



TASSQ

**Commonalities Between Privacy & Security
and Testing of Sensitive Systems**

**Keith Jonah
Trusted By Design Inc.**

TASSQ – February 16, 2011

Quote of the Day

“I cannot imagine any condition which could cause this ship to flounder. I cannot conceive of any vital disaster happening to this vessel.”

E.J. Smith, Captain of the Titanic, 1912

It's All about Trust

Whether we are talking “Security” or “Testing”, building trusted systems requires --

- Well defined system-level requirements and specifications
- Well designed component products
- Sound systems engineering practices
- Competent systems developers and assessors
- Appropriate metrics for product/system testing, evaluation, and certification
- Comprehensive system planning and life cycle management

Introduction

The disciplines of Privacy & Security (Security) are more closely related to Testing and Quality Assurance (Testing) than most people might think. We will explore commonalities in:

- Software Assurance & Security
- Trends in software development and acquisition, and application security
- Risk Management and our respective roles within it
- Security and Testing in the System Development Life Cycle, when and how much?
- The Threat and Risk Assessment (TRA) as a test planning program
- Using Control Frameworks
- Preparing for Vulnerability Assessments and Testing
- Vulnerabilities as input to test cases
- Security testing techniques
- Risk: prioritizing vulnerabilities within a business context

Software Assurance Includes Security

The term software assurance is typically relating to three software properties:

- quality (i.e., “software assurance” as the short form of “software quality assurance”);
- reliability (along with reliability’s most stringent quality—safety); and
- security (comparable to the assured security of information that is expressed by the term “information assurance”).

Overall, "software assurance" must provide a reasonable level of justifiable confidence that the software will function correctly and predictably in a manner consistent with its documented requirements; and that the function of the software cannot be compromised either through direct attack or through sabotage by maliciously implanted code.

Software Security

- Software Security could be defined as a planned and systematic set of multidisciplinary activities to ensure the conformance of both software and processes to security requirements, standards, and procedures.
- The goal is to ensure that software is free from vulnerabilities, regardless of whether they are intentionally designed into the software or accidentally inserted later in its life cycle, and that the software functions in the intended manner.

Application Security

- Application security requirements and safeguards are specified almost exclusively at the level of the system and network architecture rather than the individual application's software architecture. They are primarily implemented during the application's deployment and operation.
- Application security combines system engineering techniques, such as defense-in-depth (DiD) measures (e.g., application layer firewalls, eXtensible Markup Language (XML) security gateways, sandboxing, code signing) and secure configurations, with operational security practices, including patch management and vulnerability management.

Secure Software

- Secure software cannot be intentionally subverted or forced to fail, it remains correct and predictable in spite of intentional efforts to compromise its dependability.
- Secure software is designed, implemented, configured, and supported in ways that enable it to:
 - Continue operating correctly in the presence of most attacks by either resisting the exploitation of faults or other weaknesses in the software by the attacker, or tolerating the errors and failures that result from such exploits; and
 - Isolate, contain, and limit the damage resulting from any failures caused by attack-triggered faults that the software was unable to resist or tolerate, and recover as quickly as possible from those failures.

What is a Sensitive System?

- A "sensitive" system contains or processes information related to assets of significant concern to the organization
 - personal information (financial, purchasing history, VIP, race, location, etc.),
 - personal health information,
 - intellectual property, etc.
- “Concern” to an organization could be expressed as “risk”
- Risk is a function of Impact and Likelihood

Trends in software development and acquisition, and application security

TASSQ – February 16, 2011

Where Do Your Applications Come From?

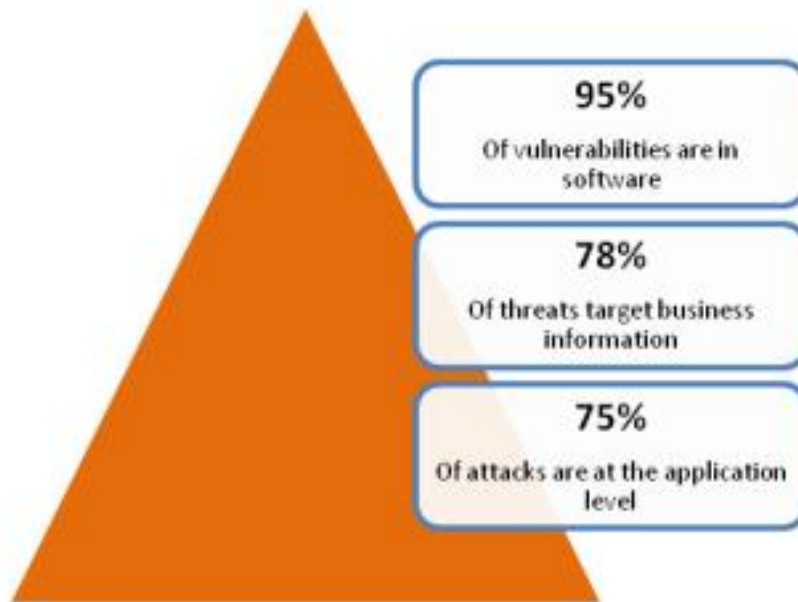
- Legacy
 - Limited or no access to original requirements, source code, testing plans, cases, results, or developers
- In-house
 - Some control and trust over your internal processes
 - Should have access to requirements, source code, etc.
- Commercial
 - Typically access only to Binary executables and configuration files
 - Pros and cons of someone else's "baby"
 - Expected to work, not usually tested as part of this "solution"
 - Generalized design, default passwords, hidden features
- Outsourced
 - Less control over their SDLC
- Open Source
 - Created by thousands, with the idea that it was "tested" by thousands is not always true
 - Usually open access to requirements, source code etc.
 - No corporate accountability

New Software Development Trends

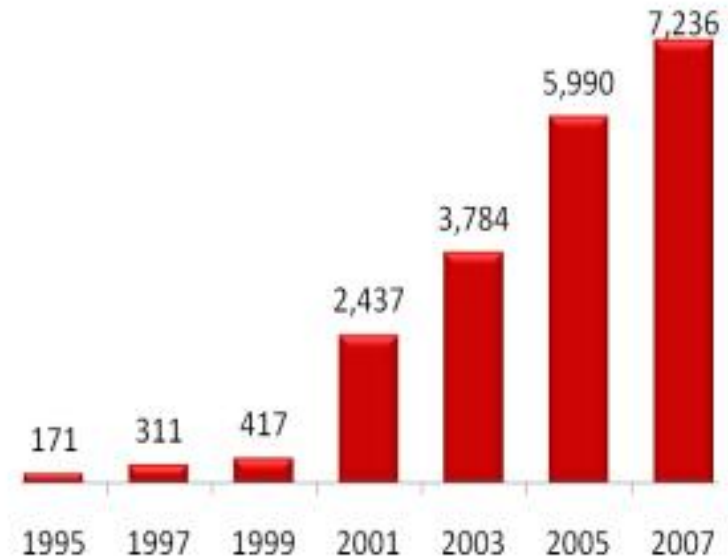
- Today's software "solutions" have significant numbers of components
- Components come from a variety of origins
- Heavier reliance on "leveraged" services and repurposing existing software outside of its original design envelop
 - Portals and Web 2.0 Mashups
 - Leveraged external web services
- Flurry of new coding tools and libraries

Software "risk" is inherited from all types of application development, procurement and management processes, big and little

Software Vulnerabilities Rising



NIST/Gartner Key Facts



CERT – Number of Software Vulnerability Disclosures per Year

Survey Time!

TASSQ – February 16, 2011

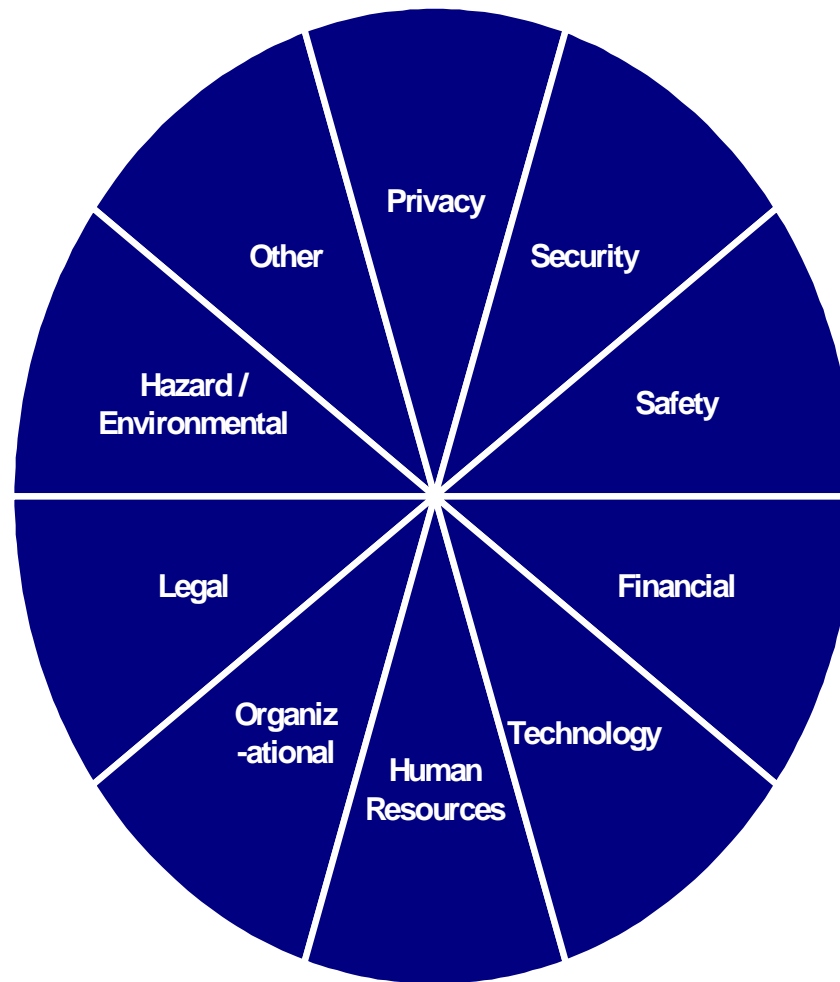
Risk Management and our respective roles within it

TASSQ – February 16, 2011

Definitions of Risk

- Objectively, risk refers to the uncertainty that surrounds future events and outcomes. It is the expression of the likelihood and impact of an event with the potential to influence the achievement of an organization's objectives.
- Some form of quantitative or qualitative analysis is required for making decisions concerning major risks to the achievement of an organization's objectives.
- Although many definitions of risk refer to the negative impact of the issue, it must be acknowledged that there are also positive opportunities arising from responsible risk-taking, and that innovation and risk co-exist frequently.

Risk Models, e.g. Risk Wheel



TASSQ – February 16, 2011

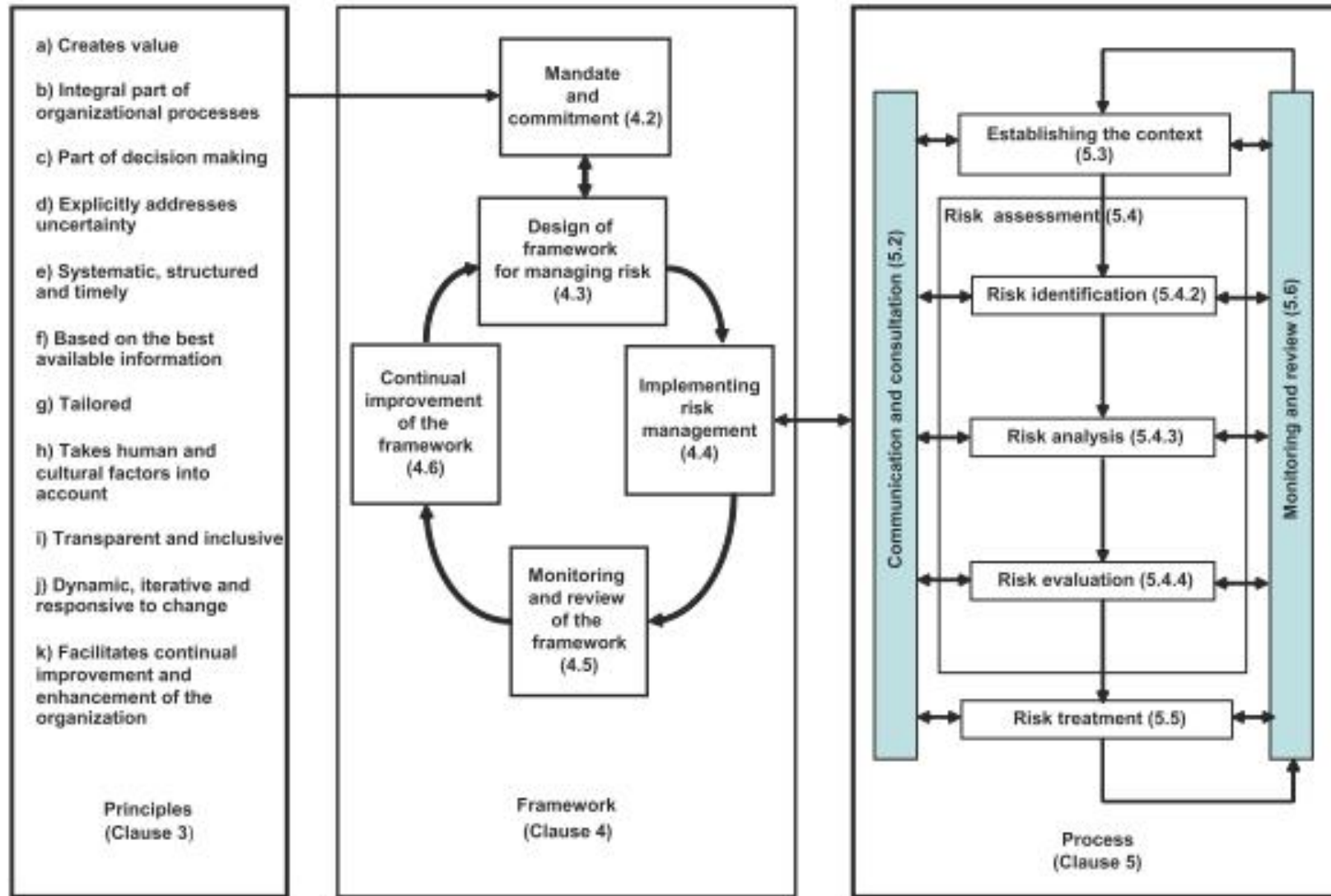
Risk and Accountability

- The things most likely to put you out of business are not the known issues of high risk, but rather the issues that remain unqualified wrt risk
- When things go bad, any semblance of Risk Management is hard to refute

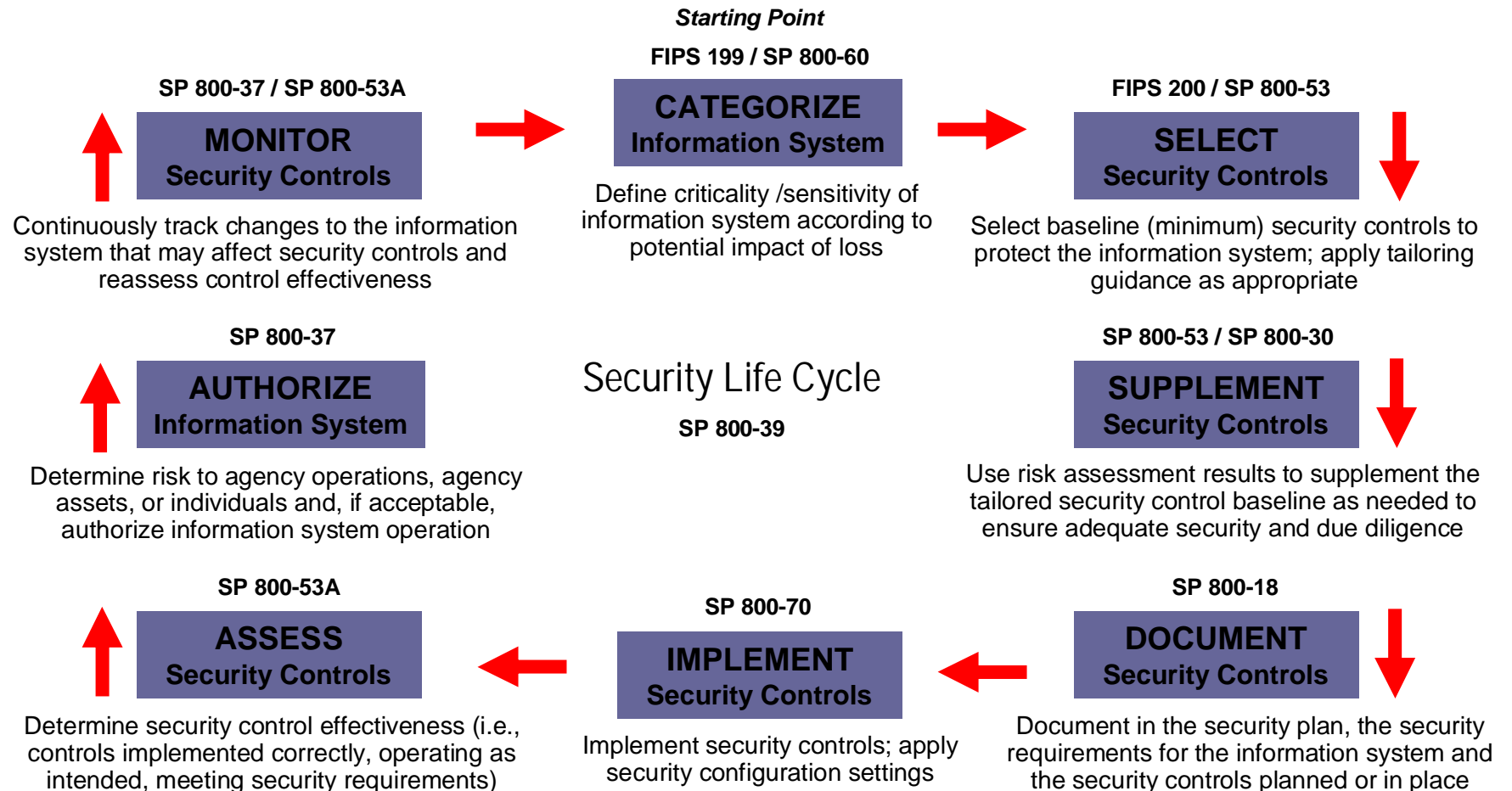
Risk-Based Decisioning

- Focusing the money for secure product development, testing and due diligence to the assets of highest risk

Risk Management Principles, Frameworks, Processes (E.g. ISO 31000:2009)



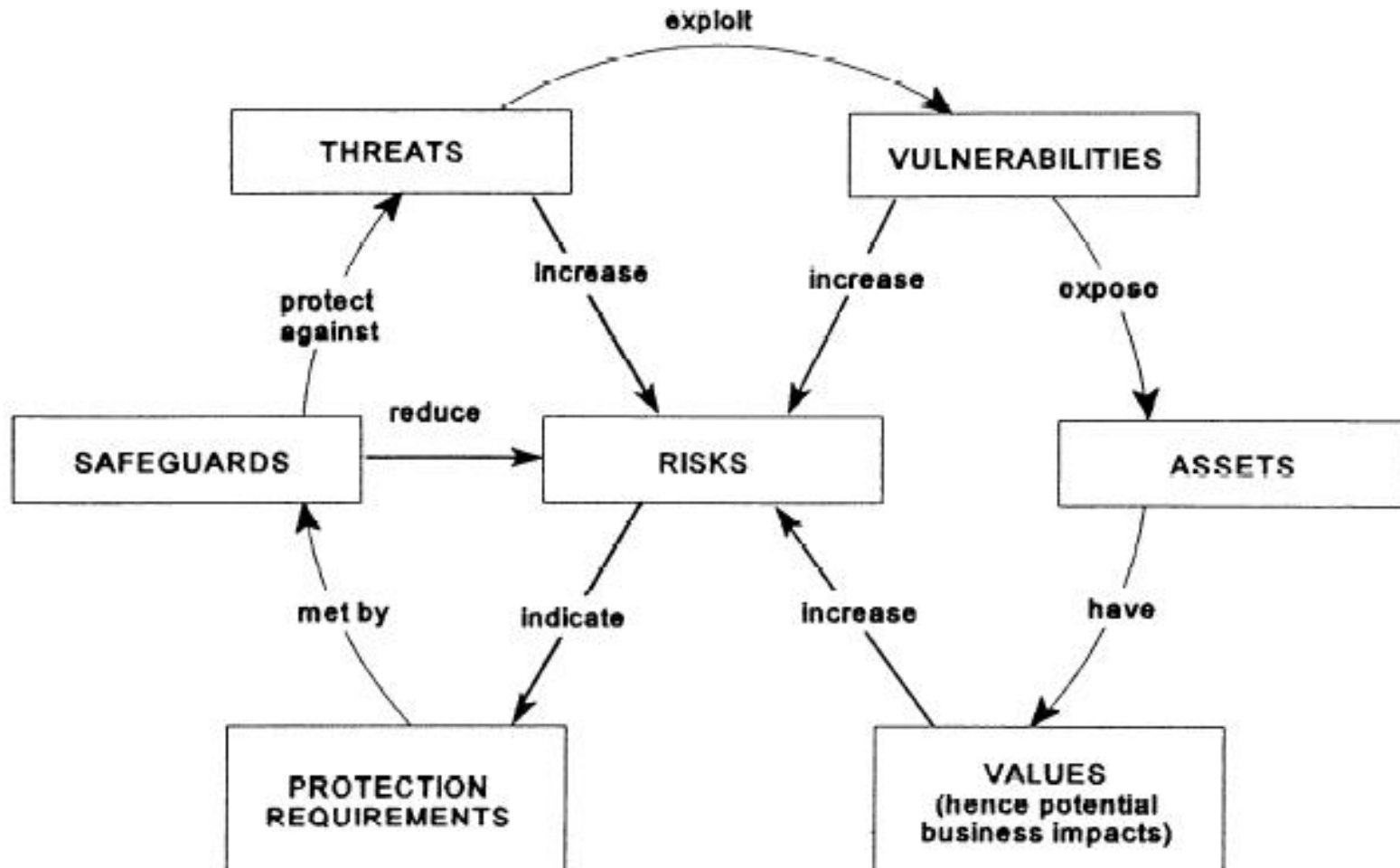
Risk Management Framework (E.g. NIST)



NIST Security Guidance

- NIST Risk framework consists of over 1200 pages of guidance
- An additional security-related mandatory 15 Federal Information Processing Standard (FIPS) Publications
- Over 100 additional security related special publications
- Over 35 Interagency Reports
- Over 65 Security Bulletins (since 2002)

Basic Risk Model (E.g. 13335)



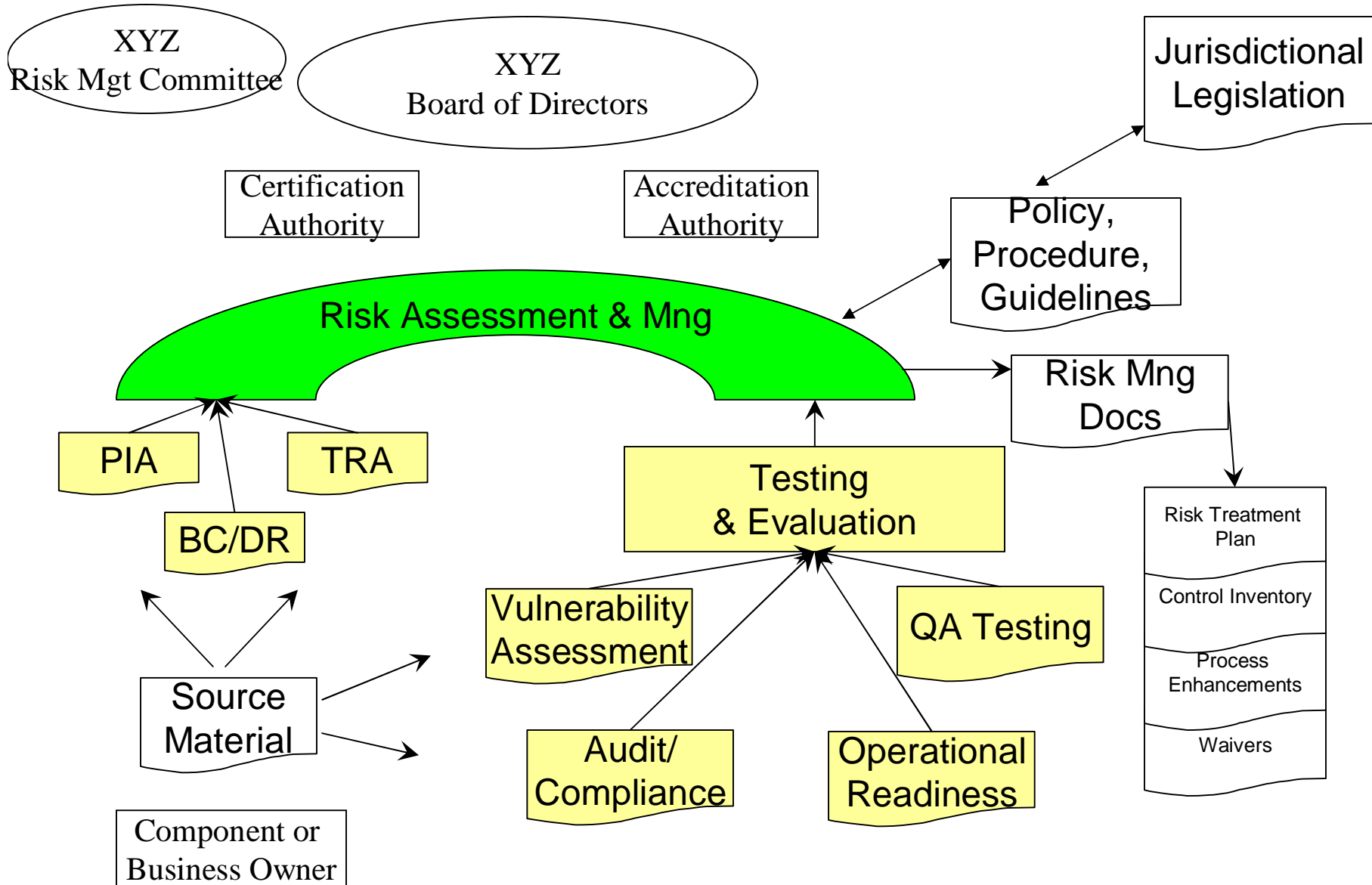
Risk Element Definitions (more detail later)

- Asset
- Safeguard
- Vulnerability (a.k.a. test issues)
- Threat
- Risk
- Target Risk
- Residual Risk
- ...

Traditional Risk Management

- Risk Management
 - Risk Assessment
 - Risk Decisioning & Acceptance
 - Risk Treatment
 - Risk Communications
 - Monitoring
 - Change Management

Risk Management Model



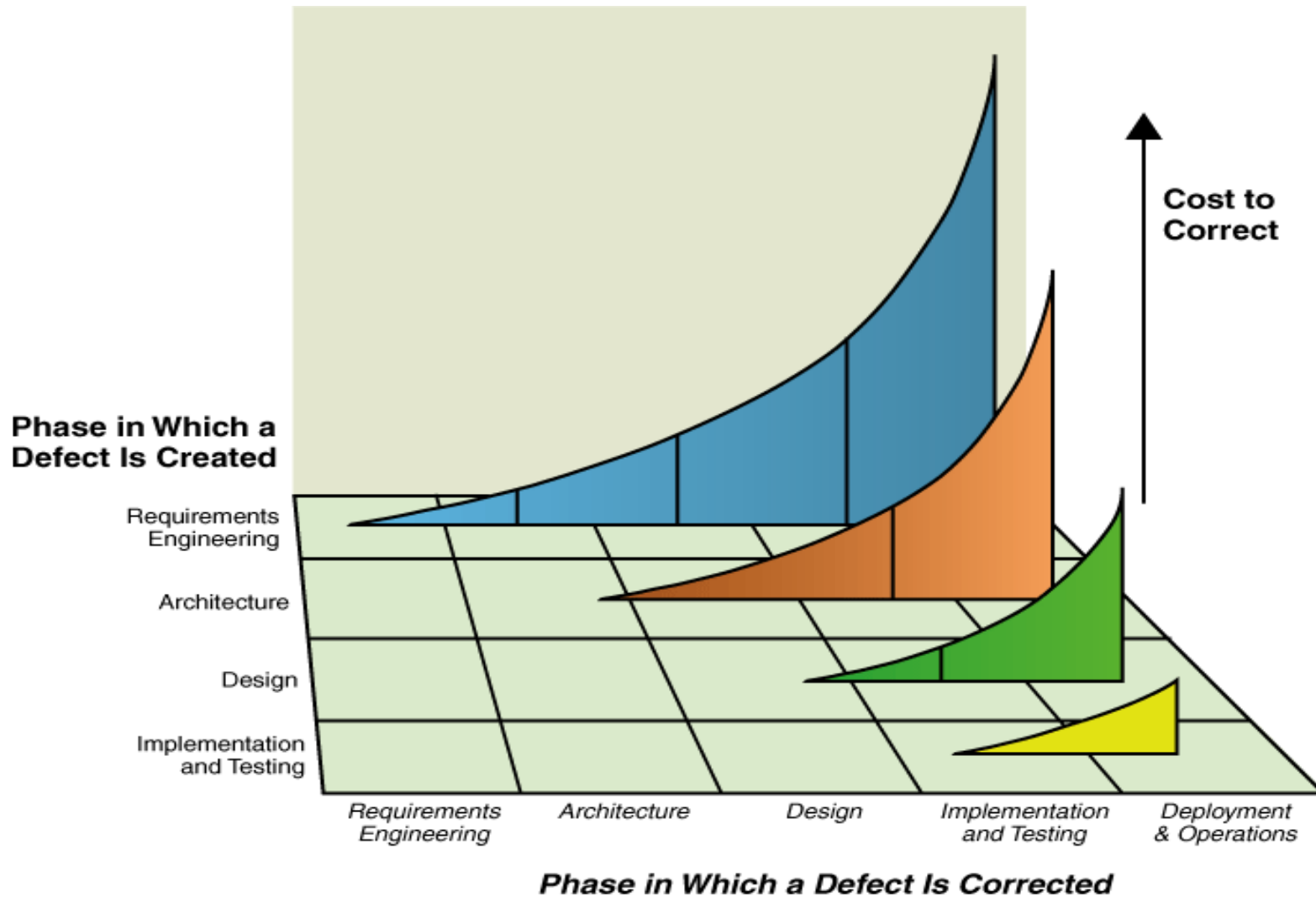
Risk Assessment

- Due diligence exercises include:
 - Architectural Reviews
 - Threat and Risk Assessments (TRA)
 - Privacy Impact Assessments (PIA)
 - Business Continuity Planning/Disaster Recovery Planning (BCP/DRP)
 - **Testing & Evaluation**
 - Detailed Vulnerability Assessment (focused on people, process or technology)
 - QA Functionality Testing
 - Operational Readiness
 - Audit and Compliance Exercises

Security and Testing in the System Development Life Cycle, when and how much?

TASSQ – February 16, 2011

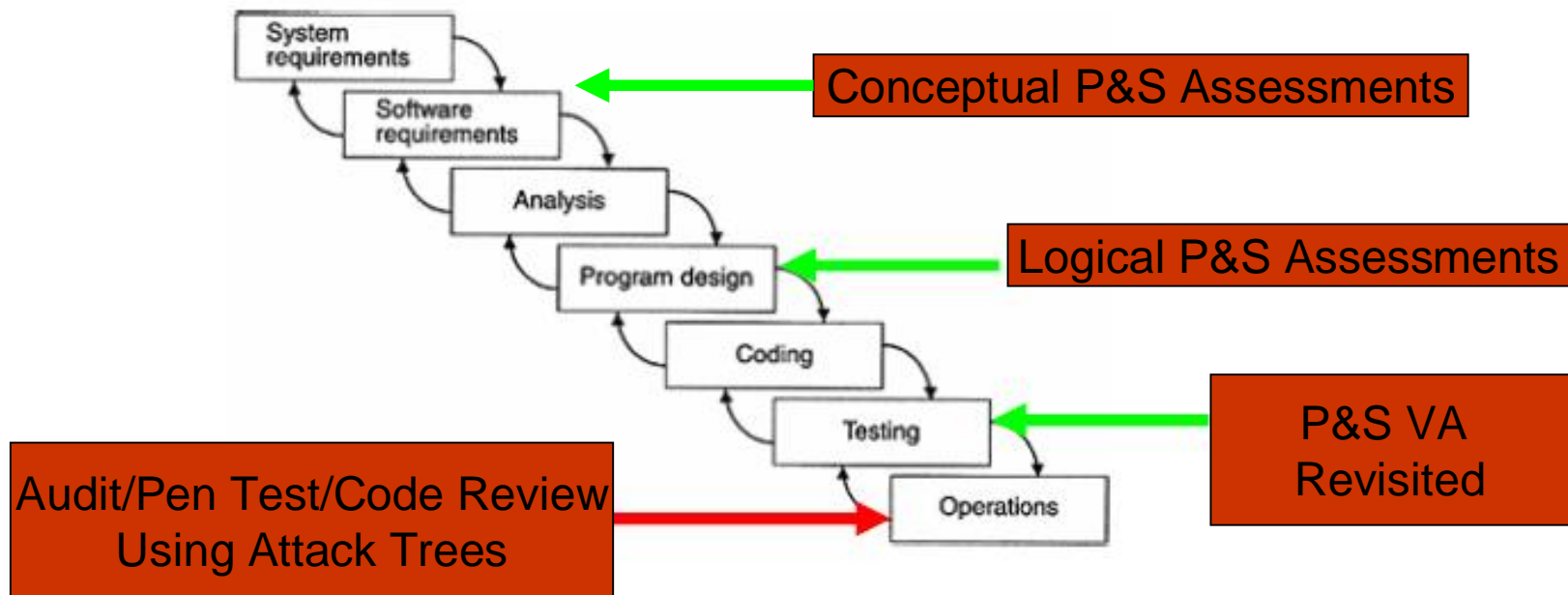
Fixing an Issue, the Sooner the Better



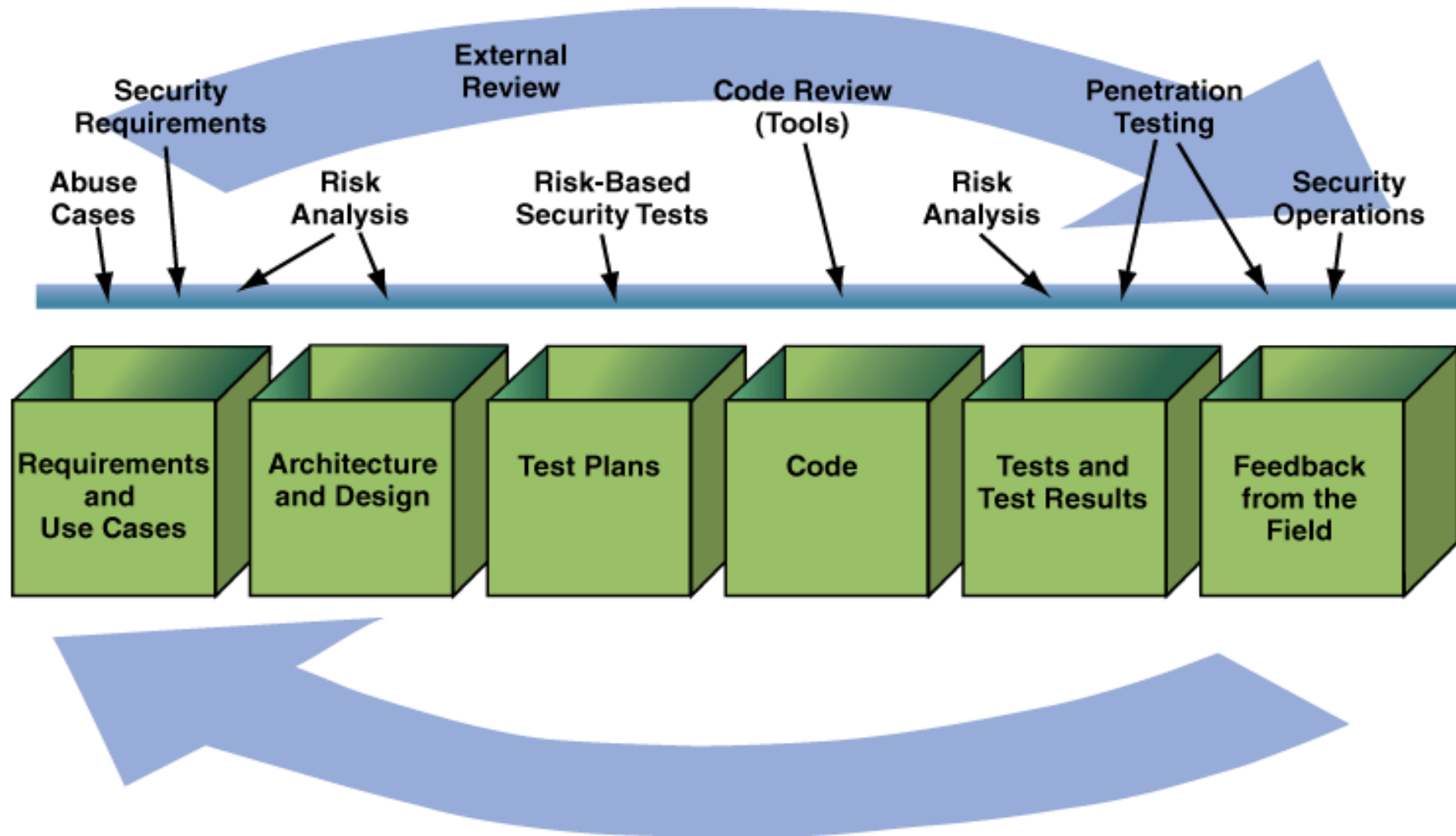
TASSQ – February 16, 2011

When to undertake Privacy & Security Assessments?

- Traditional Waterfall SDLC



Security Touch Points in SDLC



SDLC: Software Development Life Cycle

McGraw, Gary. *Software Security: Building Security In*. Boston, MA: Addison-Wesley Professional, 2006.

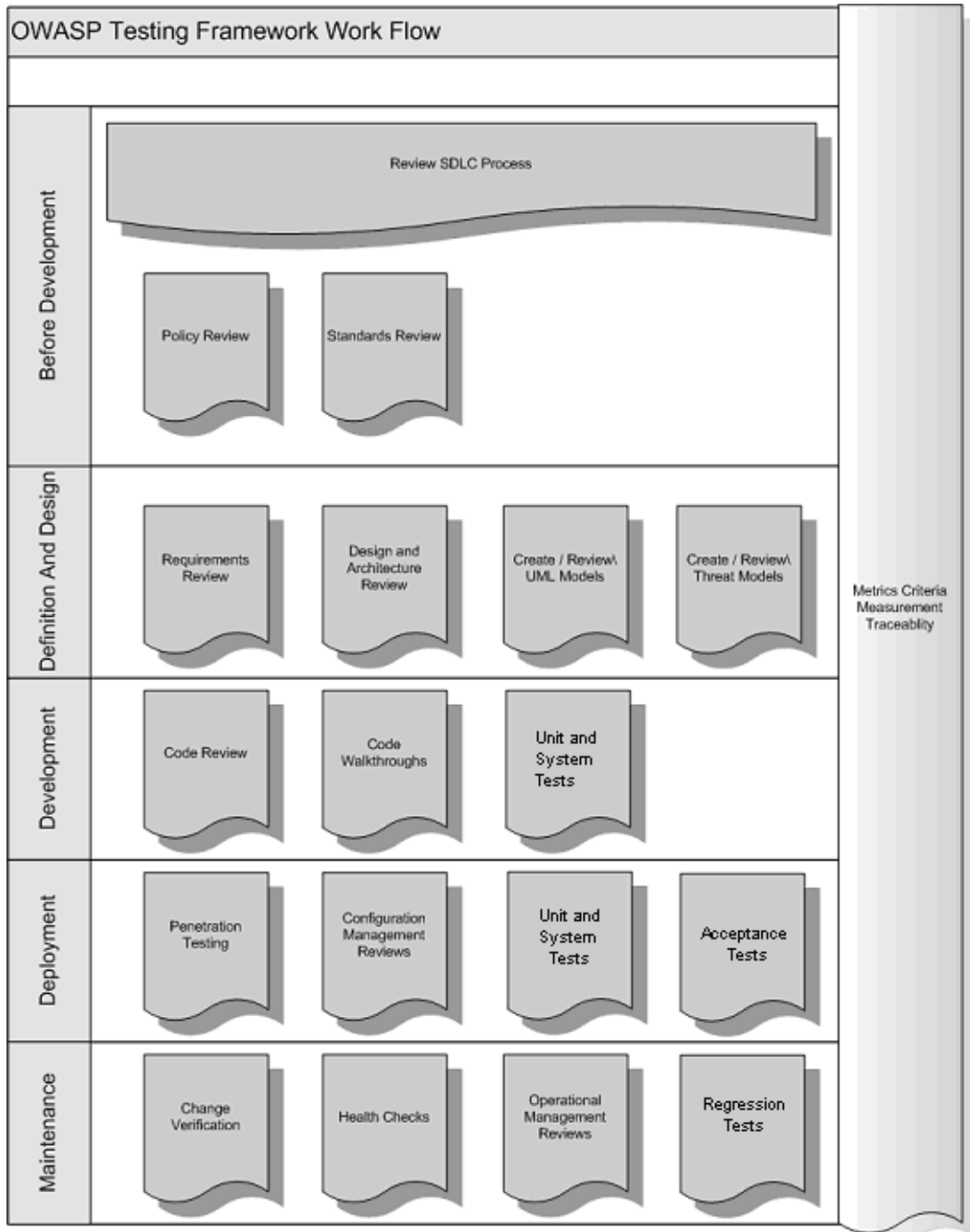
TASSQ – February 16, 2011

Open Web Application Security Project (OWASP)

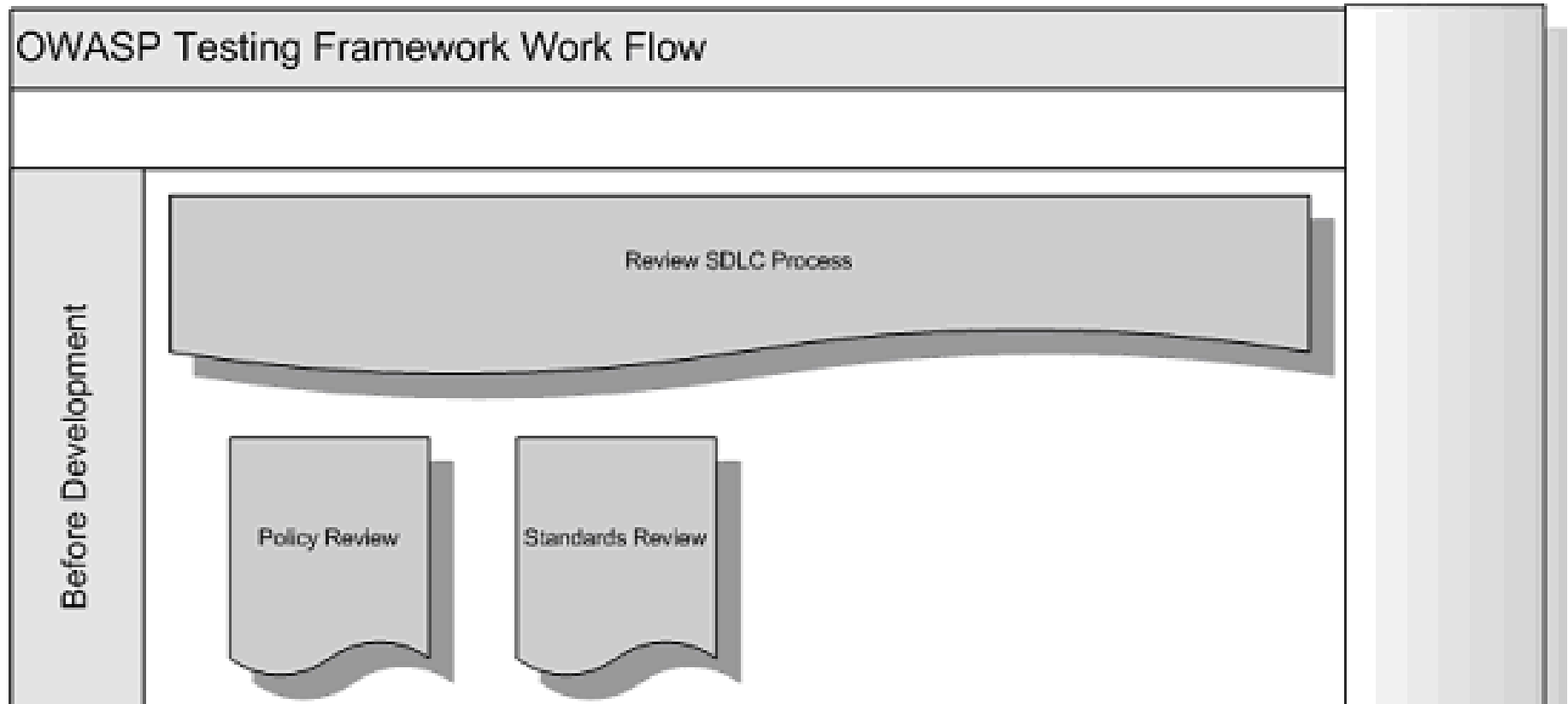
- Open Source community project staffed entirely by volunteers from across the world.
- project to develop software tools and knowledge based documentation to secure web applications and web services
- driven by discussions on the Web Application Security list at SecurityFocus.com
- All software and documentation is released under the GNU public licenses.
- Borrowed concept of “Top 10 Vulnerabilities” from SANS/FBI

OWASP Test Guide

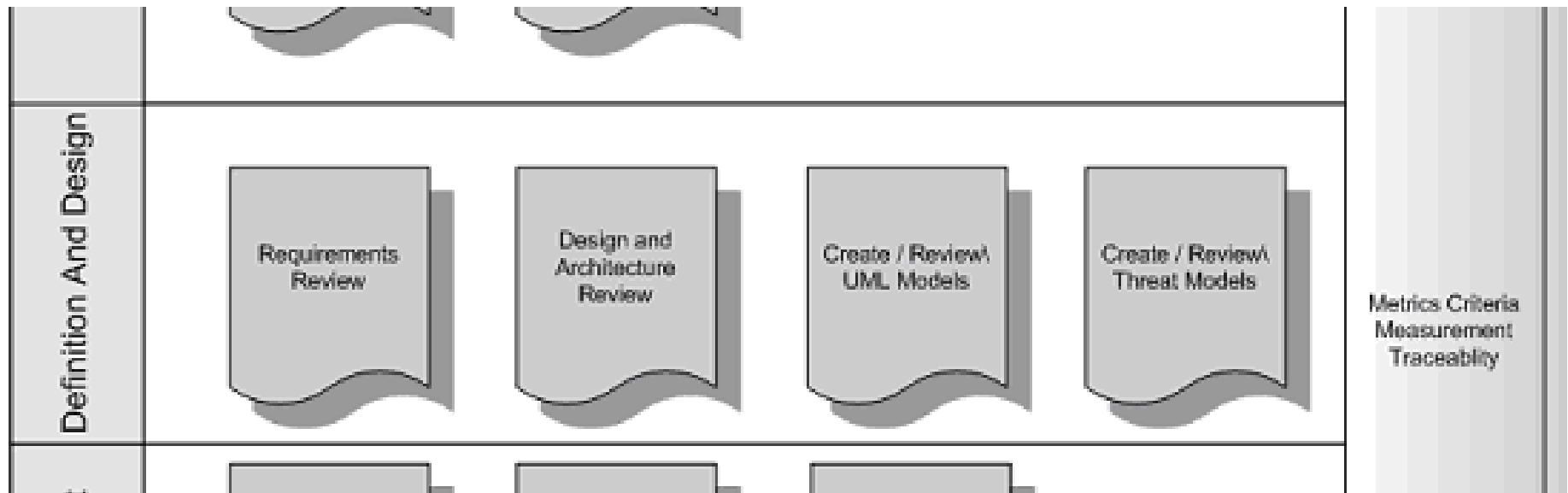
- *“Companies should inspect their overall SDLC to ensure that security is an integral part of the development process”*
- *“SDLCs should include security tests to ensure security is adequately covered and controls are effective throughout the development process.”*



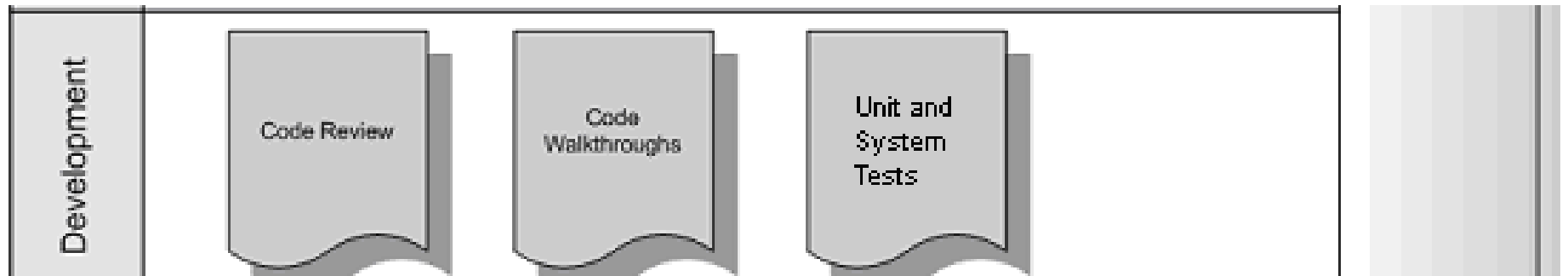
OWASP SDLC – Before Development



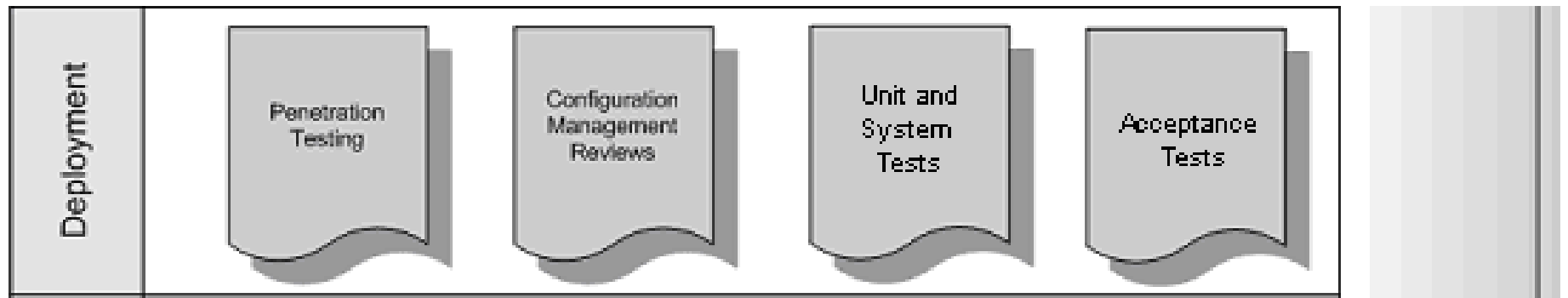
OWASP SDLC – Definition and Design



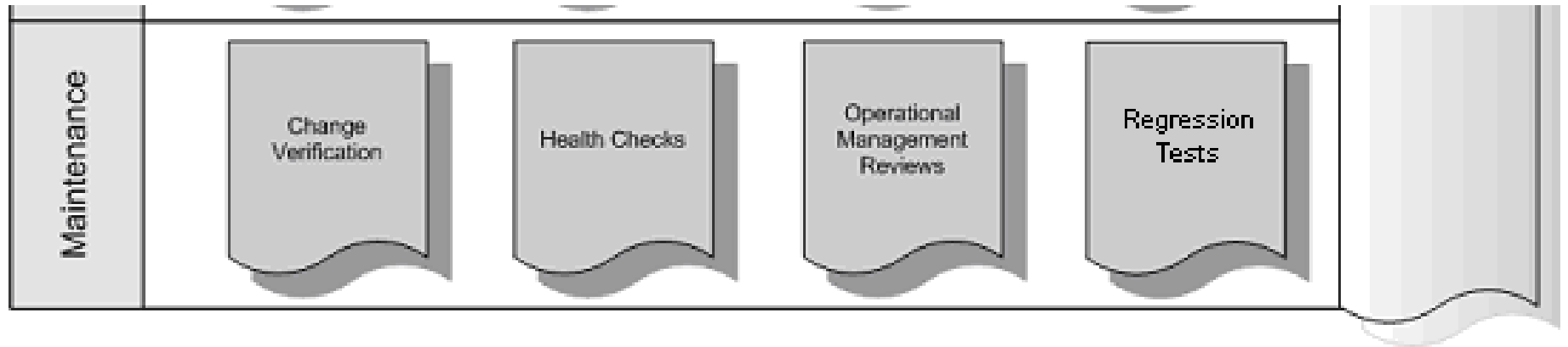
OWASP SDLC – Development



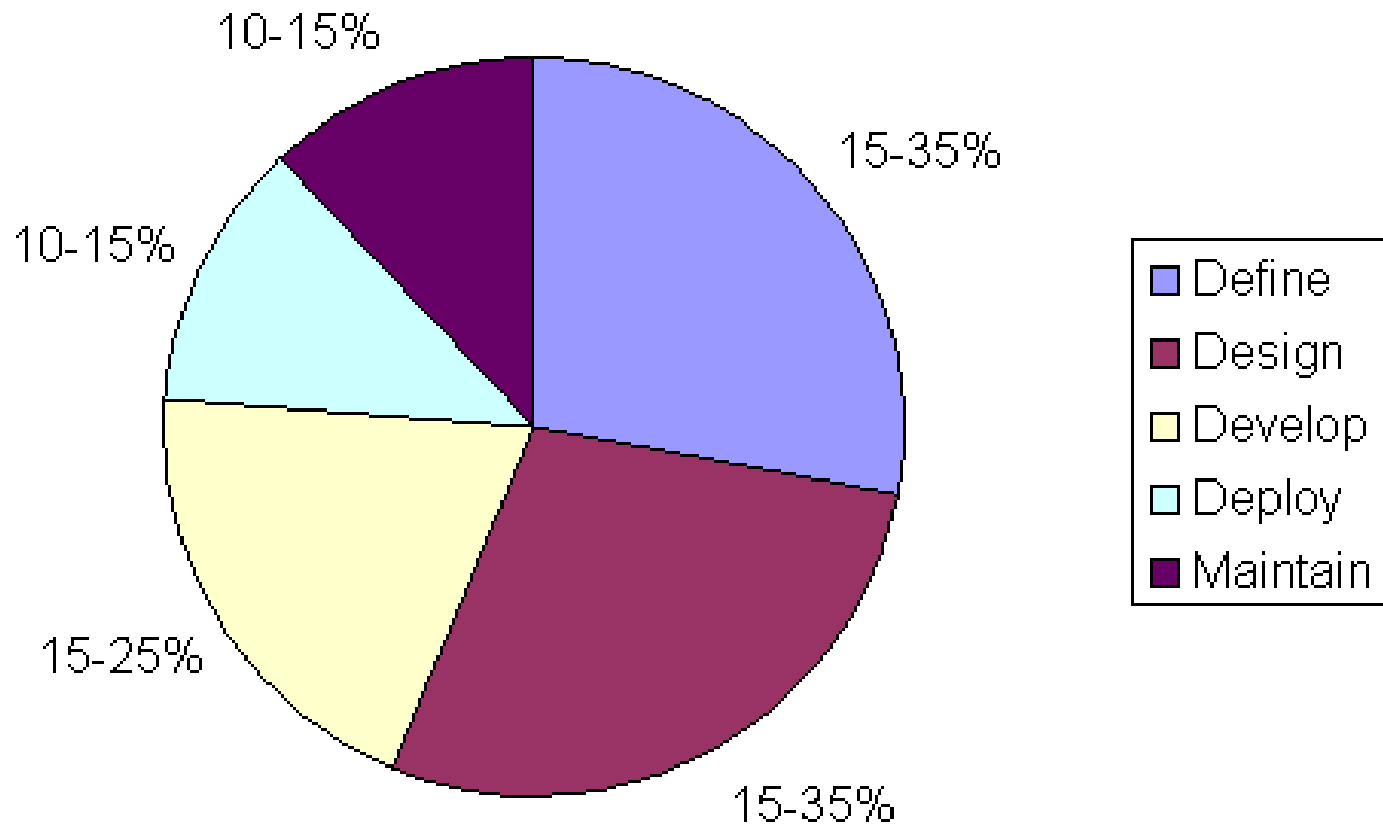
OWASP SDLC – Deployment



OWASP SDLC – Maintenance



Suggested Proportion of Security Test Effort in SDLC



The Threat and Risk Assessment (TRA) as a test planning program

TASSQ – February 16, 2011

What is a Threat and Risk Assessment (TRA)?

A TRA is a fundamental tool for Information Risk management. The objective of a TRA is to:

- Identify sensitive system assets and existing safeguards;
- Identify how these assets can be compromised by threat agents through residual vulnerabilities;
- Assess the level of risk to an asset based upon the likelihood of occurrence and impact to the organization; and
- Recommend how to proceed in the life cycle to reduce risk to an acceptable level.

What is a Threat and Risk Assessment (TRA)?

A TRA is a fundamental tool for Information Risk management. The objective of a TRA is to:

- Identify threats; agents
- Identify threats; agents
- Assess likelihood of occurrence and impact to the organization; and
- Recommend how to proceed in the life cycle to reduce risk to an acceptable level.

Sounds like Test Planning to me!

Are there different types of TRAs?

- A Conceptual TRA can be conducted early in the system development lifecycle based upon high-level specifications.
- Results of a TRA become requirements of the architecture and design which are just being developed. Old saying, “spend a dollar in analysis or six trying to fix the product later”
- A Logical TRA is conducted near the end of the build phase, just prior to production based upon detailed design and actual configurations.
- Suggest conducting both types, but requires inclusion of security into the system development life cycle!

Can one TRA cover a large system?

- An iterative approach is sometimes needed due to complexity or readiness/availability of components.
- At any one time the XYZ program will have many components that are in the conceptual, design and/or operational phases of development.
- Each iteration will cover people, process and technology related to the targeted assets.
- It is imperative that the scope and context of each iteration be carefully defined and tracked, in particular, it must detail which XYZ assets are to be evaluated.

Inheritance of Risk from 3rd Parties

- An organization's solution inherits the risks (and the underlying vulnerabilities) of all components regardless of whether they are owned and/or controlled by the organization (**remember where your applications came from**)
- Any risk assessment must take into account any previous due diligence undertaken by a component's owner or intermediary (date, scope etc. must be weighed carefully)
- This would include:
 - Application Service Provider (ASP) doing hosting or other services;
 - Third party developers
 - Third party maintainers

Assets

- All IT or business systems are composed of large numbers of tangible and intangible assets including reputation, personnel, data, facilities etc.
- The TRA takes into account all assets (tangible and intangible) however is typically focused on the following critical assets:
 - Information;
 - Services;
 - Hardware/Software; and
 - Environmental.

Assets - Sensitivities

- Rate the sensitivity in terms of:
 - confidentiality
 - integrity
 - availability
 - replacement value
- Valuing the assets allows the analyst to decide which IT areas are the highest priority, and consequently where security efforts should be focused
- Consider the loss of value (prestige, trust or business opportunity) that could result from the violation of confidentiality, integrity or availability

Using Control Frameworks

TASSQ – February 16, 2011

Control Frameworks

- Control Framework. A comprehensive set of interrelated control objectives or practices, with a particular purpose. E.g. OWASP, ISO 17799, CICA GAPP. Can be used to organize privacy & security safeguards, vulnerabilities and recommendations.
- Control Objectives. A “control” has the objective of ensuring the integrity, confidentiality, and availability of information or services. It must be understandable, implementable and measurable
- Control Practices or Activities. One or many mechanisms (a.k.a. safeguards) which together could fulfill (partially or fully) the objective
- Control Tests. A practical test of the safeguards to prove fulfillment of the objective

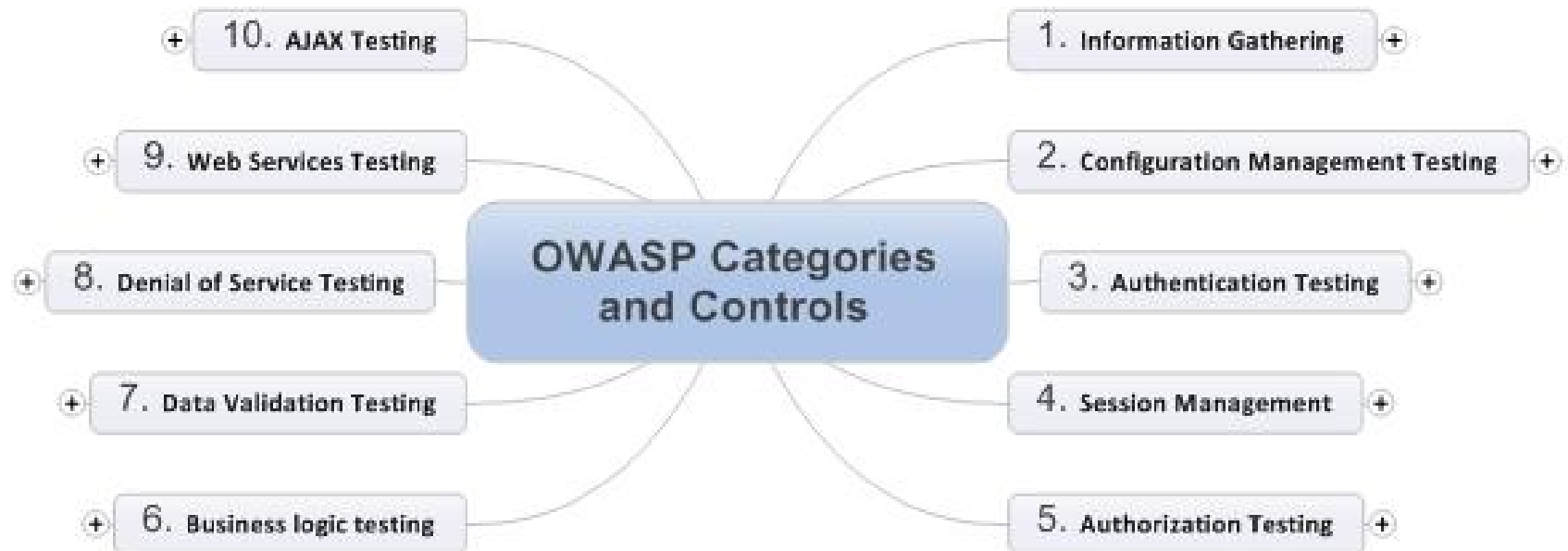
Value of Using a Control Framework

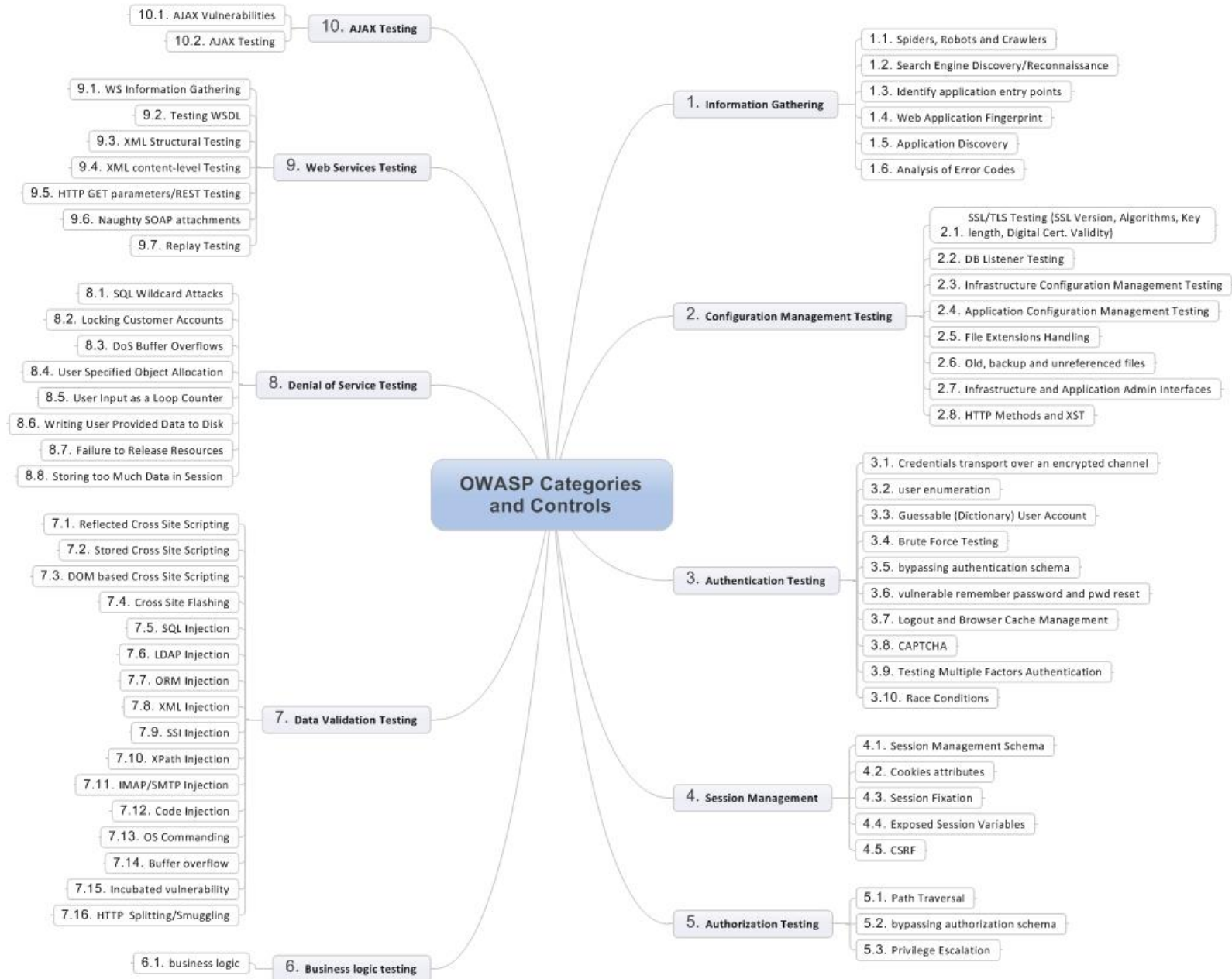
- Presenting the safeguards, vulnerabilities or risks statements in a framework allows an organization to review these three types of statements against the framework over time, against other similar organizations, and against other services.
- Consider adopting or refining a framework from the many existing candidate frameworks. The high level framework categories could be created based upon the best parts of existing Privacy and/or Security:
 - -“control frameworks” such as COBIT
 - -“principles” such as the CSA’s model code or the CICA’s Privacy Principles
 - -“services” such as the CHI HIAL P&S services
- The more granular, lower level framework sub-categories could be created based upon the best parts of existing Privacy and/or Security:
 - -“requirements” such as CHI’s P&S requirements
 - -“best practises” such as ISO 17799:2005, ITIL
 - -“guidelines” such as the OIPC Guide to PIAs Appendix B etc

Control Sources

- There are extensive sources of control objectives/requirements/criteria:
 - COSO
 - COBIT Control Objectives and Activities
 - ISO 27002, previously ISO 17799:2005
 - **Open Web Application Security Project (OWASP) Control Framework**
 - Canadian Institute of Chartered Accountants (CICA) Generally Accepted Privacy Principles (GAPP)
 - CICA's Trust Services (SysTrust and WebTrust)
 - Canadian Standard Association's (CSA) Privacy Principles
 - Carnegie Mellon's Software Engineering Institute's (SEI) Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE)
 - Information Technology Infrastructure Library (ITIL)
 - Information Security Forum's (ISF) Information Security Standards
 - International Information Security Foundation's (IISF) Generally Accepted System Security Principles (GASSP)
 - Information Systems Security Association's (ISSA) Generally Accepted Information Security Principles (GAISP)
 - NIST
 - SANS
 - ...

Example: OWASP Control Framework





SANS 20 Critical Controls

- To counter the SANS Top 20 Vulnerabilities (**we will see in a moment**), developed 20 Critical Controls. Linked to NIST Special Publication 800-53: Recommended Security Controls for Federal Information Systems
 - http://www.sans.org/critical-security-controls/press_release.pdf
 - http://csis.org/files/publication/Twenty_Critical_Controls_for_Effective_Cyber_Defense_CAG.pdf
- Embedded into Federal Information Security Management Act (FISM)
 - Leverage cyber offense to inform cyber defense – focus on high payoff areas
 - Ensure that security investments are focused to counter highest threats — pick a subset
 - Maximize use of automation to enforce security controls — negate human errors
 - Use consensus process to collect best ideas

SANS 20 Critical Controls cont'd

Critical Controls Subject to Automated Collection, Measurement, and Validation

- 1. Inventory of authorized and unauthorized hardware.**
- 2. Inventory of authorized and unauthorized software.**
- 3. Secure Configurations for Hardware and Software For Which Such Configurations Are Available.**
- 4. Secure Configurations of Network Devices Such as Firewalls And Routers.**
- 5. Boundary Defense**
- 6. Maintenance and Analysis of Complete Security Audit Logs**
- 7. Application Software Security**
- 8. Controlled Use of Administrative Privileges**
- 9. Controlled Access Based On Need to Know**
- 10. Continuous Vulnerability Testing and Remediation**
- 11. Dormant Account Monitoring and Control**
- 12. Anti-Malware Defenses**
- 13. Limitation and Control of Ports, Protocols and Services**
- 14. Wireless Device Control**
- 15. Data Leakage Protection**

SANS 20 Critical Controls cont'd

Critical Controls (not directly supported by automated measurement and validation)

- 16. Secure Network Engineering**
- 17. Red Team Exercises**
- 18. Incident Response Capability**
- 19. Disaster Recovery Capability**
- 20. Security Skills Assessment and Training To Fill Gaps**

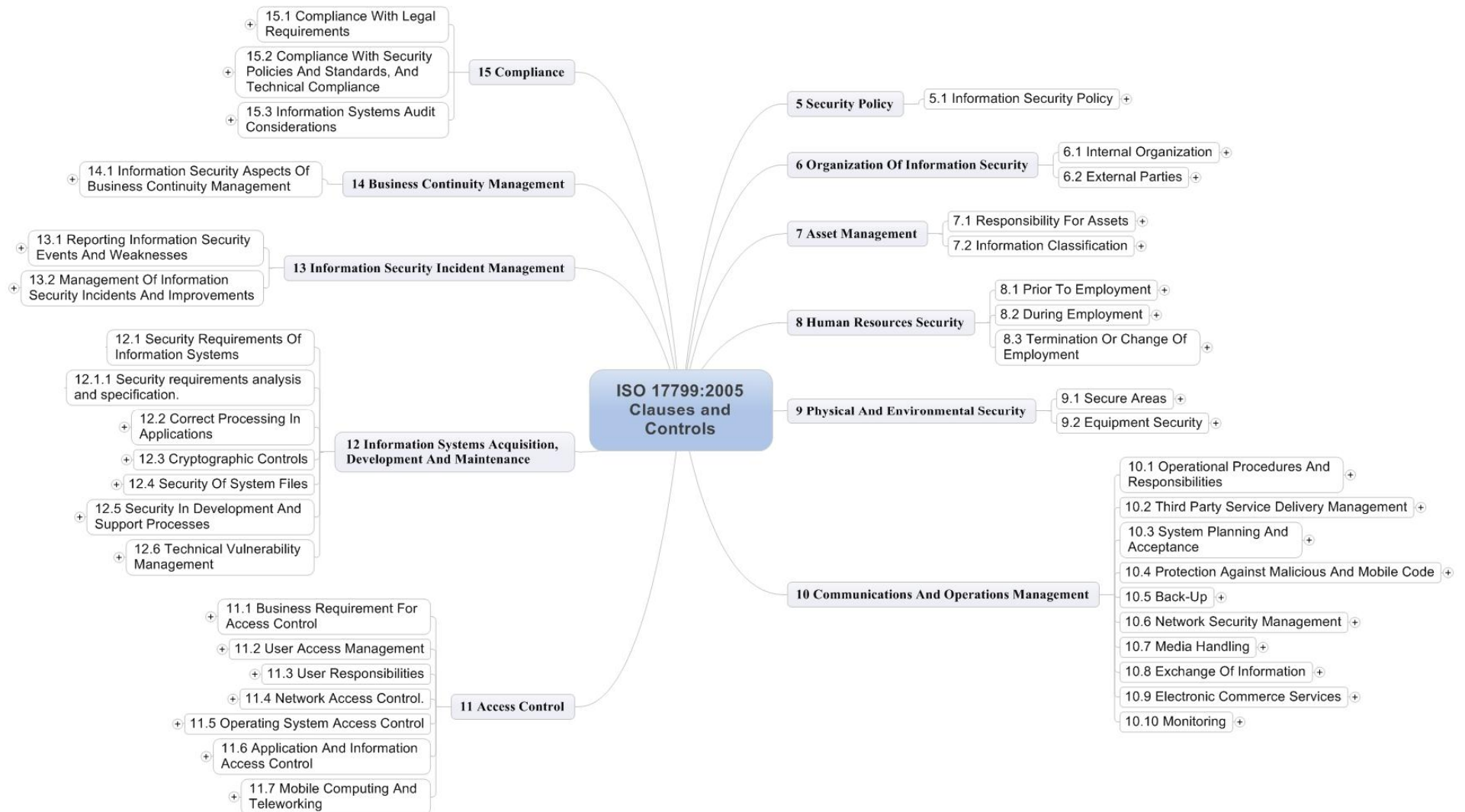
SANS 20 Critical Controls cont'd

Sample Critical Control #1

“Inventory of authorized and unauthorized hardware”

- **Control:** Accurate, up to date inventory controlled by automated monitoring and configuration management
- **Attacker Exploit:** Scan for new, unprotected systems
- **Automated Support:** Employ products available for asset inventories, inventory changes, network scanning against known configurations
- **Evaluation (Test):** Connect fully patched and hardened machine to test response from automated tools

Example: ISO 27002, Formerly 17799:2005



Preparing for Vulnerability Assessments and Testing

TASSQ – February 16, 2011

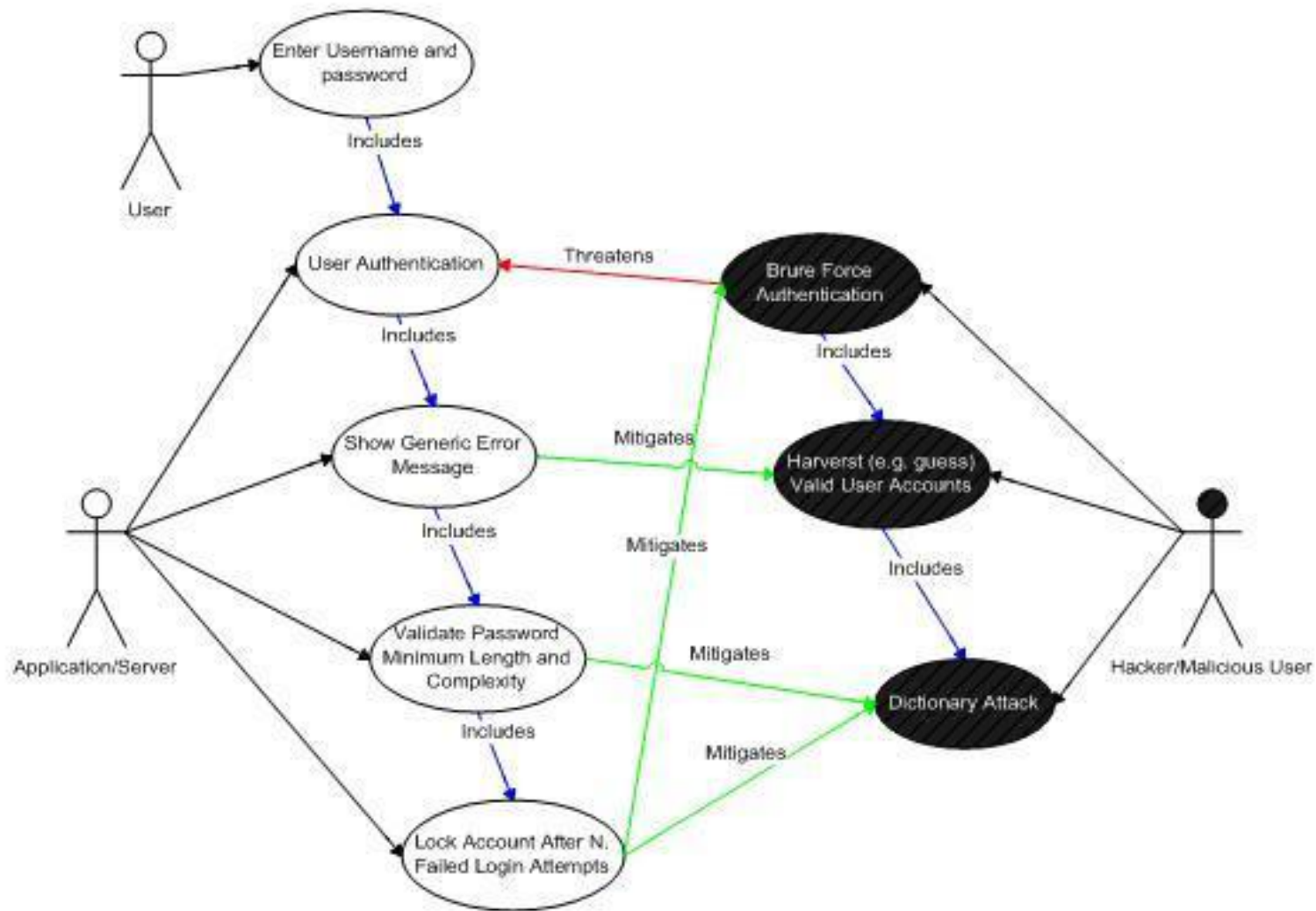
What is a Vulnerability Assessment?

- Vulnerability Assessment - the process of identifying the vulnerabilities who's exploitation could cause harm to assets, and rating their severity and exposure
- Can be very focused on just people, process, technology, or environment, a subset, or all of the above
- A VA can be a stand alone exercise however it is always a critical component of Security due diligence. The VA should be part of every Privacy and BC/DR exercise as well!

Threat Modeling, a Different Slant on a TRA

- Create model of the application (Context Diagrams, DFD, UML etc)
 - System Components, Services, Assets, Data Flows, Users, etc.
- Recognize Planned or Existing Safeguards and Their Effectiveness
- Identify and Assess Threat Agents and Vulnerabilities, Develop “Threat Scenarios” or “Threat Trees”
- Categorize Vulnerabilities
 - Example method, use STRIDE
 - Spoofing, Tampering, Repudiation, Info Disclosure, Denial of Service, Elevation of Privilege
- Rank Threat Scenarios/Risks
 - Example method, use DREAD
 - Damage potential, Reproducibility, Exploitability, Affected Users, Discoverability

Use Cases - Appropriate and Inappropriate



TASSQ – February 16, 2011

Threats

- Simplified view, adversaries or events that can adversely affects assets
- Originates from mother nature, human error, cyber events, competitors, disgruntled employees, etc.
- Sources of threats may be within the organization's control, such as employees, or they may be uncontrollable, such as a recession.
- Evaluated on capability and motivation

Threat – Source, Agent, Event, Scenario

- Threat Source
 - The funding or supporting entity (organized crime, extremist group, mother nature, competitor, foreign country, etc.) of a Threat Agent
- Threat Agent
 - an Actor (person, application proxy, environmental event, etc.) acting on behalf of a Threat Source, which could exploit a vulnerability
- Threat Event
 - a Threat Agent exploiting a vulnerability
- Threat Scenario
 - consists of one or more threat events, carried out by a threat Agent, that could result in the compromise of an asset.
 - Easily formed as a full sentence

Safeguards, a.k.a. Control Practices

- Are the properties or components of the system that are used to protect assets in terms of confidentiality, integrity or availability.
- They can encompass all aspects of people, process and technologies
- Safeguards are meant to work in concert with other safeguards to protect design vulnerabilities against exploitation by threats.
- Existing or Planned safeguards must be identified and taken into account during the Vulnerability Assessment

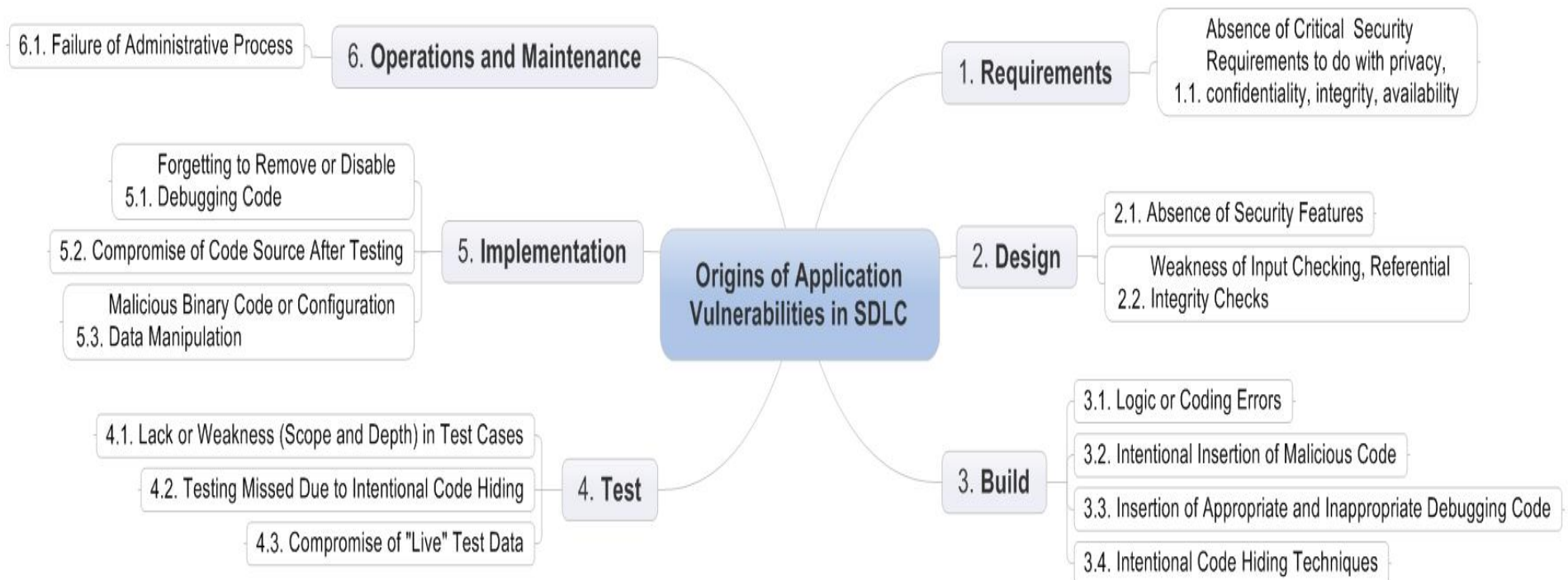
Vulnerabilities as input to test cases

TASSQ – February 16, 2011

Vulnerabilities

- Are limitations in the design or configuration of the system components that can be exploited in a threat scenario.
- Examples of vulnerabilities include:
 - Software design flaw
 - Improperly configured routers
 - Traffic exceeding a maximum bandwidth available,
 - No training of user on a particular complex software application.
- Vulnerabilities are described by:
 - Narrative description of the business/security design requirement which is not being satisfied
 - Control objectives or Control activities which are not met
 - IT system assets which may be exposed

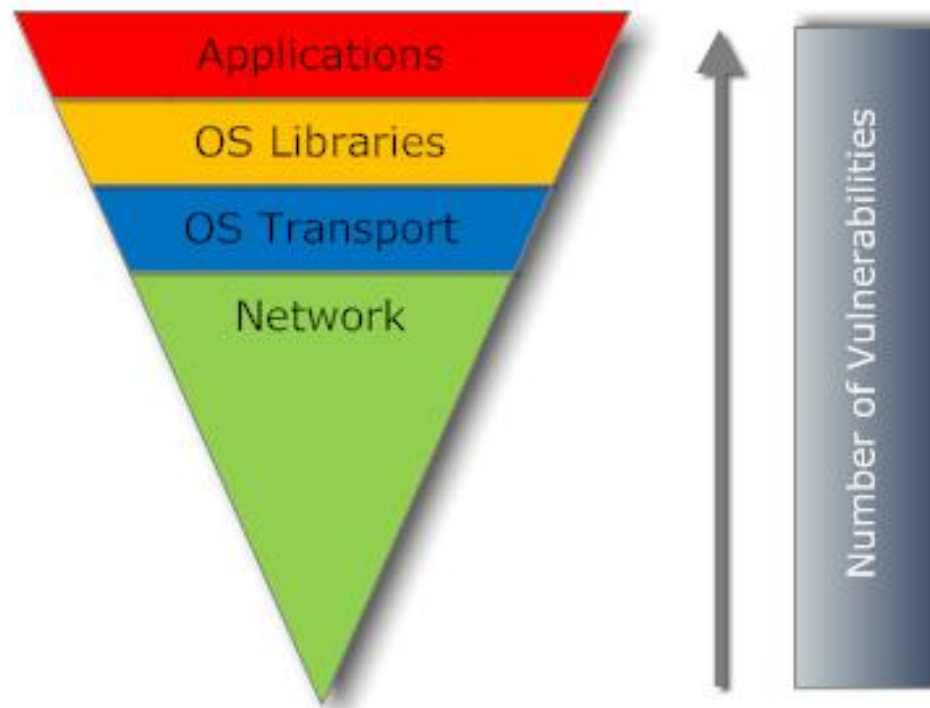
Where do Software Vulnerabilities Come From?



SANS Top 20 Vulnerabilities

- September 2009 update. “Two risks dwarf all others, but organizations fail to mitigate them”
 - Client-side software remains unpatched
 - Internet-facing web sites that are vulnerable

Application Vulnerabilities Exceed OS Vulnerabilities



TASSQ – February 16, 2011

Client Side Software that Remains Unpatched

- Client-side vulnerabilities in commonly used programs such as Adobe PDF Reader, QuickTime, Adobe Flash and Microsoft Office
 - Targeted email attacks, “spear phishing”
 - Visiting infected web sites
- Requires only download or opening of documents, music, videos

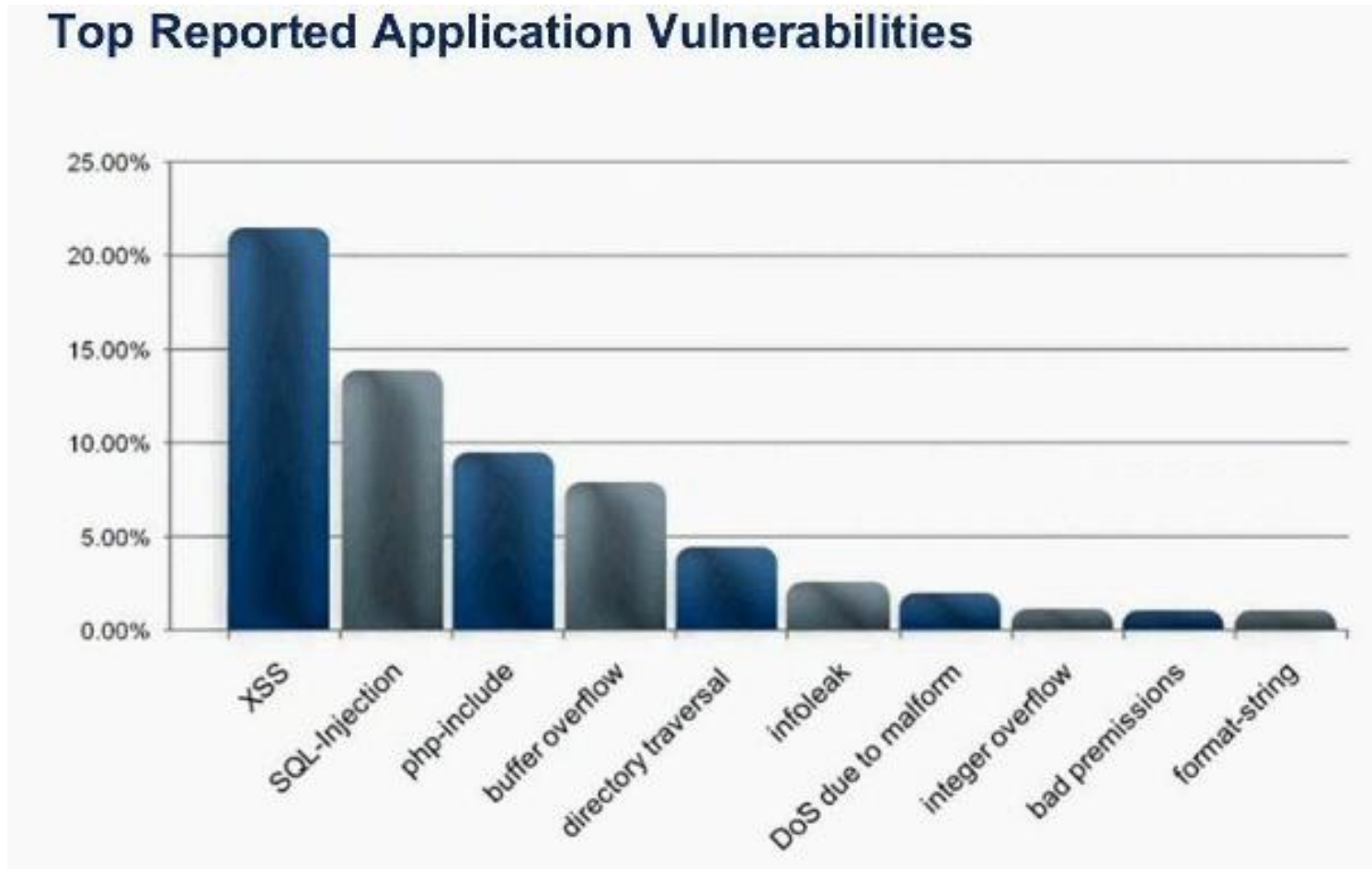
Vulnerable Internet-Facing Web Sites

- 80% of vulnerabilities come from SQL injection and Cross-Site Scripting flaws in open-source and custom-built applications

Web Application Attacks

- Two Main Avenues
 - Brute Force Password Guessing
 - MS SQL, FTP and SSH servers are popular targets (why?)
 - Web Application Attacks
 - SQL Injection, Cross-site Scripting and PHP File Include attacks are still most popular techniques

Application Vulnerability Popularity



TASSQ – February 16, 2011

Vulnerability Life-Cycle

- **Birth**

- Denote the creation of a product, system integration or process weakness

- **Discovery**

- When someone discovers that a weakness can be exploited, it becomes a vulnerability

- **Disclosure**

- The vulnerability is disclosed when the discoverer reveals details of a problem to a wider audience

- **Correction**

- A vulnerability is correctable when the vendor/designer/operator releases a modification or configuration change that corrects the underlying weakness

Vulnerability Life-Cycle (cont'd)

- **Publicity**

- The key element here is that the vulnerability becomes known on a large scale once the disclosure is out of control. May occur before or after the Correction

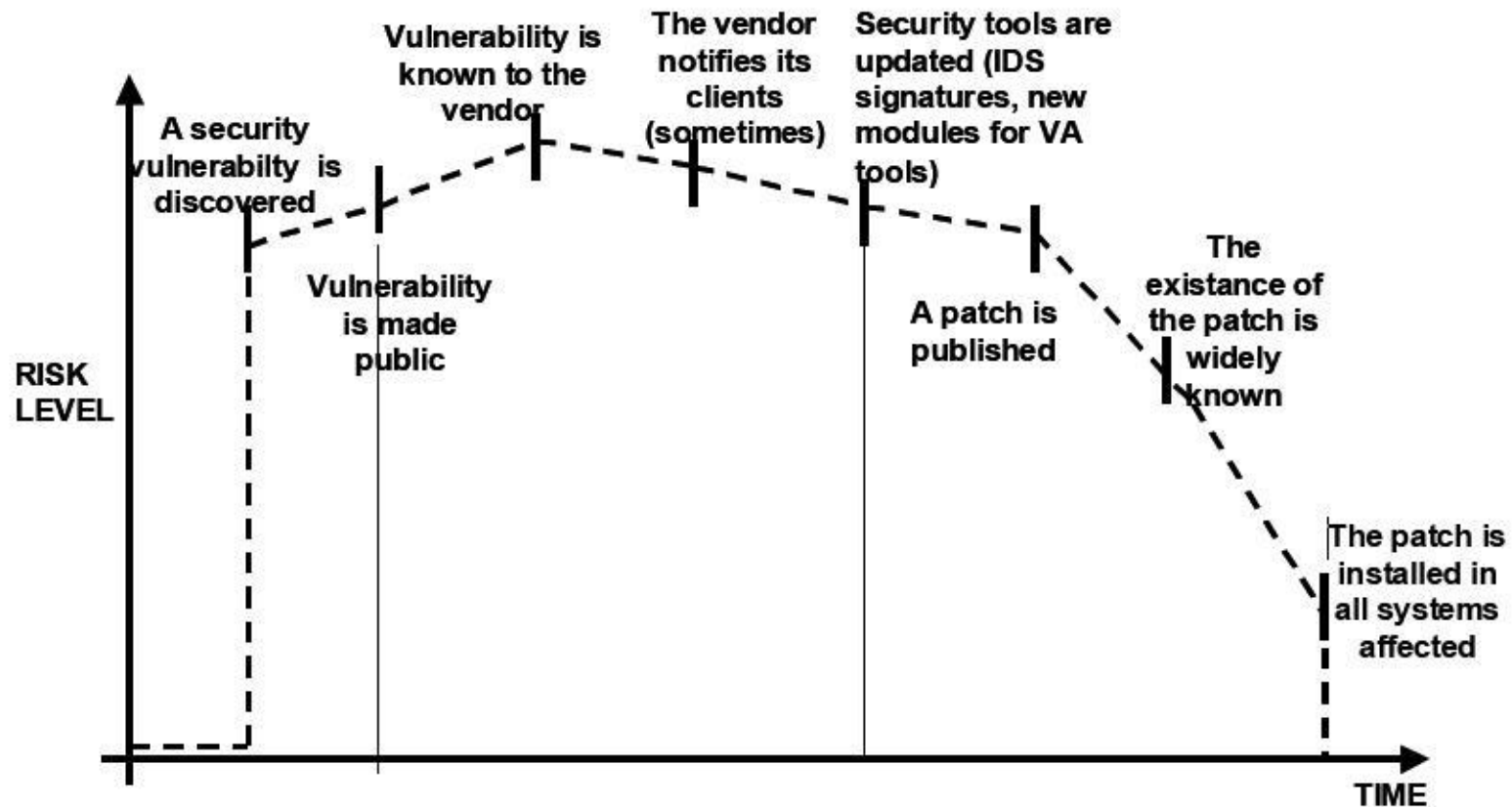
- **Scripting**

- The vulnerability has been industrialized

- **Death**

- A vulnerability dies when the number of exploitable systems shrinks to insignificance.

Vulnerability Life Cycle



Security Testing Techniques

TASSQ – February 16, 2011

Security Testing Techniques

- Static or Dynamic Analysis
- White Box, Black Box or Grey Box
- Manual Review
- Binary-code Review

When To Undertake?

Life Cycle Phase	Reviews/tests
Requirements	Security review of requirements and abuse/misuse cases
Architecture/Product Design	Architectural risk analysis (including external reviews)
Detailed Design	Security review of design. Development of test plans, including security tests.
Coding/Unit Testing	Code review (static and dynamic analysis), white box testing
Assembly/Integration Testing	Black box testing (fault injection, fuzz testing)
System Testing	Black box testing, vulnerability scanning
Distribution/Deployment	Penetration testing (by software testing expert), vulnerability scanning, impact analysis of patches
Maintenance/support	(Feedback loop into previous phases), impact analysis of patches and updates

Static Analysis

- Analysis of a program carried out without executing the program
- No need to simulate or keep track of application conditions and combinations of keystrokes
- Could apply to either source code or binary code review

Dynamic Analysis

- The process of evaluating a system or component based on its behavior during execution
- Must simulate or keep track of application conditions and combinations of keystrokes
- Applies to binary code review

White Box Testing

- Testing that is based typically on source code but now also binary code and the knowledge of how the system is implemented for intended and unintended behavior
- Includes analyzing data flow, control flow, information flow, coding practices, and exception and error handling within the system
- Occurs early in the SDLC

Black Box Testing

- Testing that is based on compiled binary executables and an analysis of the specification of the component without reference to its internal workings
- To explore the software's behavior from the outside from user inputs or external interfaces to the software
- Occurs later in the SDLC (after coding)

Grey Box Testing

- Combination of White and Black Box
- Dynamic test findings allow further static analysis or vice versa

Binary Code Analysis

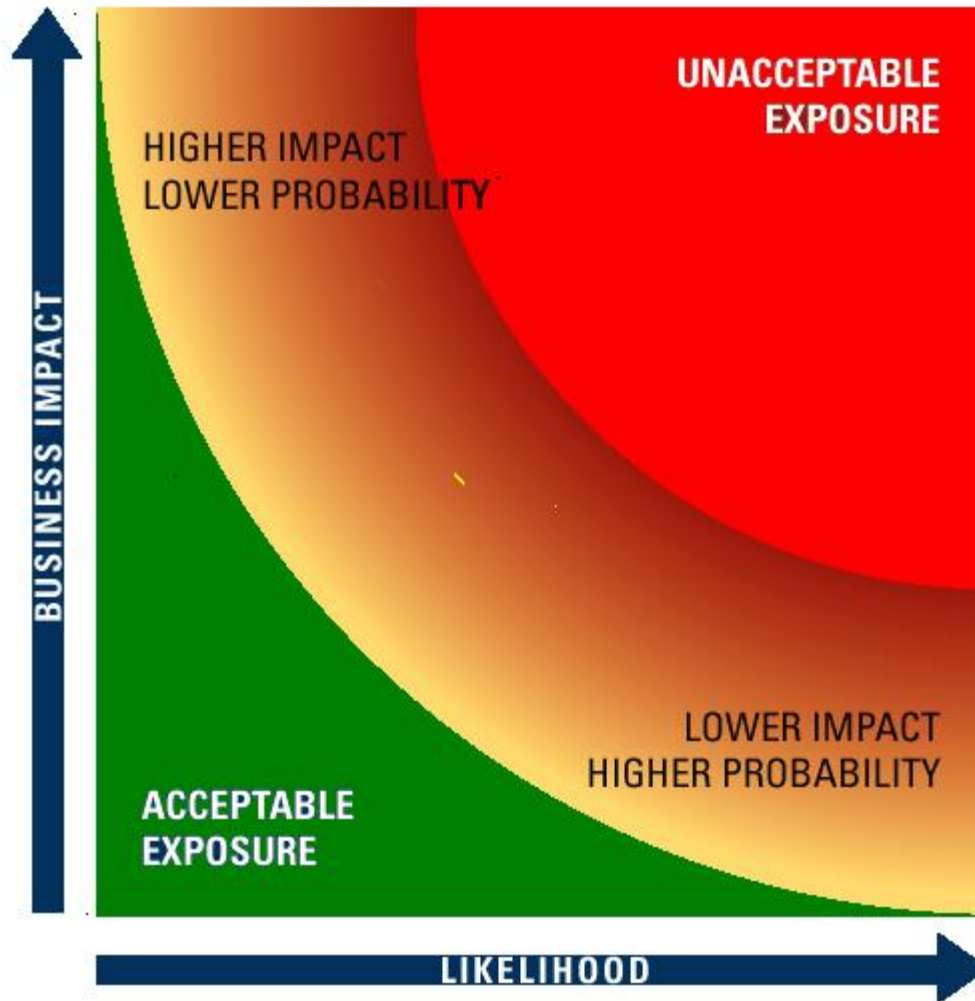
- Source code is compiled along with any externally included libraries to form a machine dependant “binary” file
- This binary file can be modified directly without modification to its source code
- Thus, analysis of only source code does not necessarily guarantee an associated binary file; and vice versa
- Analysis of only the binary file does not necessarily guarantee the associated source code
- However, the binary file is what is ultimately installed and runs on the target system

Recommendations By Vulnerability

- Recommended addition, modification or removal of Safeguards/Control Activities
- Sometimes more important than the supporting TRA, PIA or other is the list of recommendations
- Each Vulnerability and Threat Scenario must have at least one but usually many recommendations
- Recommendations apply to one or many Vulnerabilities or Threat Scenarios
- Important to “slice and dice” recommendations by “Point of View” and also look for patterns

Risk: prioritizing vulnerabilities within a business context

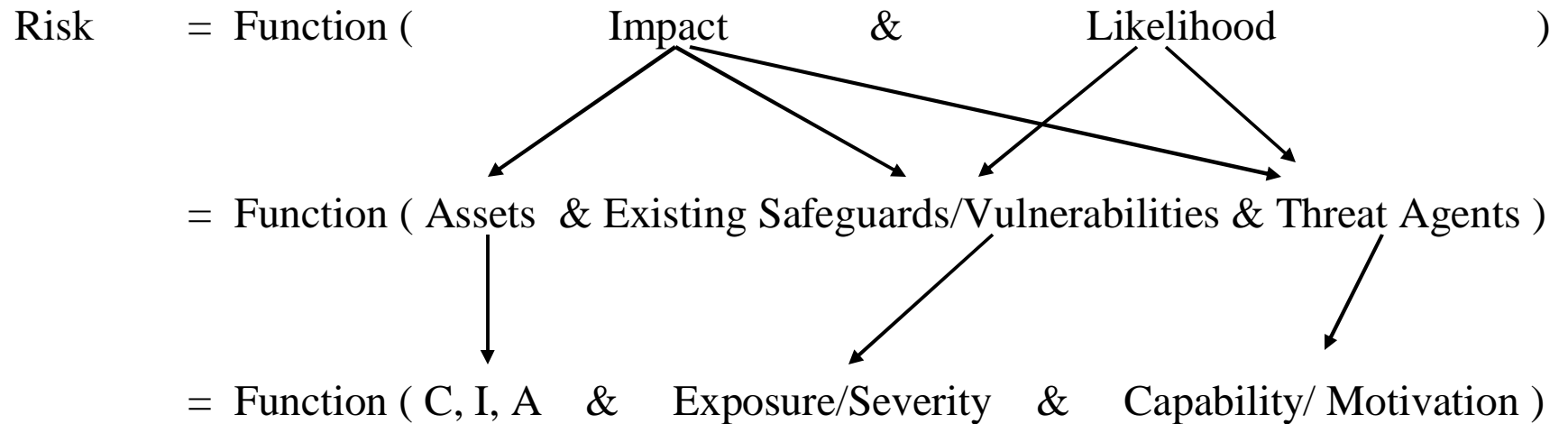
It's All About Risk



Risk - Control = Exposure

- Risk is defined as the impact and likelihood that threats can adversely affect an organization's business strategies and objectives.
- Exposure is the potential adverse impact when control does not fully mitigate risk.

Asset or Service Based Risk Equation



Impact

- Is the quantified, undesirable effect that a threat scenario can have on an IT system asset
- Impact is related to asset value (sensitivity to confidentiality, integrity and availability) and the type of exploitation results of a successful threat scenario
- Impacts are typically described by:
 - Narrative description of damage caused, both tangible and intangible
 - Assets involved
 - Cost in time, dollars, reputation
 - Severity rating (High, Medium, Low)

Likelihood

- Is the chance of success that the Threat Agent undertaking a particular Threat Scenario can exploit specific vulnerabilities to gain some form of access to an IT system asset.
- Likelihood is typically described by:
 - Frequency of past occurrences,
 - Percentage chance of occurrence over a specified number of days, months or years

Risk Rating

- Risk Rating is a quantification of the likelihood and potential impact of a threat agent executing a threat scenario that successfully exploits one or more vulnerabilities in components of the business system that is in place to protect assets.
- Risk Rating really indicates priority of doing something to mitigate the situations
- The risk associated with the assets can never be reduced to zero.

Example: OWASP Top 10 Application Risks

A1: Injection

A2: Cross Site Scripting (XSS)

A3: Broken Authentication and Session Management

A4: Insecure Direct Object References

A5: Cross Site Request Forgery (CSRF)

A6: Security Misconfiguration

A7: Failure to Restrict URL Access

A8: Unvalidated Redirects and Forwards

A9: Insecure Cryptographic Storage

A10: Insufficient Transport Layer Protection



OWASP

The Open Web Application Security Project
<http://www.owasp.org>

http://www.owasp.org/index.php/Top_10

TASSQ – February 16, 2011

Risk Decisioning & Acceptance Process

- Once the Risk Assessment is completed with a focus on BC/DR, Priv and/or Sec, the output becomes the input to the Risk Acceptance Process
- Need to accept (once verified) Risk results as valid
- Need to make Risk Decisions as to whether or not to action any/all recommendations on all significant Risk Statements
- Must involve the governance roles which have binding authority

Risk Acceptance Concerns

- How to organize and present the Risk Statements?
- What recommended controls will be actioned in light of the larger organizational needs? Look at the broader Enterprise Risk Model
- Begin process of recommendation evaluations for feasibility, cost/benefit, ...these take on a life of their own
- Do the recommendations to be actioned still mitigate the threat scenarios?
- What constraints cannot be changed?
- What waivers must be put into place and for how long?
- What are the interim mitigations until the recommendations are implemented?
- What will be the actual Residual Risk now and once implemented?

Suggested Management Actions Tied to Risk Rating

Impact	Risk Management Actions		
Significant	Considerable management required	Must manage and monitor risks	Extensive management essential
Moderate	Risks may be worth accepting with monitoring	Management effort worthwhile	Management effort required
Minor	Accept risks	Accept, but monitor risks	Manage and monitor risks
<div style="display: flex; justify-content: space-around;"> Low Medium High </div> <p style="text-align: center;">Likelihood</p>			

Risk Treatment, a.k.a. IT Security Plan

- Plan contains the results of the TRA and Risk Acceptance processes
- Describes actions to be undertaken in the short, medium and long term to achieve an appropriate level of risk for all significant risk statements
- Includes an implementation schedule along with expected costs

References

- Information Assurance Technology Analysis Center (IATAC) and Data and Analysis Center for Software (DACS). Software Security Assurance. July 31, 2007.
- Building Security In. <https://buildsecurityin.us-cert.gov/daisy/bsi/home.html>
- Howard, Michael, and David LeBlanc. Writing Secure Code: Practical Strategies and Proven Techniques for Building Secure Applications in a Networked World. CD-ROM Edition. Redmond, WA: Microsoft Press, 2001.
- Howard, Michael, David LeBlanc, and John Viega. 19 Deadly Sins of Software Security (Security One-Off): Programming Flaws and How to Fix Them. New York; London: McGraw-Hill/Osborne Media, 2005.
- Viega, John, and Gary McGraw. Building Secure Software: How to Avoid Security Problems the Right Way. Boston: Addison-Wesley Professional, 2001.
- Vineet Kumar Maurya, Santhosh Babu G, Jangam Ebenezer, Muni Sekhar V, Asoke K Talukder, Alwyn Roshan Pais. Suraksha: A Security Designers' Workbench: Dept. of Computer Engineering, National Institute of Technology Karnataka, Surathkal, March 2009.
- F. Swiderski and W. Snyder, Threat Modeling. Washington: Microsoft Press, 2004.

References cont'd

- J.D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla and Anandha Murukan, “Improving Web Application Security: Threats and Countermeasures,” msdn.microsoft.com, June 2003.
- NIST, National Institute of Standards and Technology, Computer Security Division, extensive information security publications as a result of NIST studies, investigations and research; csrc.nist.gov
- The SANS (SysAdmin, Audit, Network, Security) Institute, countless articles and guides; www.sans.org
- The Open Web Application Security Project (OWASP), Application Security Desk Reference, Developer’s Guide, Testing Guide, Code Review Guide and Control Framework; www.owasp.org
- Veracode, Chris Wysopal, Chris Eng, Guide to Software Risk Assessments and Audits; Static Detection of Application Backdoors and other whitepapers; www.veracode.com/resources





Keith Jonah
Director
Trusted By Design Inc.

kjonah@ca.inter.net

416-727-3809

TASSQ – February 16, 2011