Pulse2011

New Techniques in Web 2.0 Security

Steven “Schmidtty” Schmidt
Technical Specialist, IBM Security Solutions

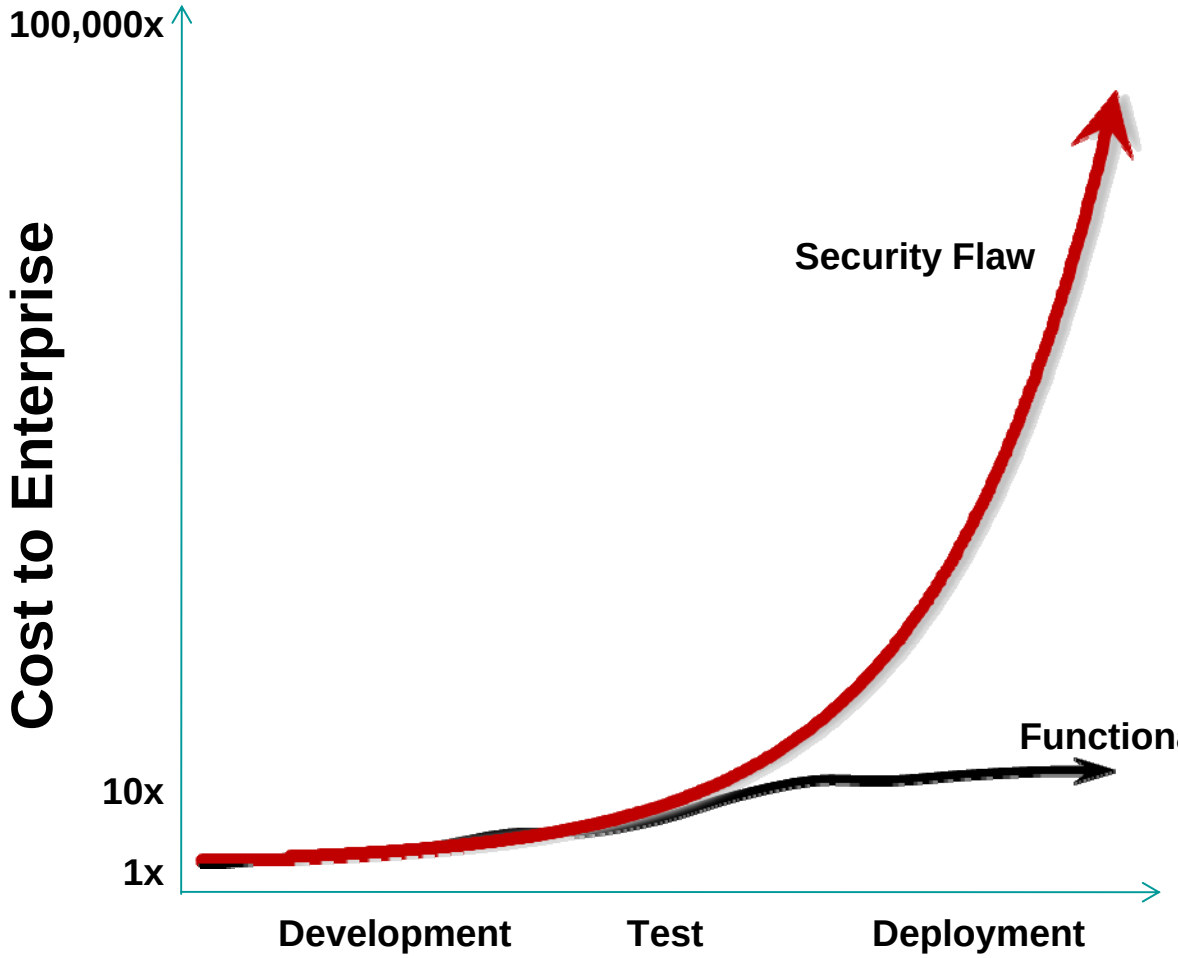
Accelerating Awareness and Progress: Secure by Design

Secure by Design is a **cost-effective** approach to constructing **safe and reliable systems** by applying IBM's experience with security technologies and best practices in all phases of system creation, from conception through system design, construction and deployment.

Being **Secure by Design** reduces the **cost**, **risk**, and **unpredictability** of integrating new technologies.

© 2011 IBM Corporation

Incremental and Unbudgeted Costs of Critical Breaches



Unbudgeted Costs:

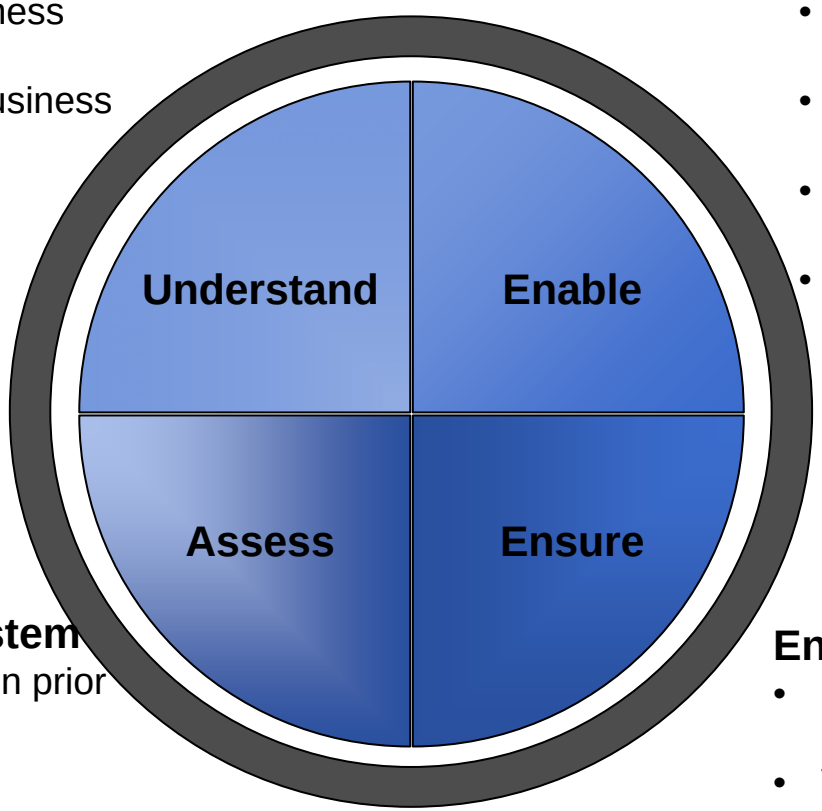
- Customer notification / care
- Government fines
- Litigation
- Reputational damage
- Brand erosion
- Cost to repair

© 2011 IBM Corporation

Components of a Successful Strategy to be Secure by Design

Understand Security Drivers

- Recognize specific business opportunity and priority
- Assess system risk to business
- Model likely threats and impacts



Enable System Security

- Mandate appropriate security controls within applications
- Implement system monitoring/logging
- Specify secure platform & configuration
- Document update management plan

Assess Production System

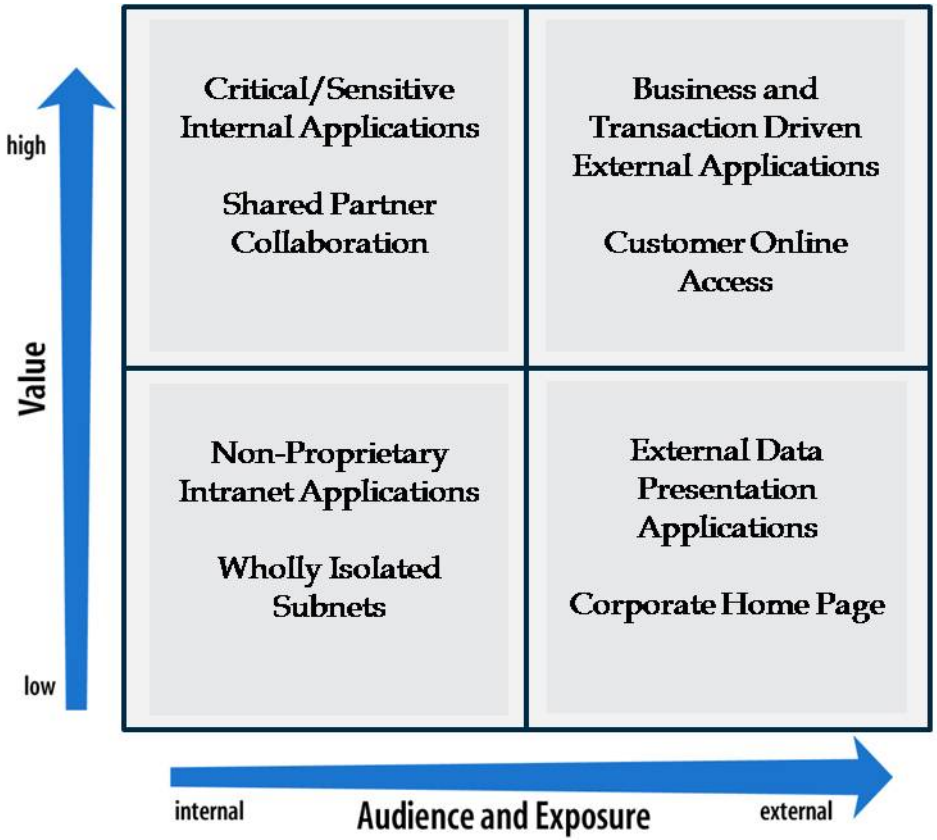
- Test composed application prior to deployment
- Verify security of platform configuration
- Establish process and schedule for regular checking

Ensure secure development

- Protect development infrastructure
- Verify safety of third-party software
- Check system for security during coding/build/integration stages

© 2011 IBM Corporation

Scoping and Prioritizing Security Improvements : Inventory



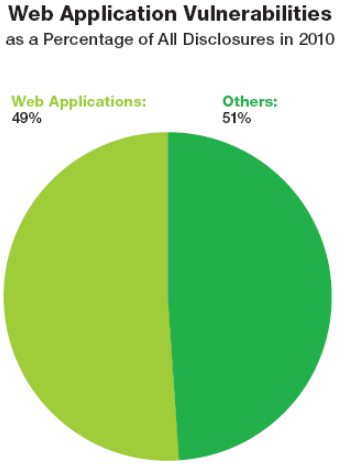
Focus on actual needs:
- Confidentiality of data
- Reliability of systems
- Integrity of data and services

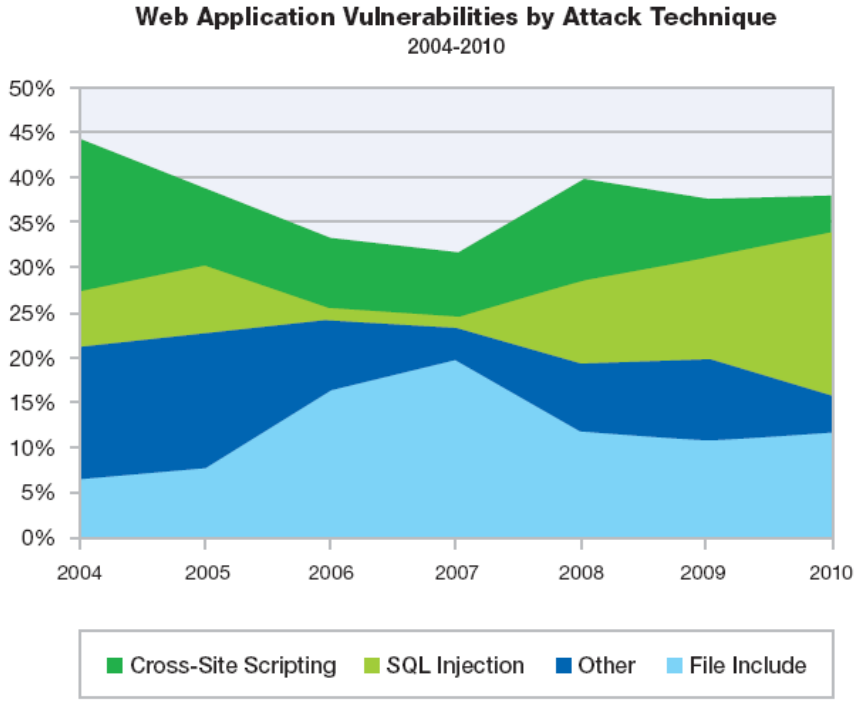
Understand where to focus:
- New Development
- Systems in Maintenance
- Legacy Systems

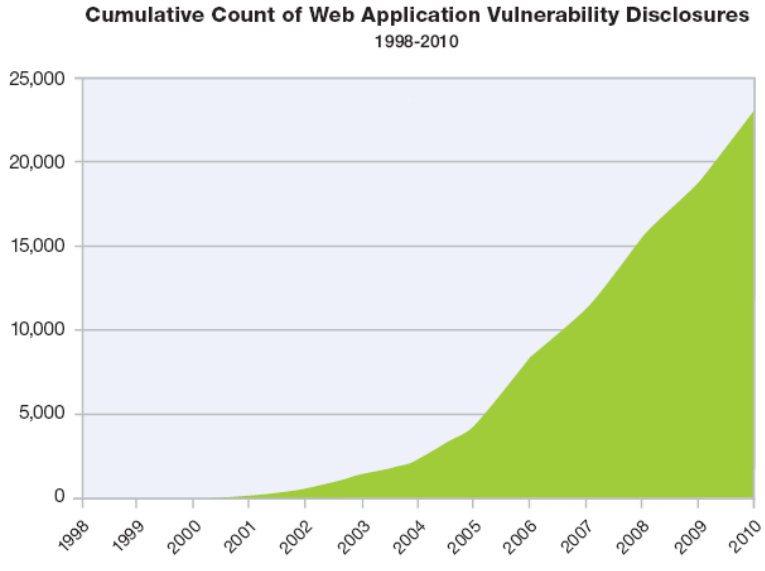
© 2011 IBM Corporation

Web App Vulnerabilities Continue to Dominate

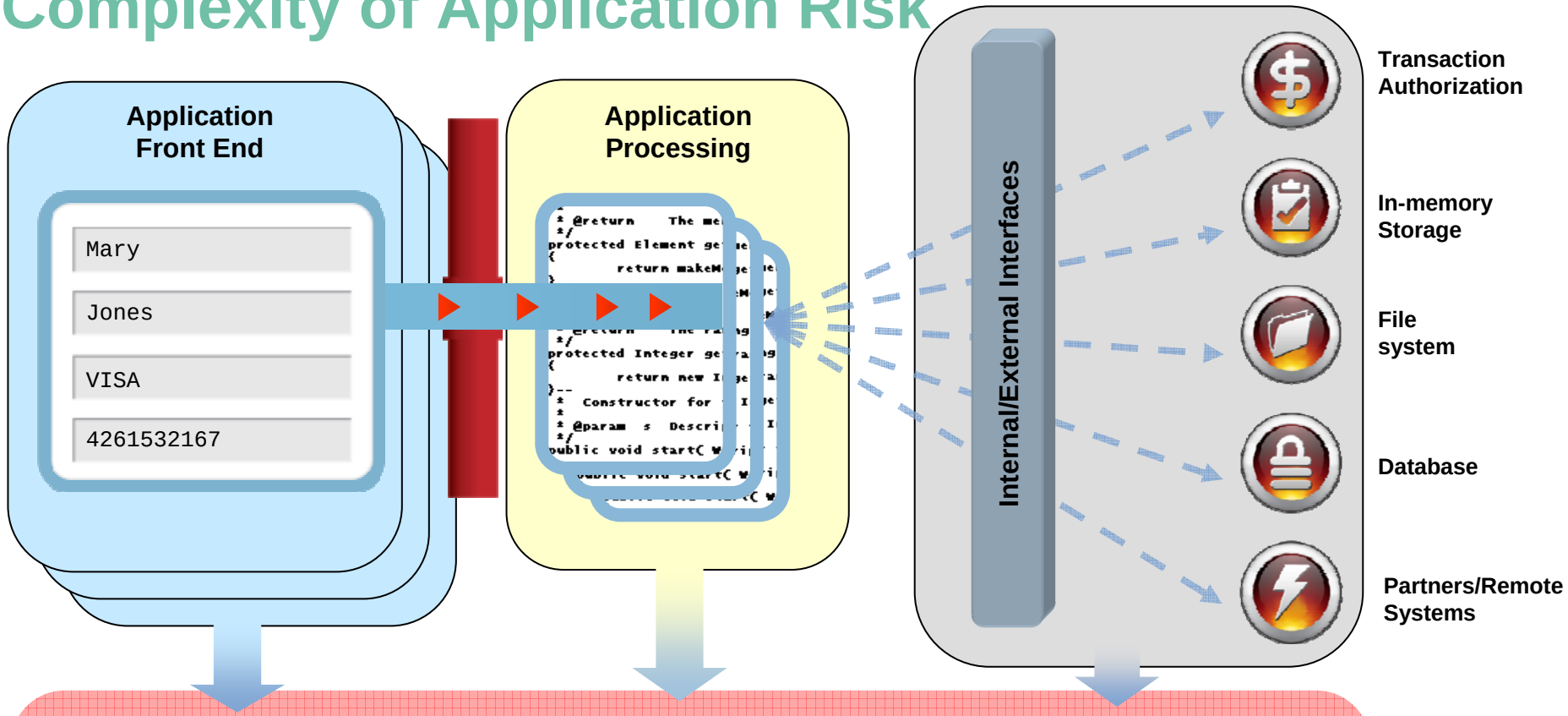
- Nearly half (**49%**) of all vulnerabilities are Web application vulnerabilities.
- Cross-Site Scripting & SQL injection vulnerabilities continue to dominate.







Complexity of Application Risk



External Vulnerabilities

- Web-based Attacks
- Misconfiguration
- Data Leakage
- Insufficient Input/Output Validation
- Number & Variety of access points/devices

Application Issues

- Programming Errors
  - Buffer Overflows
  - Race Conditions
- Insecure Data Handling
- Weak Access Control
- Malicious Code

Integrated Systems Risks

- Insecure Data Transmission
- Inappropriate Data Storage
- Incomplete Data Destruction
- Unmanaged Data Sharing

© 2011 IBM Corporation

New Techniques in Application Security

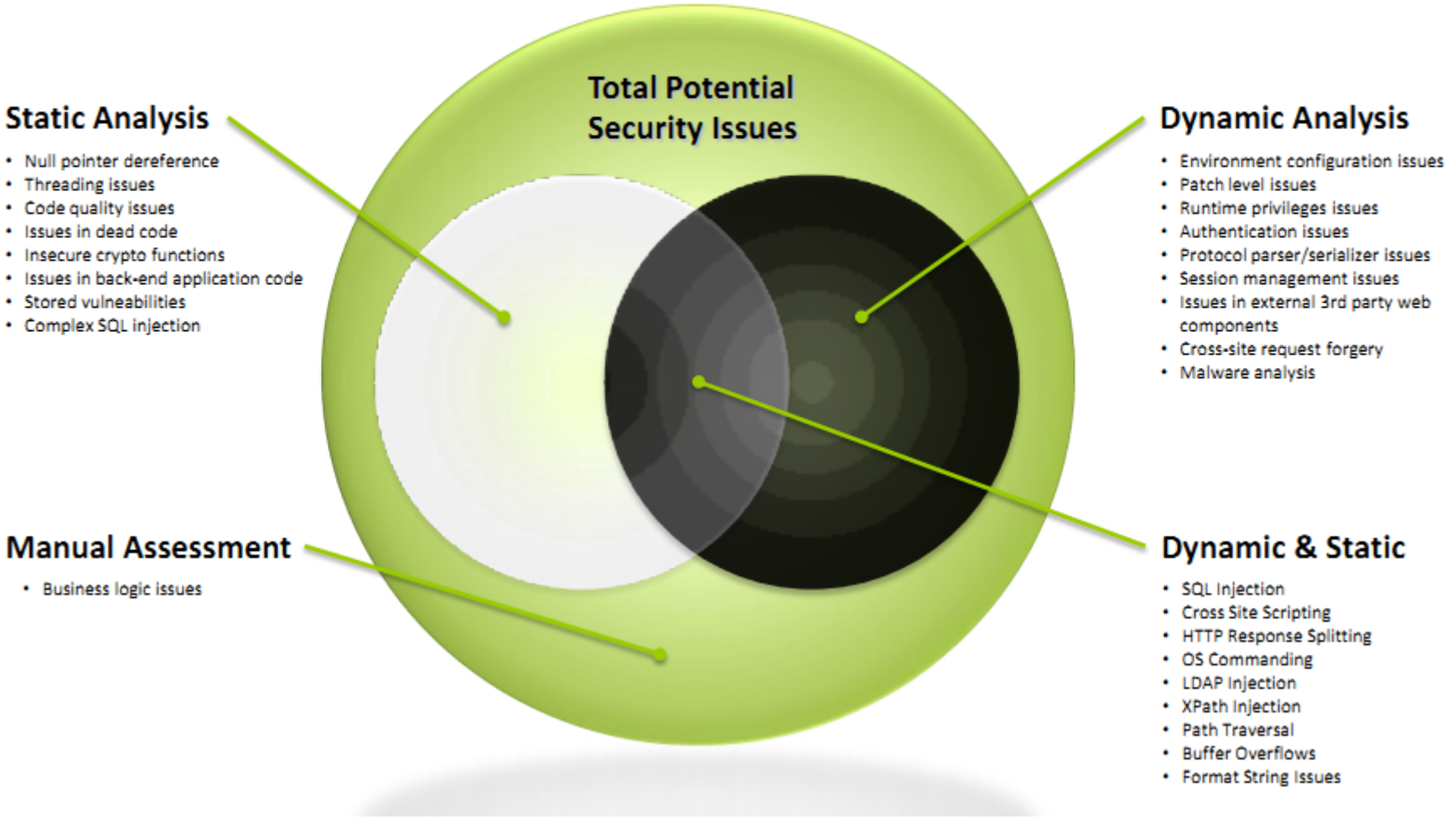
© 2011 IBM Corporation

Differences Between DAST and SAST Approaches

	Static Analysis	Dynamic Analysis
Scan input	Scans source code and bytecode for security and quality issues. Requires access to source or bytecode	Scans running web applications. Requires starting point URL, and login credentials where relevant
Assessment techniques	Uses "taint analysis" and pattern matching techniques to locate issues	Tampering of HTTP messages to locate application and infrastructure layer issues
Where does it fit in application development lifecycle	Fits best during application development	Fits best during build automation, QA and security verification of applications
Results & Output	Results are presented by line of code, source to sink functions flow	Results are presented as HTTP messages (exploit requests)

© 2011 IBM Corporation

DAST and SAST – Issue Type Coverage



© 2011 IBM Corporation

Path to real hybrid analysis – how far are we?

- Definitions
  - Aggregation
    - Combining in a single report or repository the issues reported using both DAST and SAST techniques
  - Correlation
    - Associating issues that have similar key indicators as the same issue (more on this later)
  - Hybrid
    - Combining the analysis techniques of both SAST and DAST to provide new analysis capabilities that couldn't (or is hard) to do using only a single technique.

© 2011 IBM Corporation

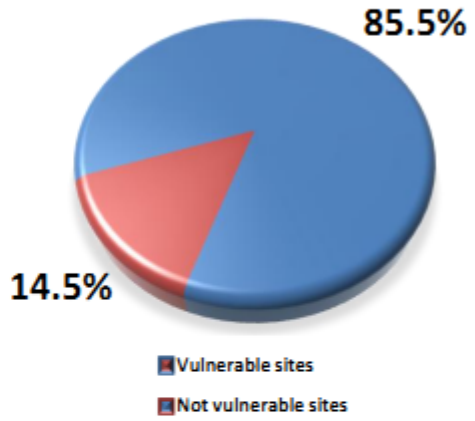
JavaScript Security Analyzer (JSA) – Overview

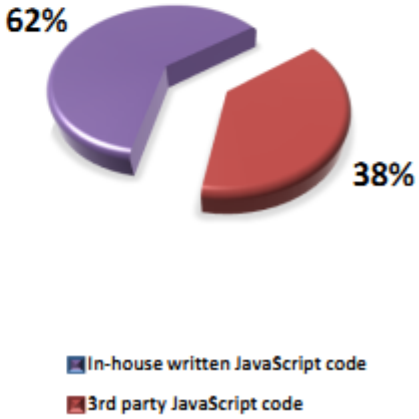
- What is JSA?
  - An extension of AppScan Standard, developed in collaboration with IBM Research, that does static taint analysis of JavaScript, detecting a range of client-side security issues:
    - DOM-based XSS
    - Code Injection
    - Open Redirect
    - CSRF Bypass
    - Dual Session
    - Port Manipulation
    - Protocol Manipulation
- Why is this significant?
  - The role of JavaScript in modern web applications becomes greater as technologies such as AJAX, JS Frameworks and HTML5 become more common
  - It makes AppScan the first tool in the world capable of detecting a range of client-side security issues. These issues are thought to be very common but nobody knows for sure because no tools exist today that can find them
  - Additionally, it makes AppScan the first scanner that applies DAST and SAST in the same scan (hybrid analysis)

© 2011 IBM Corporation

Statistics Gathered with JSA on Fortune500+ Sites

- 14.5% of the sites were found to be infested with severe client-side JavaScript issues
- These types of issues could not be automatically located up until now – JSA is the first automated tool capable of locating client-side JavaScript issues
- 38% of the issues were found in 3rd party JavaScript code such as:
  - Marketing campaign JavaScript snippets
  - Flash embedding JavaScript snippets
  - Deep linking libraries for Flash and AJAX
  - Social networking JavaScript snippets

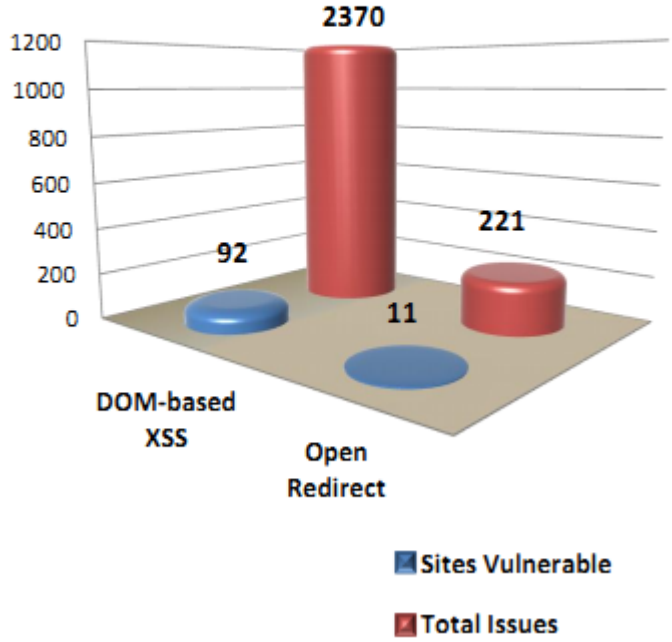




© 2011 IBM Corporation

Statistics Gathered with JSA on Fortune500+ Sites

- We have run JSA looking for either DOM-based XSS or Open Redirects
  - 92 had DOM-based XSS (94% of vulnerable sites)
  - 11 sites had Open Redirects (11% of vulnerable sites)
  - In total, we have found 2370 DOM-based XSS vs. 221 Open Redirects



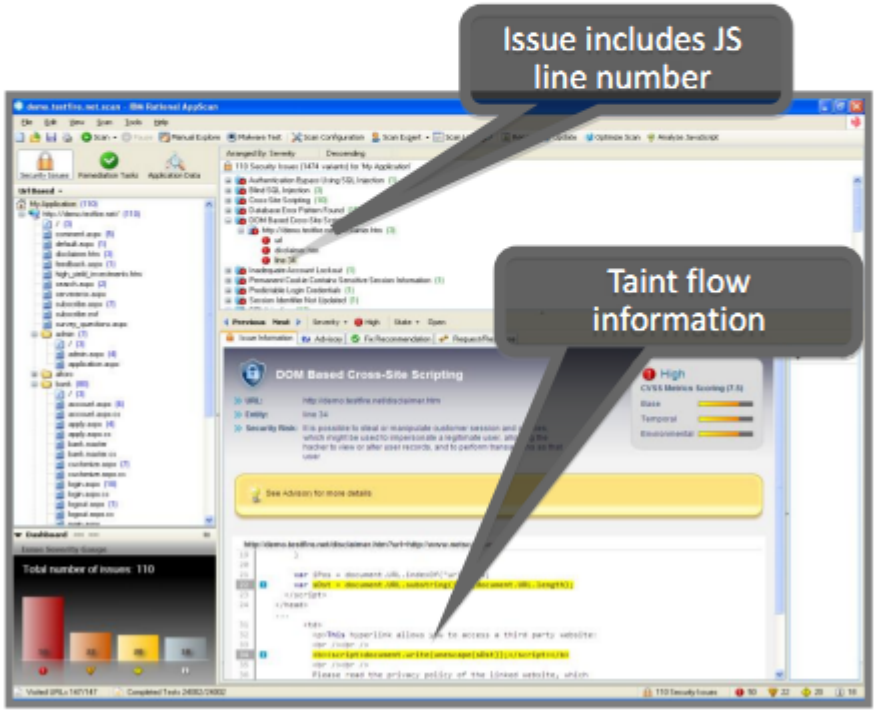
© 2011 IBM Corporation

Hybrid Analysis - JavaScript Security Analyzer (JSA)

- **What is JSA?**
  - An example of real Hybrid Analysis
    - DOM-based XSS, Code Injection, Open Redirect, CSRF Bypass, Dual Session, Port Manipulation, Protocol Manipulation

- **Why JSA?**
  - These types of issues could not be automatically located up until now – JSA is the first automated tool capable of locating client-side JavaScript issues

- **Unique Ability to AppScan**
  - First scanner that applies dynamic and static analysis in the same scan (hybrid analysis)
  - This makes AppScan the first tool in the world capable of detecting a range of client-side security issues



© 2011 IBM Corporation

Adopting and Evangelizing a Practical Secure Development Lifecycle

© 2011 IBM Corporation

A Practical Cycle Described

Design Phase

- Consideration is given to security requirements of the application
- Issues such as required controls and best practices are documented on par with functional requirements

Development Phase

- Software is checked during coding for:
  - Implementation error vulnerabilities
  - Compliance with security requirements

Build & Test Phase

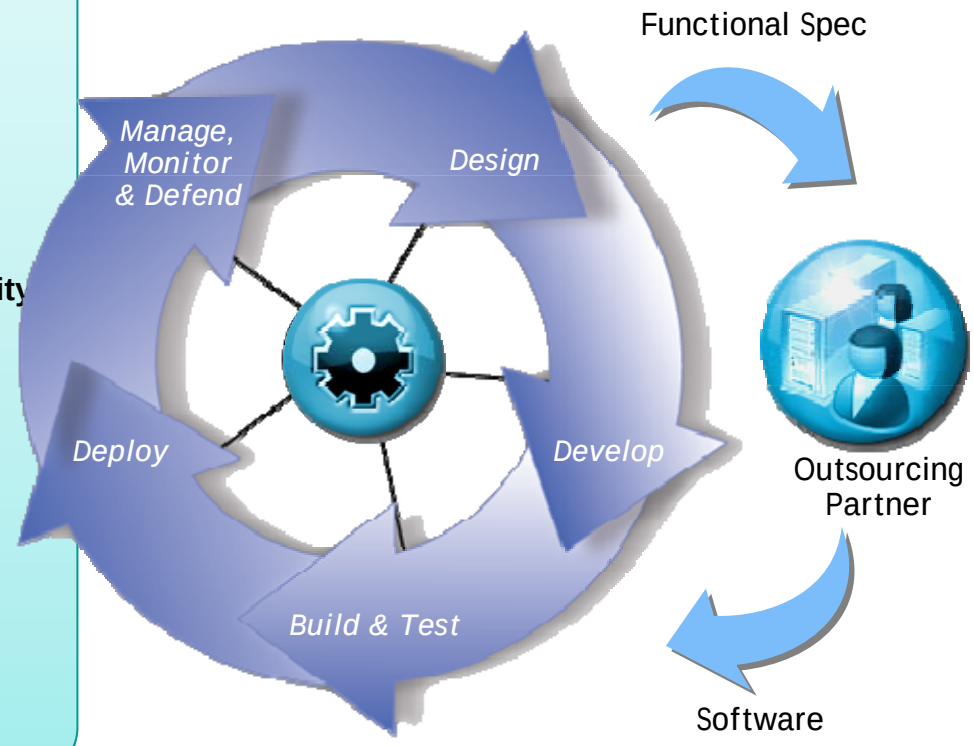
- Testing begins for errors and compliance with security requirements across the entire application
- Applications are also tested for exploitability in deployment scenario

Deployment Phase

- Configure infrastructure for application policies
- Deploy applications into production

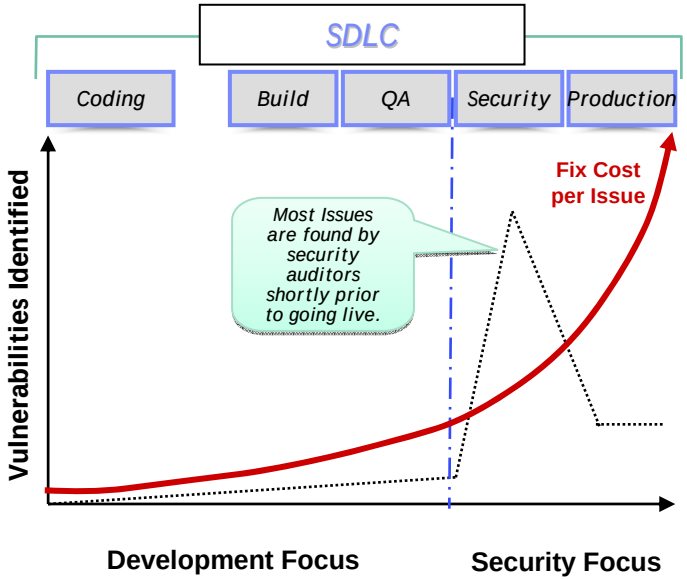
Operational Phase

- Continuously monitor applications for appropriate application usage, vulnerabilities and defend against attacks

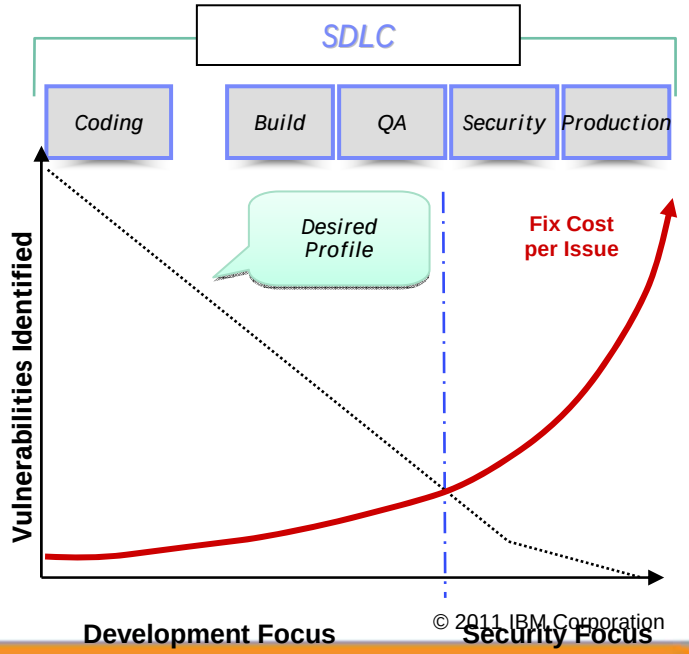


© 2011 IBM Corporation

Moving to a Desirable End State



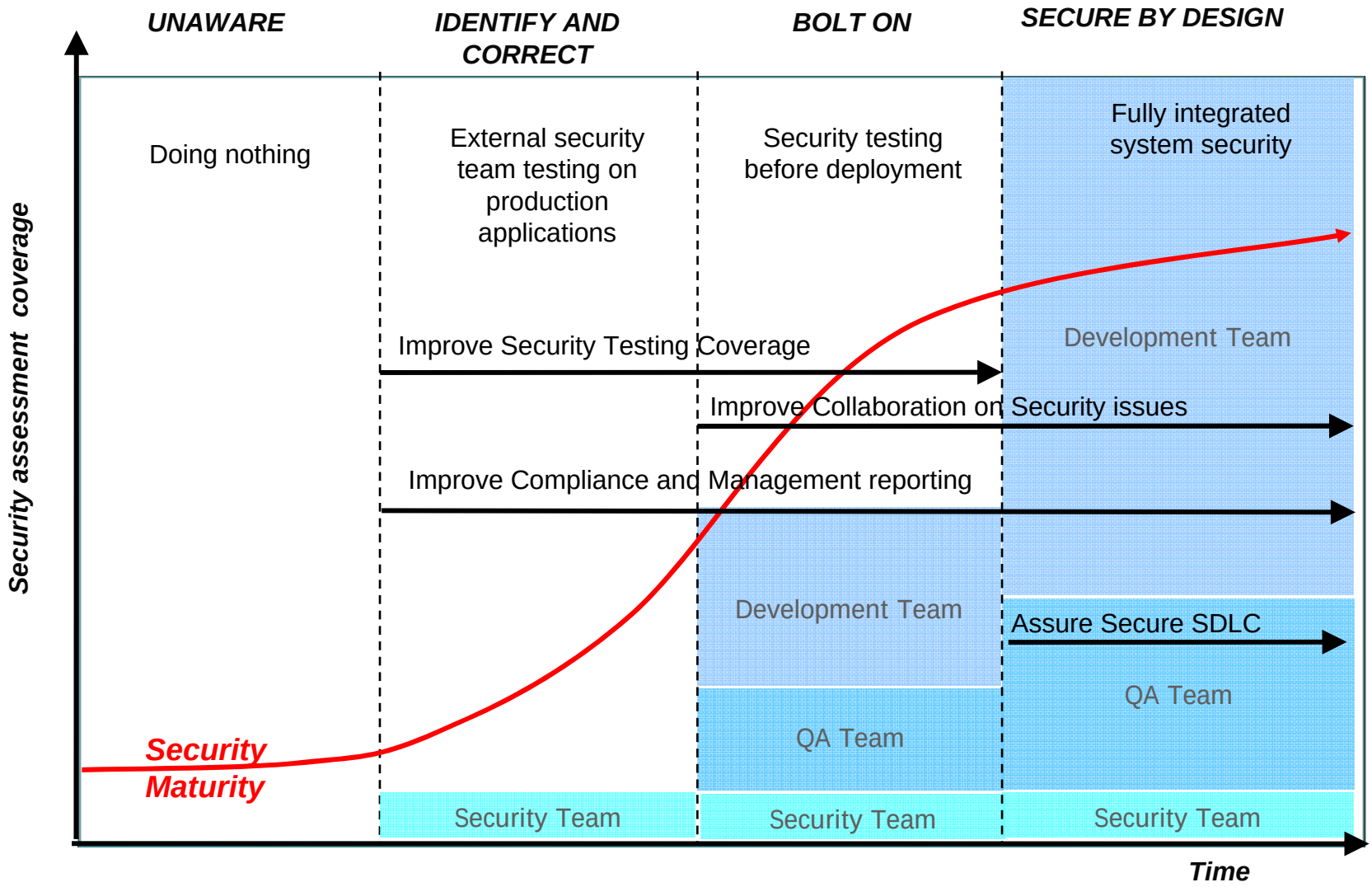
In early stages of adoption, security practitioners will assess applications during pre-deployment testing. **Costs are higher and window is shorter** to mitigate any issues found.



By integrating security into requirements, development and build/test/integration cycles, **identification occurs much earlier**, increasing find rate at a time when **fix costs are lowest**.

© 2011 IBM Corporation

Evolution of Software Security Issue Mitigation



© 2011 IBM Corporation

Qualities of a Successful Application Security Program

1. **There is a Marketing Plan (*Really*)**
2. **Assessment begins with an organization, not an application**
3. **Initial goals are modest introductions**
4. **Initial projects are carefully chosen**
   - Impact
   - Exposure
   - Acceptance
5. **Outcomes are demonstrable results**
   - Justify focus
   - Refine budgets and timelines
   - Coalesce key metrics and reporting
6. **Implementations focus on automation and integration**
   - Decreases disruption to existing processes
   - Increases value and leverage of existing investments in personnel and systems
7. **Process is consistent and repeatable to scale**

© 2011 IBM Corporation

Questions?

Steven Schmidt

schmidtty@au.ibm.com

© 2011 IBM Corporation

© 2011 IBM Corporation

Trademarks and disclaimers

© Copyright IBM Australia Limited 2011 ABN 79 000 024 733 © Copyright IBM Corporation 2011 All Rights Reserved.
TRADEMARKS: IBM, the IBM logos, ibm.com, Smarter Planet and the planet icon are trademarks of IBM Corp registered in many jurisdictions worldwide. Other company, product and services marks may be trademarks or services marks of others. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml

The customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list pricesand performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Some information addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Prices are suggested U.S. list prices and are subject to change without notice. Starting price may not include a hard drive, operating system or other features. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Photographs shown may be engineering prototypes. Changes may be incorporated in production models.

© 2011 IBM Corporation