IBM Software

UKInnovate2010

The Rational Software Conference

Smarter software for a smarter planet.

IBM Software

UKInnovate2010

The Rational Software Conference

Quality Management Future Forward

John Smith, Paul Murray
and Hazel Woodcock

Smarter software for a smarter planet.

IBM Secure by Design

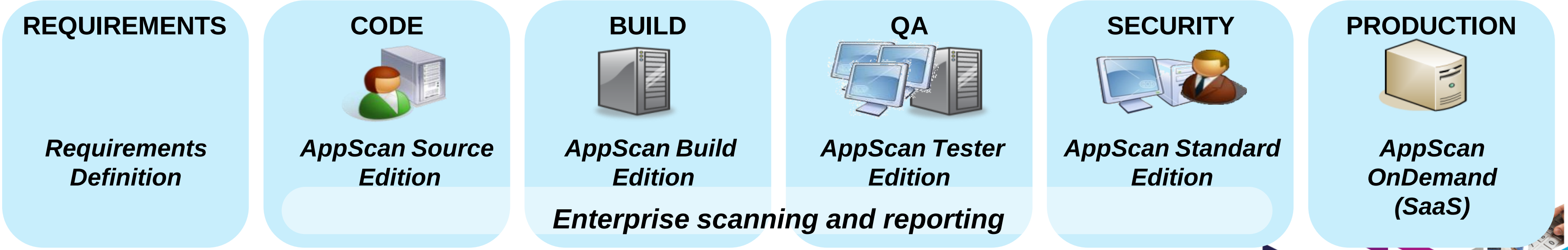
Embed security early in the development lifecycle

- Address **today's biggest threat** by efficiently identifying, triaging and remediating **application vulnerabilities** throughout the development lifecycle
- Experience **72% reduction in remediation costs** of application vulnerabilities by implementing pro-active, automated approach
- **Avoid repercussions from failed compliance audits** and breaches with consistent policies across organization

Deliver New Services Faster

Innovate Securely

Reduce Costs



REQUIREMENTS — *Requirements Definition*

CODE — *AppScan Source Edition*

BUILD — *AppScan Build Edition*

QA — *AppScan Tester Edition*

SECURITY — *AppScan Standard Edition*

PRODUCTION — *AppScan OnDemand (SaaS)*

Enterprise scanning and reporting

People - 2010 Rational Learning Roadmap

Take control of your learning
Virtually -when and where you want

Self-paced virtual classroom (SPVC):

- Essentials of IBM Rational AppScan Standard Edition V7.9
- Free trial! **http://tinyurl.com/ASCspvc**
- Get the high-quality content, hands-on lab experience, and instructor support of traditional classroom training, without the cost and hassles of travel

Web-based courses (WBT) available:

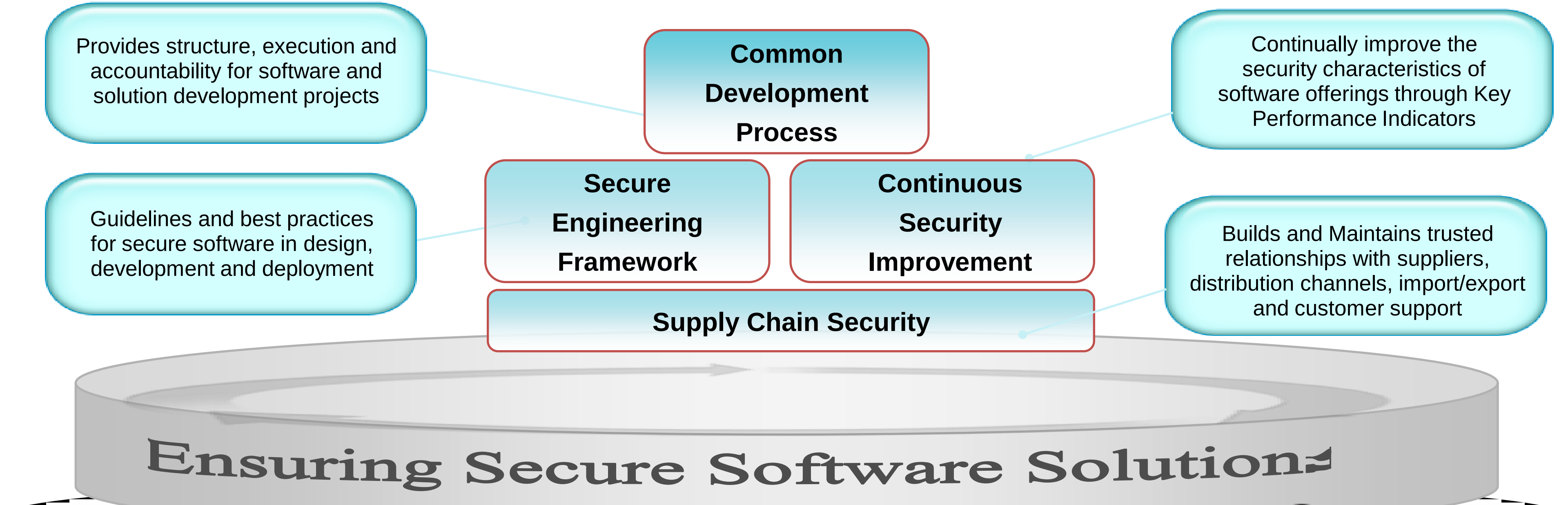
- IBM Rational AppScan Standard Edition
- IBM Rational AppScan Source Edition (Q3-10)
- IBM Rational Enterprise Edition
- IBM Rational Reporting Console
- IBM Rational Policy Tester
- Web Application Security
- Web Accessibility

http://tinyurl.com/ASCtraining
http://tinyurl.com/ASCspvc

Instructor-led courses (ILT) available:

- All WBT courses listed, plus in Q2-2010:
- Architecture Risk Analysis
- Attack and Defense
- Defensive Programming – C and C++
- Defensive Programming – C# in ASP.NET
- Defensive Programming – JavaEE
- Defensive Programming – VB.NET
- Foundations and Core Principles
- Risk-Based Security Testing Strategy
- Threat Modeling
- Web Security Testing

Process - IBM Secure Engineering Framework



Provides structure, execution and accountability for software and solution development projects

Guidelines and best practices for secure software in design, development and deployment

Common Development Process

Secure Engineering Framework

Continuous Security Improvement

Supply Chain Security

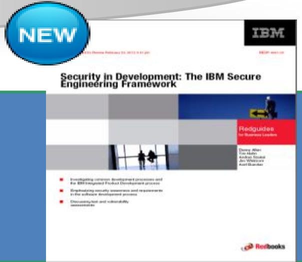
Continually improve the security characteristics of software offerings through Key Performance Indicators

Builds and Maintains trusted relationships with suppliers, distribution channels, import/export and customer support

Ensuring Secure Software Solutions

Link to Security Engineering Framework: http://www.redbooks.ibm.com/redpieces/abstracts/redp4641.html?Open

- IBM develops products and solutions for sale.
- IBM develops and operates solutions and services for its own internal use.
- IBM develops and operates solutions and services on behalf of customers.



NEW

IBM

Security in Development: The IBM Secure Engineering Framework

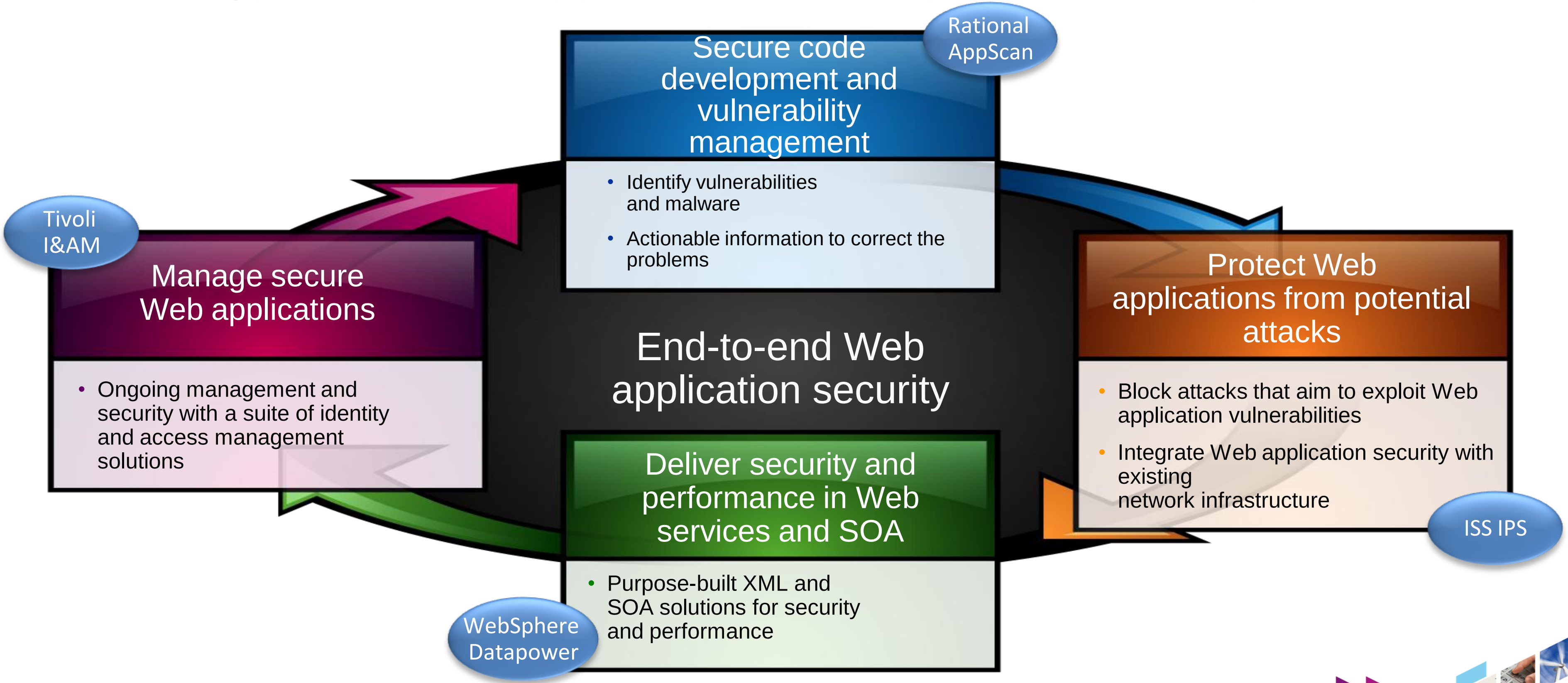
Redguides for Business Leaders

Danny Allan
Tim Hahn
Andras Szakal
Jim Whitmore
Axel Buecker

- Investigating common development processes and the IBM Integrated Product Development process
- Emphasizing security awareness and requirements in the software development process
- Discussing test and vulnerability assessments

Redbooks

Technology - IBM Web application security for a smarter planet



Rational AppScan

Secure code development and vulnerability management

- Identify vulnerabilities and malware
- Actionable information to correct the problems

Tivoli I&AM

Manage secure Web applications

- Ongoing management and security with a suite of identity and access management solutions

End-to-end Web application security

Protect Web applications from potential attacks

- Block attacks that aim to exploit Web application vulnerabilities
- Integrate Web application security with existing network infrastructure

ISS IPS

Deliver security and performance in Web services and SOA

- Purpose-built XML and SOA solutions for security and performance

WebSphere Datapower

A Path to Secure Applications

Operational Risk Mgmt

Proactive Risk Mitigation

Application & resource protection in operation

Secure application development across design, code, build, test phases

Web Application Protection
IBM Security

Identity & Access Management
IBM Security

Secure Web Services
WebSphere software

Production-Site Monitoring
Rational software

Deploy Application

Final Security Audit
Rational software

Vulnerability Assessment Functioning Application
Rational software

Vulnerability Assessment of Source Code
Rational software

Policy & Requirements Definition
Rational software
IBM Security Services

Risk Assessment
IBM Security Services

IBM Rational Investment in Application Security

Acquisitions:

- Watchfire acquisition 2007
- Ounce acquisition 2009

Global R&D Team

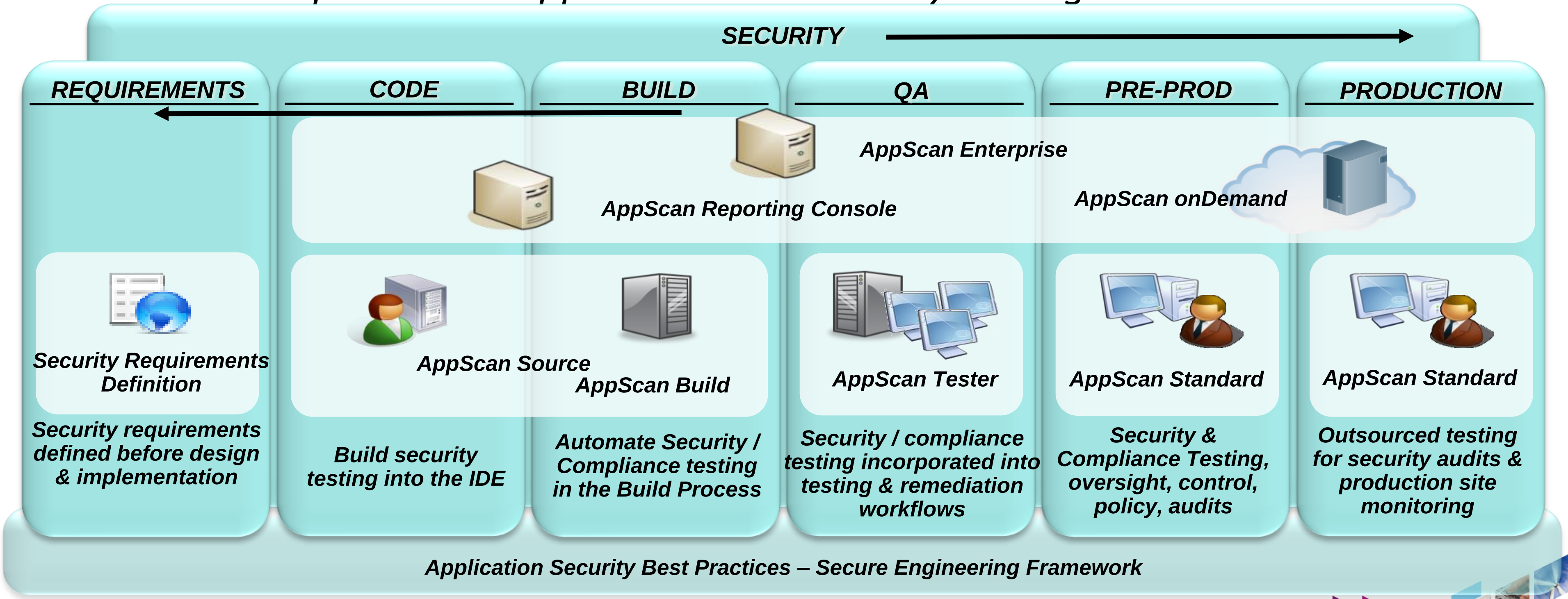
- Hawthorn NY research lab
- Tokyo research lab
- Ottawa development lab
- Toronto development lab
- Boston development lab
- Israel development lab

Gartner's take on the Ounce Labs Acquisition.

Of the major application development platform vendors, IBM made the first move to incorporate security testing into SLC with its acquisition of leading DAST tool vendor Watchfire (as well as a data-masking vendor, Princeton Softech) in 2007. **IBM now extends this leadership in 2009 with its acquisition of a leading SAST tool vendor, Ounce Labs.** SAST and DAST techniques are complementary and shouldn't have to come from separate vendors, and in the longer term they won't.

Furthermore, vendors have greater vision if they integrate static and dynamic testing to increase the breadth of application life cycle coverage and the accuracy of vulnerability detection, thus better serving enterprises' strategic security needs.

IBM Rational AppScan Suite – *Comprehensive Application Vulnerability Management*



SECURITY

REQUIREMENTS

CODE

BUILD

QA

PRE-PROD

PRODUCTION

AppScan Enterprise

AppScan Reporting Console

AppScan onDemand

Security Requirements Definition

Security requirements defined before design & implementation

AppScan Source

AppScan Build

Build security testing into the IDE

Automate Security / Compliance testing in the Build Process

AppScan Tester

Security / compliance testing incorporated into testing & remediation workflows

AppScan Standard

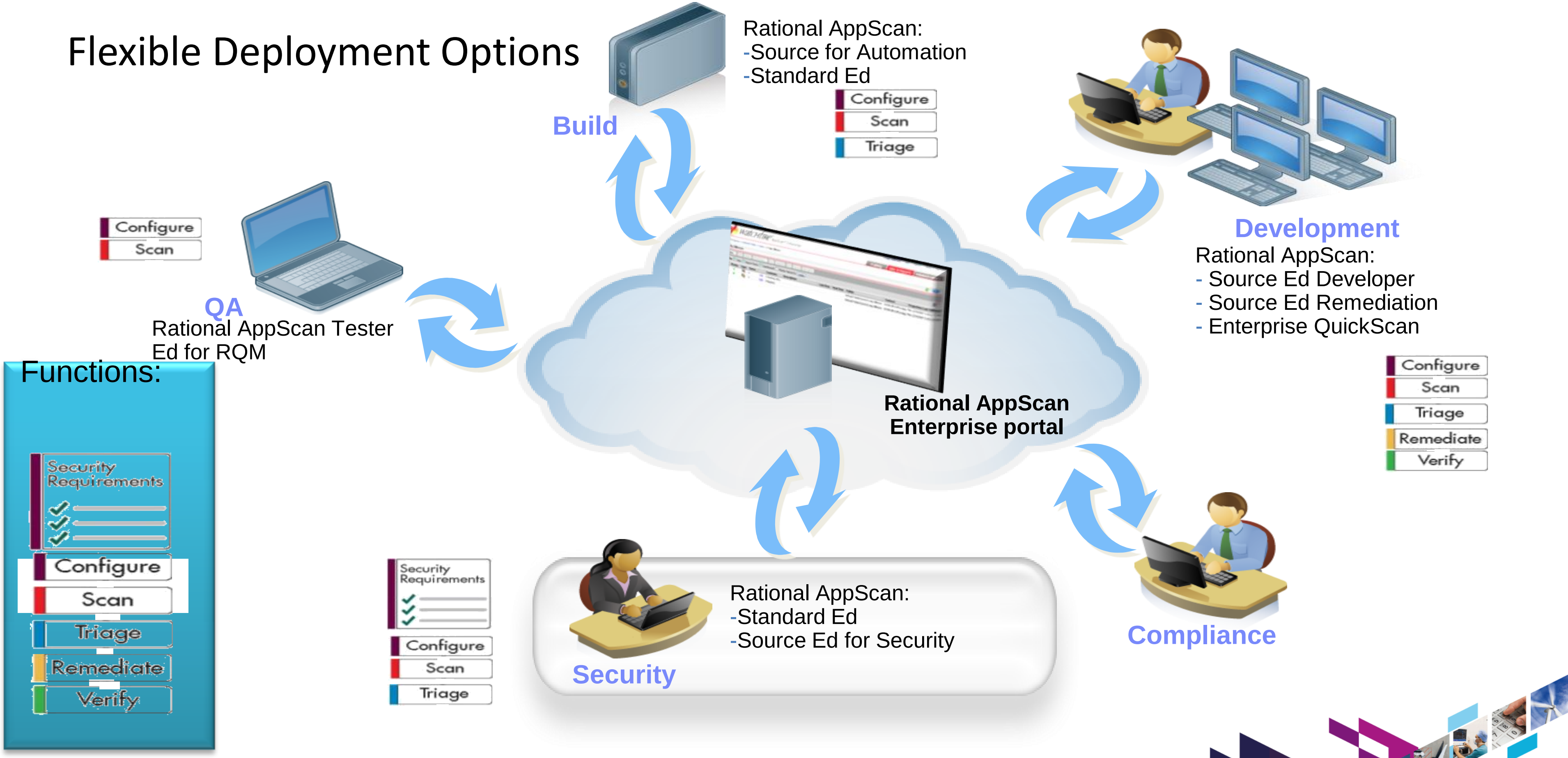
Security & Compliance Testing, oversight, control, policy, audits

AppScan Standard

Outsourced testing for security audits & production site monitoring

Application Security Best Practices – Secure Engineering Framework

Flexible Deployment Options



Build

Rational AppScan:
-Source for Automation
-Standard Ed

Configure
Scan
Triage

Development

Rational AppScan:
- Source Ed Developer
- Source Ed Remediation
- Enterprise QuickScan

Configure
Scan
Triage
Remediate
Verify

Configure
Scan

QA

Rational AppScan Tester Ed for RQM

Rational AppScan Enterprise portal

Functions:

Security Requirements
Configure
Scan
Triage
Remediate
Verify

Security Requirements
Configure
Scan
Triage

Security

Rational AppScan:
-Standard Ed
-Source Ed for Security

Compliance

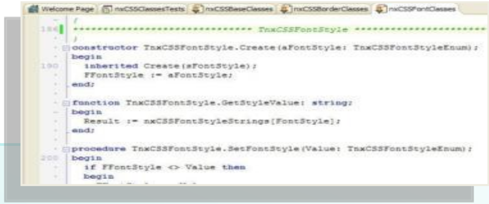
R&D Priorities

1. Integration of whitebox and blackbox technologies
2. Integration of security and quality code scanning
3. Expanding support for new platforms and languages
4. Security testing for developers

Security Testing Technologies... Combination of the Two Delivers Comprehensive Solution

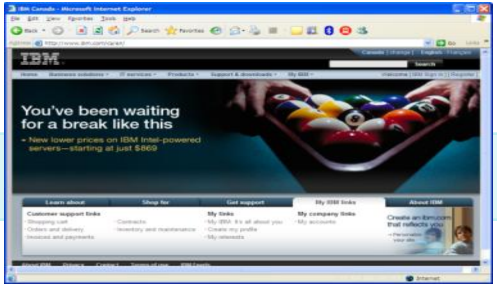
Static Code Analysis (Whitebox)

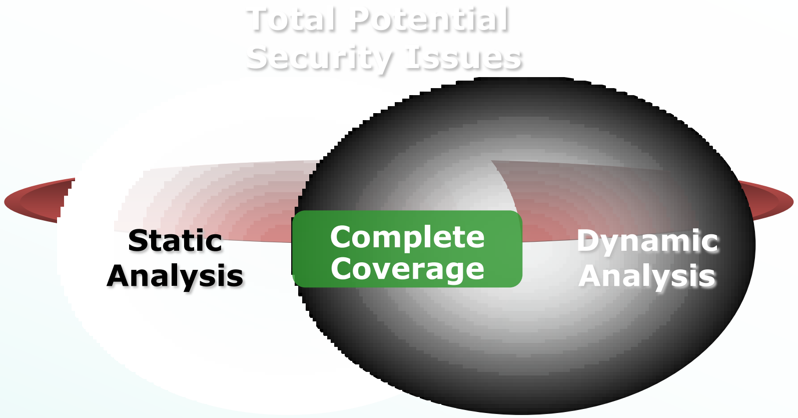
Scanning source code for security issues



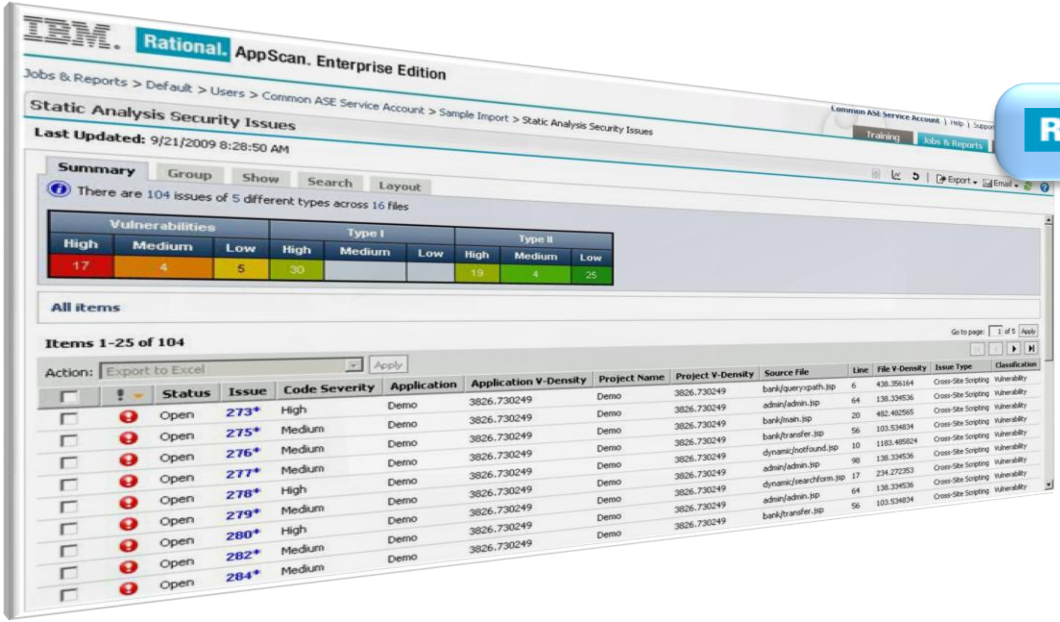
```
{
****************************** TnxCSSFontStyle ******************************
}
constructor TnxCSSFontStyle.Create(aFontStyle: TnxCSSFontStyleEnum);
begin
  inherited Create(aFontStyle);
  FFontStyle := aFontStyle;
end;

function TnxCSSFontStyle.GetStyleValue: string;
begin
  Result := nxCSSFontStyleStrings[FontStyle];
end;

procedure TnxCSSFontStyle.SetFontStyle(Value: TnxCSSFontStyleEnum);
begin
  if FFontStyle <> Value then
  begin
```

- **Dynamic Analysis (Blackbox)**
  - *Performing security analysis of a compiled application*


IBM Canada - Microsoft Internet Explorer
IBM
You've been waiting for a break like this
New lower prices on IBM Intel-powered servers—starting at just $869
Learn about
Shop for
Get support
My IBM links
About IBM



AppScan Source Edition Reporting


IBM. Rational. AppScan. Enterprise Edition
Common ASE Service Account | Help | Suppor
Training
Jobs & Reports
Export
Email

Jobs & Reports > Default > Users > Common ASE Service Account > Sample Import > Static Analysis Security Issues

Static Analysis Security Issues

Last Updated: 9/21/2009 8:28:50 AM

Summary | Group | Show | Search | Layout

There are 104 issues of 5 different types across 16 files

Vulnerabilities			Type I			Type II		
High	Medium	Low	High	Medium	Low	High	Medium	Low
17	4	5	30			19	4	25

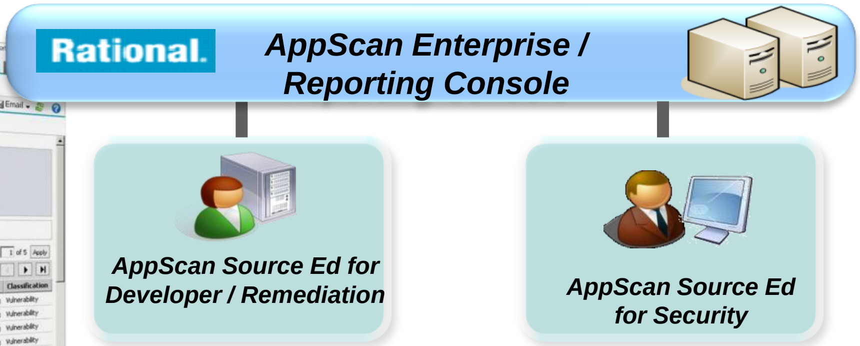
All items

Items 1-25 of 104

Go to page: 1 of 5 Apply

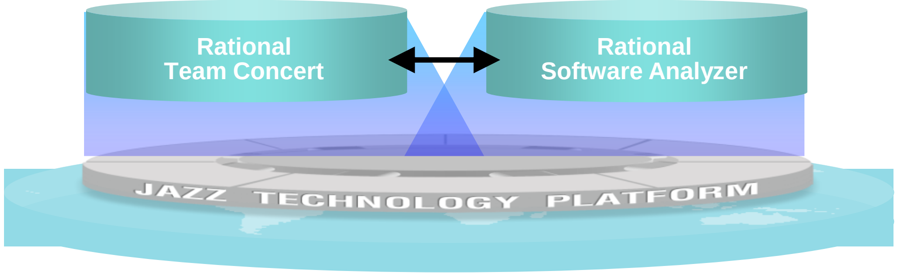
Action: Export to Excel Apply

□	!	Status	Issue	Code Severity	Application	Application V-Density	Project Name	Project V-Density	Source File	Line	File V-Density	Issue Type	Classification
□	●	Open	273*	High	Demo	3826.730249	Demo	3826.730249	bank/queryxpath.jsp	6	438.356164	Cross-Site Scripting	Vulnerability
□	●	Open	275*	Medium	Demo	3826.730249	Demo	3826.730249	admin/admin.jsp	64	138.334536	Cross-Site Scripting	Vulnerability
□	●	Open	276*	Medium	Demo	3826.730249	Demo	3826.730249	bank/main.jsp	20	482.482565	Cross-Site Scripting	Vulnerability
□	●	Open	277*	Medium	Demo	3826.730249	Demo	3826.730249	bank/transfer.jsp	56	103.534834	Cross-Site Scripting	Vulnerability
□	●	Open	278*	High	Demo	3826.730249	Demo	3826.730249	dynamic/notfound.jsp	10	1183.489824	Cross-Site Scripting	Vulnerability
□	●	Open	279*	Medium	Demo	3826.730249	Demo	3826.730249	admin/admin.jsp	98	138.334536	Cross-Site Scripting	Vulnerability
□	●	Open	280*	High	Demo	3826.730249	Demo	3826.730249	dynamic/searchform.jsp	17	234.272353	Cross-Site Scripting	Vulnerability
□	●	Open	282*	Medium	Demo	3826.730249	Demo	3826.730249	admin/admin.jsp	64	138.334536	Cross-Site Scripting	Vulnerability
□	●	Open	284*	Medium	Demo	3826.730249	Demo	3826.730249	bank/transfer.jsp	56	103.534834	Cross-Site Scripting	Vulnerability
□													


Rational.
AppScan Enterprise / Reporting Console
AppScan Source Ed for Developer / Remediation
AppScan Source Ed for Security

Rational Software Analyzer - Automated Code Quality

Leveraging Rational Software Analyzer and Rational Team Concert



Implement Code Quality Governance directly into the development stream

Developers must run configured rule sets in Software Analyzer before checking into repository

Measure team performance against best practice metrics

Utilize Team Concert to view adherence through quality review reporting in Software Analyzer

Improved productivity through reduced re-work and maximized reuse of code

Automated code quality reviews throughout the development lifecycle improve code quality best practices

Expanding Support for New Platforms and Languages

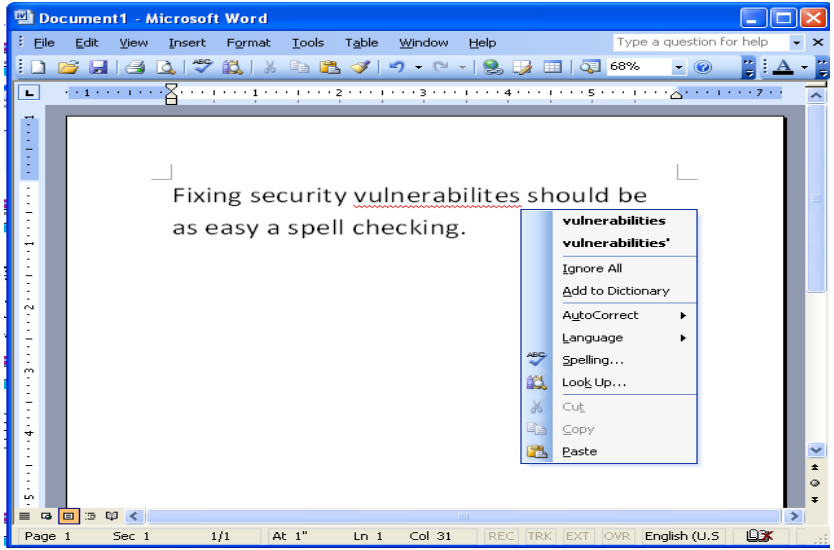
Blackbox

- Stronger JavaScript and AJAX support
- Flash Action Script 3

Whitebox

- PHP, JavaScript, Cold Fusion, Perl
- Cobol
- SAP/ABAP

Vision



Document1 - Microsoft Word

Fixing security vulnerabilites should be as easy a spell checking.

Our vision is for secure software delivery to become an intrinsic property of the developer’s IDE and development environment. Much like spell checking highlights errors as you are writing a document, security checking should highlight errors at the earliest point in the SDLC, when you are writing the code itself.

Technology Challenges

1. Accuracy

- For many types of injection issues the tool needs to understand that not only is sanitization and validation is occurring, but that the sanitization and validation logic is actually correct. This currently requires the developer to not only manually configure the tool to understand where validation or sanitization is occurring, but also to manually inspect the validation and sanitization routines to ensure they are correct. This approach can be very time consuming, error prone, and in the worst case introduce false negatives.

2. Partial vs. full analysis

- The “Ping Pong” effect
  - Developer only works on small part of code but tools require them to analyze everything (so instead of delivering functionality they are playing ping-pong as the scan is running).
- Playing “Which Security Bug is Mine” game

Developers are given a report for the entire application and are forced to weed through a long list of issues to find which bugs are relevant to them.

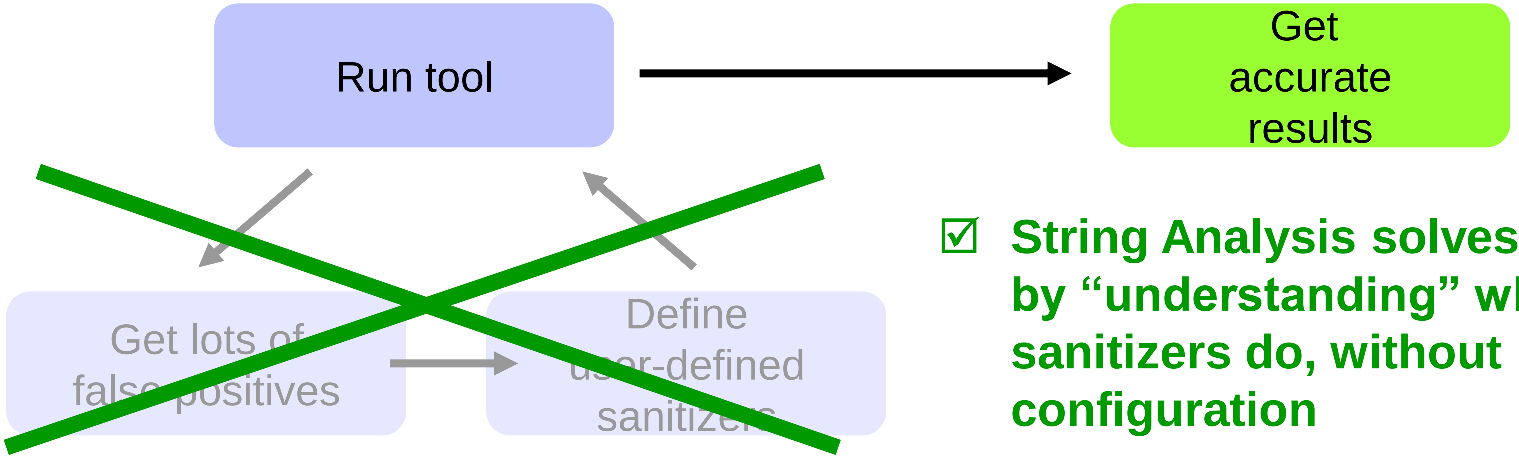
3. AutoCorrect

- Nothing like this exists. Developer needs to understand problem and manually fix it with no level of confidence that not only is the fix correct, but is using an approved mitigation strategy within the software security guidelines of the business.

IBM R&D Projects to Address These Road Blocks

Problem	Solution
Accuracy	String analysis technology
Partial vs. full analysis	Incremental analysis technology
AutoCorrect	Autofix technology

Using Static Analysis in the Real World



Run tool

Get accurate results

~~Get lots of false positives~~

~~Define user-defined sanitizers~~

☑ **String Analysis solves this by "understanding" what sanitizers do, without configuration**

Top complaints from users of static analysis tools:

#1: Lots of false positives

#2: Configuration of sanitizers is time consuming

String Analysis Summary

IBM’s next-gen static analysis technology

World’s smartest static analyzer

✓ No need to define what the sanitizers are

✓ Understands inline sanitization

✓ Understands validators

✓ Verifies your sanitizers really do what they’re supposed to

What this means for you

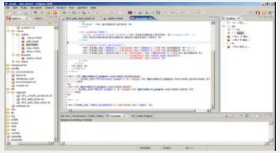
Greater accuracy out-of-the-box

Less configuration

More reliable results

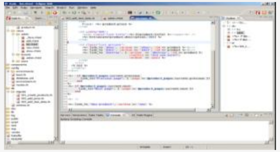
Easier to use

Incremental Analysis Enables Day-to-day Use by Developers



See security issues in IDE

Fix issue, save



List of issue updates within 1-2 seconds

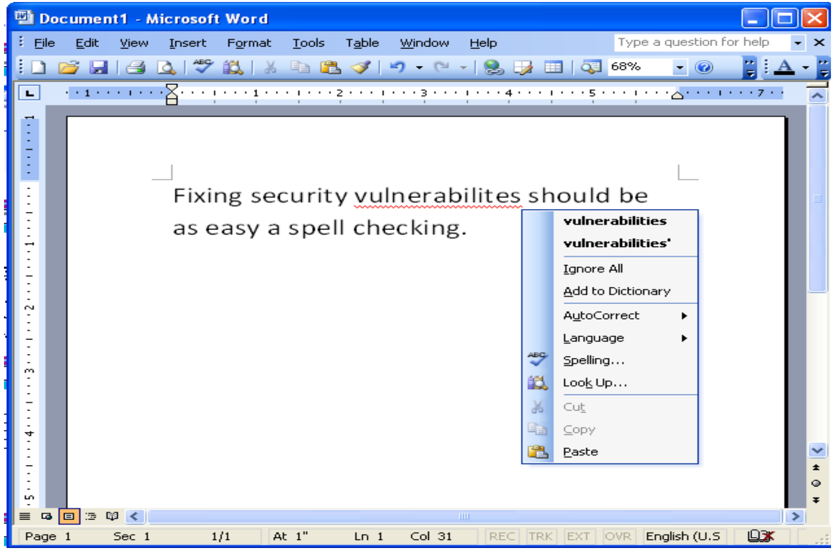
including deep data-flow analysis

Negates the need to scan all code for every analysis

Incremental analysis allows static analysis to be performed in-line as the developer is writing code as only portions of the code are modified

This also ensures that the developer that introduces a security issue is notified at the earliest point in the development process possible and at the best time to include a fix.

Auto-fix



Fixing security vulnerabilites should be as easy a spell checking.

Document1 - Microsoft Word
File Edit View Insert Format Tools Table Window Help
Type a question for help
68%
vulnerabilities
vulnerabilities'
Ignore All
Add to Dictionary
AutoCorrect
Language
Spelling...
Look Up...
Cut
Copy
Paste
Page 1 Sec 1 1/1 At 1" Ln 1 Col 31 REC TRK EXT OVR English (U.S

Auto-fix

- One-click to apply automatic, complex code-transformations for fixing security issues
- User can review code changes before accepting
- Fix can be immediately verified using incremental analysis

▪**Product benefits:**

- ✓ Ease of use
- ✓ Reduces need for security training
- ✓ Fix code the right way
- ✓ Immediately verifiable

Rational Security & Compliance Customer Experience Program

The Customer Experience Program is the way to…

- validate that your long term needs match with IBM's long term plans
- ensure your key requirements are communicated to the IBM development teams
- influence the plans and designs of the next release of the AppScan portfolio
- demo early drivers of the next release of the AppScan portfolio
- fully engage in the development process

A single customer feedback program spanning the entire Security & Compliance portfolio allowing IBM and our valued customers to interact any time, in any way, & on any topic!

For more information, or to join the program, contact Rick Goldberg – CEP Program Manager <rickmg@ca.ibm.com>

Summary

Application security continues to grow in importance

- Interconnected , smarter planet will drive need for stronger security

Consider a phased approach to addressing application security

- Create multiple security checkpoints throughout your development process

Security by Design is the answer to creating secure applications

- People, Process and Technology

IBM has a comprehensive solution for application security

- X – IBM solutions for pre and post deployment
- Integrating testing solutions for all phases of SDLC

IBM Software

UKInnovate2010

The Rational Software Conference

TMM*i*

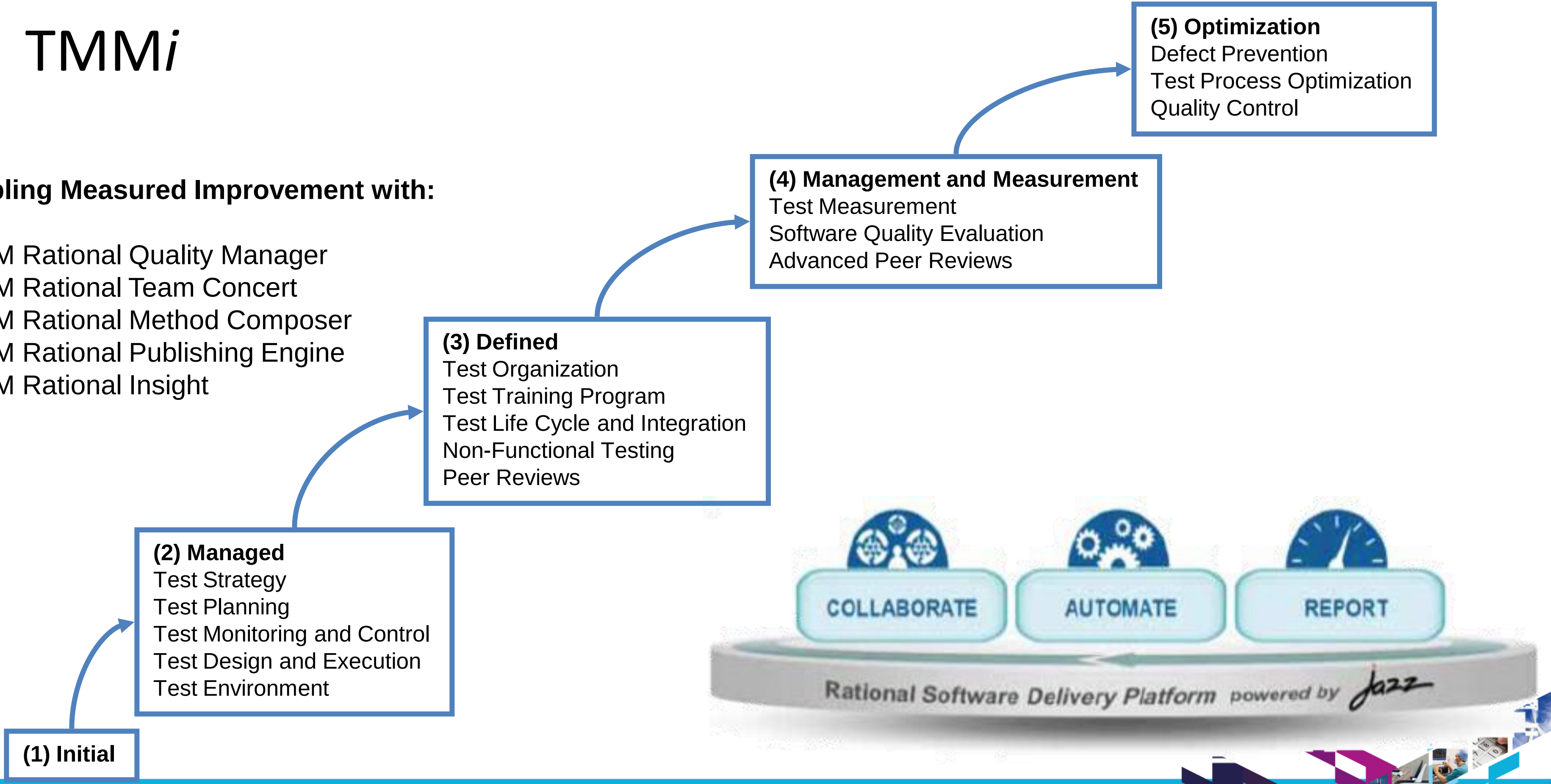
Hazel Woodcock

Smarter software for a smarter planet.

TMM*i*

Enabling Measured Improvement with:

- ✓ IBM Rational Quality Manager
- ✓ IBM Rational Team Concert
- ✓ IBM Rational Method Composer
- ✓ IBM Rational Publishing Engine
- ✓ IBM Rational Insight



(1) Initial

(2) Managed
Test Strategy
Test Planning
Test Monitoring and Control
Test Design and Execution
Test Environment

(3) Defined
Test Organization
Test Training Program
Test Life Cycle and Integration
Non-Functional Testing
Peer Reviews

(4) Management and Measurement
Test Measurement
Software Quality Evaluation
Advanced Peer Reviews

(5) Optimization
Defect Prevention
Test Process Optimization
Quality Control

PA 2.1 Test Policy and Strategy	Rational Quality Manager	Rational Method Composer	Rational Team Concert	Rational Publishing Engine	Rational Insight
SG 1 Establish a Test Policy	🟦 ✓	🟩 ✓	✓		
SG 2 Establish a Test Strategy	🟦 ✓	🟩 ✓	✓		
SG 3 Establish Test Performance Indicators	🟦 ✓	🟩 ✓			

🟩 ✓ Primary Solution

🟦 ✓ Supporting Solution

PA 2.2 Test Planning	Rational Quality Manager	Rational Method Composer	Rational Team Concert	Rational Publishing Engine	Rational Insight
SG 1 Perform Product Risk Assessment	🟩 ✓				
SG 2 Establish a Test Approach	🟩 ✓				
SG 3 Establish Test Estimates	🟩 ✓				
SG 4 Develop a Test Plan	🟩 ✓				
SG 5 Obtain Commitment to the Test Plan	🟩 ✓		🟦 ✓		

PA 2.3 Test Monitoring and Control	Rational Quality Manager	Rational Method Composer	Rational Team Concert	Rational Publishing Engine	Rational Insight
SG 1 Monitor Test Progress against Plan	🟦 ✓	🟦 ✓	🟦 ✓		🟩 ✓
SG 2 Monitor Product Quality against Plan and Expectations	🟦 ✓				🟩 ✓
SG 3 Manage Corrective Actions to Closure			🟦 ✓	🟦 ✓	🟩 ✓

PA 2.4 Test Design and Execution	Rational Quality Manager	Rational Method Composer	Rational Team Concert	Rational Publishing Engine	Rational Insight
SG 1 Perform Test Analysis and Design Using Test Design Techniques	🟩 ✓	🟦 ✓			
SG 2 Perform Test Implementation	🟩 ✓				
SG 3 Perform Test Execution	🟩 ✓		🟦 ✓		
SG 4 Manage Test Incidents to Closure			🟦 ✓	🟦 ✓	

PA 2.5 Test Environment	Rational Quality Manager	Rational Method Composer	Rational Team Concert	Rational Publishing Engine	Rational Insight
SG 1 Develop Test Environment Requirements	🟩 ✓				
SG 2 Perform Test Environment Implementation	🟩 ✓				
SG 3 Manage and Control Test Environments	🟩 ✓		🟦 ✓		

🟩 ✓	Primary Solution
🟦 ✓	Supporting Solution

PA 3.1 Test Organization	Rational Quality Manager	Rational Method Composer	Rational Team Concert	Rational Publishing Engine	Rational Insight
SG 1 Establish a Test Organization		🟩✓			
SG 2 Establish Test Functions For Test Specialists		🟩✓			
SG 3 Establish Test Career Paths		🟩✓			
SG 4 Determine, Plan and Implement Test Process Improvements		🟩✓	🟦✓		
SG 5 Deploy Organizational Test Processes and Incorporate Lessons Learned	🟦✓	🟩✓	🟦✓		

PA 3.2 Test Training Program	Rational Quality Manager	Rational Method Composer	Rational Team Concert	Rational Publishing Engine	Rational Insight
SG 1 Establish an Organizational Test Training Capability		🟩✓			
SG 2 Provide Necessary Test Training		🟩✓			

🟩✓ Primary Solution

🟦✓ Supporting Solution

PA 3.3 Test Lifecycle and Integration	Rational Quality Manager	Rational Method Composer	Rational Team Concert	Rational Publishing Engine	Rational Insight
SG 1 Establish Organizational Test Process Assets	🟦 ✓	🟩 ✓			
SG 2 Integrate the Test Life Cycle with the Development Models		🟩 ✓	🟦 ✓		
SG 3 Establish A Master Test Plan	🟦 ✓	🟩 ✓			

PA 3.4 Non-Functional Testing	Rational Quality Manager	Rational Method Composer	Rational Team Concert	Rational Publishing Engine	Rational Insight
SG 1 Perform a Non-Functional Product Risk Assessment	🟩 ✓				
SG 2 Establish a Non-Functional Test Approach	🟩 ✓	🟦 ✓			
SG 3 Perform Non-Functional Test Analysis and Design	🟩 ✓				
SG 4 Perform Non-Functional Test Implementation	🟩 ✓				
SG 5 Perform Non-Functional Test Execution	🟩 ✓				

PA 3.5 Peer Reviews	Rational Quality Manager	Rational Method Composer	Rational Team Concert	Rational Publishing Engine	Rational Insight
SG 1 Establish a Peer Review Approach	🟩 ✓	🟦 ✓	🟦 ✓	🟦 ✓	
SG 2 Perform Peer Reviews	🟩 ✓		🟦 ✓	🟦 ✓	

🟩 ✓ Primary Solution

🟦 ✓ Supporting Solution

IBM Software

UKInnovate2010

The Rational Software Conference

Smarter software for a smarter planet.