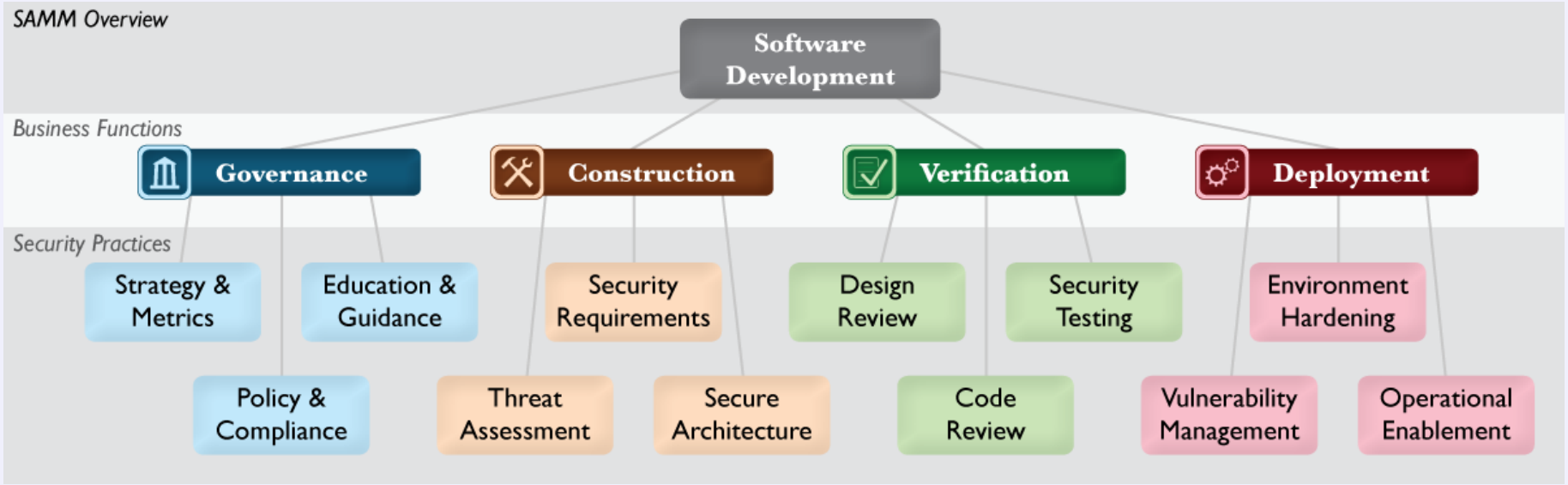
Verification

Deployment

- 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

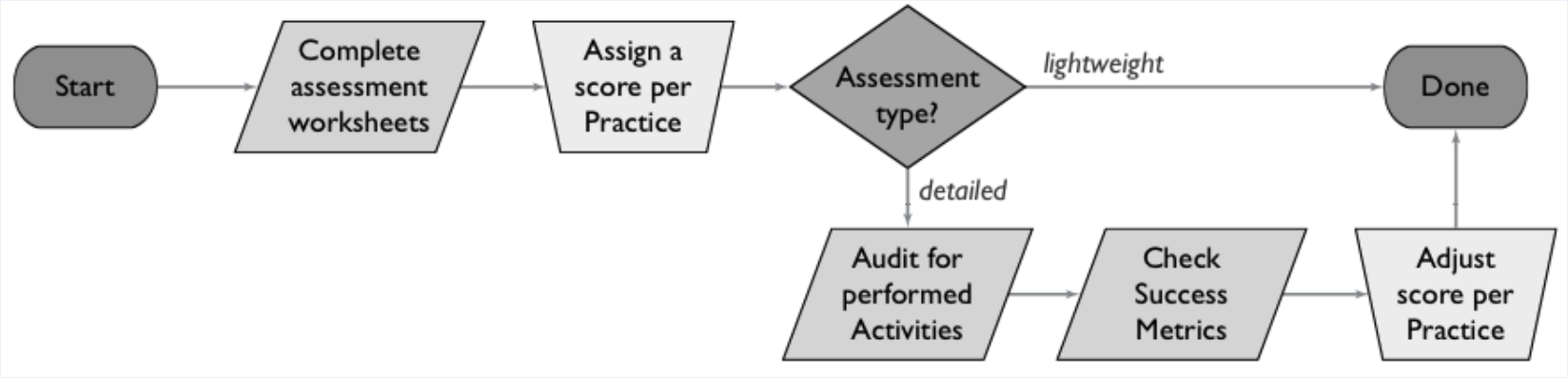
- From each of the Business Functions, 3 Security Practices are defined
- The Security Practices cover all areas relevant to software security assurance



Assessment process

OWASP
The Open Web Application Security Project

- Supports both lightweight and detailed assessments
- Organizations may fall in between levels (+)



Threat Modeling – The Basics

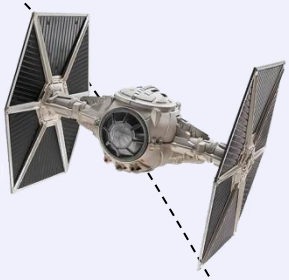
OWASP
The Open Web Application Security Project

Threat:

Causes harm

Risk:

Chance of harm occurring

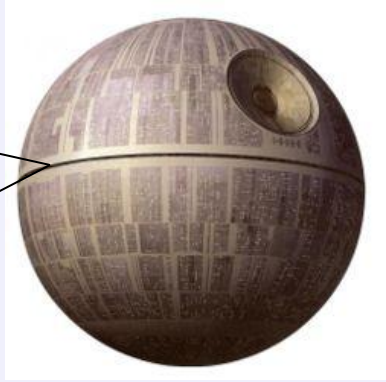


Countermeasure:

Reduces risk

Asset:

Valuable resource



Vulnerability:

Exploitable weakness

Why start again?

OWASP
The Open Web Application Security Project

Threat

Risk is low

Asset

Countermeasure

Dependency's Threat

Dependency's Countermeasure

Dependency

OWASP
The Open Web Application Security Project

- **Applications are about information**
  - Confidentiality, Integrity & Availability
- **Explicit security requirements**
  - Make security verifiable!
- **Security in depth**
  - Security considered through the whole application
  - Propagation of credentials
- **Security by default**
  - Who may do what?

>> More code == more bugs! <<

OWASP
The Open Web Application Security Project

Functional Designers & Architects:

- It is not only about what functionality the application has to supply, it also what it may not!

Engineers:

- Quality is not just '*does it work*' .

Testers:

- Security weaknesses are not different from other, functional, bugs. They can be traced down the same way.

Managers:

- Reserve project time for security
- Understand security as manditory value of an application

Security Analyst:

Involve a security Analyst at the beginning of the design phase.

That's it…

..thank you!