IBM Innovate 2011
“Look Ma, No Hands!”
A Practical Guide to Automated Source Code Scanning

Rahul Pandey
QM – Specialist
Rational Software- IBM

IBM Software
Innovate2011
The Premier Event for Software and Systems Innovation

Software. Everyware.

Aug 9-11 Bangalore India

Please Note

IBM’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM’s sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Agenda

- A successful security initiative
- The security policy
- Is automation the answer for you
- Cost and coverage of automation
- Demo
- Q&A

Qualities of a Successful Application Security Program

1. There is a Marketing Plan (*Really*)
2. Security begins with an organization, not an application
3. Organizational structure is well understood
4. Initial projects are carefully chosen
   - Impact
   - Exposure
   - Acceptance
5. Outcomes are demonstrable
   - 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. Processes are consistent and repeatable to scale

Prerequisite: Craft a Starter App Sec Policy and Plan

- You decide: what is and is not acceptable in your production applications
- Consider application security as a whole: policy, organization, tools, resources and related support and training
- Limit the scope of the initial implementation, but imagine and plan for the complete portfolio
- Build capabilities through a sequence of steps that build upon each other
- Help communicate realistic expectations about how and when business benefits accrue

Phase I: Walk Before Run

- Select 3 or 4 vulnerability types to identify and correct

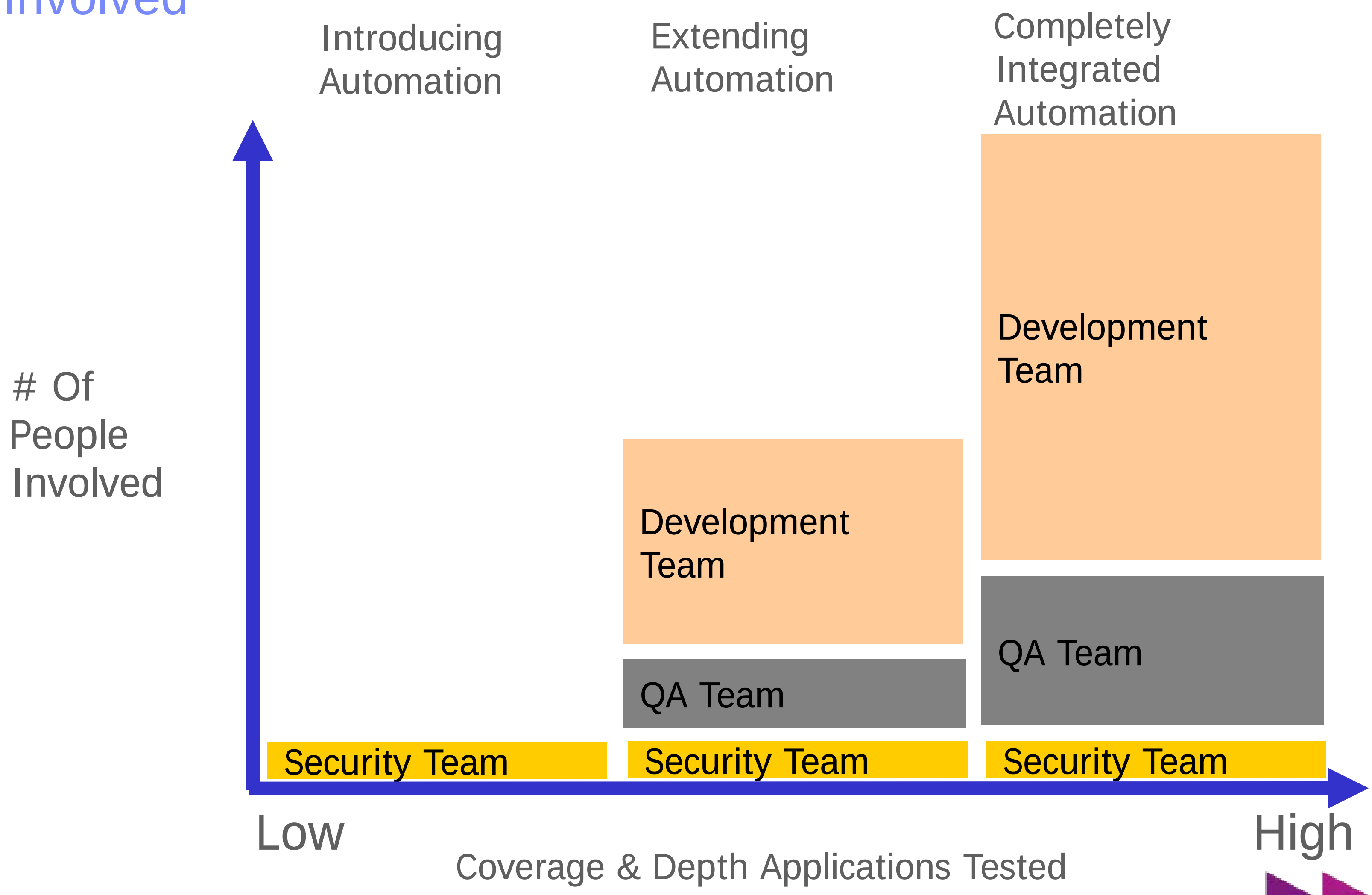
- Criteria
  - Easy to find (easy to hack)
  - Easy to fix
  - Ensure remediation assistance fits organizational requirements

- Candidates
  - Cross Site Scripting
  - SQL Injection
  - Command Injection
  - Authentication/Password problems (e.g. hardcoded, weak, clear text)

Phase II: Walk a Little Faster/Farther

- Once you’re underway and (somewhat) comfortable, consider expanding to the OWASP Top Ten or the SANS 25
  - OWASP
  - SANS 25
- 2010 list includes
  - 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: Insecure Cryptographic Storage
  - A8: Failure to Restrict URL Access
  - A9: Insufficient Transport Layer Protection
  - A10: Unvalidated Redirects and Forwards

Phase III: Scaling Testing / Gradually Getting More Resources Involved



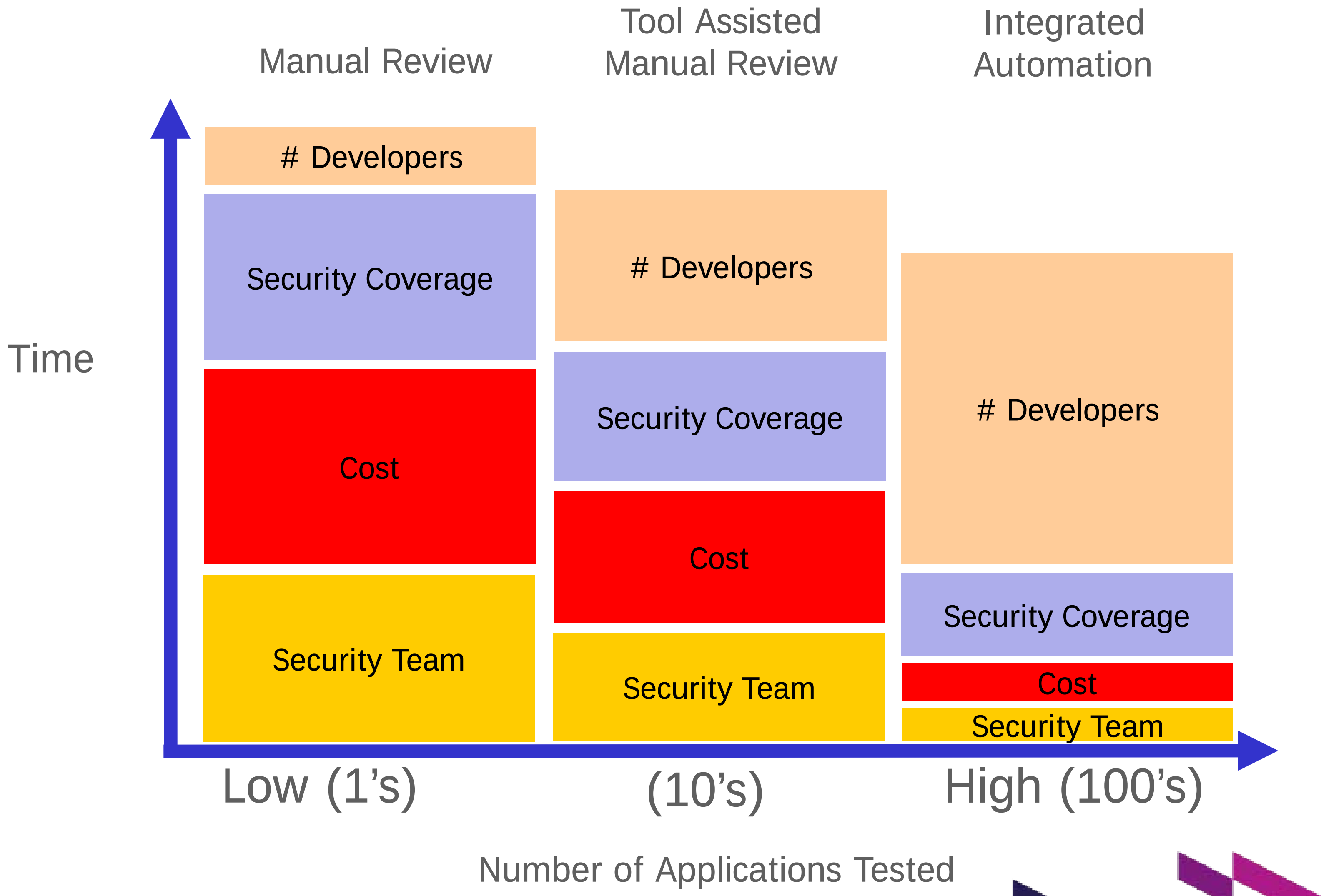
The Current State of Scanning at Many Organizations

- High degree of domain expertise required by Security and Development
  - Developers vs. Security vs. Scanning Tools
    - Need all three to get the job done
      - Scan configuration - language skill set for building
      - Deep product knowledge for triage workflow
      - Application security expertise to understand risk
- Poor data distribution and notification cycles
- Disruptive to development team
  - False positives creates low confidence in process
  - Integration of tools into development environment
- Roadblocks for the security team
  - Development environment is off limits to Security for build integration
  - Missing parts of the source code
  - Deep triage is too time consuming
- Bad assumptions and expectations
  - Identification of every security vulnerability

Goals of Automated Scanning

- Quick and easy configuration
- Scalable
- Produce high confidence results
- Provide automated notification of scan results to consumers
- Minimal disruptions for development and security
- Provide scan configuration confidence
- Ability to provide self service model

Manual Review vs Automation



Cost of Manual vs Automated Scanning

	Manual	Automated
Applications to assess	200	200
Dedicated security staff	2	1
Average application size	250 (KLoC)	250 (KLoC)
Average config & assessment time per application	40 (hours)	2 (hours)
Average remediation assessments per year	3	Unlimited
Average remediation assessment time per application	24 (hours)	0
Cost of security for 1 man year (fully loaded)	$200,000	$100,000
Total time for initial assessments	3.3 (years)	10 (weeks)
Total time for remediation assessments	3.5 (years)	0
Total LOE costs	$2,680,000	$19,230

Based on estimated labor costs only

What About Scanning on the Desktop?

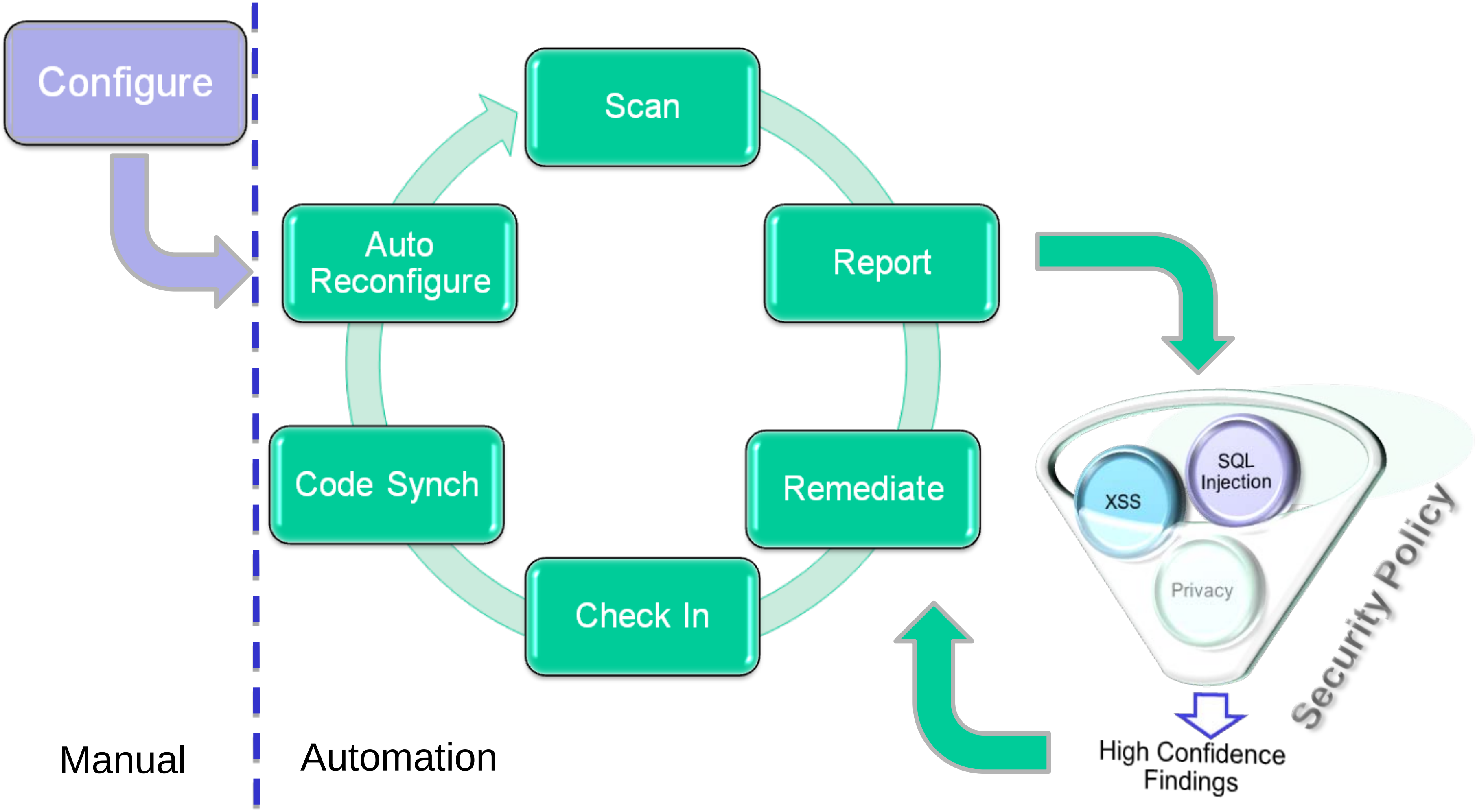
- Assume 200 Applications
  - Approximately 300 Developers
  - 20% are testing on any giving day. 60 developers per day run 1 scan
  - Average scan time: 30 Minutes

- 30 hours per day of limited productivity while developers are scanning

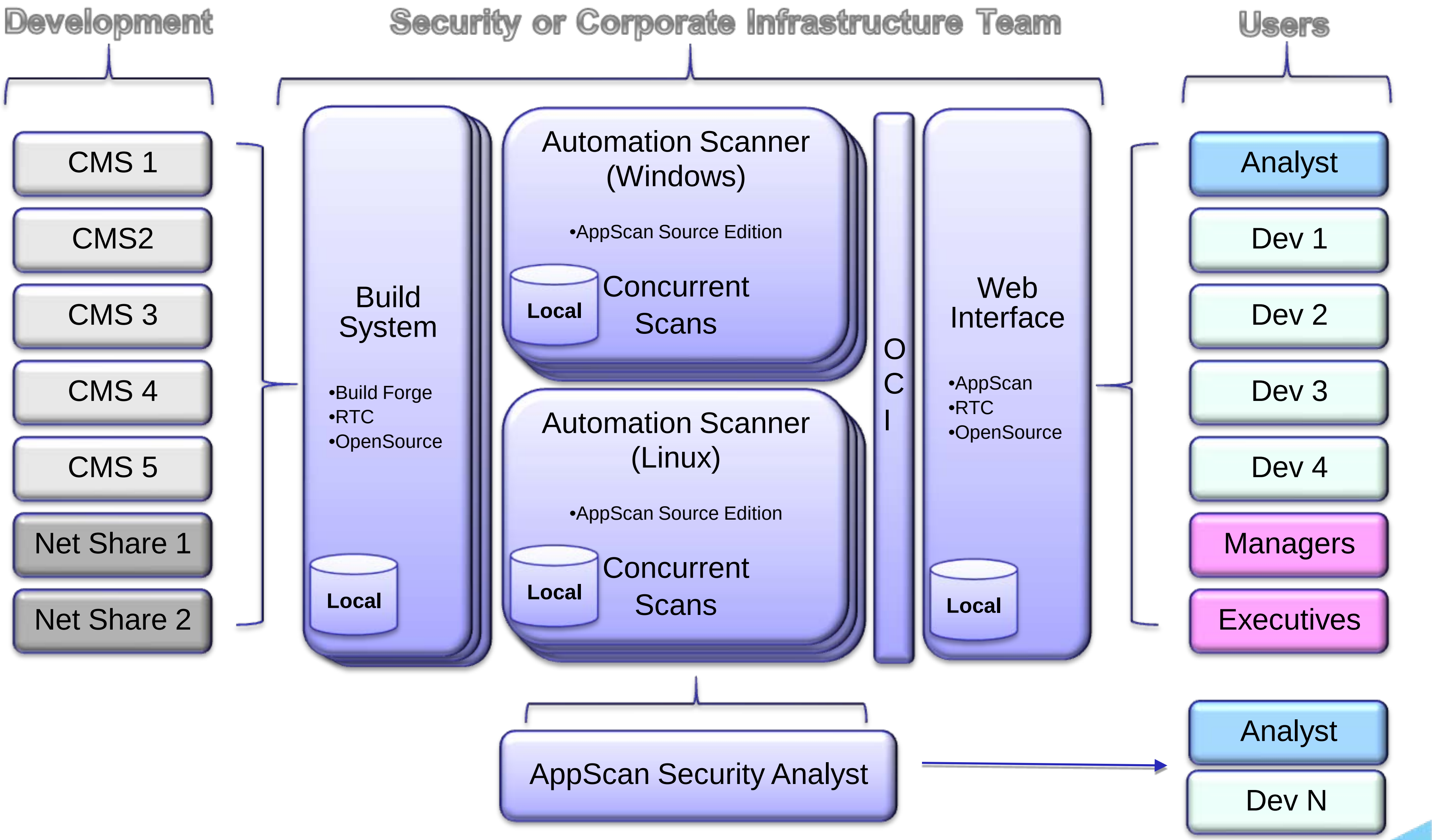
- Often don’t have ALL the pieces on the desktop

- Necessary in some environments
  - Should be incorporated into overall solution (champions)

Repeatable "Low Touch" High Confidence Findings



Scalable Solution



Development

Security or Corporate Infrastructure Team

Users

CMS 1

CMS2

CMS 3

CMS 4

CMS 5

Net Share 1

Net Share 2

Build System

- Build Forge
- RTC
- OpenSource

Local

Automation Scanner (Windows)

- AppScan Source Edition

Local

Concurrent Scans

Automation Scanner (Linux)

- AppScan Source Edition

Local

Concurrent Scans

OCI

Web Interface

- AppScan
- RTC
- OpenSource

Local

Analyst

Dev 1

Dev 2

Dev 3

Dev 4

Managers

Executives

AppScan Security Analyst

Analyst

Dev N

Automation Scanning Workflow

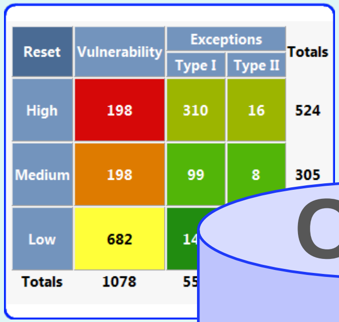
- Setup scan in build system
  - Assign repository
  - Assign notifications
  - Select filtering
- Configure application in AppScan Security client (if needed)
- Validate first scan (optional)
- Add scan to reporting console
- Assign rights to report data

- Press GO

IBM Rational AppScan Source Edition Solution

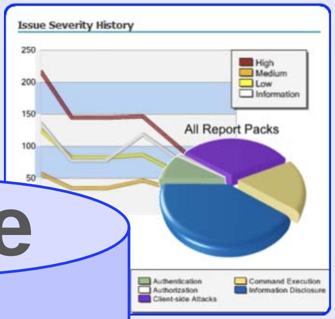
Security

- Configure Software
- Scan
- Triage Results
- Manage Security Policies



Reset	Vulnerability	Exceptions		Totals
		Type I	Type II	
High	198	310	16	524
Medium	198	99	8	305
Low	682	14		
Totals	1078	55		

Reporting Console



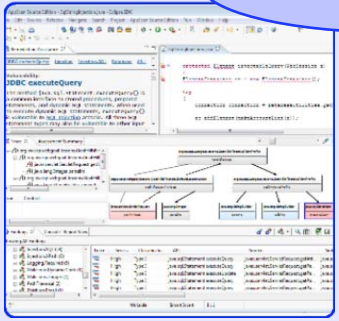
- Correlate BB/WW results
- Compare Applications
- Manage Portfolio Risk

Core

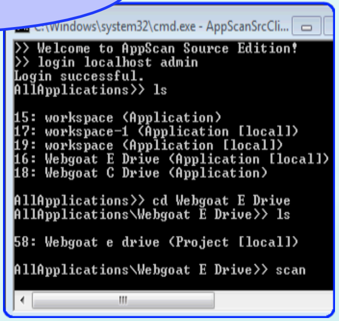
- Knowledgebase
- Assessment DB
- Custom Rules

Development

- Investigate Flaws
- Remediate with Guidance
- Scan
- Confirm Fix



Automation



```
>> Welcome to AppScan Source Edition!
>> login localhost admin
Login successful.
AllApplications>> ls

15: workspace (Application)
17: workspace-1 (Application [local])
19: workspace (Application [local])
16: Webgoat E Drive (Application [local])
18: Webgoat C Drive (Application)

AllApplications>> cd Webgoat E Drive
AllApplications\Webgoat E Drive>> ls

58: Webgoat e drive (Project [local])

AllApplications\Webgoat E Drive>> scan
```

- Build integration
- Automate Scans
- ANT, Make, Maven integration
- Data Access API

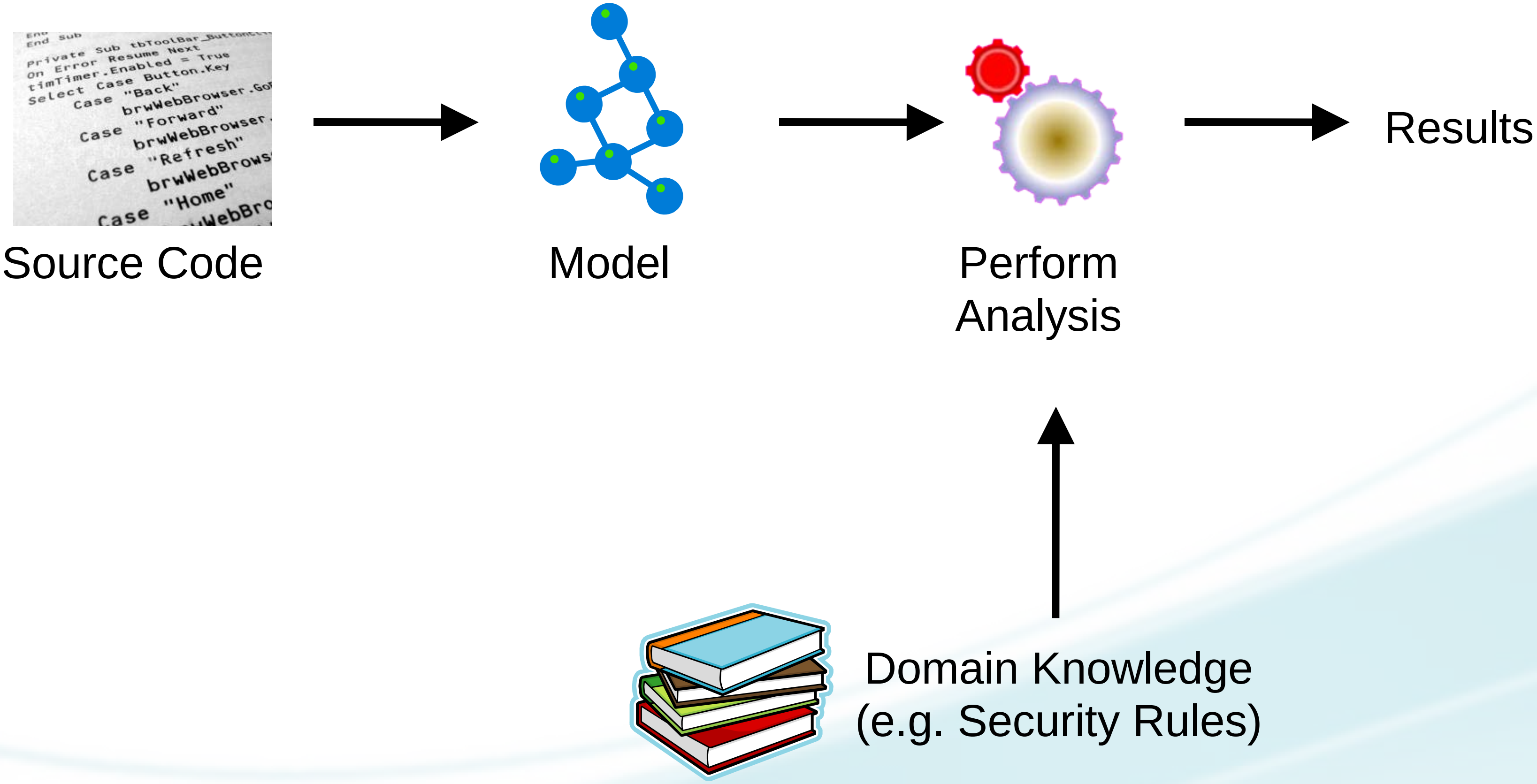
Broad Application Language Support

Out-of-the-Box

- Java
- JSP
- C
- C++
- .NET
  - C#
  - VB.NET
  - ASP.NET
- Classic ASP (VB6)
- COBOL
- PL/SQL
- T-SQL

- PHP
- HTML
- Perl
- ColdFusion
- Client-Side JavaScript
- Server-Side JavaScript
- VBScript

Static Analysis



```
End
End Sub

Private Sub tbToolBar_ButtonCl
On Error Resume Next
timTimer.Enabled = True
Select Case Button.Key
    Case "Back"
        brwWebBrowser.GoB
    Case "Forward"
        brwWebBrowser.
    Case "Refresh"
        brwWebBrows
    Case "Home"
        wWebBro
```

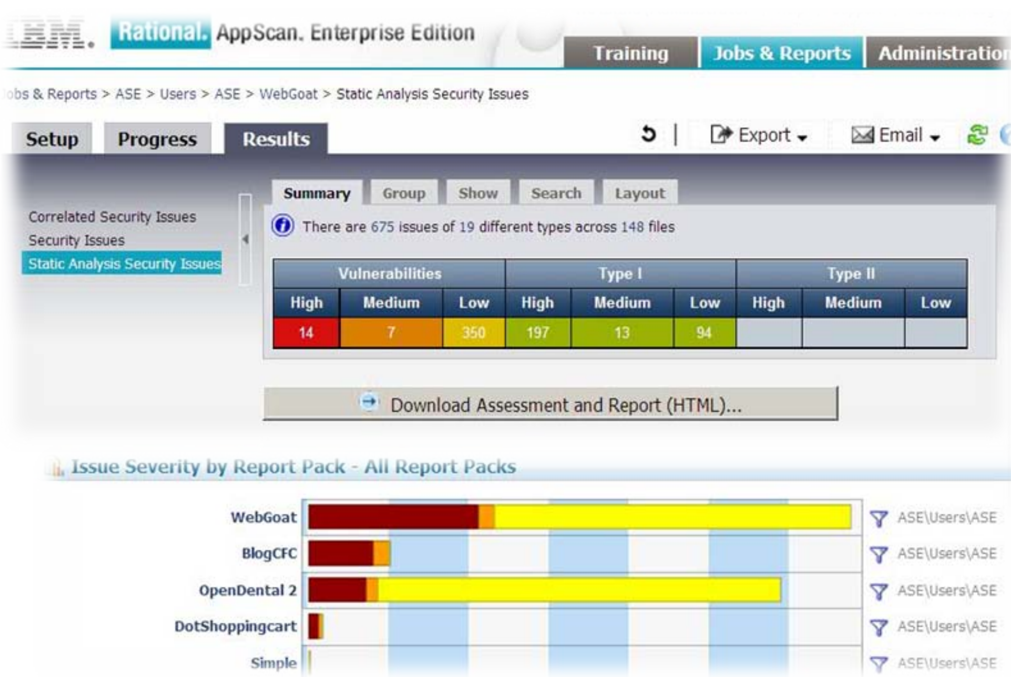
Source Code

Model

Perform Analysis

Results

Domain Knowledge
(e.g. Security Rules)

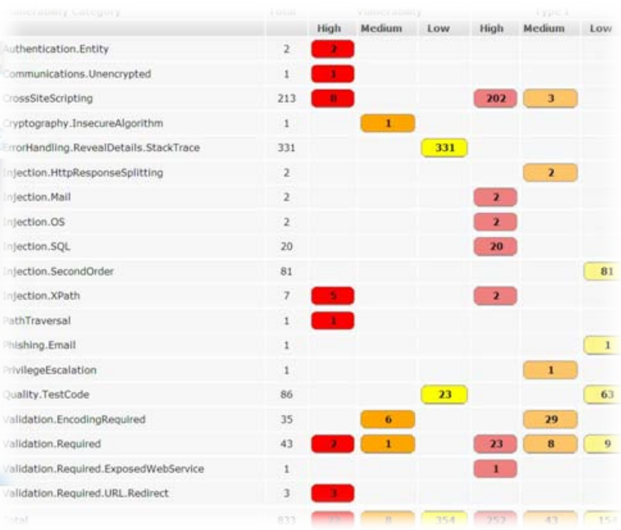


obs & Reports > ASE > Users > ASE > WebGoat > Static Analysis Security Issues

There are 675 issues of 19 different types across 148 files

Vulnerabilities			Type I			Type II		
High	Medium	Low	High	Medium	Low	High	Medium	Low
14	7	350	197	13	94			

Issue Severity by Report Pack - All Report Packs



Vulnerability Category	Total	Vulnerability			Type I		
		High	Medium	Low	High	Medium	Low
Authentication.Entity	2	2					
Communications.Unencrypted	1	1					
CrossSiteScripting	213	8			202	3	
Cryptography.InsecureAlgorithm	1		1				
ErrorHandling.RevealDetails.StackTrace	331			331			
Injection.HttpResponseSplitting	2					2	
Injection.Mail	2				2		
Injection.OS	2				2		
Injection.SQL	20				20		
Injection.SecondOrder	81						81
Injection.XPath	7	5			2		
PathTraversal	1	1					
Phishing.Email	1						1
PrivilegeEscalation	1					1	
Quality.TestCode	86			23			63
Validation.EncodingRequired	35		6			29	
Validation.Required	43	2	1		23	8	9
Validation.Required.ExposedWebService	1				1		
Validation.Required.URL.Redirect	3	3					
Total	833	22	8	354	252	43	154

Rational. Build Forge

Summary: Goals of Automated Scanning

- 2 hours per application for configuration
- High confidence reports
- Differential reporting
- Facilitate the distribution of scanning artifacts
- Automated notification of scan results to consumers
- Configuration confidence indicators for misconfigured scans
- Minimal disruptions for development and security
- Provide self serve scanning model

Benefits: Automated Scanning

- Scan large number of applications very quickly
- Scans initiated by schedule – no remediation “costs”
- Non-intrusive, low maintenance
- Immediate notifications of security issues to development staff
- Easy access to high confidence assessment data for developers
- Catches critical issues early in SDLC

- Quickly raises the overall security state of the enterprise

www.ibm.com/software/rational

www.ibm.com/software/rational