Understanding Product Characteristics Throughout the SDLC

SwA Working Groups June 22, 2010

Mike Kass, NIST

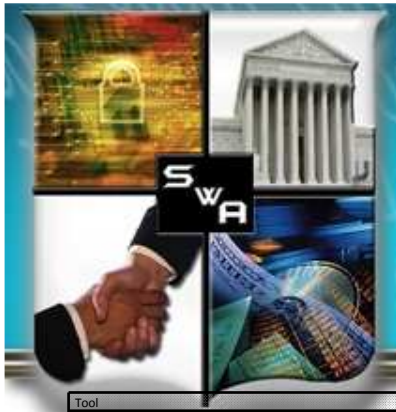
Michele Moss, Booz Allen Hamilton

- Overview of Tools for the Phases Where They Are Most Beneficial
  - Requirements
  - Design
  - Implementation
    - Keys to success
    - Potential use of Common Enumerations
    - Value to Development Stakeholders
    - Value to Acquisition Stakeholders
  - Testing
  - Acquisition
  - Operations (?)

Tool	Skills	Benefits	Drawbacks
Static Analysis Code Scanning	Understanding of the implementing language	• Reduces cost over system life • Educates developers about secure programming • Rechecks legacy code • Automates repetitive and tedious aspects of source code security audits • Checks for good programming style • No disassembly	• No architectural-level flaws
Pedigree Analysis	None	• Reduces cost over system life • Educates developers about secure programming • Rechecks legacy code • Automates repetitive and tedious aspects of source code security audits • Reduces the amount of testing necessary • No disassembly	• No architectural-level flaws • Requires use of open source software
Byte Code Analysis	Understanding of byte code Understanding of testing methodology	• Reduces cost over system life • Rechecks legacy code • Checks for good programming style • No need for source code • Guarantees that the analysis is performed on the actual product	• No architectural-level flaws • Requires additional analysis • Limited to a single language

- Adopt a secure coding standard
  - Develop and/or apply a secure coding standard for your target development language and platform [cert] [ca]
  - Make a security policy that prohibits the use of banned functions that make code weak. [atsec]
- Validate input [cert] (msdn) [safecode] [ca]
  - Assume nothing on user input [cgi]
  - Formulate your validation criteria for all user input [cgi]
  - Enforce the validation criteria on all user input [cgi]
  - Validate the data on a trusted machine (the server), not on a client [cgi]
  - Trust only what has been validated [cgi]
  - Use multiple-tier validations [cgi]
- Validate output [safecode]
  - Sanitize data sent to other systems [ca] [cert]
- Etc

- Implementation-specific CWEs
- Corresponding Attack Patterns against CWEs
- “Bug Metrics”…. For measuring development process quality



SOFTWARE ASSURANCE FORUM

BUILDING SECURITY IN

SwA Tools for the Testing Phase

Tool	Skills	Benefits	Drawbacks
Source Code Fault Injection	• Underlying source code • Implementing language • Testing methodology	• Reduces cost over system life • Increased test coverage • Increased accuracy • Improved accuracy and coverage • No disassembly	•Required expertise •Lack of tool availability •Additional analysis •Additional preparation
Dynamic Analysis	• Underlying development methodology • Testing methodology	• Reduces cost over system life • No need for source code • Improved accuracy and coverage • No disassembly to perform	• No architectural-level flaws • Thorough understanding of the software • Requires additional preparation
Binary Fault Injection	None	• Reduces cost over system life • Rechecks legacy code • Increased test coverage • No need for source code • Improved accuracy and coverage • No disassembly • Guaranteed the analysis is performed on the actual product	• Thorough understanding of the software • Additional analysis
Fuzzing			

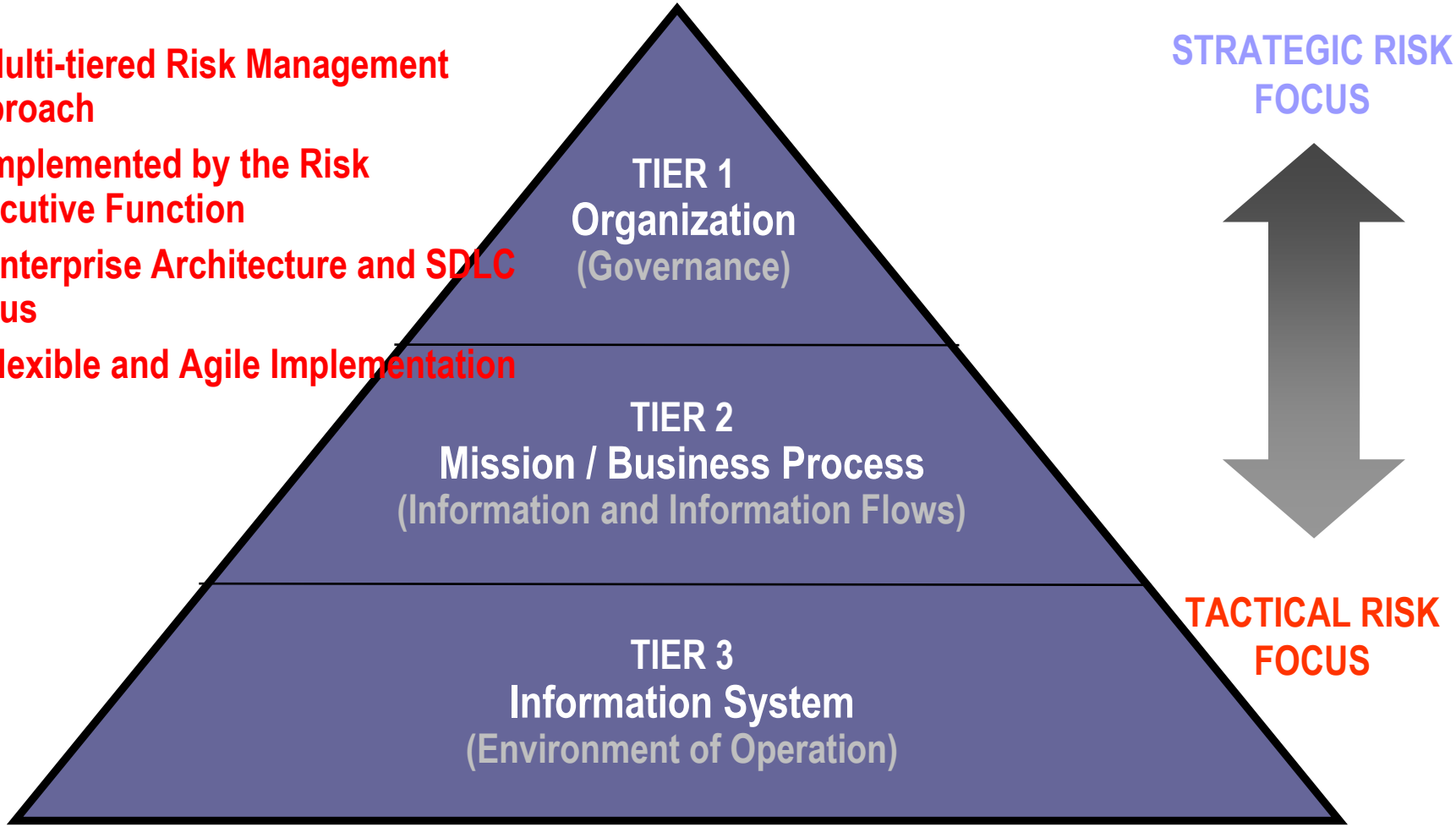
- Use effective assurance techniques by performing: [ca]
  - fuzz testing [safecode]
  - penetration testing [safecode]
  - a third party/independent assessment of the code [safecode]
- Verify that a program does not contain hidden malware by maximizing the percentage of code tested (average unit testing of software tests only about 50% of the code)
- Perform risk-driven security testing (testing under abnormal conditions based on attack patterns) and stress testing (testing under abnormal activity, inputs, etc.) [sis]
- Do not limit test cases to normal operating conditions [sis]
- Test the most important systems first [nist]
- Integrate security testing into the risk management process [nist]
- Look at the big picture. The results of routine testing may indicate that an organization should readdress its systems security architecture [nist]

- Understand the capabilities and limitations of vulnerability testing. Vulnerability testing may result in many false positive scores, or it may not detect certain types of problems that are beyond the detection capabilities of the tools [nist]
- Fix security issues correctly instead of "hacking" in a fix [owasp]
- Ensure QA supports edge/boundary value condition testing. [BSIMM]
- Integrate the attacker perspective into test plans. [BSIMM}
- Allow declarative security/security features to drive tests. [BSIMM]
- Begin to build/apply adversarial security tests (abuse cases). [BSIMM]
- Include security tests in QA automation. [BSIMM]
- Drive tests with risk analysis results. [BSIMM]
- Leverage coverage analysis. [BSIMM]
- Require "application-specific" security testing to ensure baseline security before deployment. [SAMM]

- **Value Provided to Development Stakeholders**
  - Independent verification of expected security mechanisms surrounding critical business functions [SAMM]
  - High-level due diligence toward security testing [SAMM]
  - Ad hoc growth of a security test suite for each software project [SAMM]
  - Deeper and more consistent verification of software functionality for security [SAMM]
  - Development teams enabled to self-check and correct problems before release [SAMM]
- **Value Provided to Acquisition Stakeholders**
  - Stakeholders better aware of open vulnerabilities when making risk acceptance decisions [SAMM]

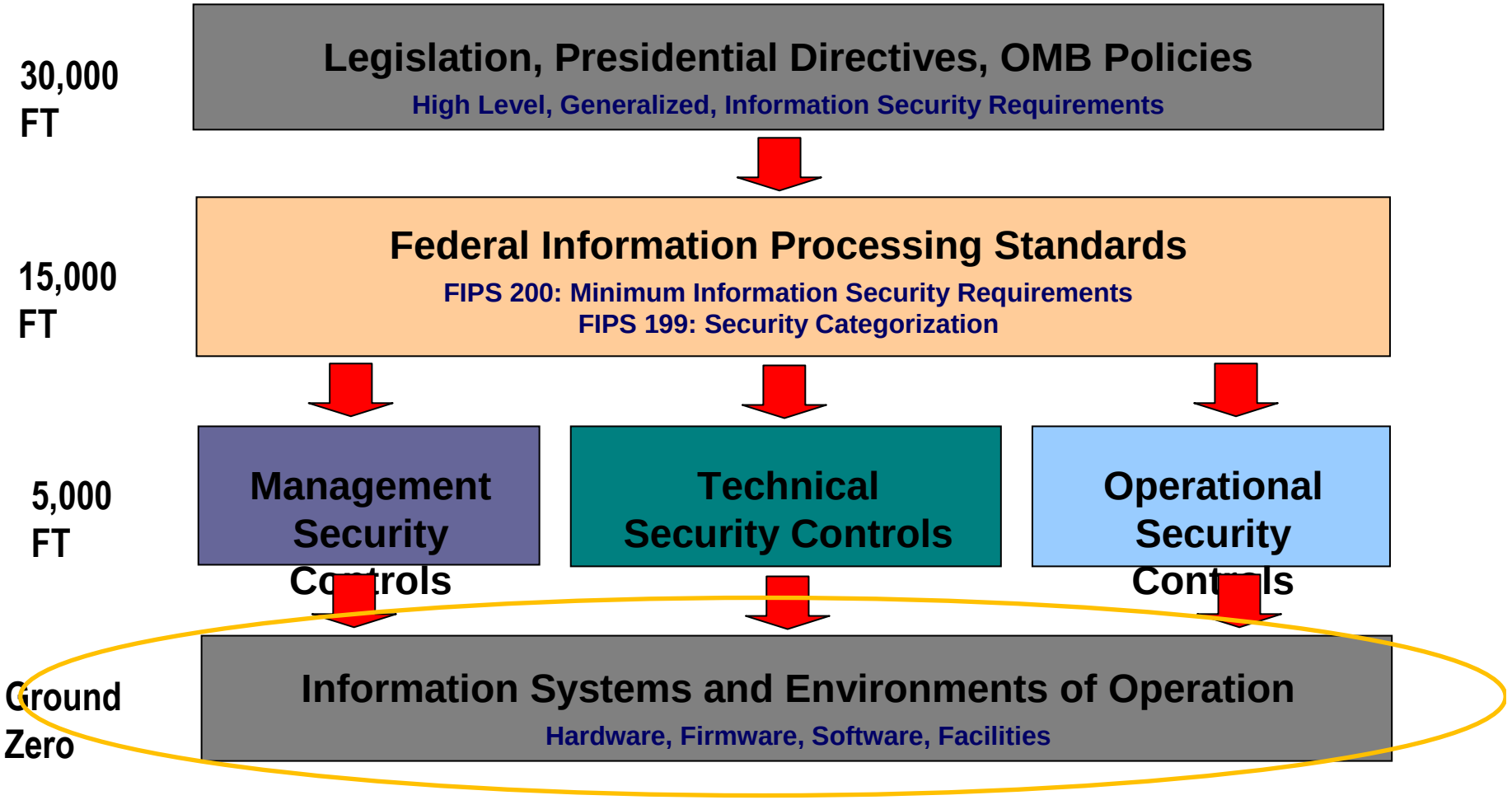
Enterprise-Wide Risk Management

- Multi-tiered Risk Management Approach
- Implemented by the Risk Executive Function
- Enterprise Architecture and SDLC Focus
- Flexible and Agile Implementation



FISMA 2010 and Beyond
Strategic and Tactical Risk Management and the Role of Software Assurance
Ron Ross, NIST
Software Assurance Workshops
June 21, 2010

Security Requirements Traceability



FISMA 2010 and Beyond
Strategic and Tactical Risk Management and the Role of Software Assurance
Ron Ross, NIST
Software Assurance Workshops

SP 800-53 Security Control Families

Supporting Software Assurance

- Configuration Management
  - Configuration Change Control
  - Security Impact Analysis
  - Access Restrictions for Change
  - Configuration Settings
  - Least Functionality
- System and Information Integrity
  - Security Functionality Verification
  - Software and Information Integrity
  - Information Input Validation
  - Error Handling
  - Predictable Failure Prevention

FISMA 2010 and Beyond
Strategic and Tactical Risk Management and the Role of Software Assurance
Ron Ross, NIST
Software Assurance Workshops
June 21, 2010

SP 800-53 Security Control Families

Supporting Software Assurance

- Program Management
  - Mission/Business Process Definition
  - Enterprise Architecture
  - Risk Management Strategy
  - Information Security Resources
  - Information Security Measures of Performance
- System and Services Acquisition
  - Resource Allocation
  - Acquisition and Life Cycle Support
  - Security Engineering Principles
  - Developer Configuration Management and Testing
  - Trustworthiness and Critical Information System Components
  - Supply Chain

SP 800-53 Control ID	Control Name
AT-3	**Security Awareness** -The organization …provides appropriate information system security training:
RA-3	**Risk Assessment -** The organization conducts assessments of the risk and magnitude of harm…
RA-5	**Vulnerability Scanning -** The organization scans for vulnerabilities in the information system
SI-2	**Flaw Remediation** - The organization identifies, reports, and corrects information system flaws.
SI-3	**Malicious Code Protection** - The information system implements malicious code protection.
SI-10	**Information Accuracy, Completeness, Validity, and Authenticity** - IS checks information for accuracy, completeness, validity, and authenticity.

- Study NIST SP 800-53 for possible contribution to efforts related to the Information System level of Enterprise Risk Management
  - Explore expanding the content through collaboration with
    - MAEC efforts
    - SwA Practice Self-Assessment efforts
    - Contributions from industry implementation
- What knowledge do we need to share with NIST?
- How to collaborate?
- Continue discussions at future SwA events