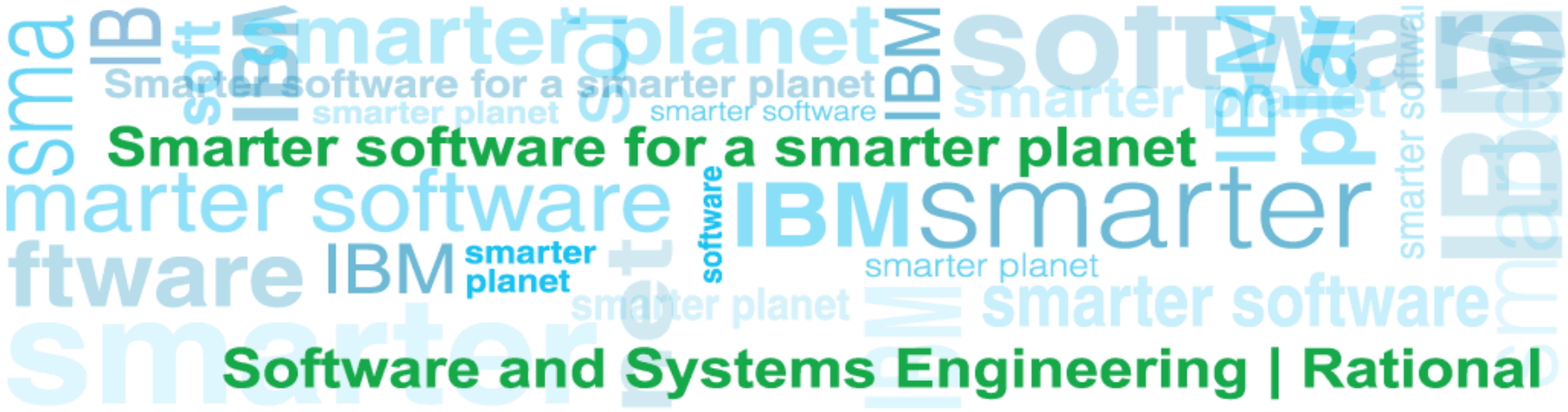
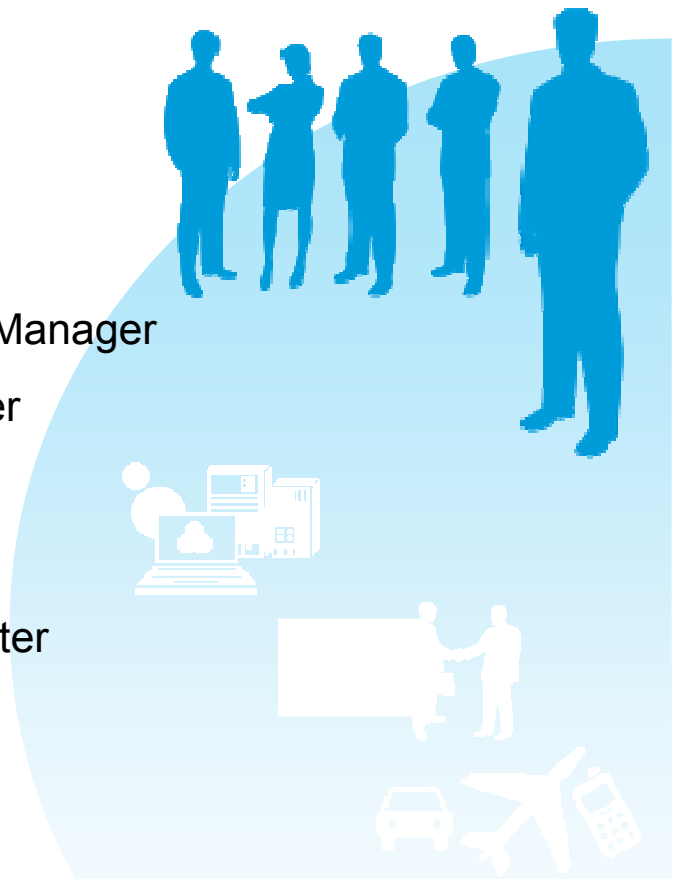
Testing and Quality

Key highlights of a broad expanding portfolio



Agenda

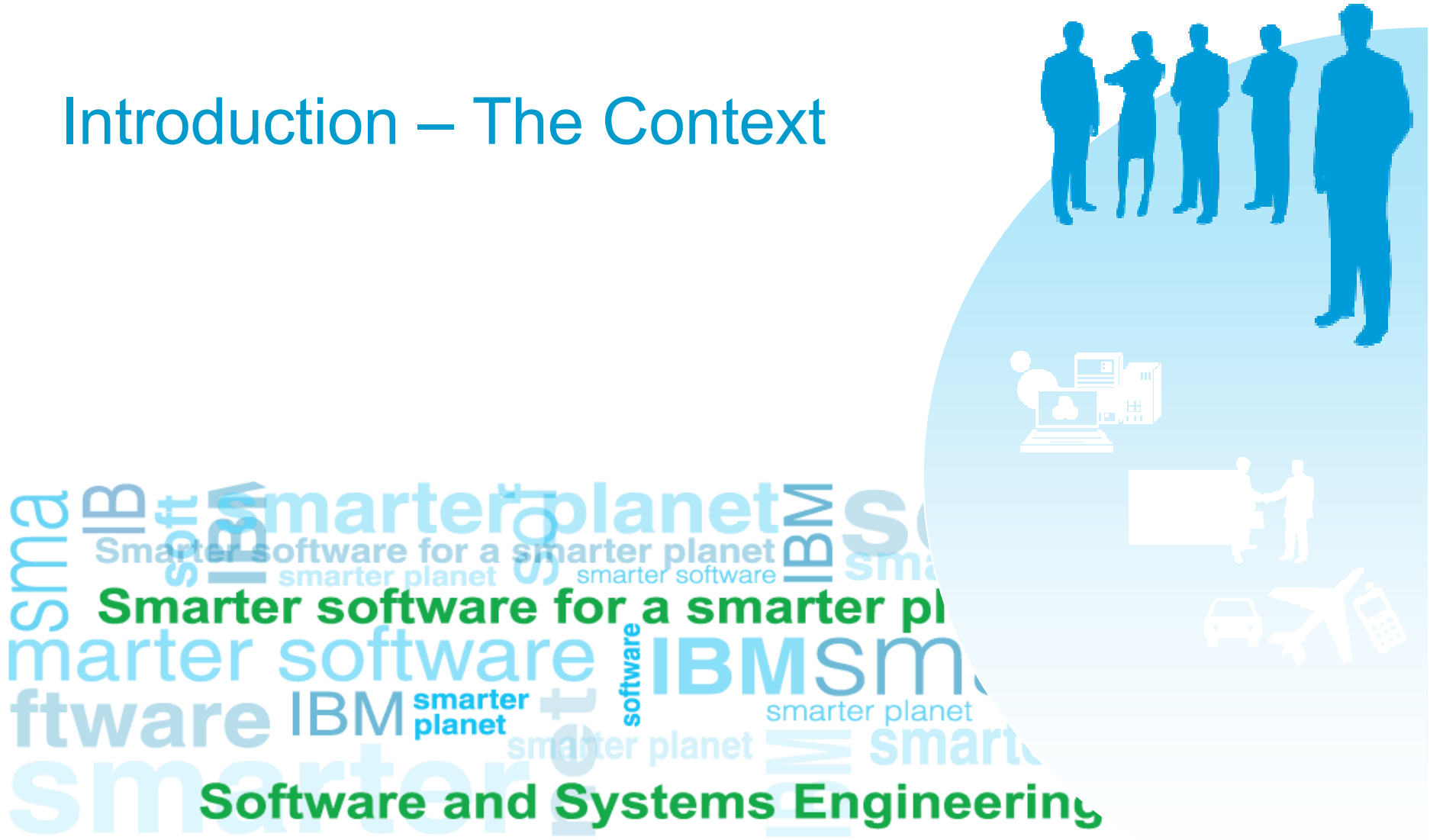


09:30	Registration & Welcome coffee
10:00	IBM Rational Quality Management Strategy
10:20	Enterprise Quality Management – Rational Quality Manager
11:15	Automated Testing – IBM Rational Functional Tester
11:30	Break
11:45	Service Testing – Rational Service Tester
12:05	Performance validation - Rational Performance Tester
12:50	Security Testing - Rational Appscan Family
12:55	Wrap-up/Next Step
13:15	Lunch
14:00	Rational Solution for SAP – SAP Connector, Worksoft, VirtualForge
15:10	Green Hat Overview
15:30	Conclusione

Tester's dream…

Fortunately It's not possible … people is better than software

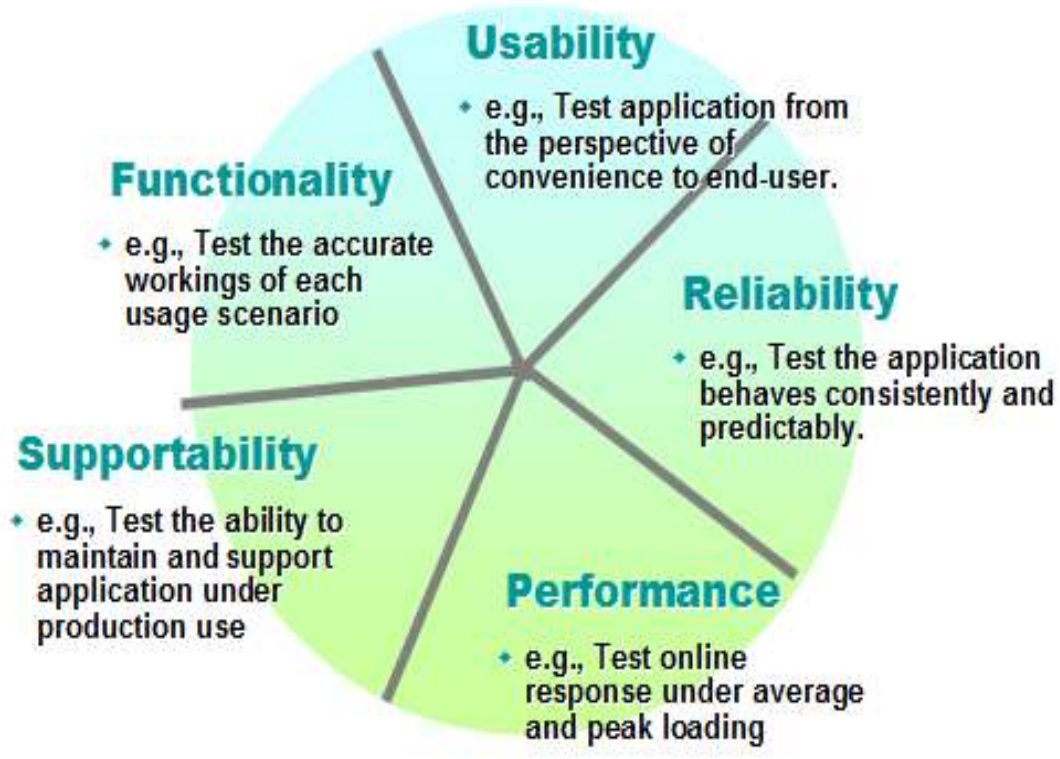
Introduction – The Context



Why testing Software ?

- To improve the quality of the product
- To decrease the rate of failures (increase the product's reliability)
- To ensure that the requirements are implemented
- To validate that the product is fit for its intended purpose
- To verify that the required standards and legal requirements are met

Dimensions of Quality: FURPS



Functionality
Feature set, Capabilities, Generality, Security

Usability
Human factors, Aesthetics, Consistency, Documentation

Reliability
Frequency/severity of failure, Recoverability, Predictability, Accuracy, Mean time to failure

Performance
Speed, Efficiency, Resource consumption, Throughput, Response time

Supportability
Testability, Extensibility, Adaptability, Maintainability, Compatibility, Configurability, Serviceability, Installability, Localizability, Portability

Quality is not a problem if…

- We have clear requisite
- We implement requisite in a right way
- We have Resouce, time and money for our project
- We have a clear Quality Assurance process
- Management know that quality is the target

- But we are not …NASA

(By: Nigel Cheshire " How good is good enough" http://java.sys-con.com/node/312718)

After a 1996 Fast Company article on the Lockheed Martin group that builds and maintains software for the space shuttle program, that software is often cited as the most expensive code on the planet, line for line. I'm not sure anyone really knows the cost per line of the space shuttle software (it's been estimated at $1,000 per line), but we do know that (as of 1996) it took roughly 260 developers to maintain 420,000 lines of code, which comes out at about 1,600 lines per person. That's expensive - but the approach seems to work: according to the article, the previous three versions of the software had only a single defect detected per release.

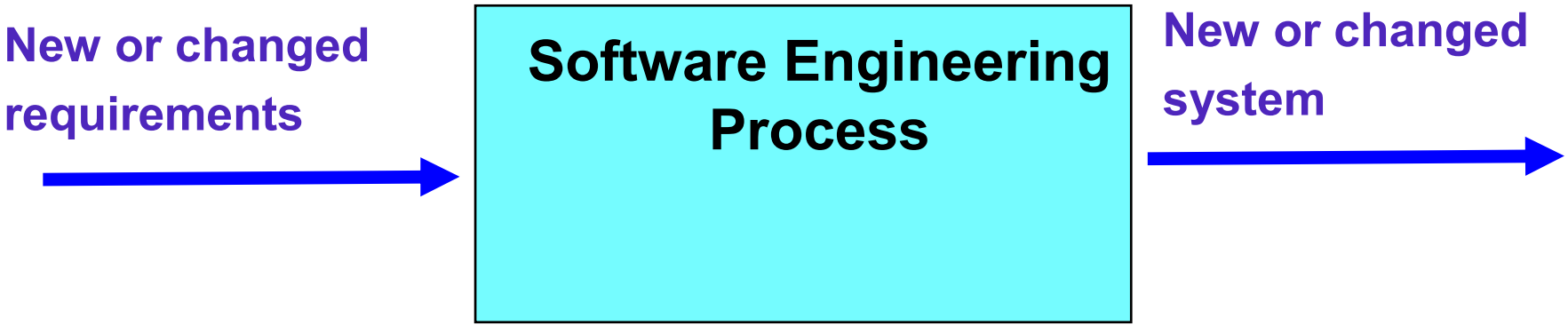
Where to start?

Tools, Process, or Something Else?

- Tools vs.. Process – is there an order?
  - Traditional approaches involve tool deployment and then adoption
    - Don't do this
    - Without a process framework, this can backfire
    - The best tools on the planet won't help you if you don't know what you're doing
- The Secret Sauce – Define Your Process first
  - Make it lightweight in terms of activities & artifacts
  - Adopt best practices – but only when it makes sense to the effort and yields high value to the team
  - Keep it simple – or it won't be used
  - Bring in experts – someone who won't use a cookie-cutter approach

What if ….better products and a bad process?

A process defines **Who** is doing **What When,** and **How**, in order to reach a certain goal.



Process or processing (verb) typically describes the action of taking something through an established and usually routine set of procedures or steps to convert it from one form to another, such as processing paperwork to grant a mortgage loan, processing milk into cheese, or converting computer data from one form to another. A process involves steps and decisions in the way work is accomplished, and may involve a sequence of events.

By Wikipedia

What is a Practice?

Guidance for software and systems development, management, governance, and more

- A **Practice** is a self contained **aspect of a process** that can be adopted to provide a set of capabilities, they are made up of:
  - Detailed tasks for executing the work
  - Work products used and produced
  - Roles and guidance in support of those tasks and work products
  - Recommended measurements/metrics (both product/project metrics and process metrics)
  - Tool guidance and configuration assets (utilities, artifact templates, report templates, etc.)
- **Practices** are designed to be **independent of** each other or any specific delivery **process or lifecycle model**
- **Practices** become the primary **building block for** developing and tailoring content, and are generally reusable across a variety of delivery **processes or lifecycle models.**

Practice - Table of Contents

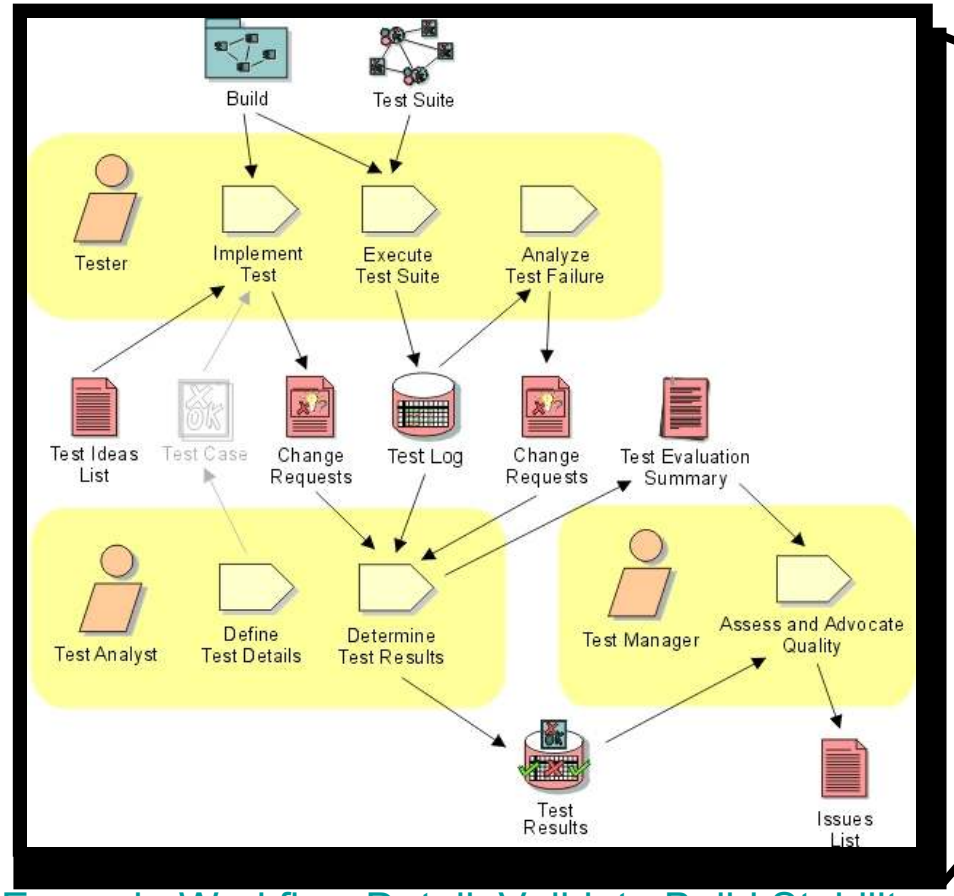
- Motivation – why do it
- How to adopt this practice
- Enablement, and reference material
- Key Concepts
- Work Products – what you produce
- Tasks – what you do
- Guidance – how you do it
- Tool guidance and configuration assets
- Recommended Metrics/Measurements
- Related practices

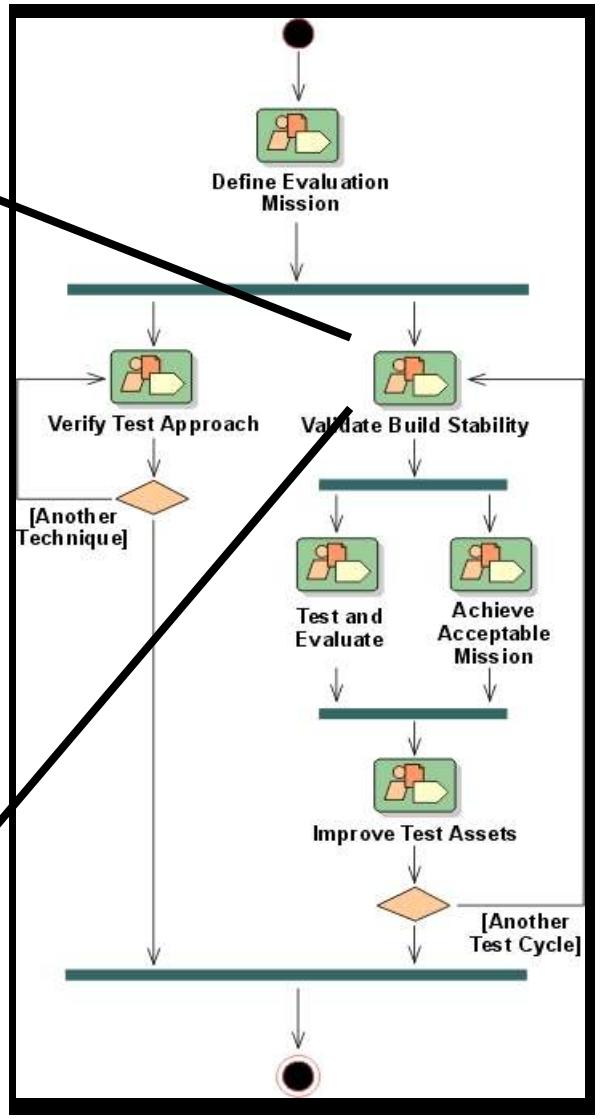
Example Practices (Teams practice…):
•Iterative Development
•Test-Driven Development
•Continuous Integration
•Requirements Management

Results:

- ✓ Avoids self-inflicting too much process
- ✓ Faster and more predictable results
- ✓ Incremental Measured Improvement

Testing Workflow from Rational Unified Process





Example Workflow Detail: Validate Build Stability

How to convince manager to invest

Often, quality professionals focus on trying to educate Management about quality.
This is the wrong starting point; instead, focus first on tactical successes that impact the short-term bottom line.
"Short-term" means from now until the product ships.
Sure, I understand that increasing quality is likely to decrease customer service calls during maintenance—I've been making that argument as a QA professional for years.
But does this mean it's good business to give you more money?
Think about the equipment purchase/rental scenario. Do you have enough data to justify a solid return on investment? If not, I'd rather apply this money where the risk is lower and the payoff is as great. How much does technical support cost versus the amount of money you think you need? Have you thought this through?

Why Collect Metrics?

- Manage
  - Identify scope of test
  - Ensure effort is on schedule
  - Determine state of product
- Communicate
  - Test status/progress
  - Product quality
  - Readiness to ship
  - Problem components
- Improve
  - Identify problem components and feedback f on release

Do We Really Need Metrics?

"If you cannot measure it, you cannot improve it."

Lord Kelvin

"In physical science the first essential step in the direction of learning any subject is to find principles of numerical reckoning and practicable methods for measuring some quality connected with it. I often say that when you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meagre and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the state of Science, whatever the matter may be." [PLA, vol. 1, "Electrical Units of Measurement", 1883-05-03]

"You cannot manage what you cannot measure"

Slide 14

I4

Manage:
How many test cases to be tested in what time period?
Are you meeting your milestones?
Project state is what are you finding? how much are you finding and how far along are you?

Communicate:

IBM_User; 29/01/2003

Measurements and Practices

Increase Defect Prevention

Measures:
- Defect density
- Defect arrival/closure rates
- Defect backlog
- Fixes failing verification
- Rework effort

Practices:
- Test-driven Dev.
- Design-driven Implem.
- C&C management
- System Component Arch.
- Whole team
- Pair Programming
- Review/Inspection

Increase Defect Detection

Measures:
- Defect density, distribution
- Defect arrival/closure rates
- Defect removal effectiv.
- Fixes failing verification
- Test coverage
- Test execution status

Practices:
- Test management
- Continuous integration
- Evolutionary Architecture
- Component Architecture
- Test-driven dev.
- Test practices
- Iterative Dev.
- Risk Value Lifecycle
- C&C Management
- Review/Inspection

Deliver on Customer Requirements

Measures:
- Post-ship problem reports
- Customer satisfaction
- Pipeline conversion
- Support / maint. costs
- Requirem. test coverage
- Requirements delivery
- Survey of feature usage

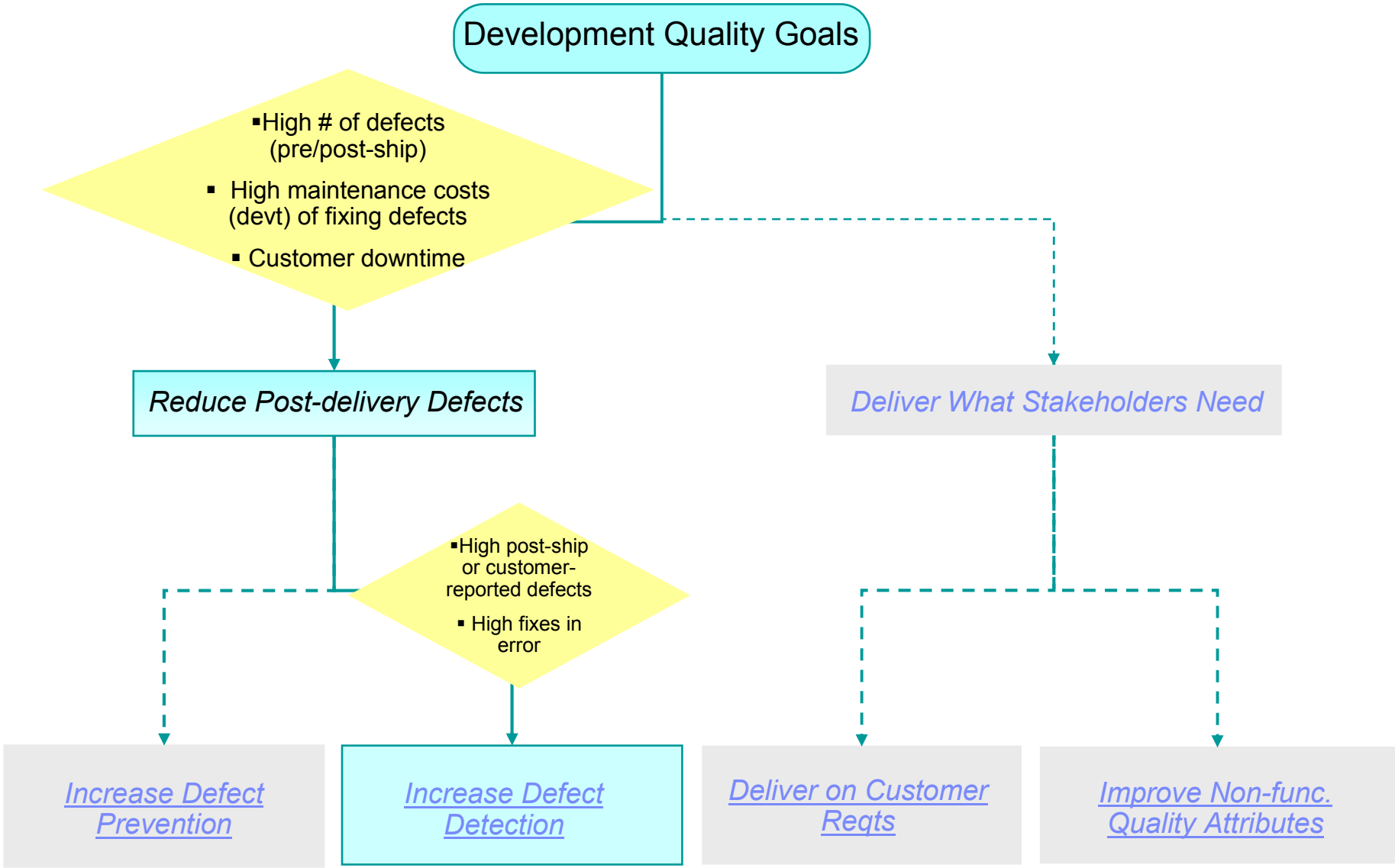
Practices:
- Shared Vision
- Use-case Driven Dev
- Requirements Mgnt.
- Whole Team
- Iterative Dev.
- Functional Testing
- C&C Management
- Review/Inspection

Improve Non-functional Quality Attributes

Measures:
- Post-ship problem reports
- Customer satisfaction
- Support / maint. costs
- Requirement test coverage
- Test execution results

Practices:
- Application Vulnerability Assessment
- Performance Testing
- Requirements Mgnt.
- Shared Vision
- Risk-Value Lifecycle
- Evolutionary Architecture
- Test-Driven Development
- Iterative Development
- Evolutionary Design
- Component Architecture
- Continuous Integration
- Concurrent Testing
- Whole Team
- Review/Inspection

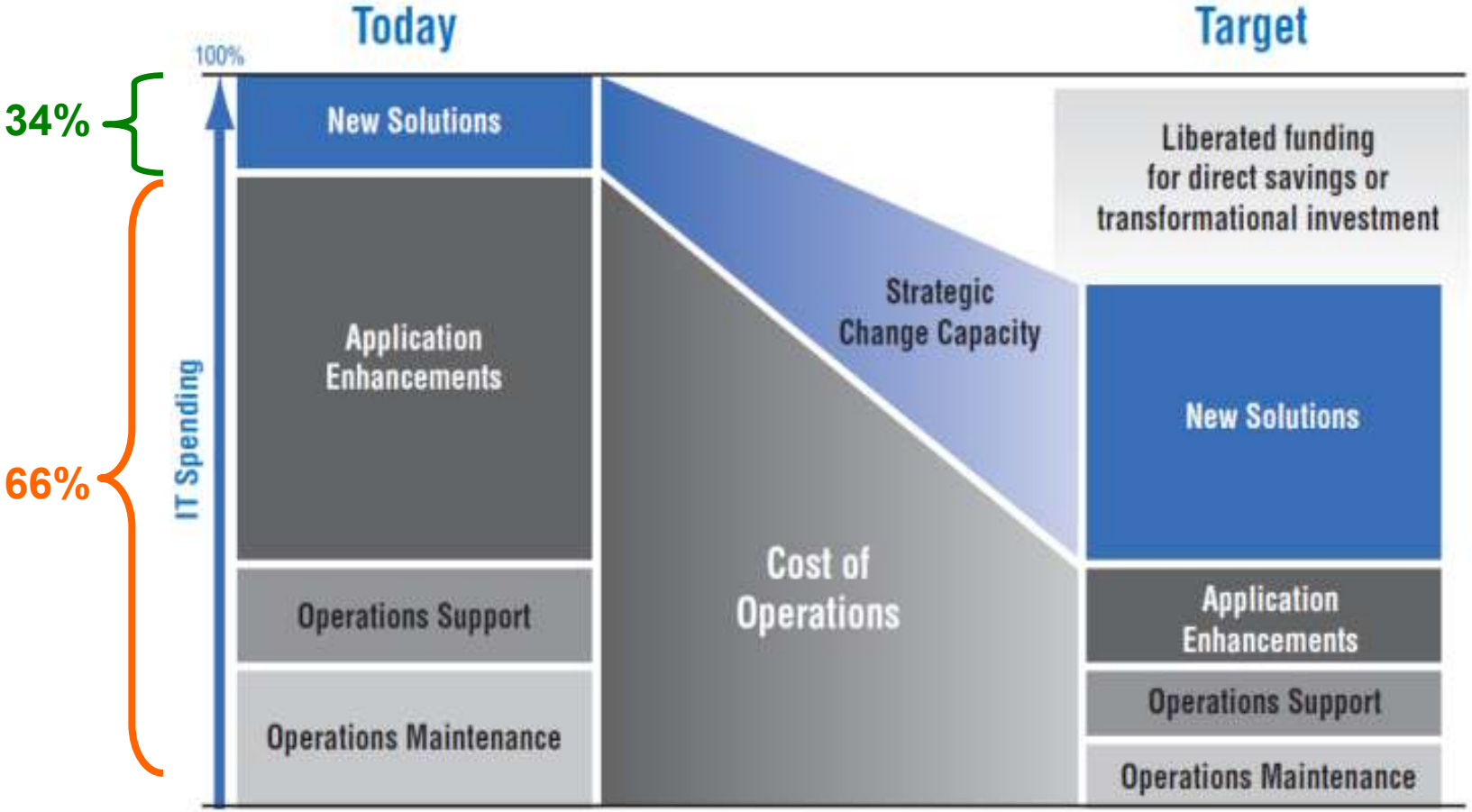
Quality Traceability Tree: Decision Criteria



IBM Rational Quality Management Strategy

Business and IT Agility: Balancing Resources to Support Business Innovation

Balance IT investments to focus on new solutions.

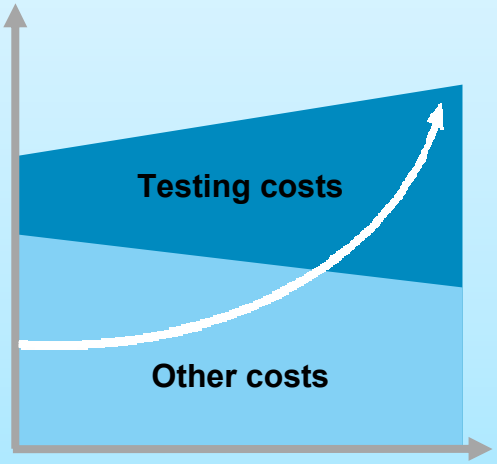


- Forrester estimates that ongoing operations and maintenance consume ***66%*** of IT budgets
- While new projects and software initiatives represent only ***34%***

Cost, complexity and velocity make today's quality paradigm impractical

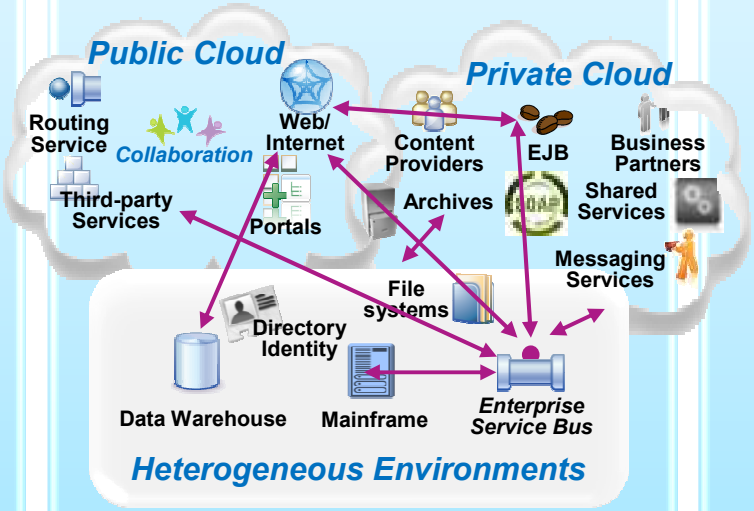
An estimated 60 - 80 percent of the cost of software development is in rework*

Increasing Cost of Quality



Outsourcing **labor** is no longer a sustainable model as global wages are increasing

Increasing Development Complexity



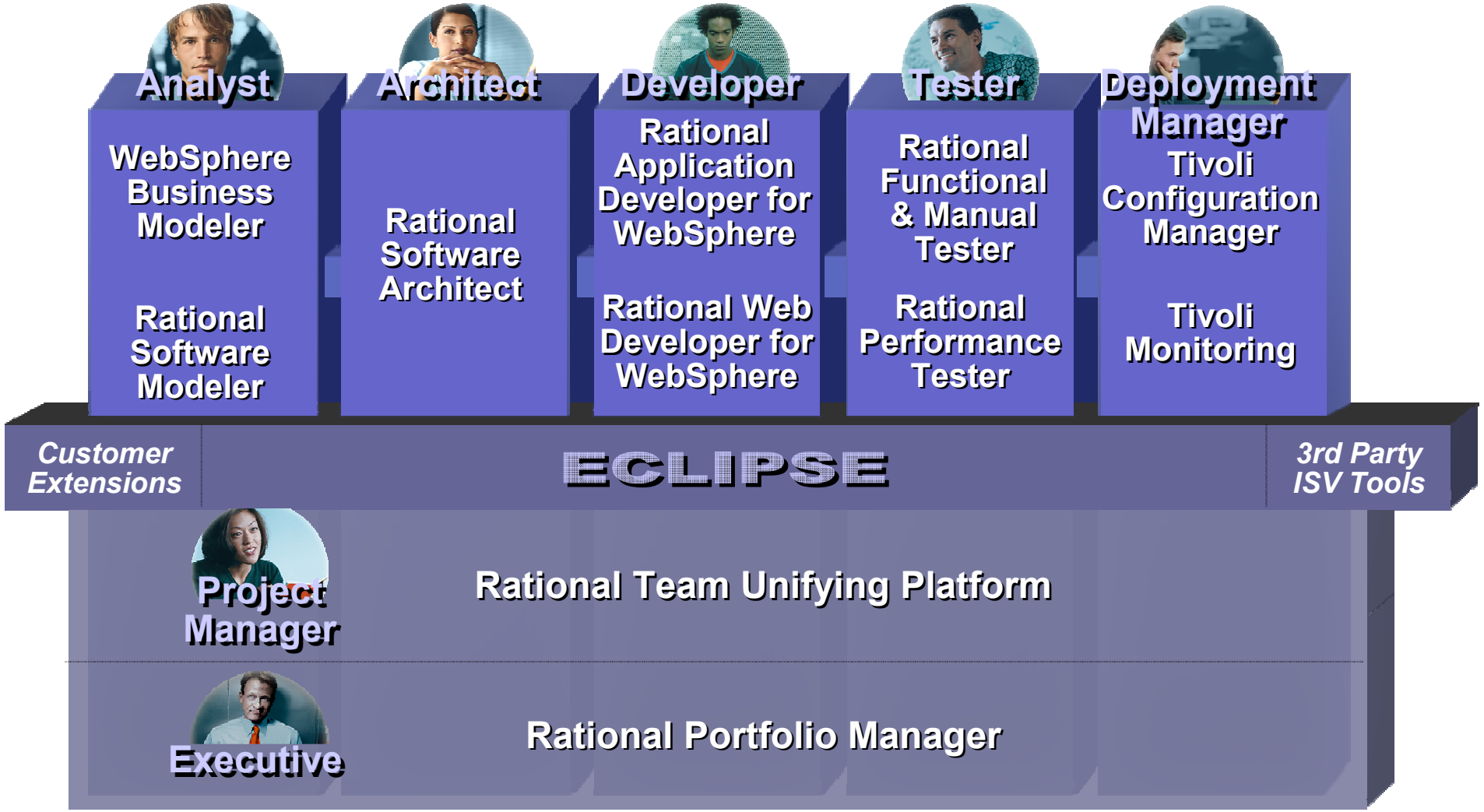
Product and application **complexity** and size are increasing

Balancing Quality and Speed

Productivity is inhibited as test teams can no longer keep up with agile development

* Source:http://www.sei.cmu.edu/about/message/

We are moving From …



Immaginiamo un *gruppo jazz* fa un *concerto* per un *pubblico pagante*

- Più che eseguire, interpretano
- Sono disciplinati anche senza direttore d'orchestra
- Collaborano e si ascoltano
- Improvvisano "consapevolmente"
- Suonano "live" di fronte al pubblico pagante
- Si divertono in quello che fanno

Proviamo a sostituire

- Gruppo jazz con **team di sviluppo SW**
- Concerto con **progetto**
- Pubblico con **cliente**

jazz

People,
not organizatios,
build great software.

…le quali introducono nuove esigenze tecniche

- Gli strumenti dovrebbero essere non intrusivi
- Non dovrebbero esserci confini tra gruppi diversi
- La collaborazione creativa deve essere possibile anche tra siti geografici e organizzazioni diverse
- I processi debbono essere personalizzabili e attivamente supportati dagli strumenti
- La Governance deve essere forte ma non oppressiva
- Le attività non creative e ripetitive vanno automatizzate

Jazz Vision -Three key actions of transforming software and systems delivery

Integrate

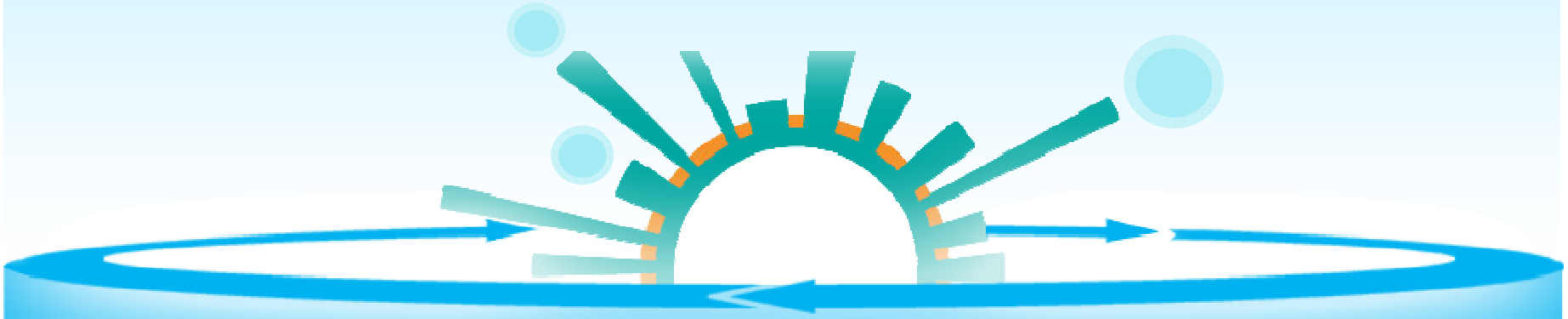
Connecting process and information
Software, data, and tools

Collaborate

Unifying teams
Projects, and organizational cultures

Optimize

Simplifying governance
Plans, scope, and measures

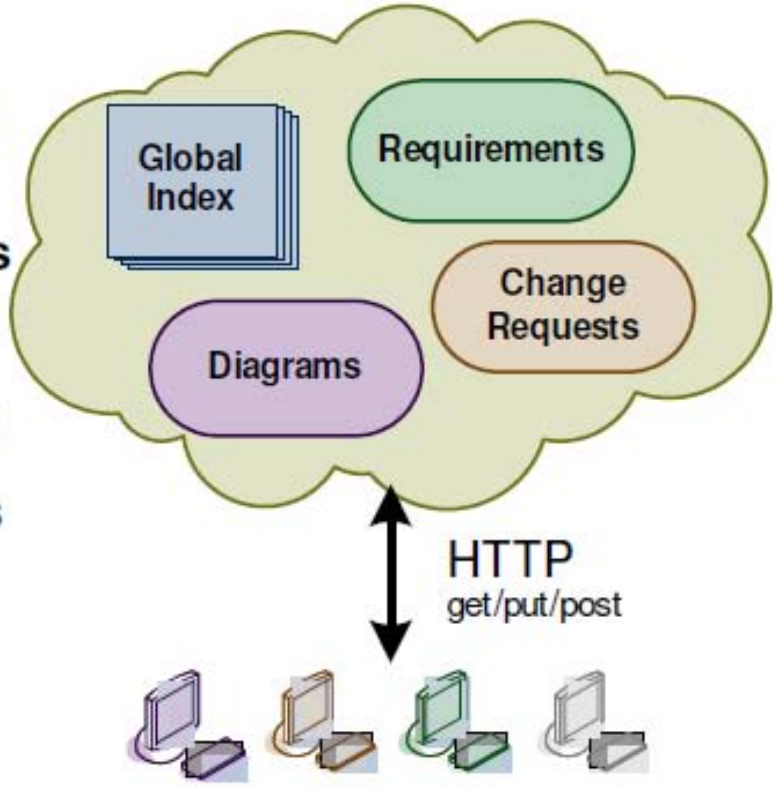


Realized benefits:

- **Improved quality and time to market**
- **Reduced risk and cost**
- **Tighter alignment to business priorities**

What does Internet Inspiration mean?

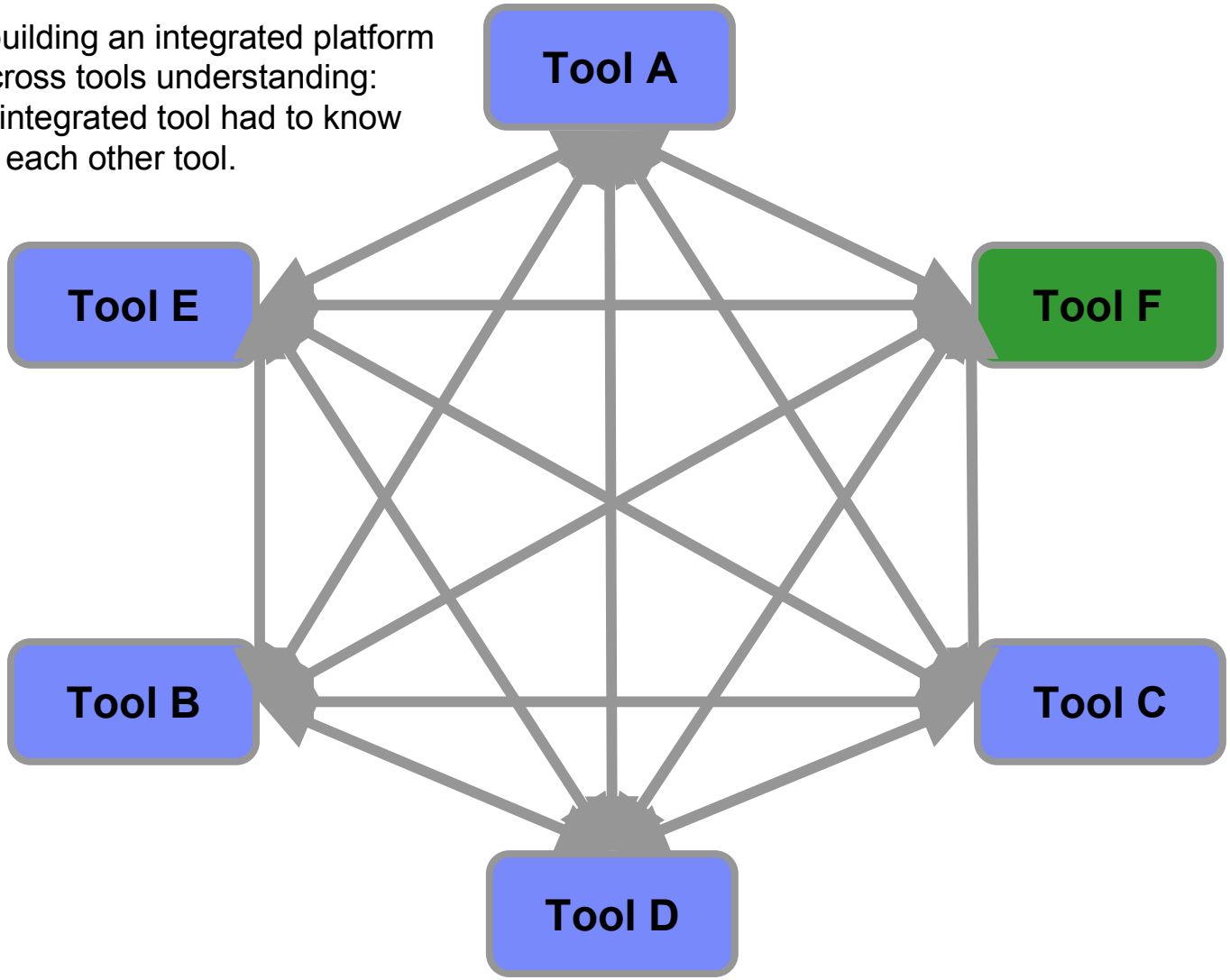
- Data specified independently of tools
- All data are resources with URLs
- Multiple Tools access data
- References are embedded URLs
- Resources have representations
- Unprecedented extensibility
- Independent search and query
- REST (Representational State Transfer)



Tools collaboration generally means exponential complexity!

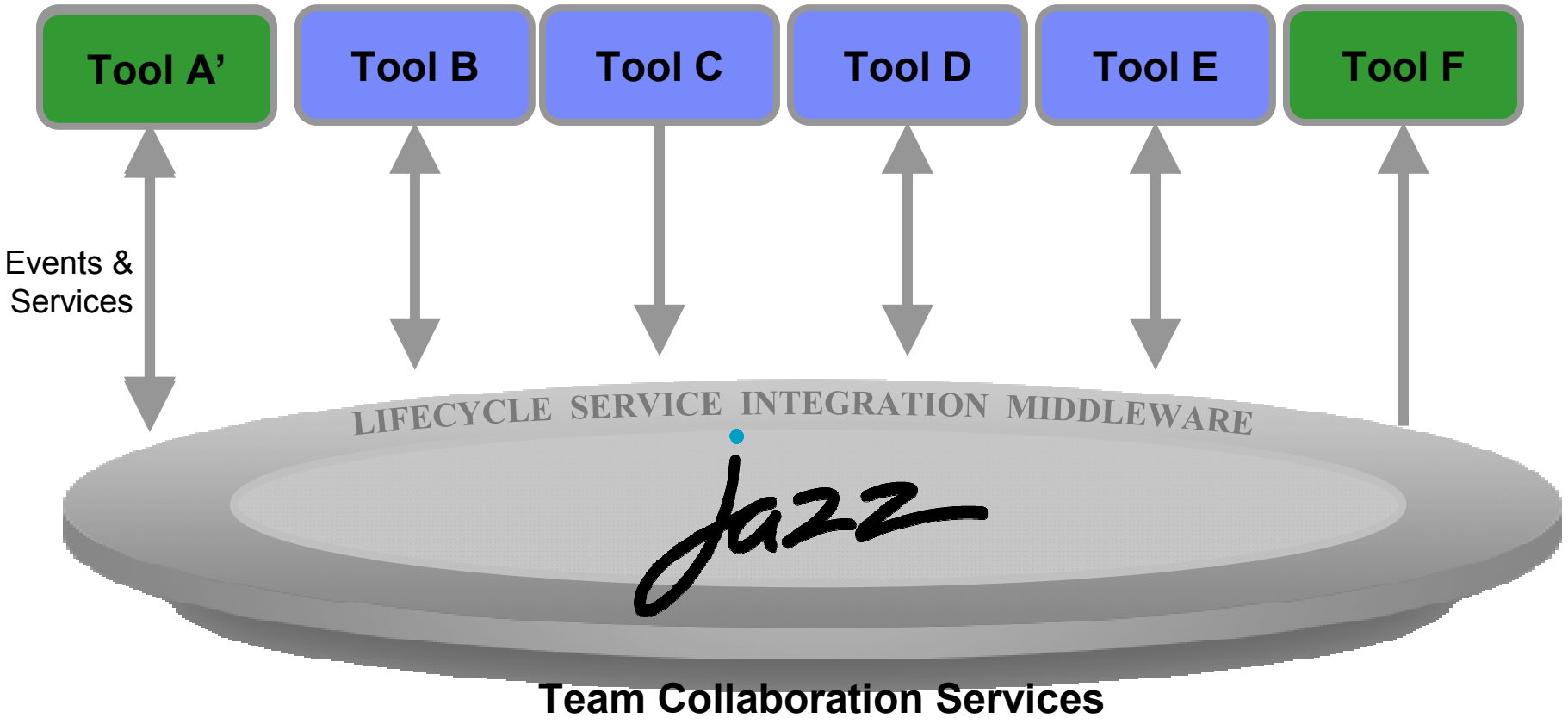
Until now, building an integrated platform required a cross tools understanding:

Each integrated tool had to know about each other tool.

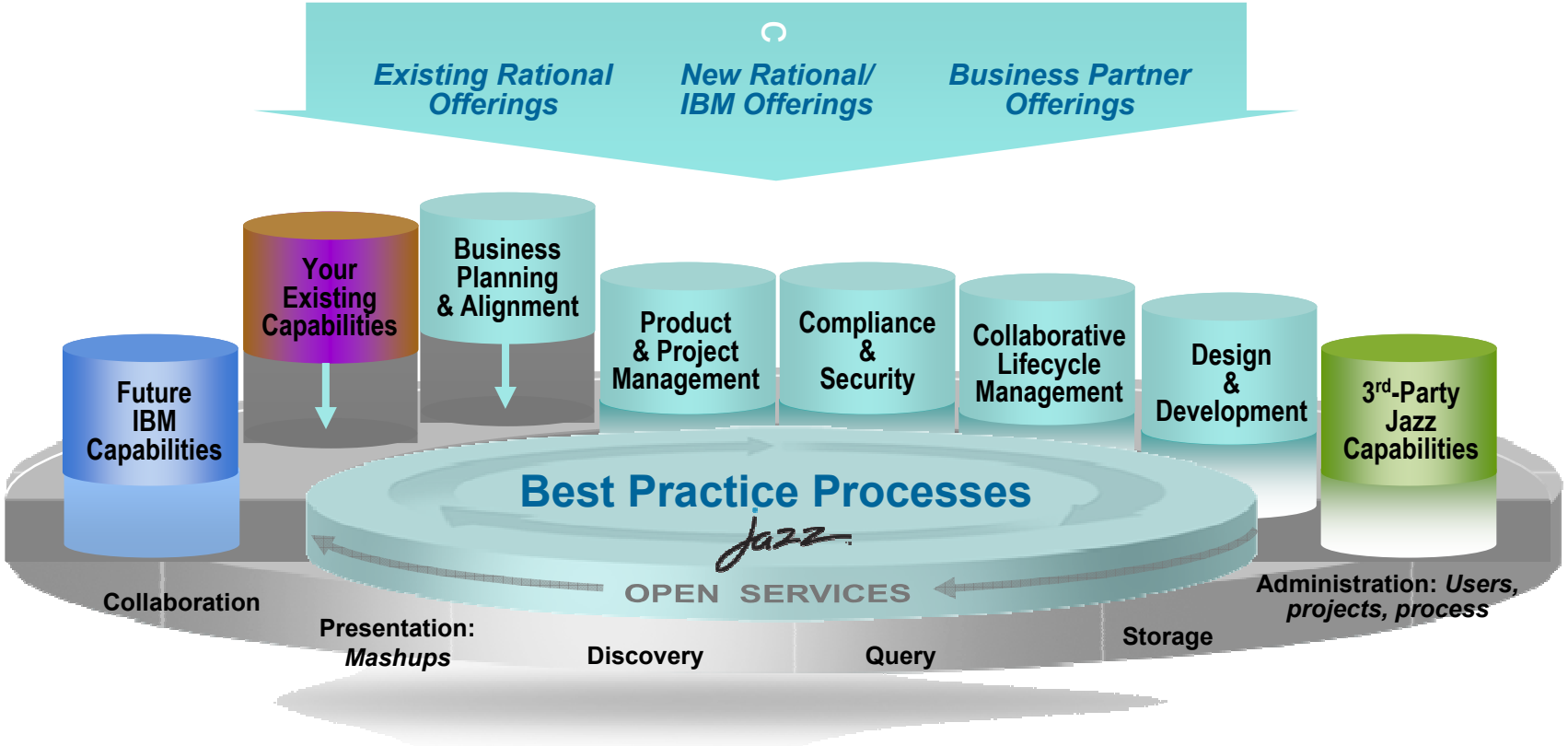


Tools collaboration based on middleware services

- With the Jazz platform, the tools communicate only with the platform:
  - By listening to normalized/standardized events from the platform
  - By sending normalized/standardized events to the platform



Jazz is an *open* platform with a shared set of *services*



- Separate the implementation of tools from the data
- Federated, open data model
- Tools can be implemented in any internet-aware programming language.
- Support multiple client technologies (Web, Eclipse and Microsoft .Net– others possible)
- Implement OSLC Specifications

An ALM solution powered by Jazz

Rational solution for Collaborative Lifecycle Management

CREATE SOFTWARE

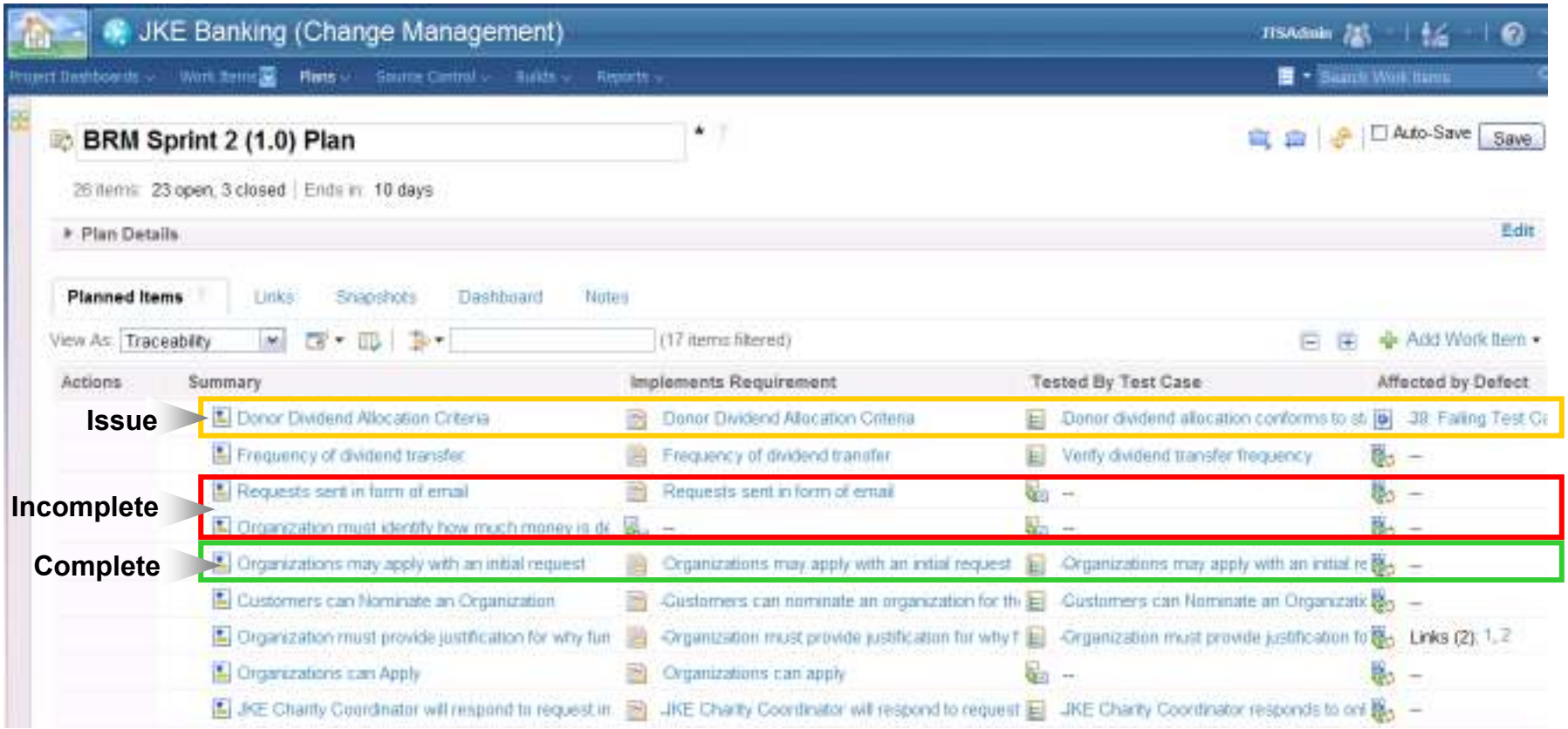
Real-time Planning, Lifecycle Traceability, Team Collaboration, Development Intelligence, Continuous Improvement

Rational Requirements Composer	Rational Team Concert	Rational Quality Manager
Requirements Management	Planning, Change, Configuration & Build Management	Quality Management

*extensions

Team leads improve release quality & predictability

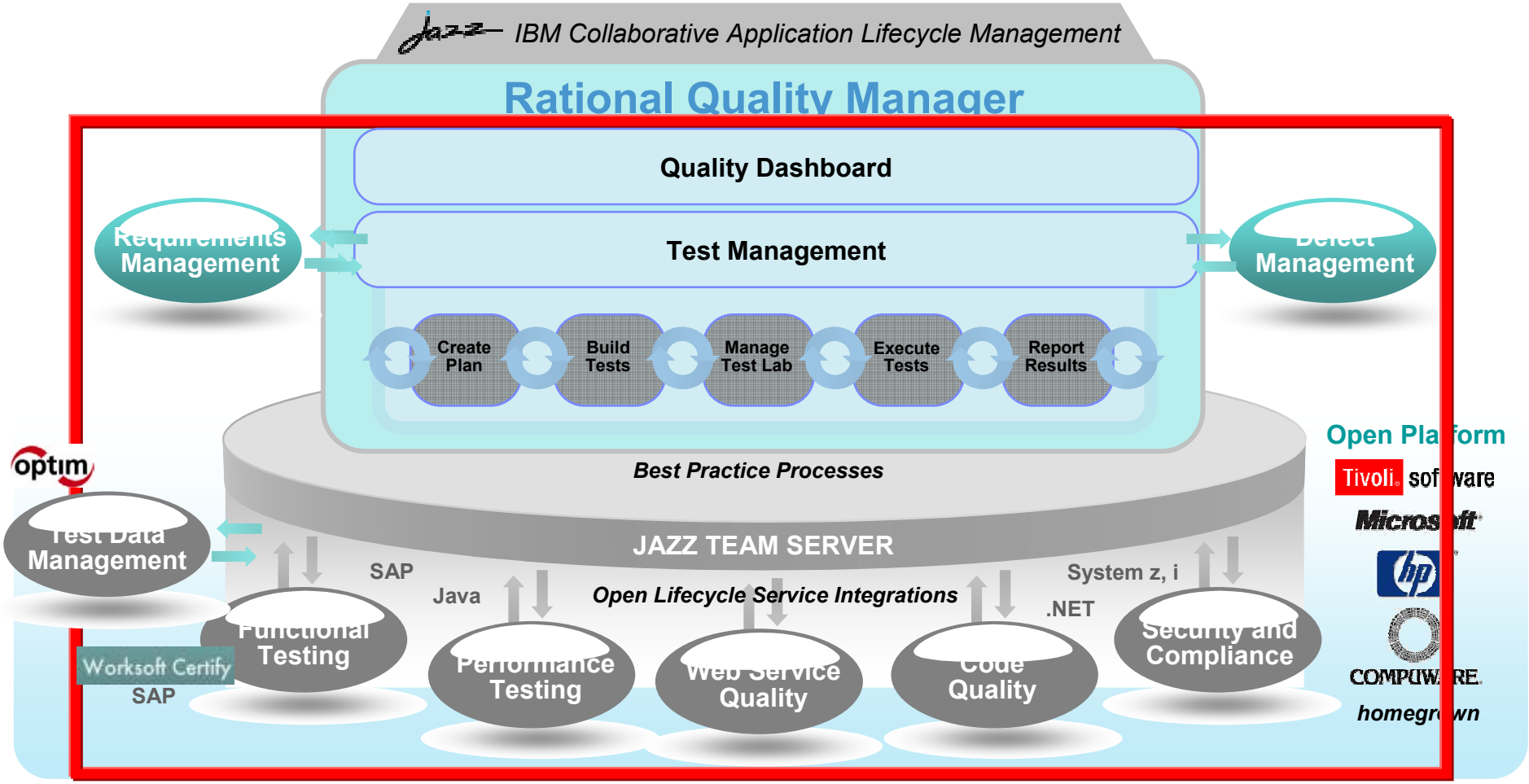
- Proactively respond to gaps as they surface through out the project
- Issues quickly highlighted and resolved



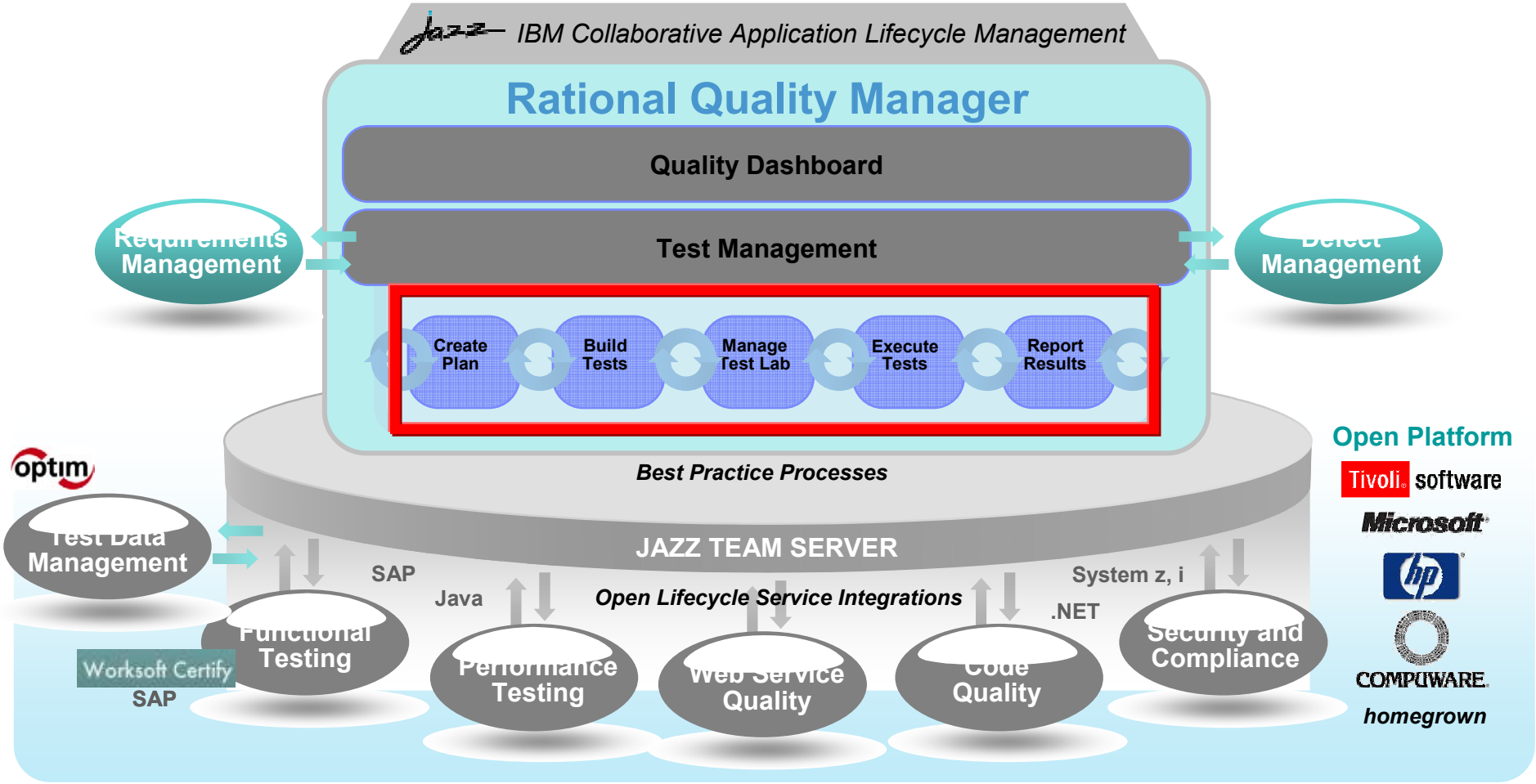
Enterprise Quality Management
Rational Quality Manager

Today's High Level QM Segment Architecture

Optimize software quality with a centralized test management hub and integrated full lifecycle support across any platform and type of testing



Centralized test management offering allowing full lifecycle support across all types of testing and platforms



IBM Rational Quality Manager

A central hub for business-driven software quality

Mitigate business risk with collaboration

- ✔ Stakeholder and team coordination reduces mistakes
- ✔ Risk identification and management leads to educated prioritization decisions
- ✔ Test traceability linked to business requirements improves customer satisfaction

Improve operational efficiency with automation

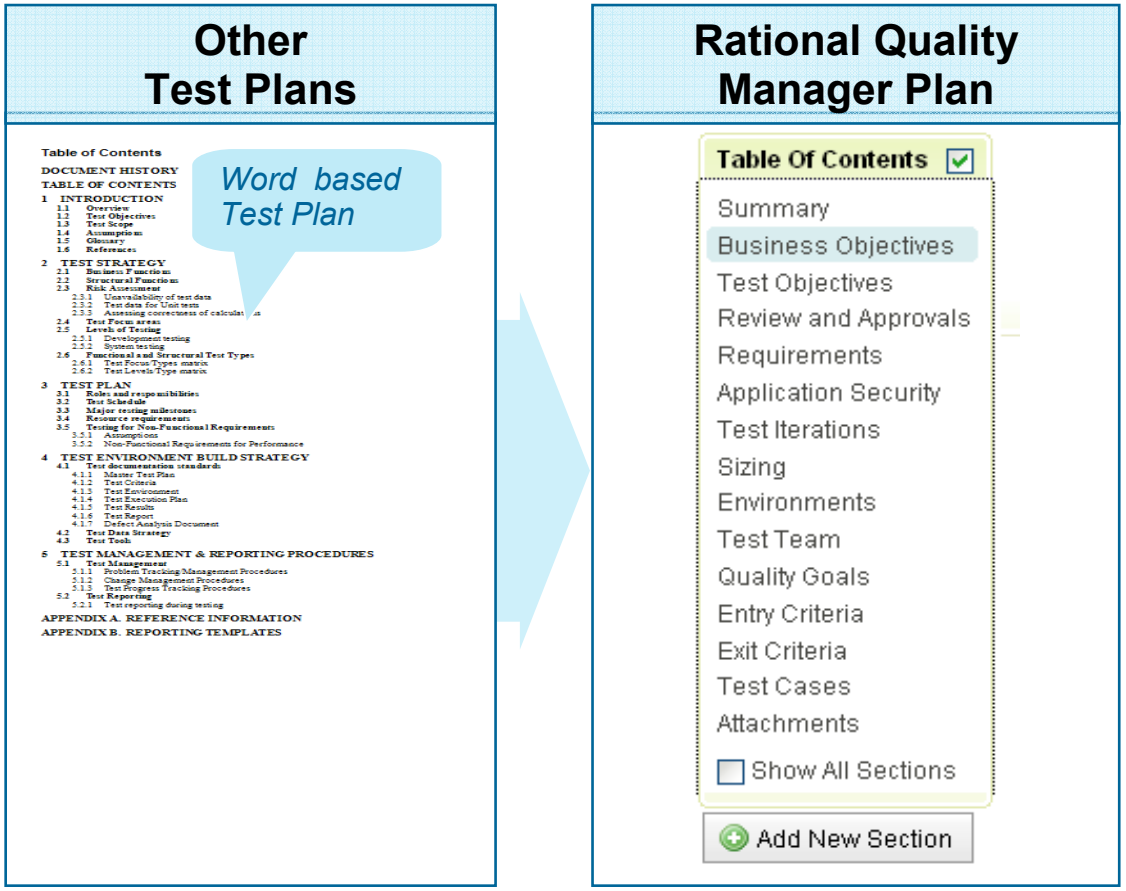
- ✔ Running tests earlier leads to reduced repair costs
- ✔ Running more tests in less time improves coverage
- ✔ Reducing manual labor leads to fewer testing errors
- ✔ Lab configuration automation improves efficiency and asset utilization

Make confident decisions with effortless reporting

- ✔ Real-time dashboards enable proactive risk management
- ✔ Customizable reports facilitate ongoing process improvement

Comprehensive dynamic planning and updates

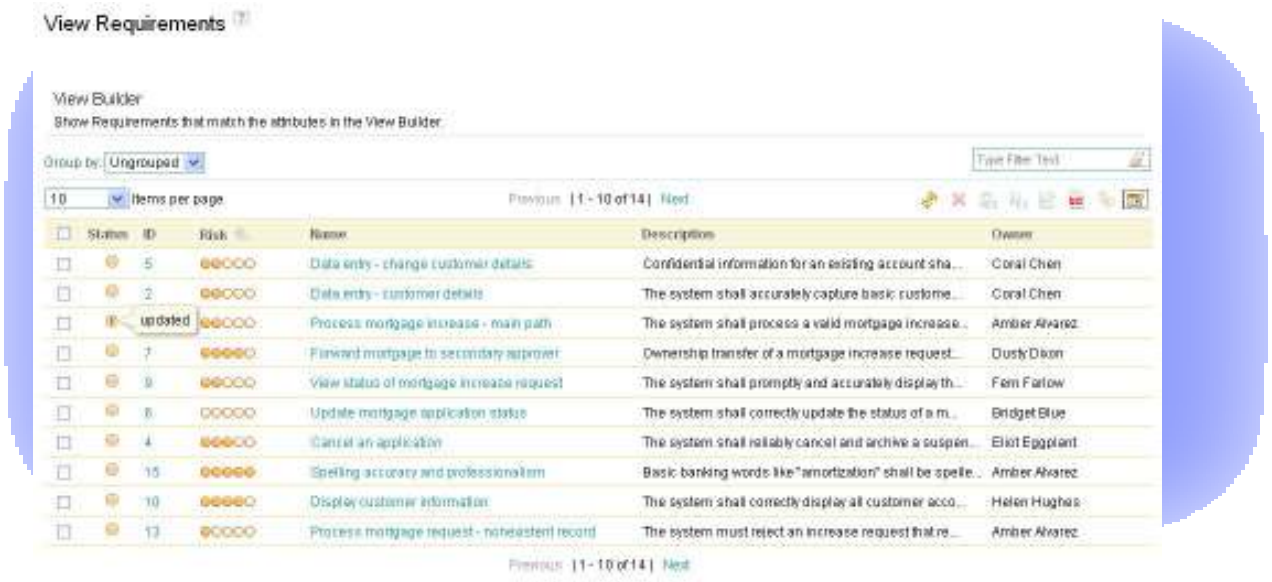
Process flow, not artifacts drives team activities



- Live dynamic documentation
- Defines test process and strategy
- Defines responsibilities
- Activity based versus hierarchy
- Business level reporting against quality objectives

Requirements driven testing

Knowing what to test

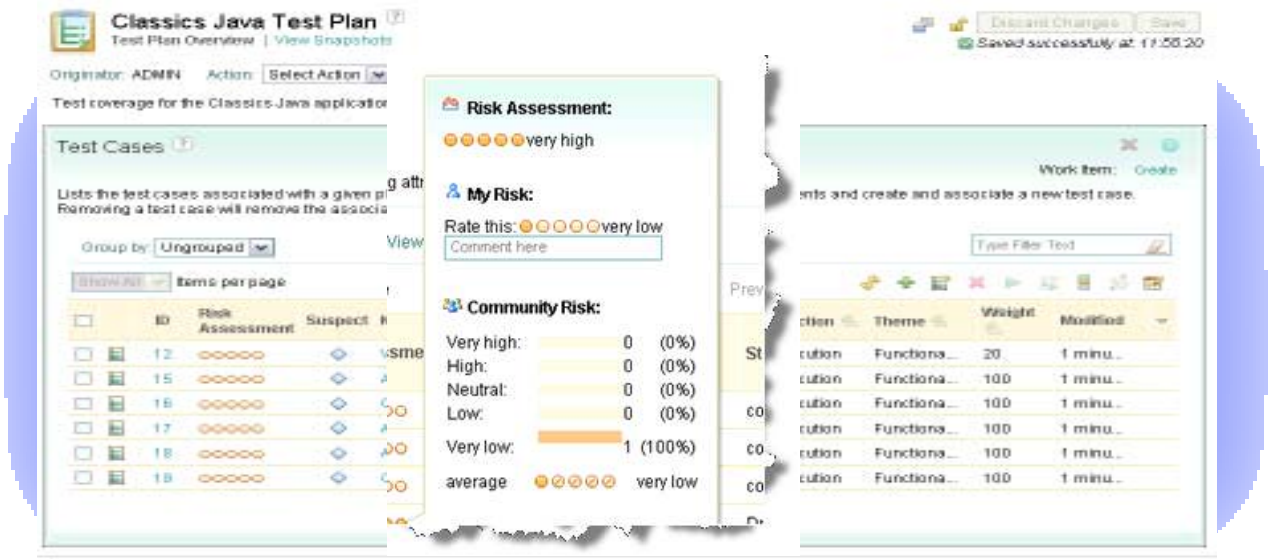


- **Requirements tracking built into the test management tooling**
- **Customizable attributes enable you to track what is important to your team**
- **Real-time impact analysis of requirements changes**
- **Traceability of test results to user needs**

Know you are testing the right things

Collaborative risk based testing

Risk management and prioritization



- **Risk assessments captured in Test Plan and Test Cases**
- **Collaboration planning of risk mitigation strategy**
- **Test Case will contain a risk failure score and a risk priority score**
- **Documented risk related decisions**

Base project decisions on qualitative risk analysis

Test coverage optimization

Focus resources on testing the right combinations

Test Configurations

4 languages
4 browser types
5 databases
5 application servers

400 Combinations!

Pairwise Optimizations

Test the right 20 combinations

OS	Browser	Protocol	CPU	DBMS
XP	IE	IPv4	Intel	MySQL
XP	Firefox	IPv6	AMD	Sybase
XP	IE	IPv6	Intel	Oracle
OS X	Firefox	IPv4	AMD	MySQL
OS X	IE	IPv4	Intel	Sybase
OS X	Firefox	IPv4	Intel	Oracle
RHL	IE	IPv6	AMD	MySQL
RHL	Firefox	IPv4	Intel	Sybase
RHL	Firefox	IPv4	AMD	Oracle
OS X	Firefox	IPv6	AMD	Oracle

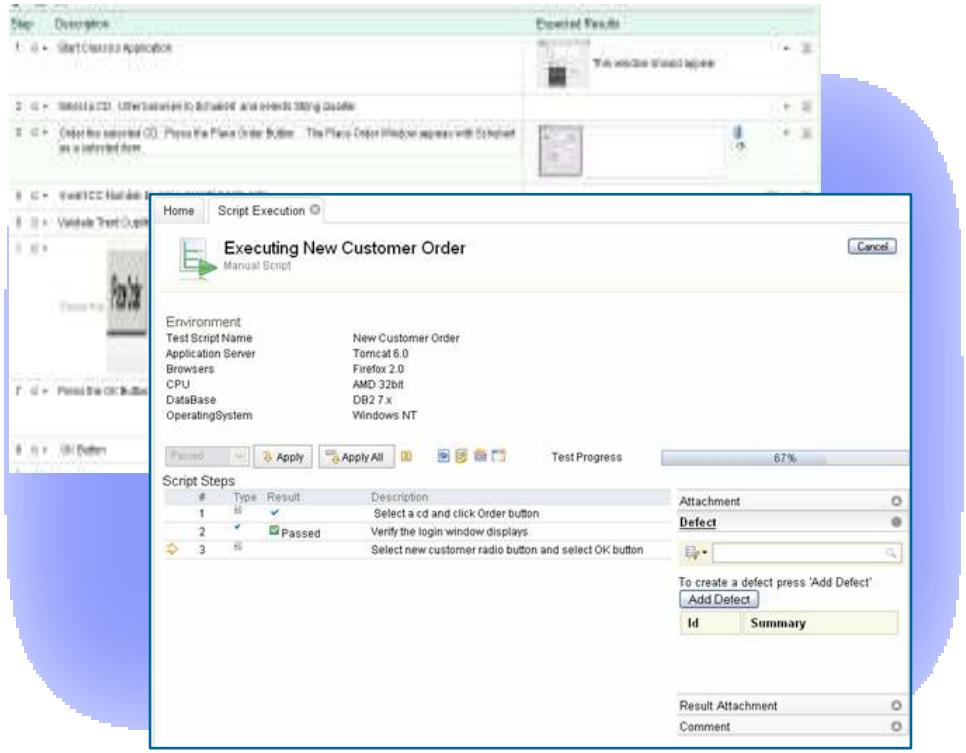
Configuration awareness

Test Platform Environment Management

- Focus your environment coverage
- Document your environment coverage
- Gain agreement across the project

Test the right cases instead of everything
Plan optimal execution

Integrated Manual test authoring and execution

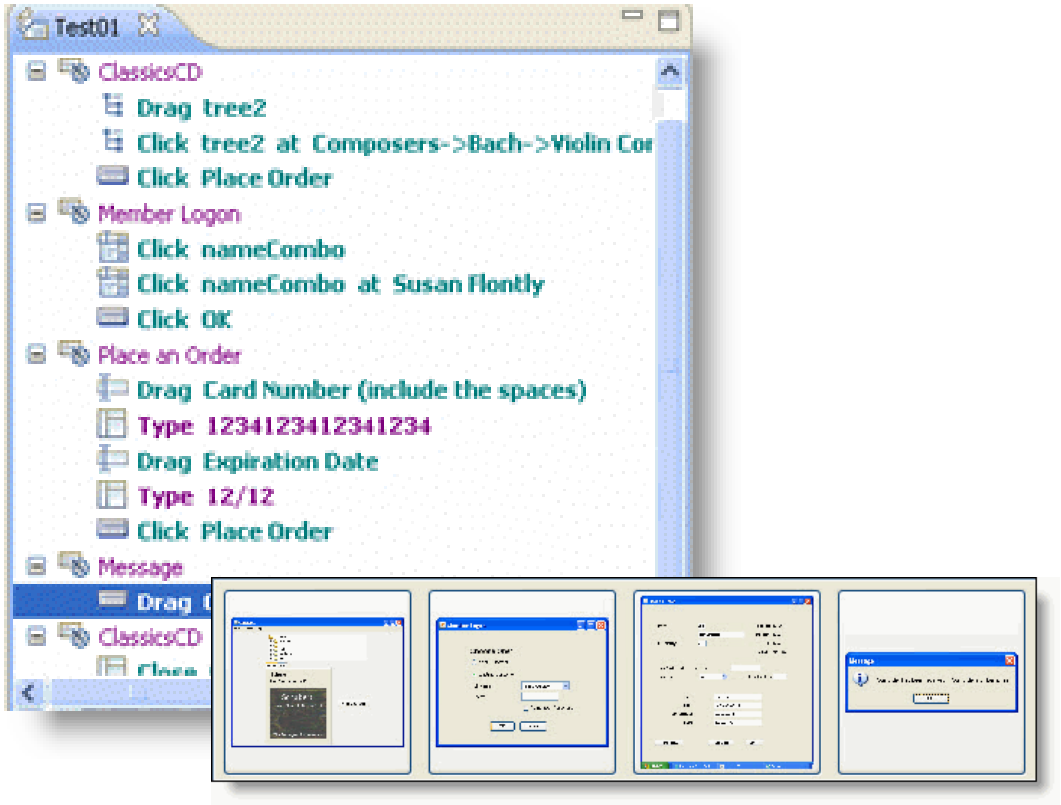


- Manual test author and execute
  - Step by step capture and execution of manual tests
  - Assisted data entry
  - Keyword support for integrated manual and automated testing
  - Rich defect capture during execution, including screenshot and attachments
  - Simple intuitive interface for quick test execution

Maximizing efficiency of manual testing

Integrated Functional and Regression test execution

1. Increase repeatability through automated test playback
2. Test more critical functions faster with automation
3. Automatically deploy your test environment and schedule the execution of your test Suites
4. Track and communicate progress and regressions throughout the testing lifecycle

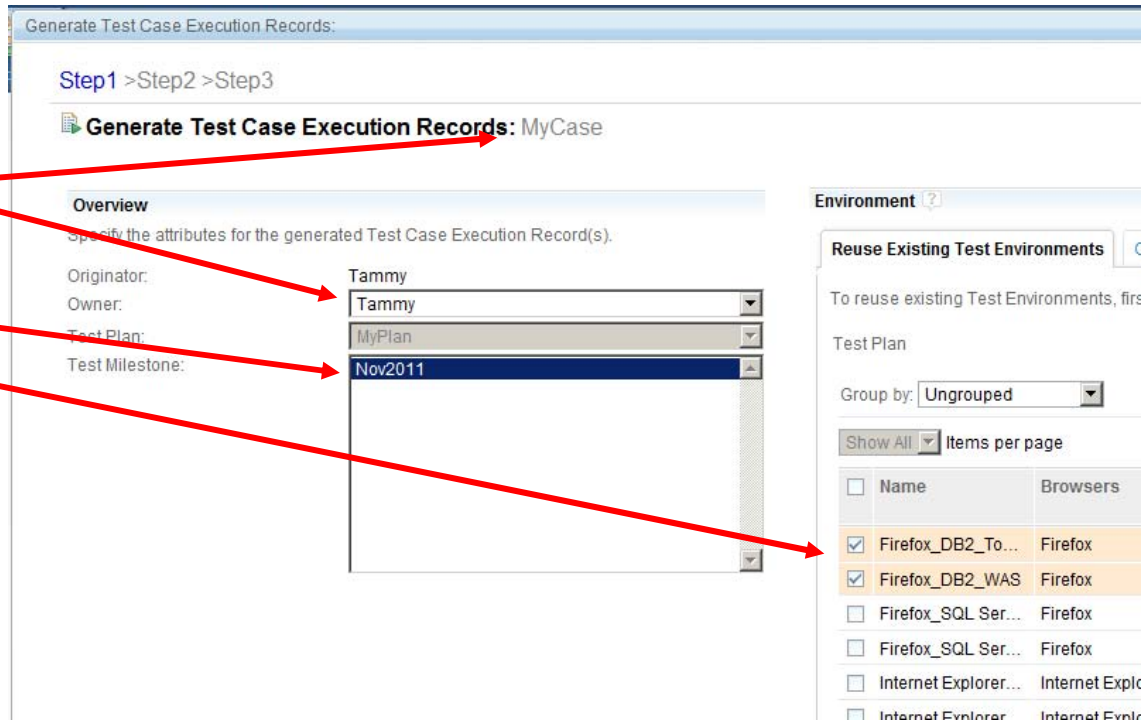


Accelerate test execution and deepen test coverage through automated test execution

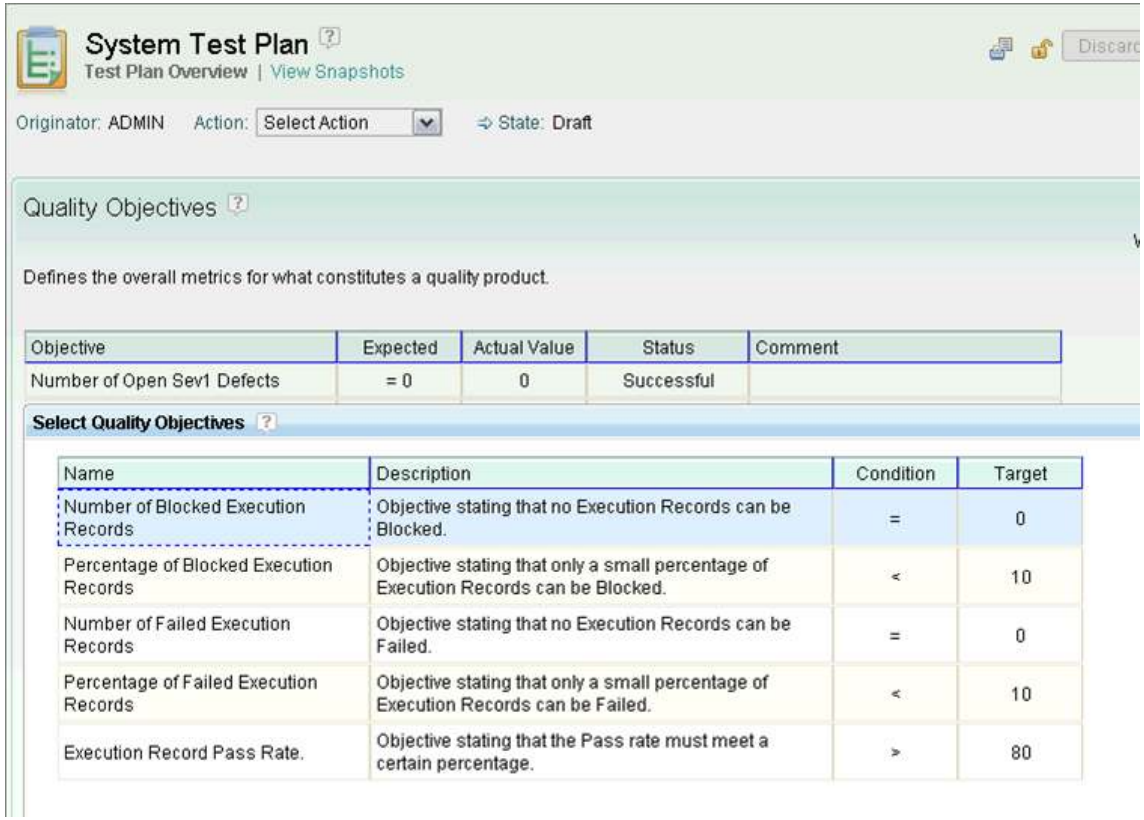
My Favorite: Test Execution Record

‘Test order’

- Who is
- Doing what
- When
- In which environment



Assess and measure against Organizational policies



System Test Plan

Test Plan Overview | View Snapshots

Originator: ADMIN Action: Select Action State: Draft

Quality Objectives

Defines the overall metrics for what constitutes a quality product.

Objective	Expected	Actual Value	Status	Comment
Number of Open Sev1 Defects	= 0	0	Successful	

Select Quality Objectives

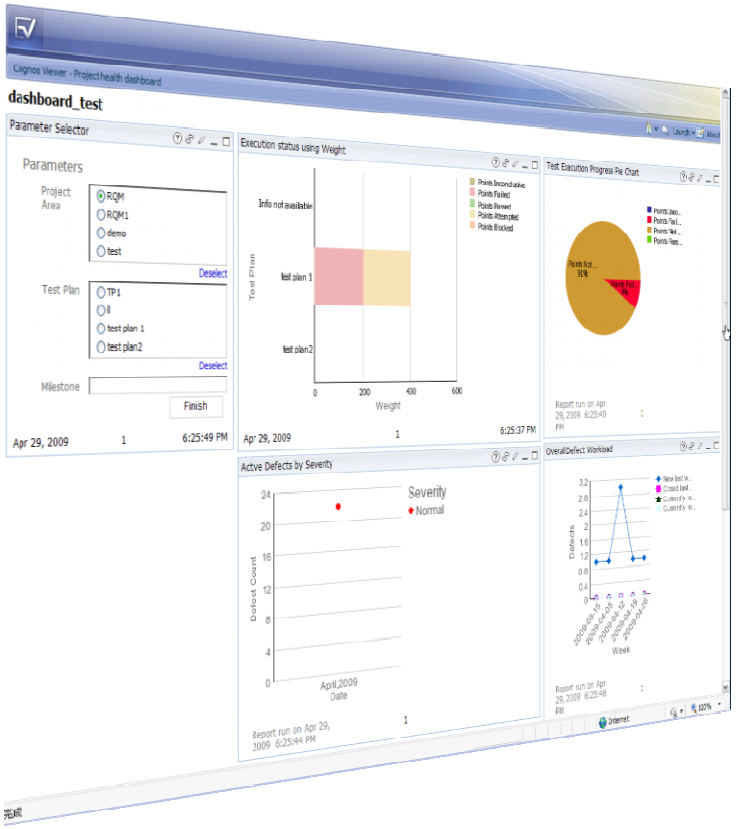
Name	Description	Condition	Target
Number of Blocked Execution Records	Objective stating that no Execution Records can be Blocked.	=	0
Percentage of Blocked Execution Records	Objective stating that only a small percentage of Execution Records can be Blocked.	<	10
Number of Failed Execution Records	Objective stating that no Execution Records can be Failed.	=	0
Percentage of Failed Execution Records	Objective stating that only a small percentage of Execution Records can be Failed.	<	10
Execution Record Pass Rate.	Objective stating that the Pass rate must meet a certain percentage.	>	80

- Assessing status
  - Standard Objectives
  - Reuse across Test Plans
  - All working toward same objectives
  - Measures against business objectives

Drive continuous and measured improvement

Make confident decisions with effortless reporting

Closed Loop Analysis & Reporting



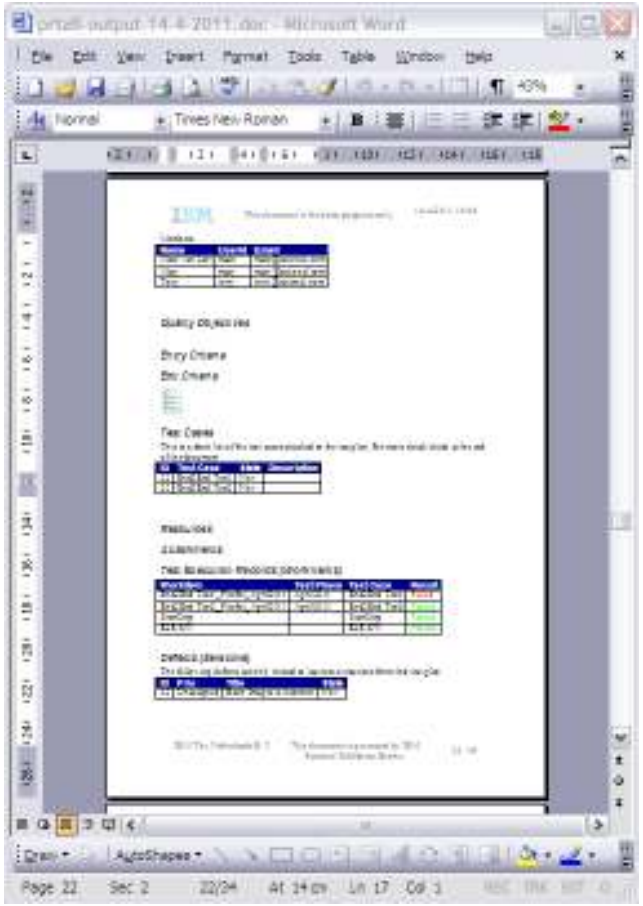
- Customizable reports and dashboards
  - Reduce escalating cost of information gathering
  - Reduce risk by identifying trends before they become issues
  - Raise enterprise visibility and transparency to reduce costs and risk
  - Measures the effectiveness of processes and practices to improve organizational and business outcomes

Make the right decisions at the right time

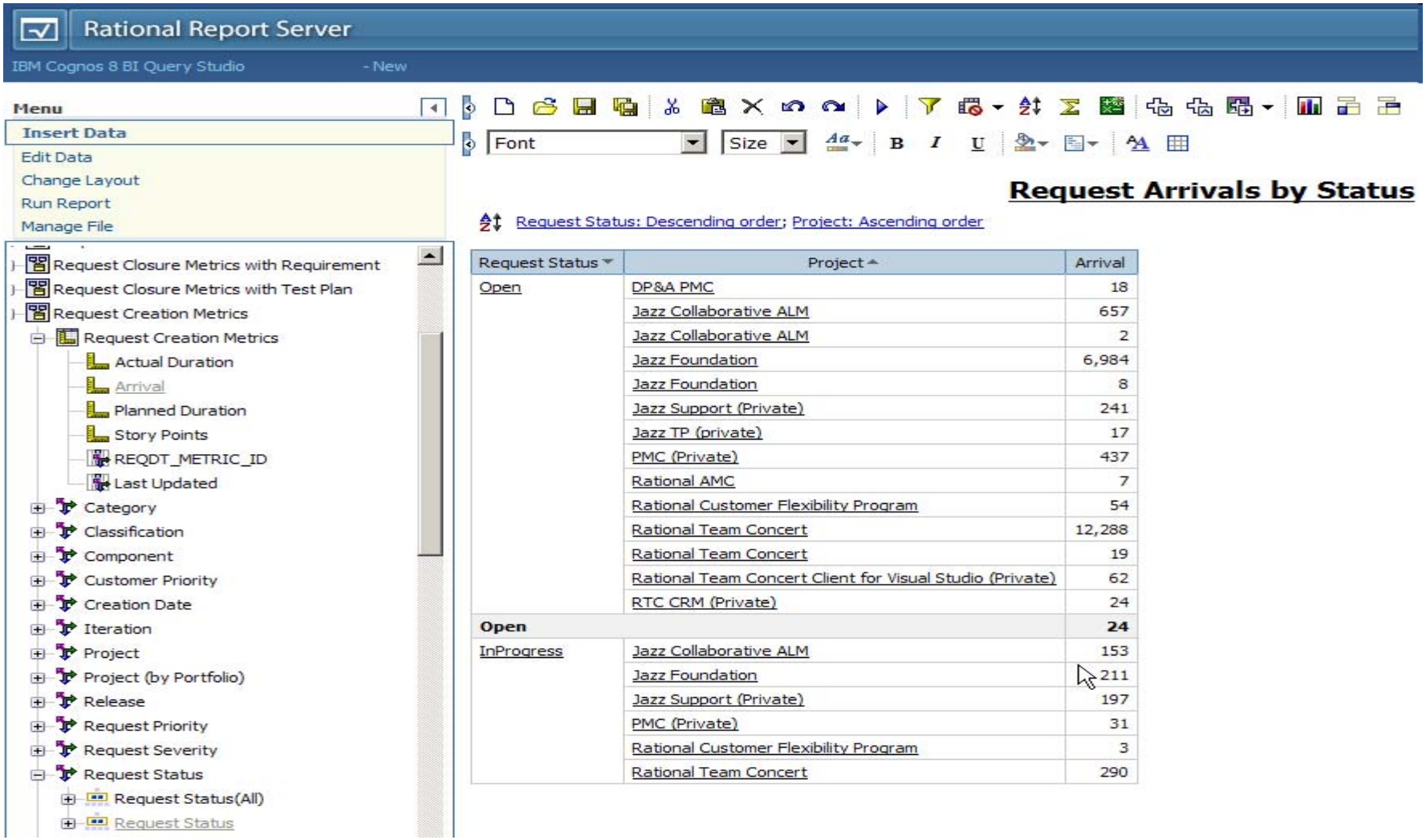
Reporting with IBM Rational

Generating Documents – Rational Publishing Engine

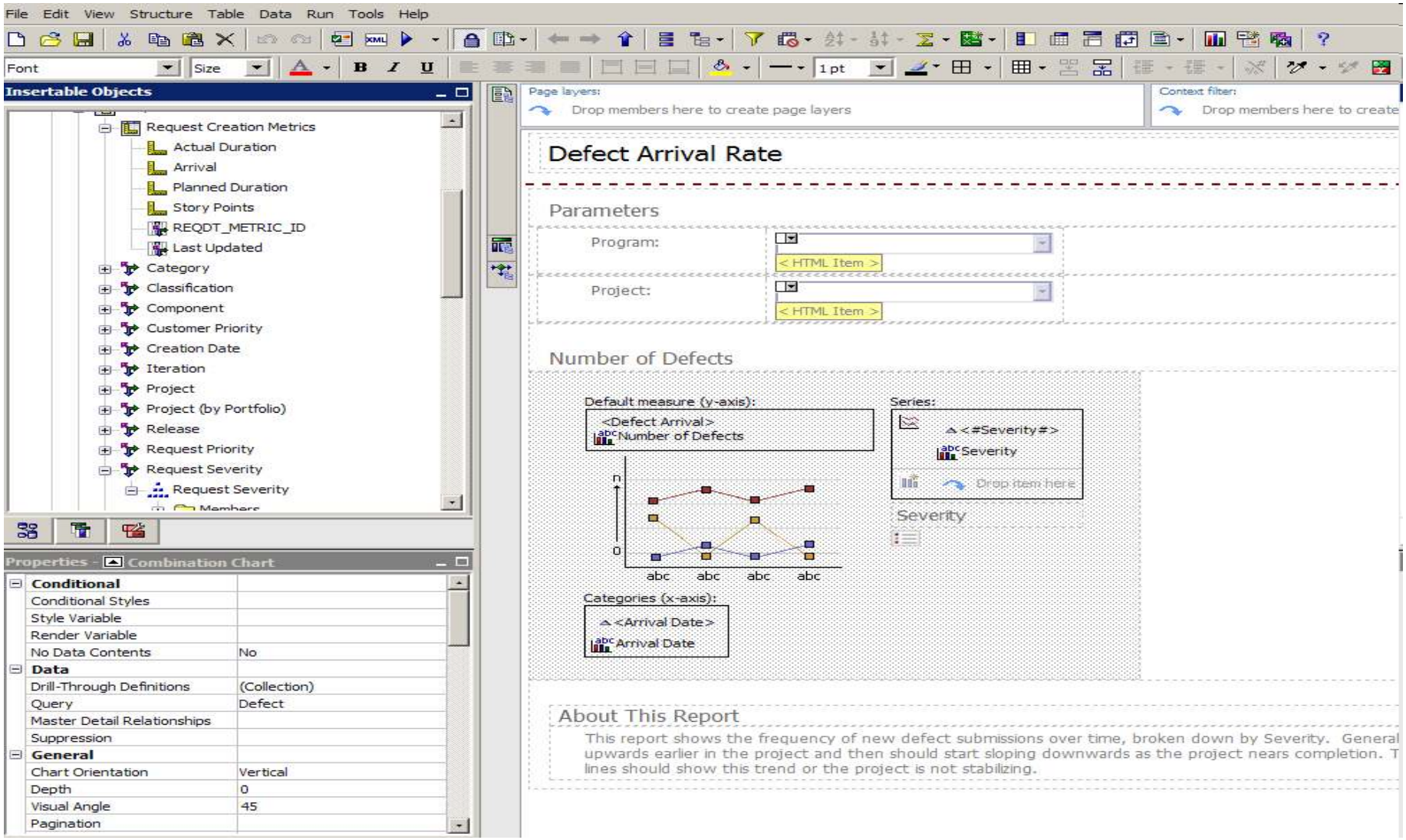
- Generates high quality documents with complete flexibility in formatting
- Generates composite reports containing data from multiple sources
- Supports multiple output formats
- Includes predefined templates for rapid adoption
- Provides an easy-to-use graphical template editing environment for custom report design
- Supports concurrent document generation to multiple target formats from a single template



Rational Reporting for Developer Intelligence – Query Studio



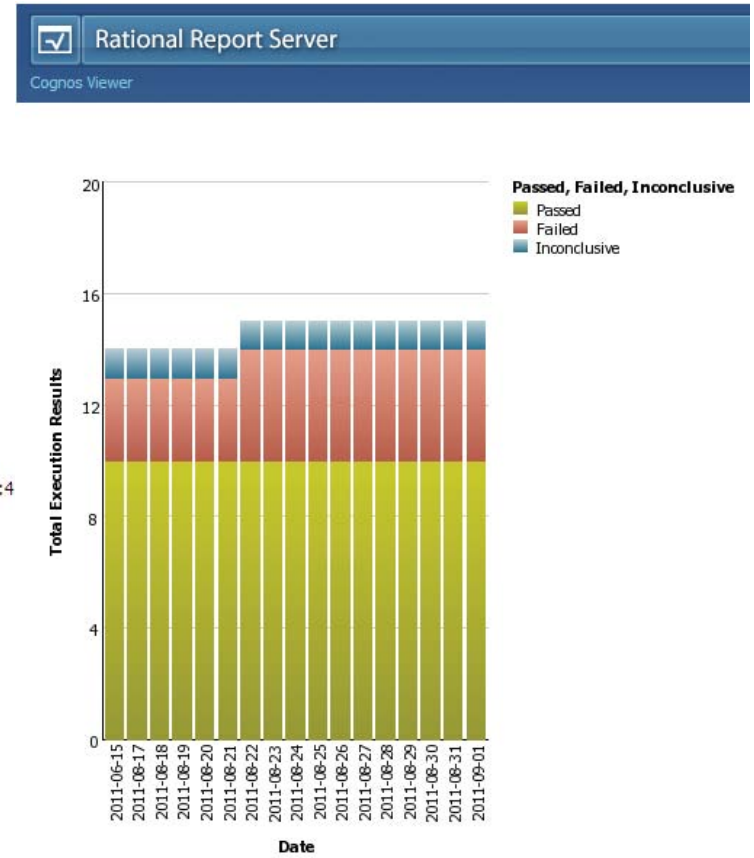
Rational Reporting for Developer Intelligence - Report Studio



Name	Test Plan State	Name
JKE Banking Release 1	New	Allocate dividends by amount and frequency
	New	Allocate Dividends by Percentage
	New	Customers can Nominate an Organization
	New	Dividend Allocation by Percentage
	New	Donation amount limits
	New	Donor dividend allocation conforms to stated criteria
	New	Donors Can Choose to Support an Organization
	New	Donors Choose an Organization
	New	Donors Deposit Money Into a Pooled Assistance Fund
	New	Donors will receive confirmation and receipt
	New	JKE Charity Coordinator responds to online request
	New	Organization must identify how much money is desired
	New	Organization must provide justification for why funds are needed
	New	Organizations can Apply
	New	Organizations may apply with an initial request
	New	Process email requests
	New	Process hard copy requests
	New	Verify dividend transfer frequency
JKE Banking Sprint 1	New	Allocate dividends by amount and frequency
	New	Allocate Dividends by Percentage

Aug 30, 2011 1 1:4

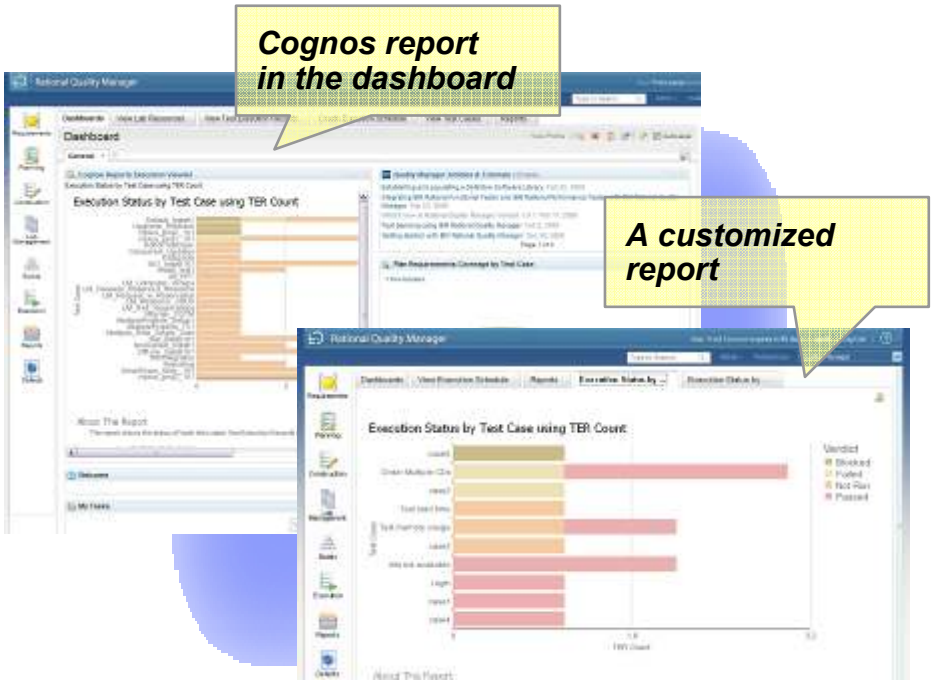
Top Page up Page down Bottom

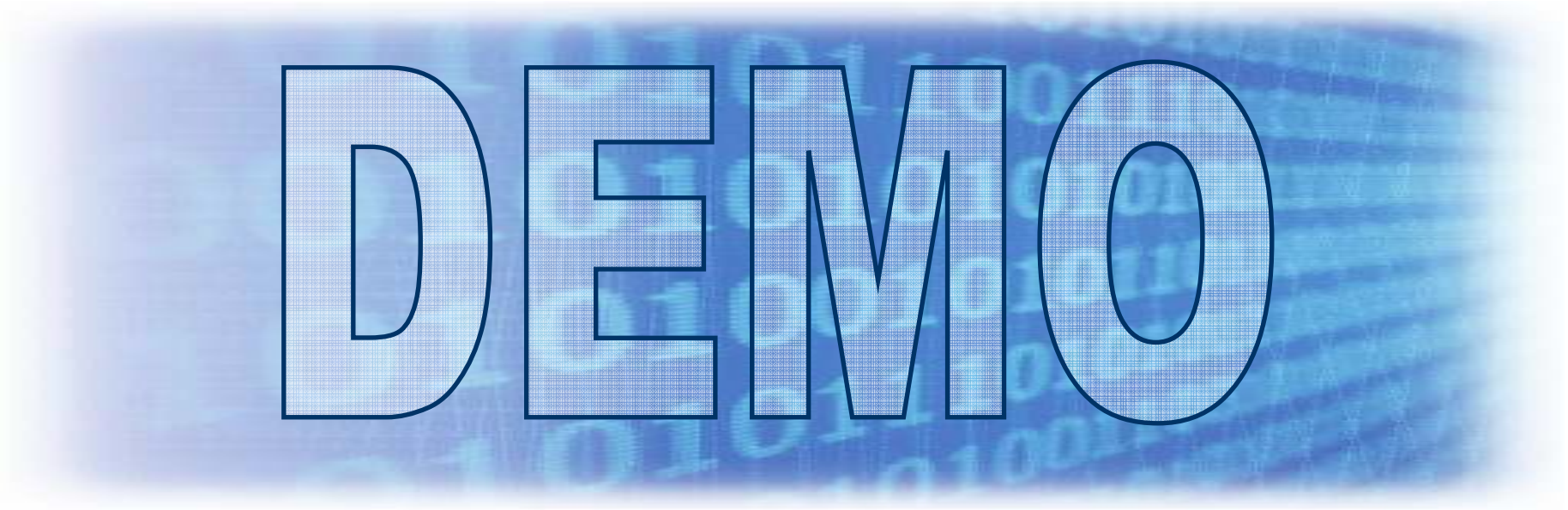


Make informed decisions and proactively change with real-time analysis and actionable reporting

Measure and manage quality, project and team status performance and results

- **Measure developement process** and project outcomes
  - Real-time intelligence based on IT industry best-practice metrics, dashboards and models
- **Inform quality decisions** and drill into issues
  - Alerts and automated analysis focuses owner to take action on root causes
  - **52** out-of-the-box, customizable Cognos test management reports
- **Take real-time action** on relevant quality and project data
  - Proven business intelligence backbone automates collection and analysis to improve lifecycle productivity



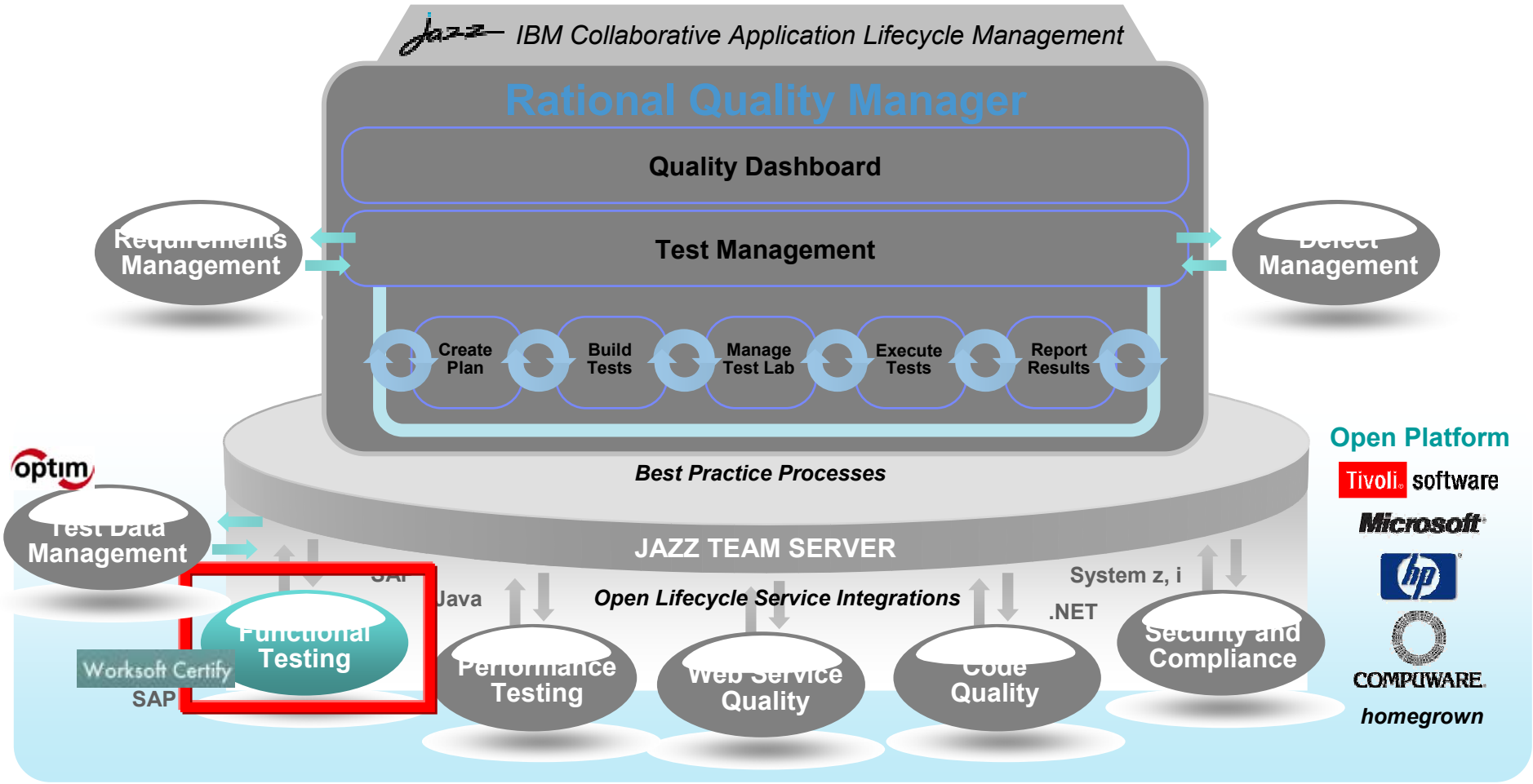


What You'll See:

- Rational Quality Manager

Automated Testing
IBM Rational Functional Tester

Centralized test management offering allowing full lifecycle support across all types of testing and platforms



Test Automation from Wikipedia

- Test automation is the use of software to control
  - the execution of tests
  - the comparison of actual outcomes to predicted outcomes
  - the setting up of test preconditions
  - other test control and test reporting functions

Promise of Test Automation

- Run existing tests on a new version of a program
  - Minimal effort involved in performing regression tests
- Run more tests more often
  - Run more tests in less time
  - Make it possible to run more often
- Perform tests that are difficult / impossible to do manually

Promise of Test Automation

- Better use of resources
  - Machines
  - Skilled testers
- Consistency and repeatability of tests
  - Tests repeated exactly every time
  - Insure consistent standards both in testing and in development
- Reuse of tests

Limitations of Test Automation

- Does not replace manual testing
  - Tests that are run only rarely
  - Where the software is very volatile
  - Tests where the result is easily verified by a human
  - Tests that involve physical interaction

Limitations of Test Automation

- Manual tests find more defects than automated tests
  - A test is most likely to reveal a defect the first time it is run
  - Test execution tools are “re-testing” tools
- Great reliance on the quality of the tests
  - A tool can only identify differences between the actual and expected outcomes
  - Great reliance on the correctness of the expected outcomes

Limitations of Test Automation

- Test automation does not improve effectiveness
  - Automation can eventually improve the efficiency only
- Test automation may limit software development
  - Automated tests take more effort to set up than manual tests
- Tools have no imagination
  - What if expected outcomes are wrong
  - What if unexpected events happen

A Fairy Tale

- Once upon a product cycle, there were four testers who set out on a quest to test software…

Warning: The fairy tale you are about to read is a fib—but it's short, and the moral is true.

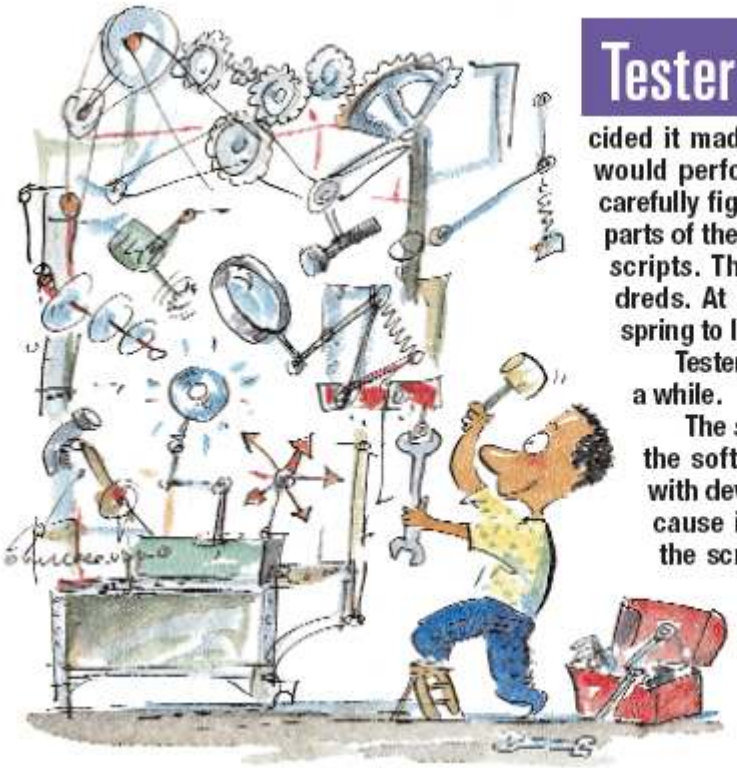
Once upon a product cycle, there were four testers who set out on a quest to test software.

Tester 1 started hands-on testing immediately, and found some nice bugs. The development team happily fixed these bugs, and gave Tester 1 a fresh version of the software to test. More testing, more bugs, more fixes.

Tester 1 felt productive, and was happy—at least for a while.

After several rounds of this find-and-fix cycle, he became bored and bleary-eyed from running virtually the same tests over and over again by hand. When Tester 1 finally ran out of enthusiasm—and then out of patience—the software was declared "ready to ship."

Customers found it too buggy and bought the competitor's product.

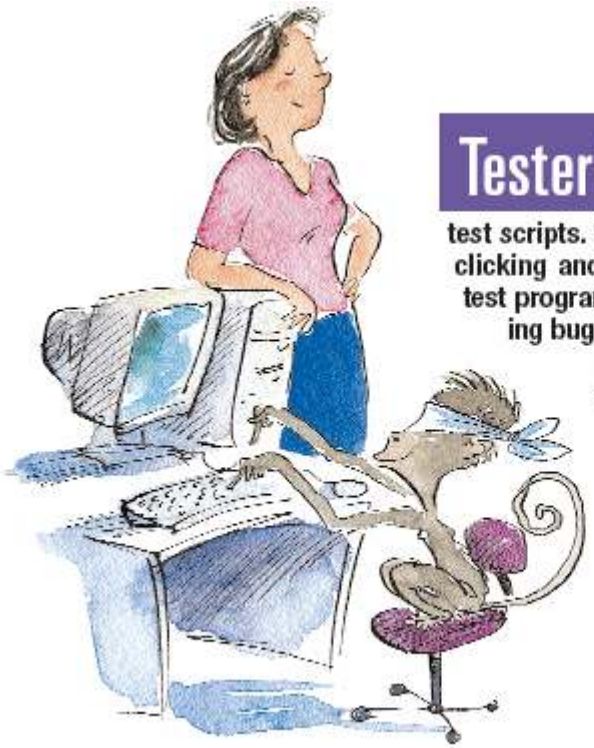


Tester 2 started testing by hand, but soon decided it made more sense to create test scripts that would perform the keystrokes automatically. After carefully figuring out tests that would exercise useful parts of the software, Tester 2 recorded the actions in scripts. These scripts soon numbered in the hundreds. At the push of a button, the scripts would spring to life and run the software through its paces.

Tester 2 felt clever, and was happy—at least for a while.

The scripts required a lot of maintenance when the software changed. He spent weeks arguing with developers to stop changing the software because it broke the automated tests. Eventually, the scripts required so much maintenance that there was little time left to do testing.

When the software was released, customers found lots of bugs that the scripts didn't cover. They stopped buying the product and decided to wait for version 2.0.



Tester 3 didn't want to maintain hundreds of automated test scripts. She wrote a test program that went around randomly clicking and pushing buttons in the application. This "random" test program was hypnotic to watch, and it found a lot of crashing bugs.

Tester 3 enjoyed uncovering such dramatic defects, and was happy—at least for a while.

Since the random test program could only find bugs that crashed the application, Tester 3 still had to do a lot of hands-on testing, getting bored and bleary-eyed in the process. Customers found so many functional bugs in the software when it was released that they lost trust in the company and stopped buying its software.

Tester 4 began with hands-on, exploratory testing to become familiar with the application—and used the knowledge gained during the hands-on testing to create a very simple behavioral model of the application. Tester 4 then used a test program to test the application's behavior against what the model predicted. The behavioral model was much simpler than the application under test, so it was easy to create. Since the test program knew what the application was supposed to do, it could detect when the application was doing the wrong thing.

As the product cycle progressed, developers wrote new features for the application. Tester 4 quickly updated the model, and the tests continued running. The program ran day and night, constantly generating new test sequences. Tester 4 was able to run the tests on a dozen machines at once and get several days of testing done in a single night.

After several rounds of testing and bug fixes, Tester 4's test generator began to find fewer bugs. Tester 4 upgraded the model to test for additional behaviors and continued testing. Tester 4 also did some hands-on testing and static automation for those parts of the application which were not yet worth modeling.

When Tester 4's software was released, there were very few bugs to be found. The customers were happy. The stockholders were happy.

And Tester 4 was happy.

Maximize your investment in test automation

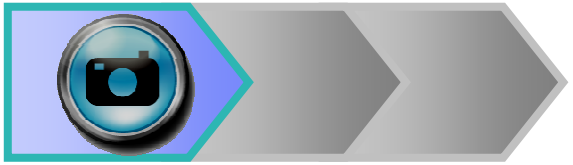
With IBM Rational Functional Tester

- Achieve success quickly and minimize maintenance
  - Simplified natural language scripting with Storyboard testing
  - Eclipse based or Visual Studio .net
  - Easy to learn
  - Maximize reuse
- Complete test coverage
  - Supports testing for Java, Web, Visual Basic .Net, SAP, Siebel, Web 2.0, Power Builder and Terminal Based applications
  - Ability to support custom controls

Effective Test Automation

Recording an automated script

- Automated script capture
  - Test scripts are recorded on the fly, as user navigates application
  - Verification points are inserted to validate system response
  - Test data can be specified and parameters created while recording



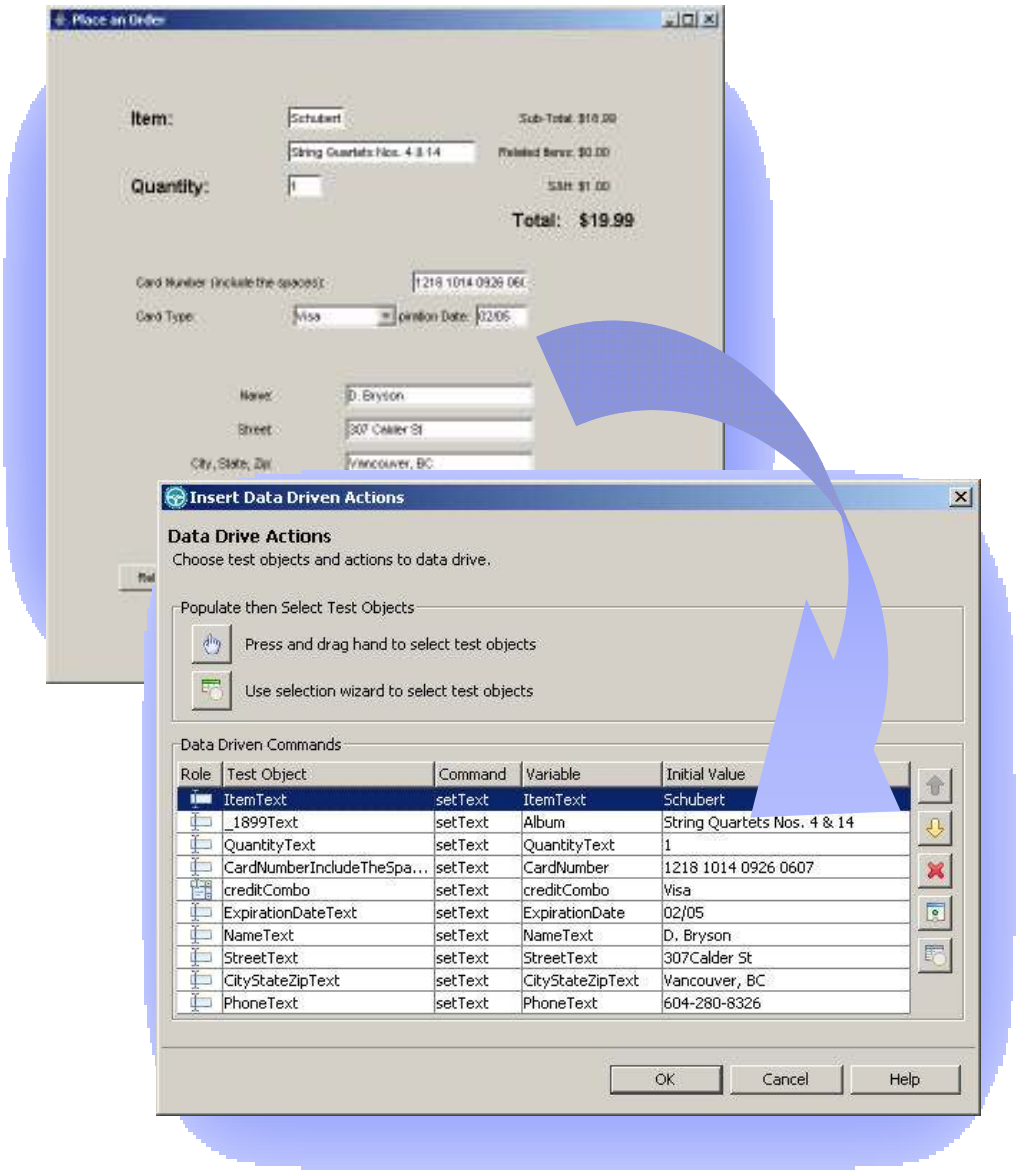
Functional Tester Highlights

- Tool mentors and process advisors accelerate training
- Broad environment support
- Create data driven tests without coding
- Static data and properties verification
- Dynamic data validation without coding

Recording Scripts

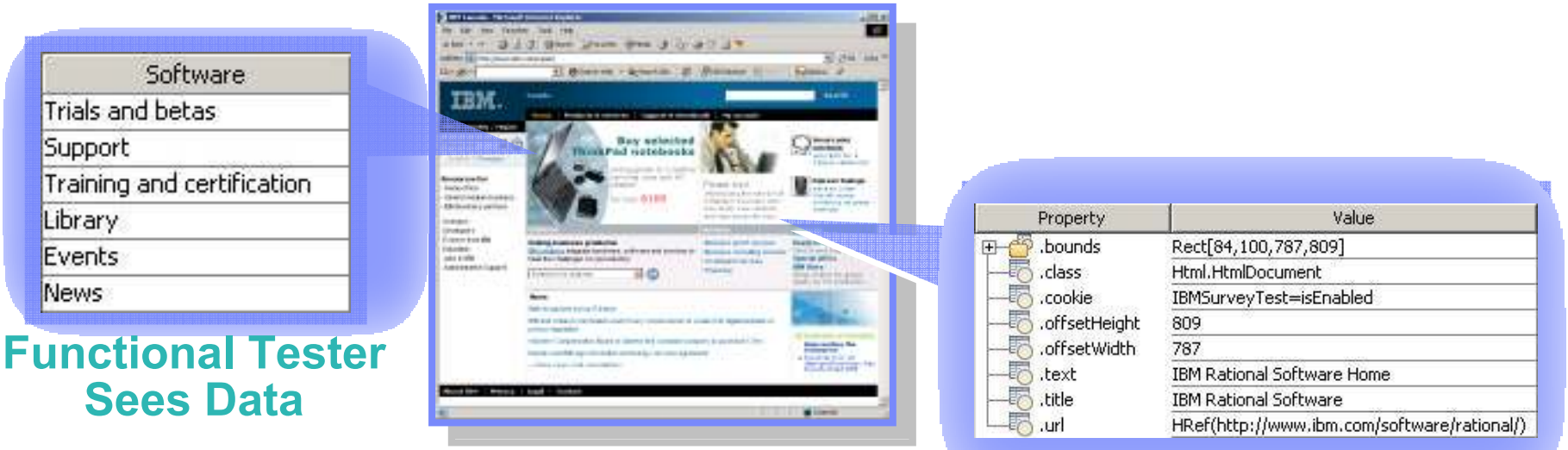
Data Driven Testing

- **Data Driven Testing**
  - Separates test data from test script
  - Enables a single script to run multiple tests by using multiple data sets
- **Wizard driven process**
  - No programming involved
  - Import data from external sources



Recording Scripts

Verification Points



- **Automated Validation**
  - Functional Tester captures data and properties that can be invisible to users
  - During script execution, current results are compared to stored baselines
  - Discrepancies are flagged and reported to user in an HTML based test log

Recording Scripts

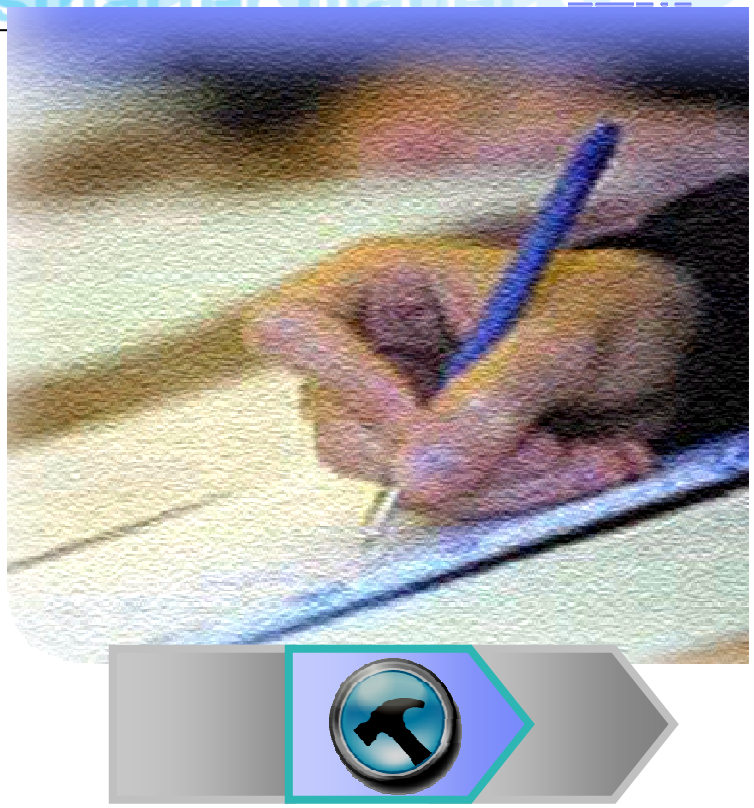
Validating Dynamic Data

- Use pattern matching technique to verify dynamic data and create robust tests
- E.g. Instead of validating “Order ID 230”, validate “Order ID ###”
  or Order ID 2##, etc.
- This allows for a wide variety of acceptable responses as well as restrictions on acceptable responses when validating the application’s behavior

Effective Test Automation

Enhancing Scripts

- Enhancing Scripts with basic coding extends their value and reach
  - VB.net or Java code is added to perform a variety of functions
  - Typical Modifications: Conditional branching, datapooling, refactoring, adding additional dynamic data patterns



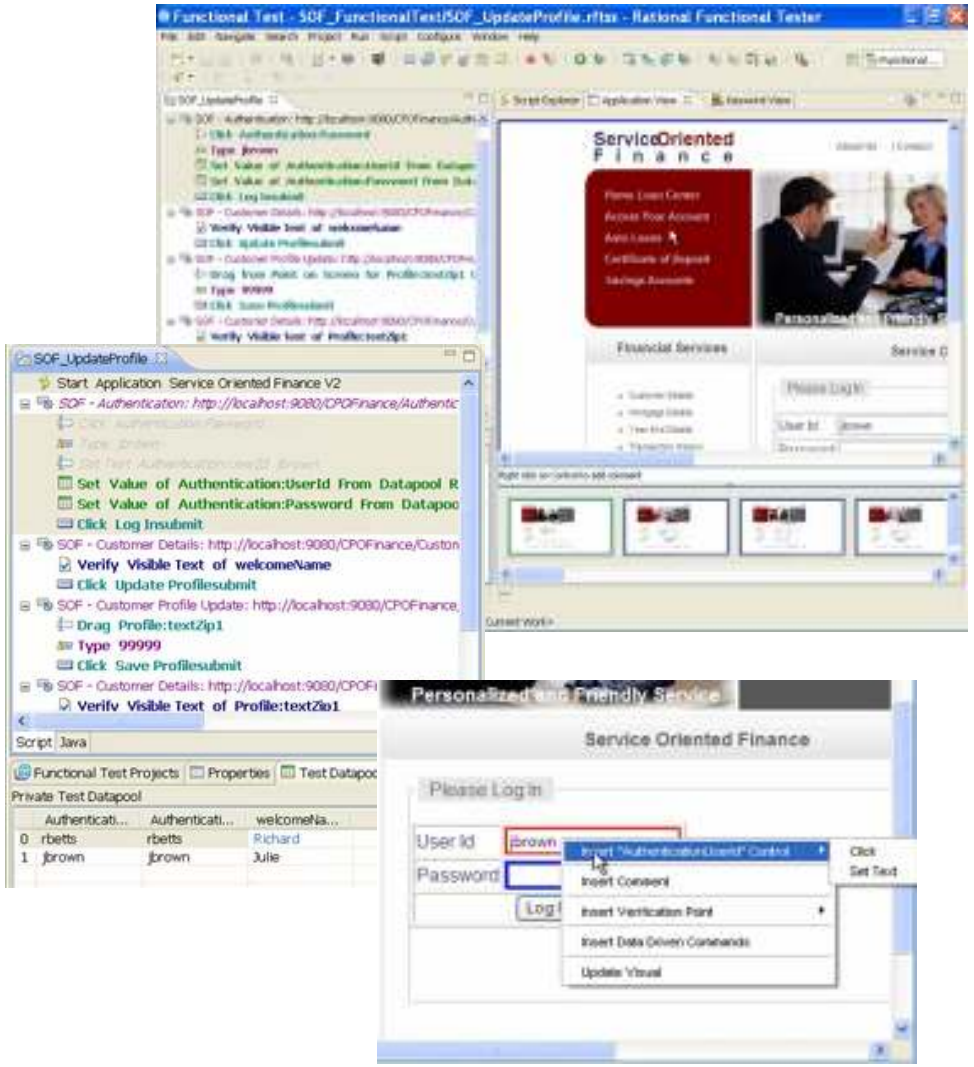
Functional Tester Highlights

- Pure Java and VB.Net provides flexibility
- Professional debugger
- Central object map to minimize rework
- RTC-ready for version control
- Dynamic data validations without coding

Enhancing Scripts

Storyboard testing simplification

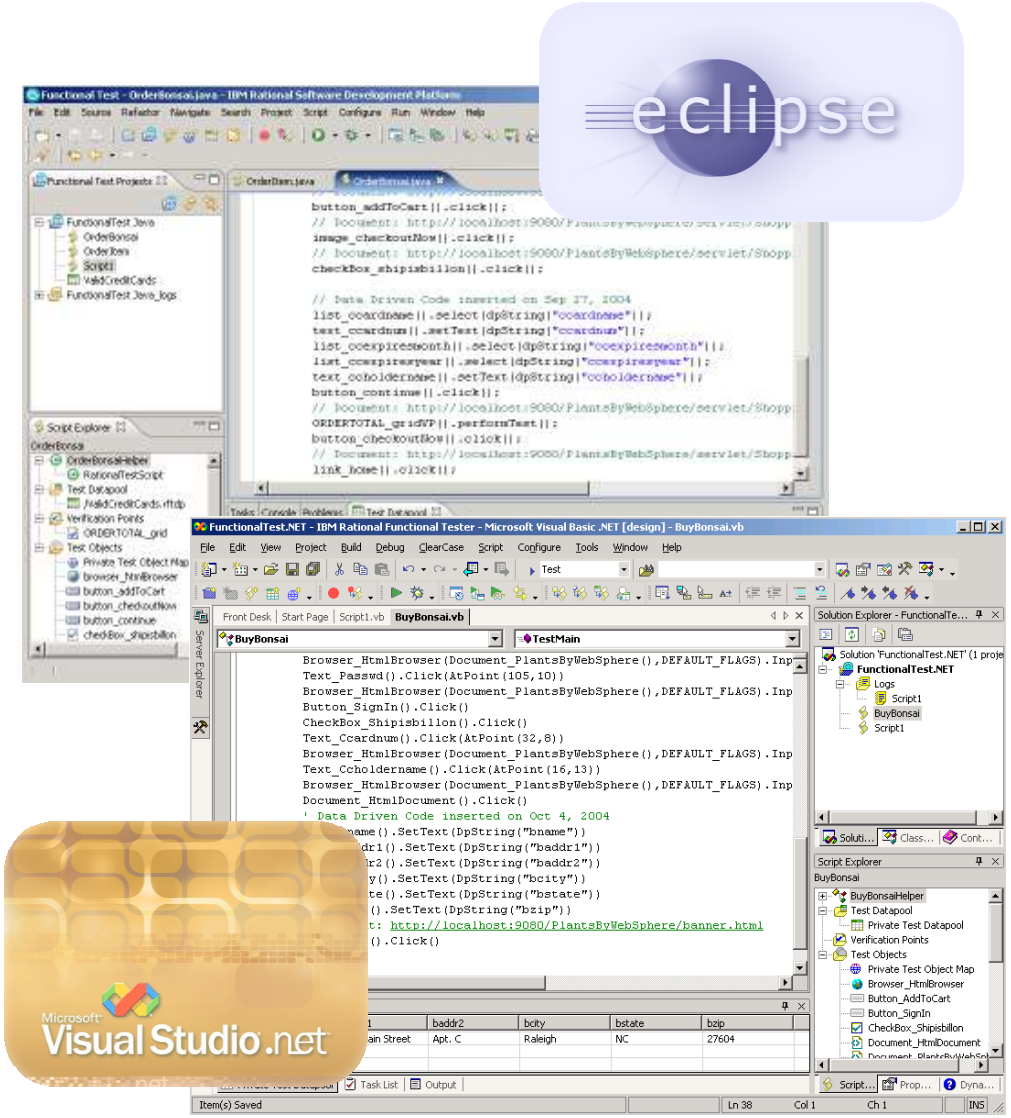
- Enable novice and professional testers alike to easily understand and communicate test flows
- Natural language script view
- Storyboard test visualization
  - Application snapshots are captured and displayed as thumbnails
  - Insert verification points
  - Maintain test datapools



Enhancing Scripts

Powerful, Professional Debugger

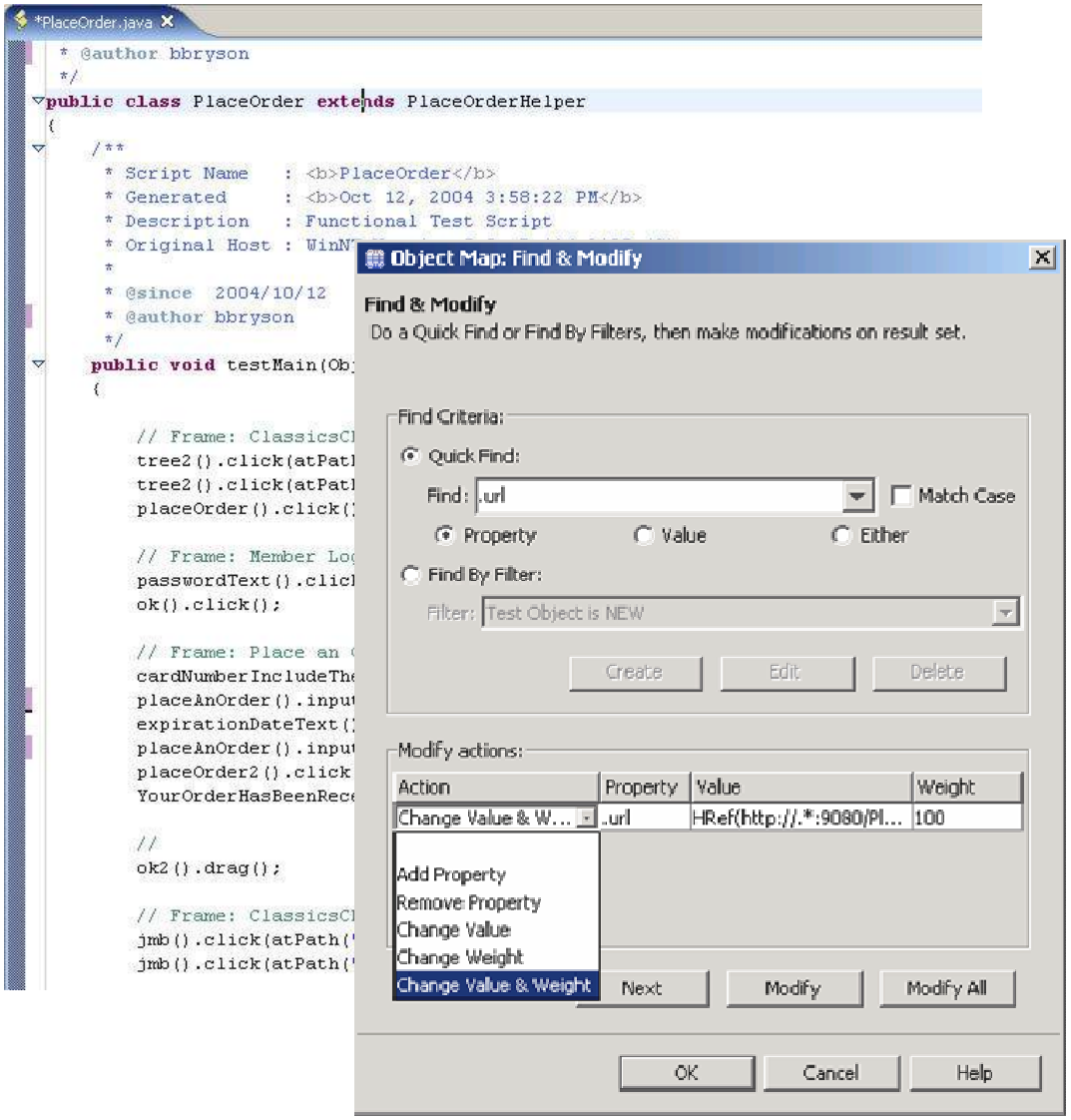
- Functional Tester offers two development environments
  - Eclipse based IBM Software Development Platform
  - Visual Studio .net
- Both environments offering powerful debugging features
  - Code assist editors
  - Step debugging
  - Variable watches
  - More…



Enhancing Scripts

Object map editing flexibility

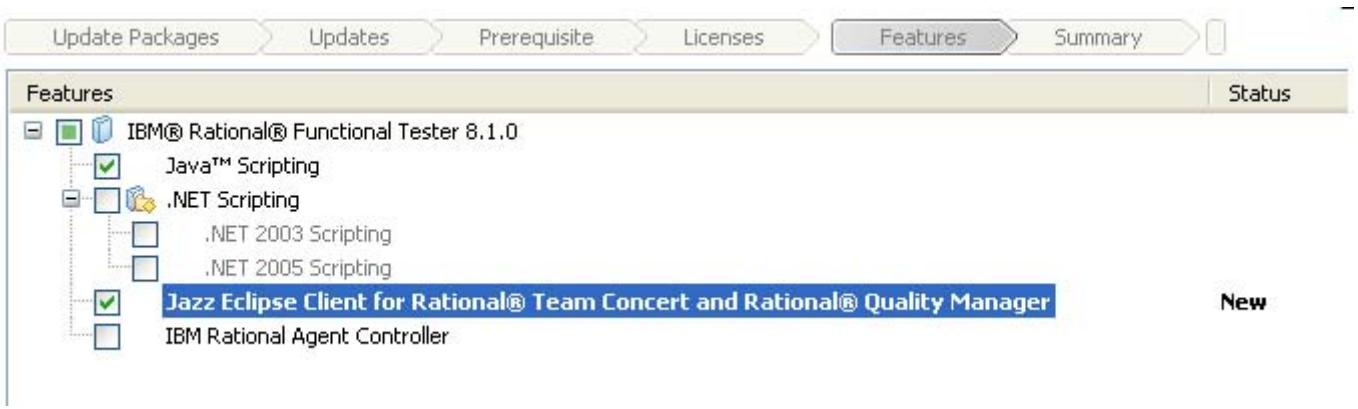
- Script Maintenance can outpace script development as the volume of tests grows
- Functional Tester includes an Object Map update tool which enables batch updates to a centralized object map
  - Reduces time spent fixing individual scripts
  - Frees up more time for script development



Enhancing Scripts

Jazz Team Collaboration

- Rational Team Concert enabled
  - Assets maybe managed within Jazz SCM
  - RFT user can use standard source control features available in RTC – Install RFT and RTC in same package group
  - RTC 2.0 as an optional feature in RFT



Effective Test Automation

Executing Tests

- Scripts are executed, discrepancies are noted
  - Scripts are executed and test logs created
  - Test logs are highlight differences between actual and expected results

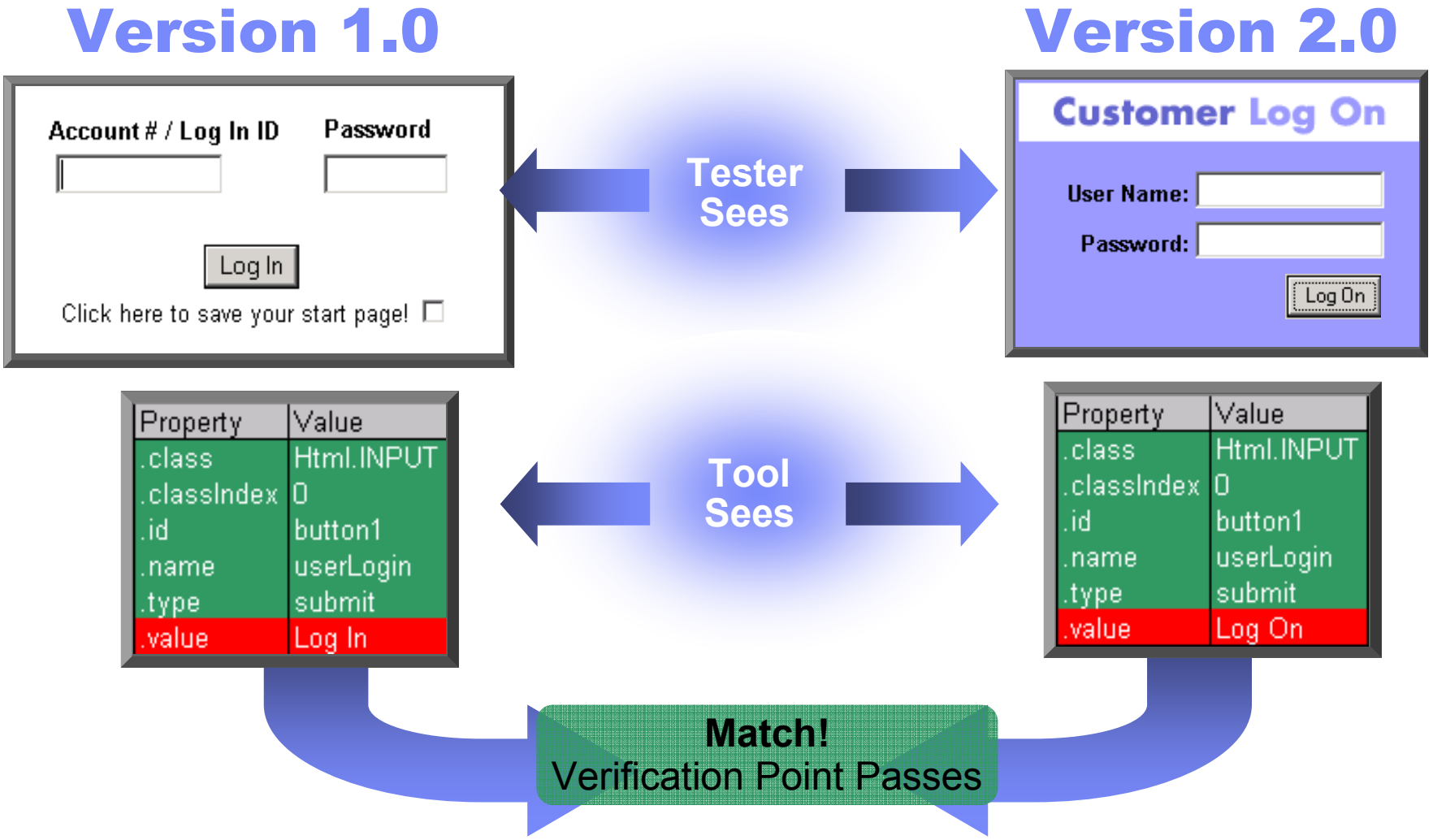
- Key considerations when executing scripts
  - Reliable playback with ScriptAssure
  - Remote and local playback on various platforms

Functional Tester Highlights

- Central object map with ScriptAssure object weighting
- Flexible results reporting
- Dynamic data validations without coding

Reduce Test Script Maintenance

Reliable Playback with Script Assure

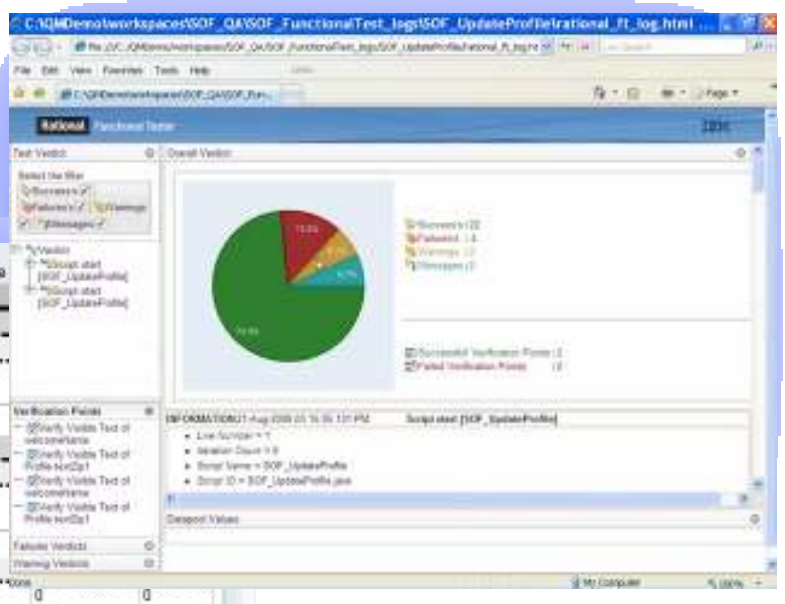


Executing Scripts

Reporting fits your organization's needs

- Following execution results can be viewed and stored in many ways:
  - Viewed and stored in an XML or HTML format
  - Centralized in Rational Quality Manager
  - Adobe® PDF 7 and 8 documents

Functional Tester XML

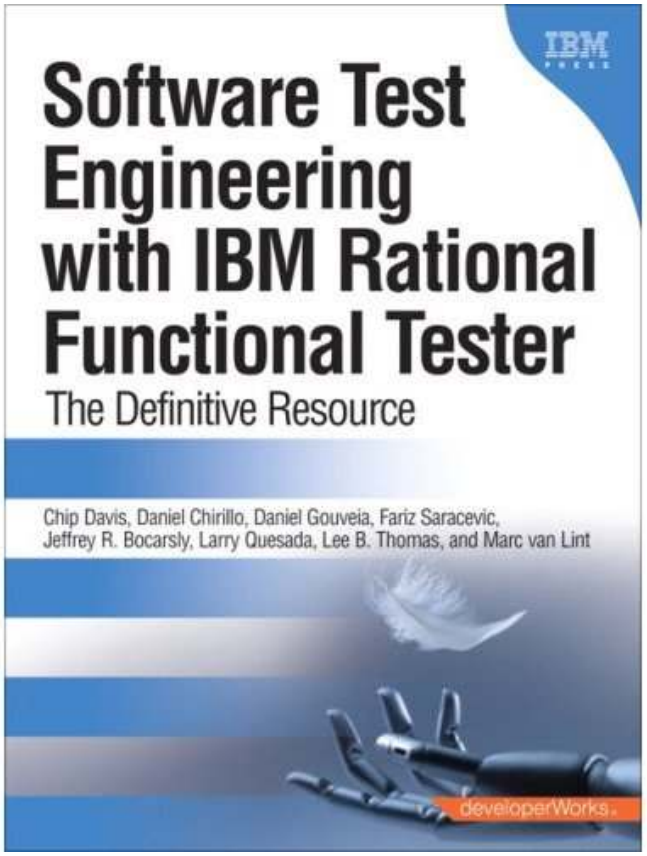


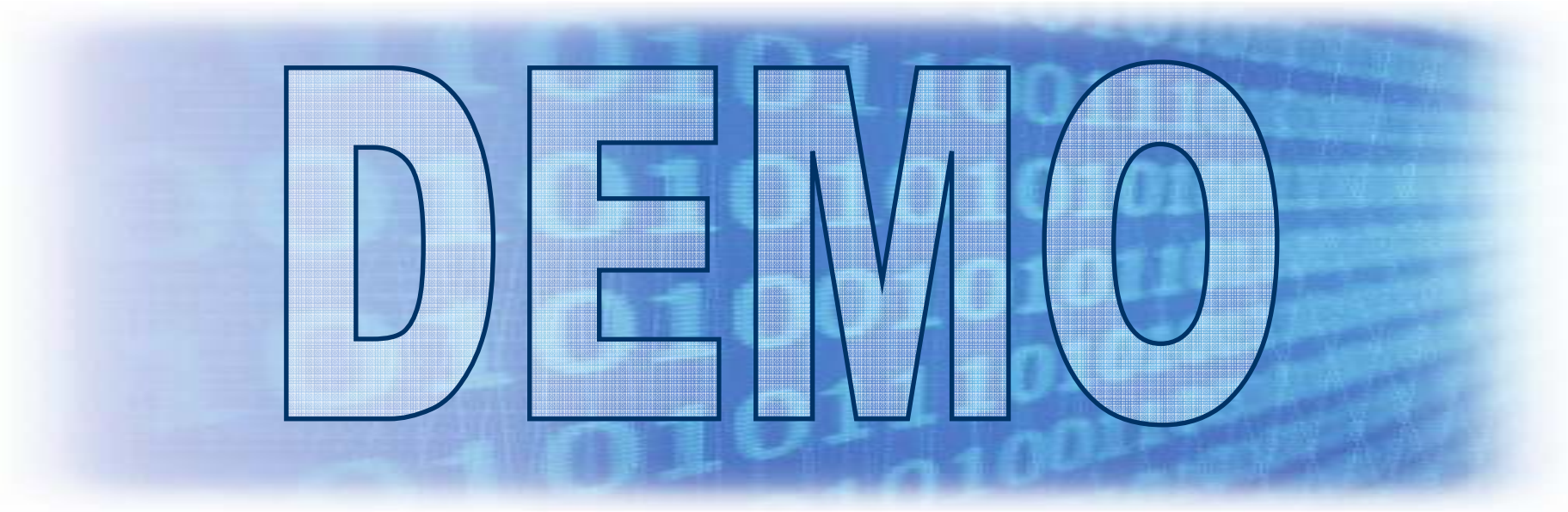
Rational Quality Manager

Test Iteration	Tester	Configuration	Test case	EWI ID	Weight	Points Passed	Points Failed	Points Blocked	Points Inconclusive		
M0					320	160	0	0	0		
	ADMIN				95	95	0	0	0		
		SAMPLE_AMDx86_WinXP_IE			95	95	0	0	0		
			Accessibility Web UI Test 3		95	95	0	0	0		
				3	95	95	0	0	0		
	donald				65	0	0	0	0		
		SAMPLE_Intelx86_WinXP_Firefox			45	0	0	0	0		
			Performance Web Services Test 2		45	0	0	0	0		
				2	45	0	0	0	0		
		SAMPLE_x86_Linux_Firefox			20	0	0	0	0		
			Accessibility Web UI Test 4		20	0	0	0	0	0	0
				4	20	0	0	0	0	0	0
	larry				65	65	0	0	0	0	5
		SAMPLE_Intelx86_WinXP_Firefox			65	65	0	0	0	0	5
			Functionality Security Test 8		65	65	0	0	0	0	5
				8	65	65	0	0	0	0	5

Supporting resources

- IBM Internet – Technotes
- DeveloperWorks – Forum
- Publication – Software Test Engineering with IBM Rational Functional Tester



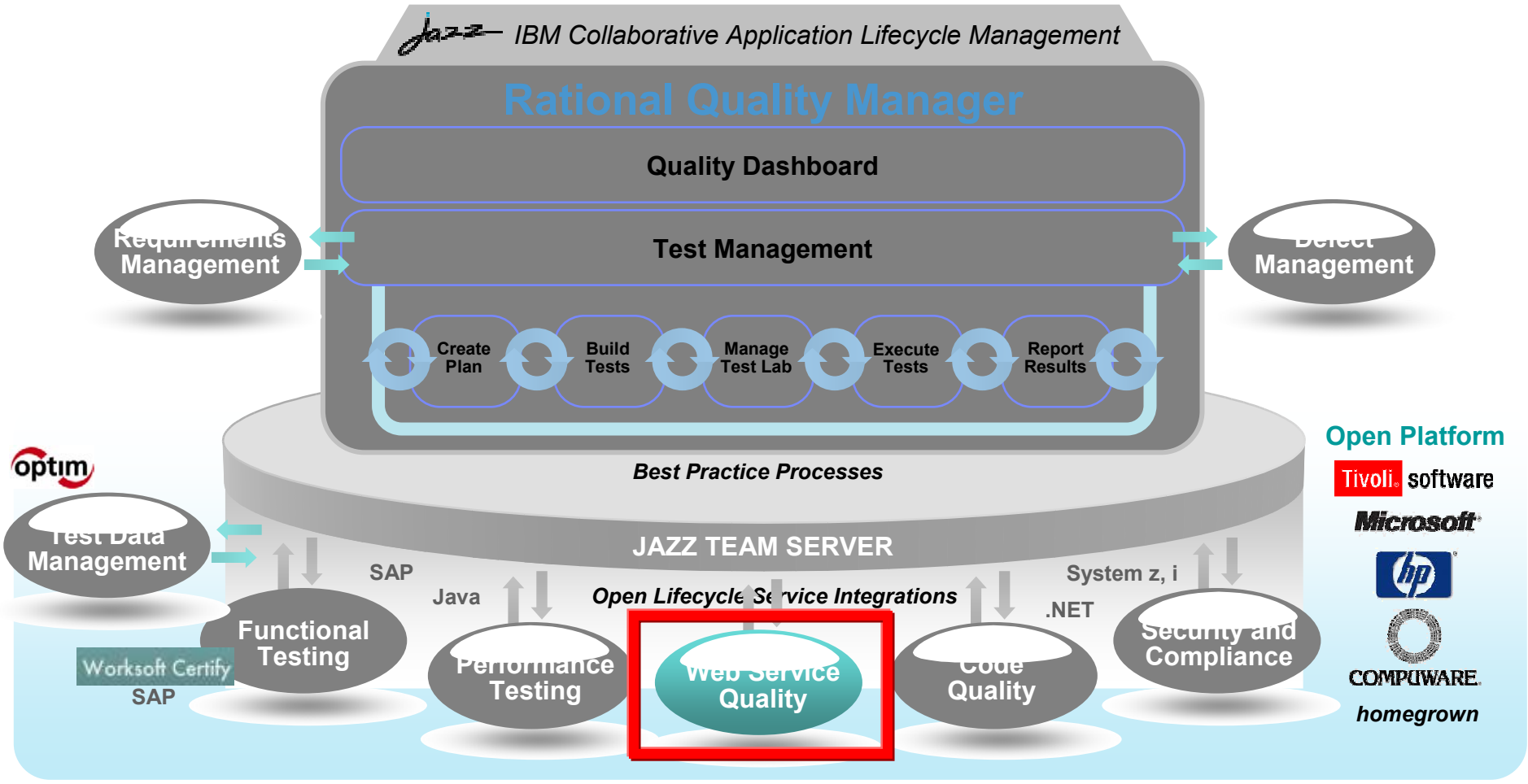


What You'll See:

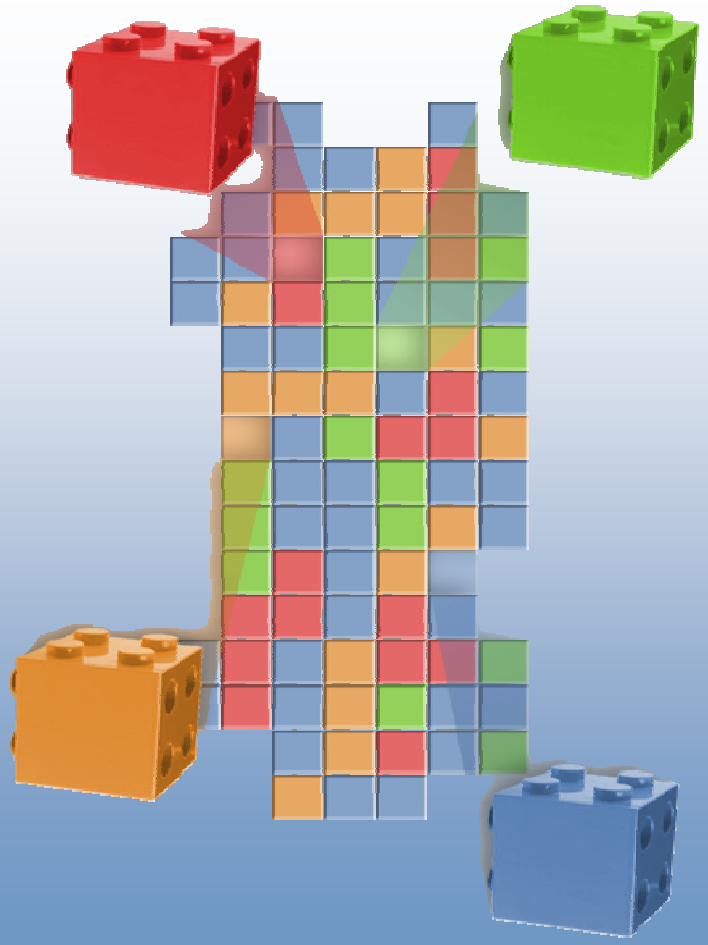
- Rational Functional Tester

Service Testing
Rational Service Tester

Centralized test management offering allowing full lifecycle support across all types of testing and platforms



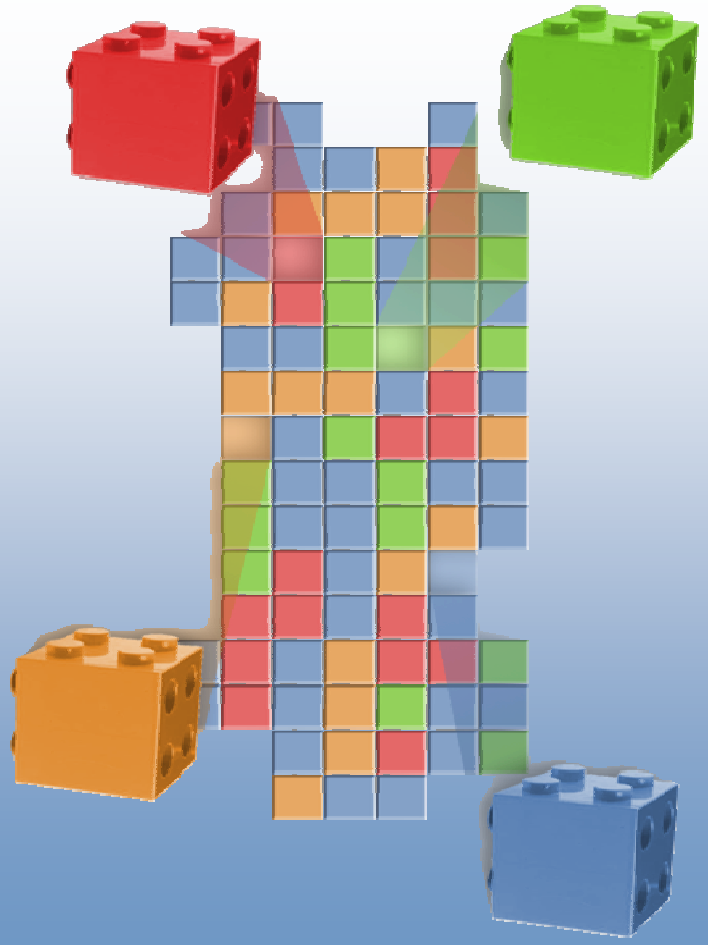
SOA: Service Oriented Architecture Definitions



To the IT Executive

Flexible applications built upon **re-usable** building blocks that are **easily connected**

SOA: Service Oriented Architecture Definitions



To the IT Executive

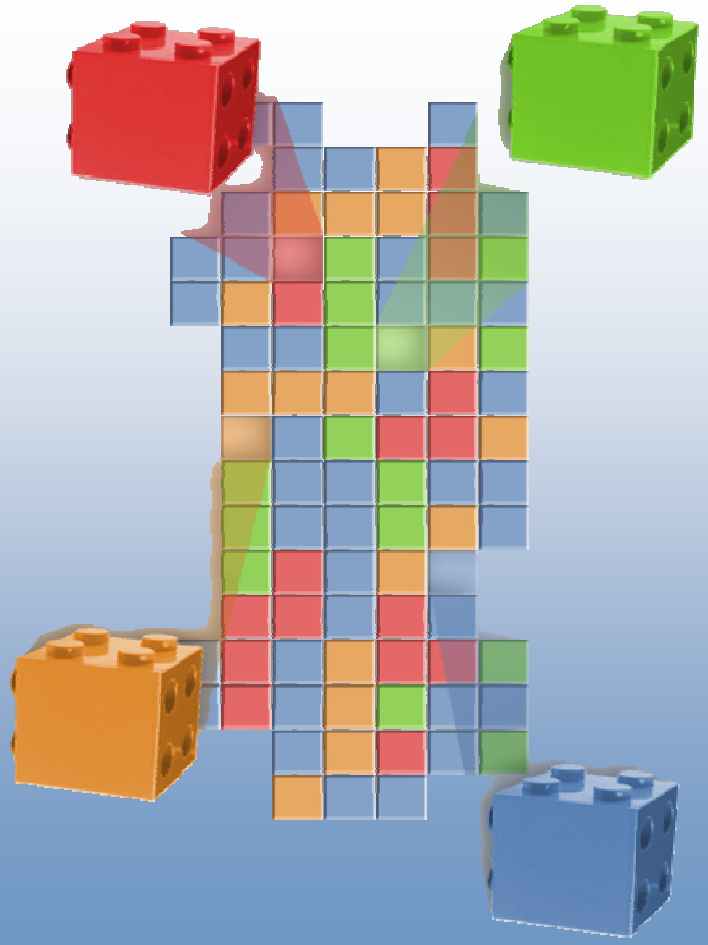
Flexible applications built upon **re-usable** building blocks that are **easily connected**

To the Developers and Testers

Web Services.

Period.

SOA: Service Oriented Architecture Definitions



To the IT Executive

Flexible applications built upon **re-usable** building blocks that are **easily connected**

To the Software Architect

An IT **architectural style** which assembles loosely coupled distributed services to implement a business process

To the Developers and Testers

Web Services.

SOA: Implications for Quality Management

To the IT Executive

Flexible applications built upon **re-usable** building blocks that are **easily connected**

Validate Business Process

Challenges

- **Identifying test cases**
- **Managing Data Complexity**
  - **Requirements, Test Cases, Defects**
- **Ensuring optimal test & configuration coverage**

SOA: Implications for Quality Management

To the Developers and Testers

Web Services.

→ **Validate Web Services**

Challenges

- **No user accessible interface**
- **Multiple test case data cases per test**
- **Service interaction testing**

SOA: Implications for Quality Management

To the Software Architect

An IT **architectural style** which assembles loosely coupled distributed services to implement a business process

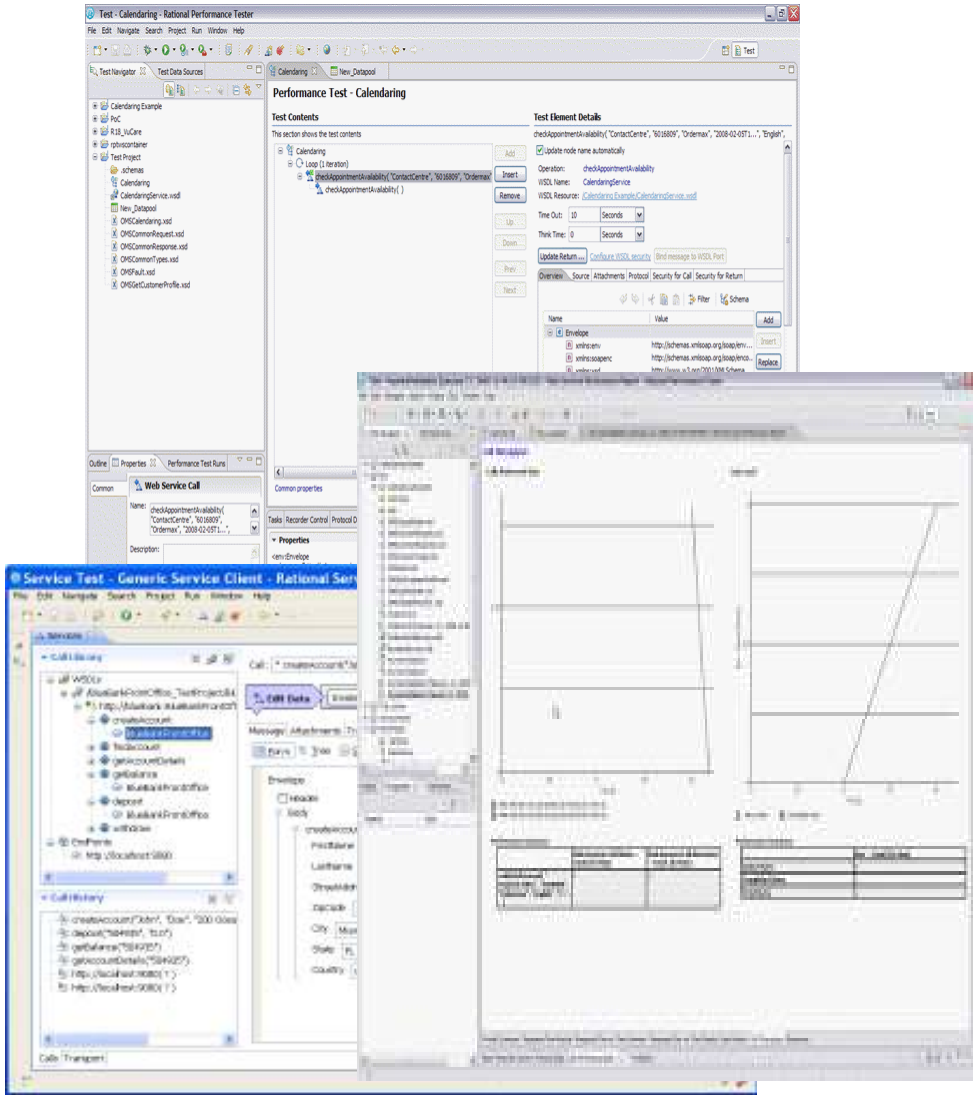
Validate Infrastructure

Challenges

- **Ensuring service operability post deployment**
- **Service upgrade & interoperability management**
- **Service Performance**

Rational Service Tester for SOA Quality

- Used to test web services
- Key features
  - Generic Web Services Client
  - XML editing, viewing
  - WSDL/Schema validation
  - Messaging and logging
  - Load and stress functions
  - Data driven testing
  - Java scripting
  - Automated Response validation
  - Performance Testing and Analysis

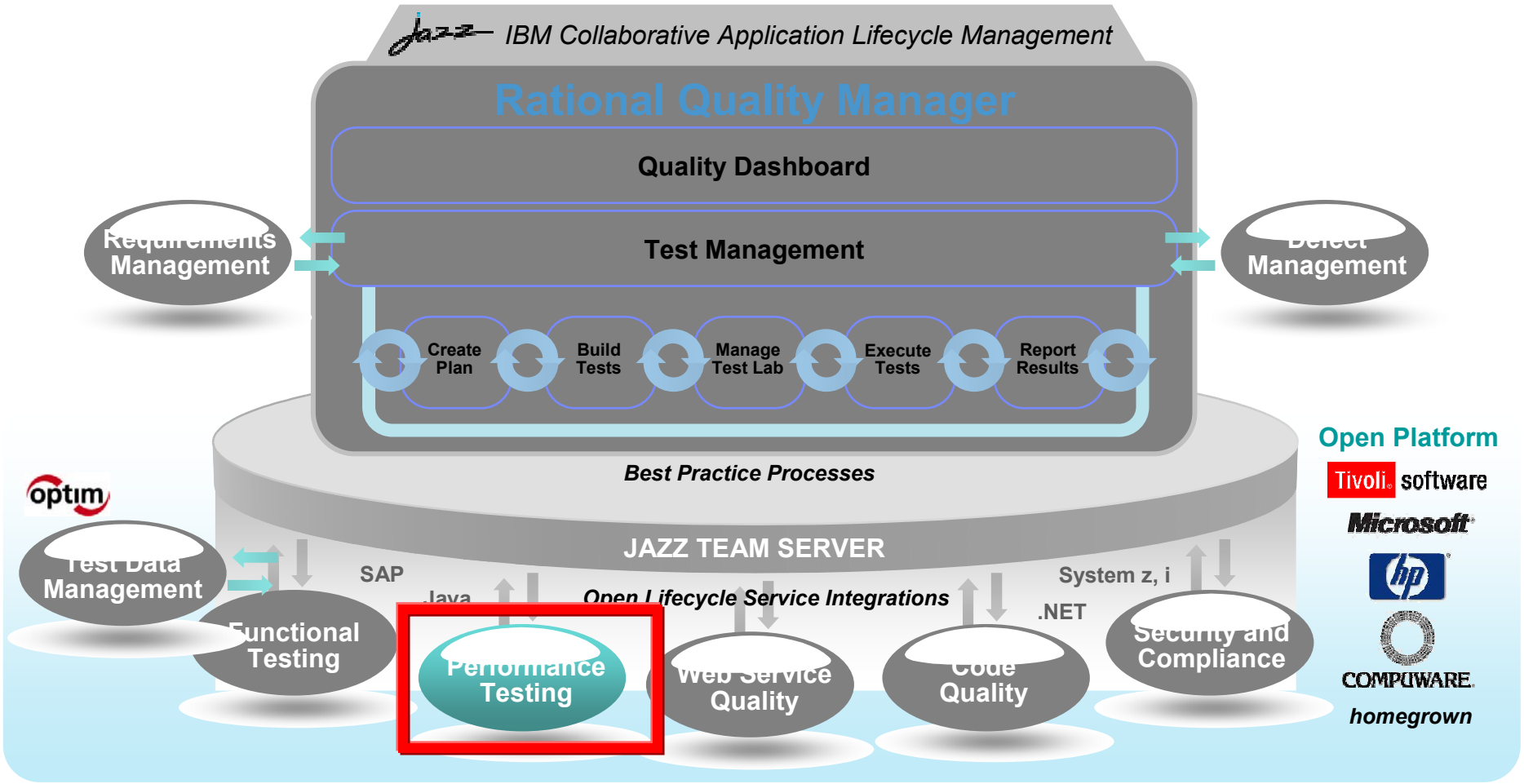


Rational Service Tester for SOA Quality

	Features	Benefits
Packaging	• Performance and functional testing personas • Monitoring/Response Time Breakdown Package	• Improved consumability of functional testing capabilities • Improved visibility and support, and leverage value of performance problem determination features
Environment Support	• Support for additional WS-* standards • Text / JSON message formats • Support for IPv6	• Extend the range of supported SOA environments • Meet government requirements for IPv6 support
Enterprise readiness	• Improved support for multi-day runs with the ability to capture and process large volume of performance measurements	• Ability to address larger and more complex performance test opportunities
Usability	• Universal Service Test Client • Improved functional testing capabilities (creation, execution, reporting)	• Simple and unique user experience to create tests for all supported protocols • Improved consumability of functional testing capabilities
Product Integrations	• Support for Rational Quality Manager • Support for Rational Test Lab Manager	• Support quality throughout the life cycle through integration with Quality Management and Lab Management solutions

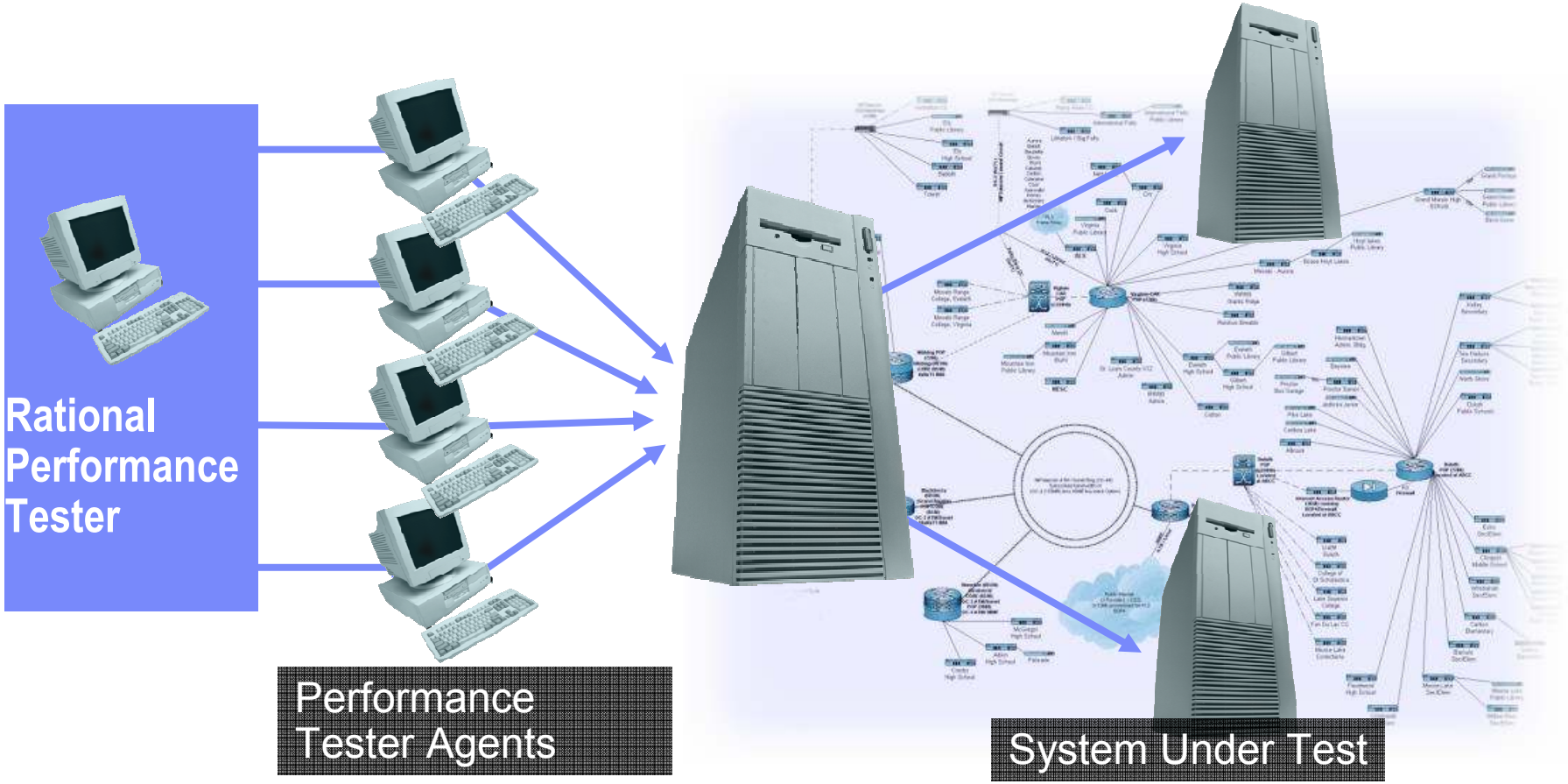
Performance validation
IBM Rational Performance Tester

Centralized test management offering allowing full lifecycle support across all types of testing and platforms



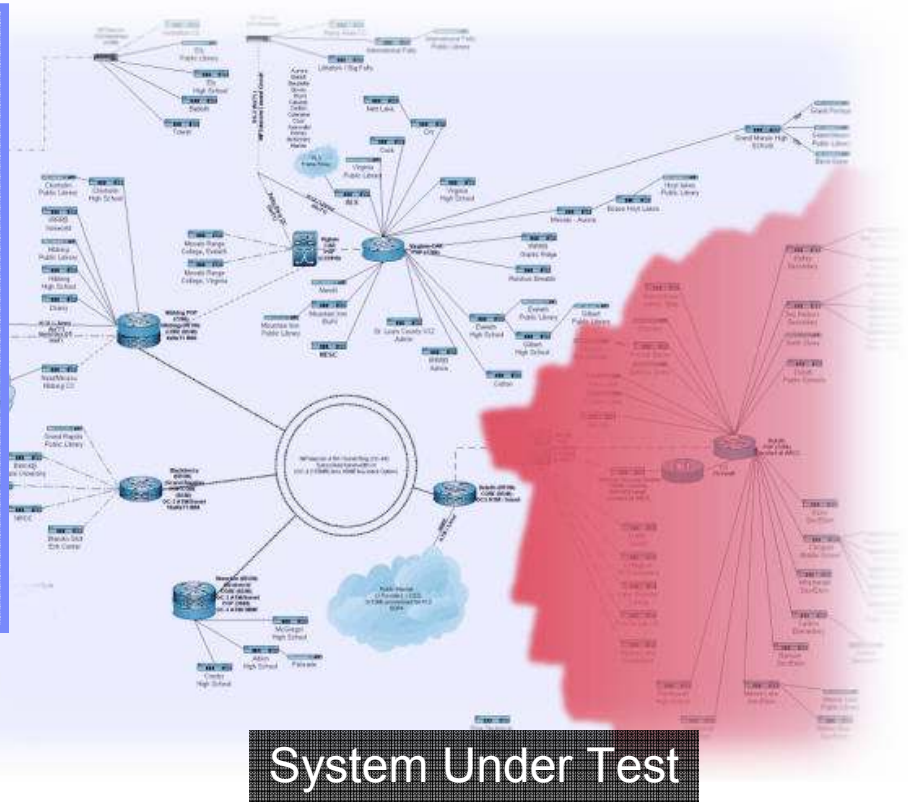
What Is Performance Testing?

- The process of exercising an application by emulating actual users with a load generation tool for the purpose of finding system bottlenecks



Why do Performance Testing?

- Because a break at any point in your system means your customers are not getting the service you think they are



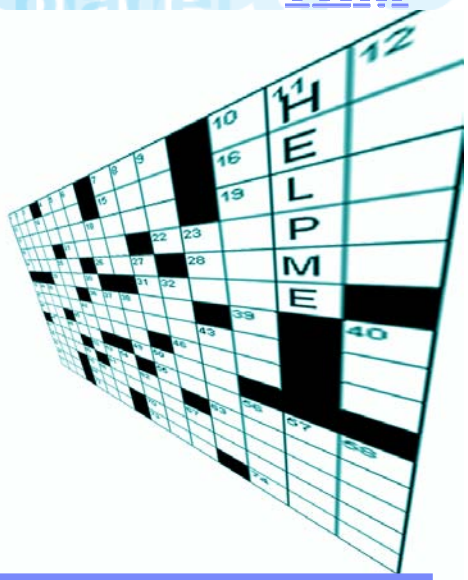
Performance Testing with IBM Rational Performance Tester

Test automation for the novice and the professional

- **IBM Rational Performance Tester**
  - Performance problem identification and diagnosis for Web, SAP, 3270, Siebel, Oracle and Citrix based applications

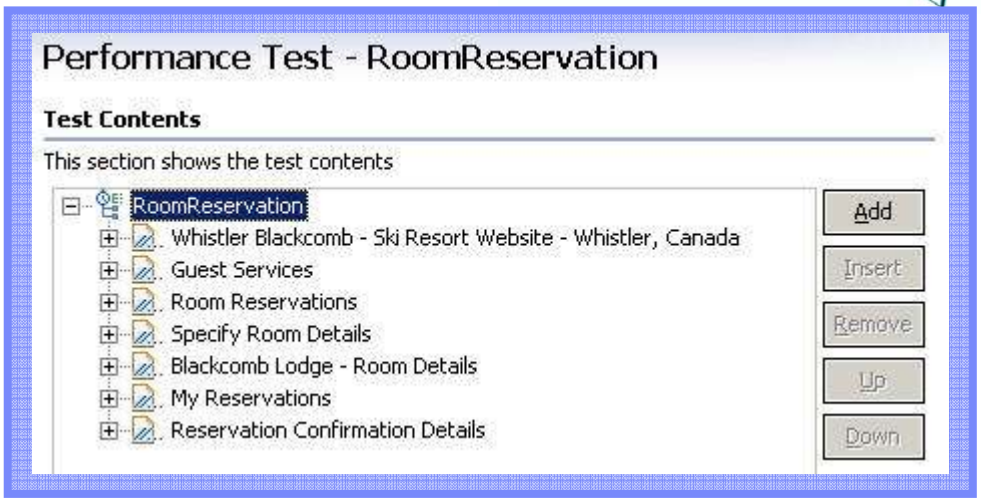
- **Performance test automation**
  - **Built for Day 1 Productivity**
    - Mask complexity to get the job done
  - **Advanced Data Access & Manipulation**
    - Automated data variation and synchronization
  - **Root Cause Analysis**
    - Identifies location and **root cause** of performance problem in hardware and software

Challenge 1: No in-house experience

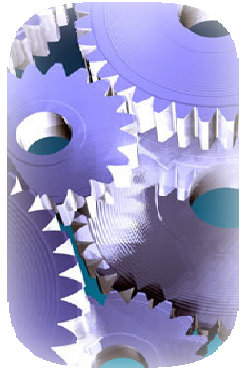


- **Challenge**
  - Tool complexity and lack of experience intimidates many first time users

- **Resolution**
  - Represent tests as a tree view of sequential flow through application
  - Simplify test editing with wizards
    - Looping
    - Conditional events
    - Data validation
  - Integrate Java code to handle unique performance challenges

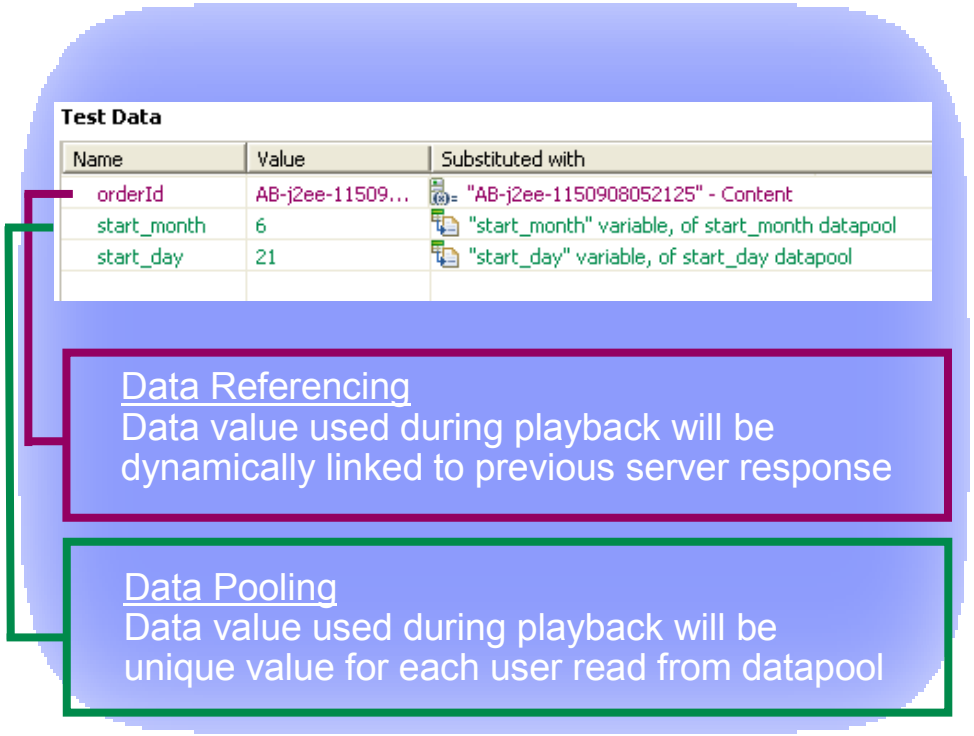


Challenge 2: Complexity of System Under Test



- **Challenge**
  - Complexity of system under test prohibits simple record and playback

- **Resolution**
  - Integrate data pooling technology to ensure each unique data for each user
  - Identify data relationships to dynamically reference server generated data during playback
  - Utilize unique TCP/IP addresses for each user to ensure realistic load



Challenge 2: Tools Lack Insight

- **Challenge**
  - Tool can find the problem, but not diagnose the root cause
- **Resolution**
  - Root Cause Analysis features provide additional insight to diagnose the cause of a bottleneck
  - Resource Monitoring data monitors hardware during test
  - Response Time Breakdown report breaks down response times into

Page Performance > Response Time Breakdown Statistics

demo:9080/ab/checkout.do

Component	Base Time (seconds)
CASPIAN	311.512
IBM Rational Performance Test	311.512
Delivery Time	26.500
Response time	208.748
text/html;charset=ISO-8859-1	76.264
demo	2,109.879
J2EE/WebSphere/6.0.0.1/demoNode01	2,109.879
Filter	39.632
JDBC	1,673.199
JSP	33.572
RMI-IIOP	5.280
Servlet	26.112
Session EJB	160.628
Web Services Provider	2.840
Web Services Requestor	168.616

Creating a Performance Test

Creating a performance test is a three step process

Build Scripts

- **Script Creation Considerations**
  - Visual test editor, varying input data & correlating server responses

Creating a Performance Test

Creating a performance test is a three step process

Build Scripts **Schedule Workload**

- **Script Creation Considerations**
  - Visual test editor, varying input data & correlating server responses
- **Scheduling Considerations**
  - Accurately representing a true user workload

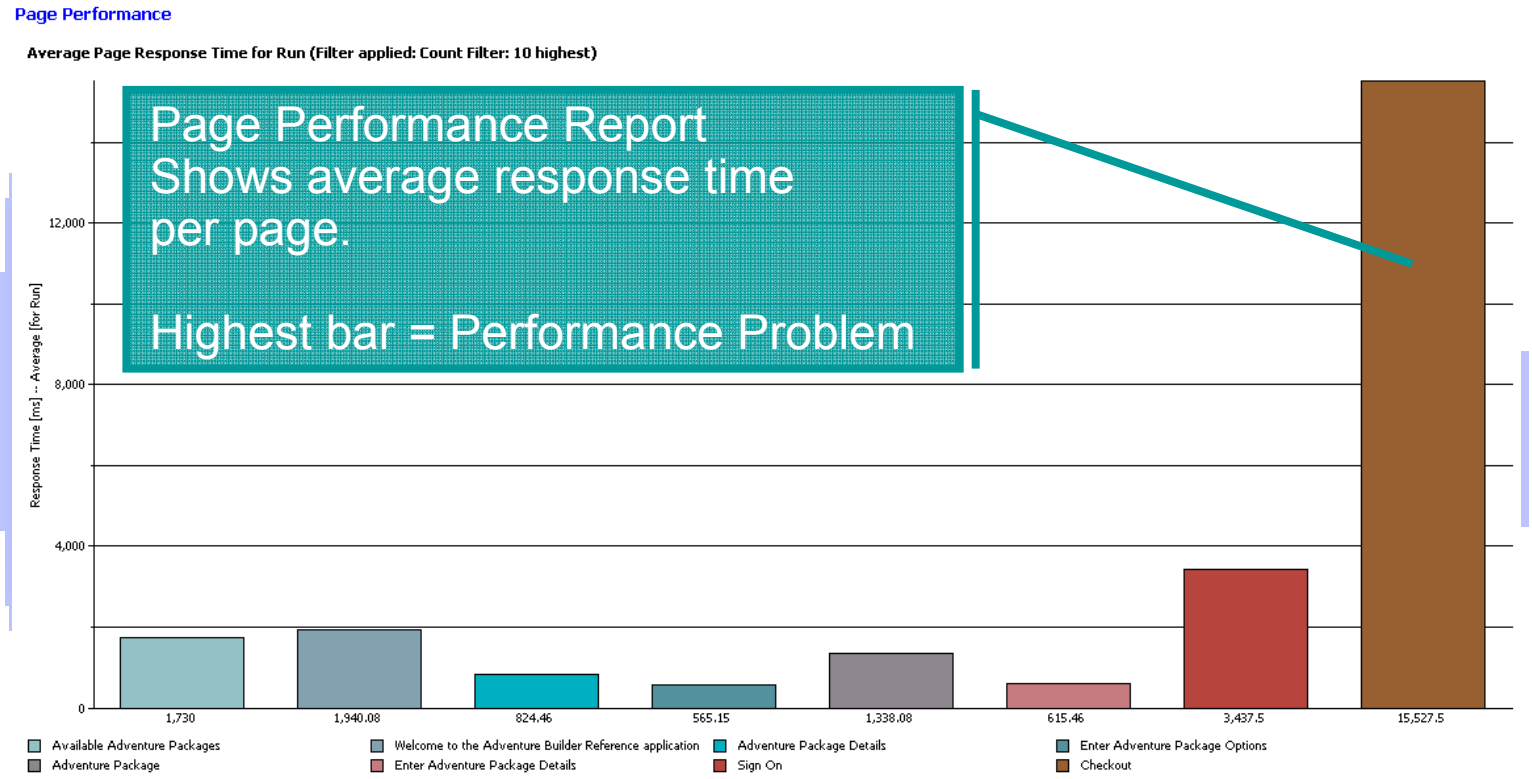
Creating a Performance Test

Creating a performance test is a three step process

Build Scripts **Schedule Workload** **Execute & Analyze**

- **Script Creation Considerations**
  - Visual test editor, varying input data & correlating server responses
- **Scheduling Considerations**
  - Accurately representing a true user workload
- **Execute and Analyze Considerations**
  - Validating responses & finding the bottleneck

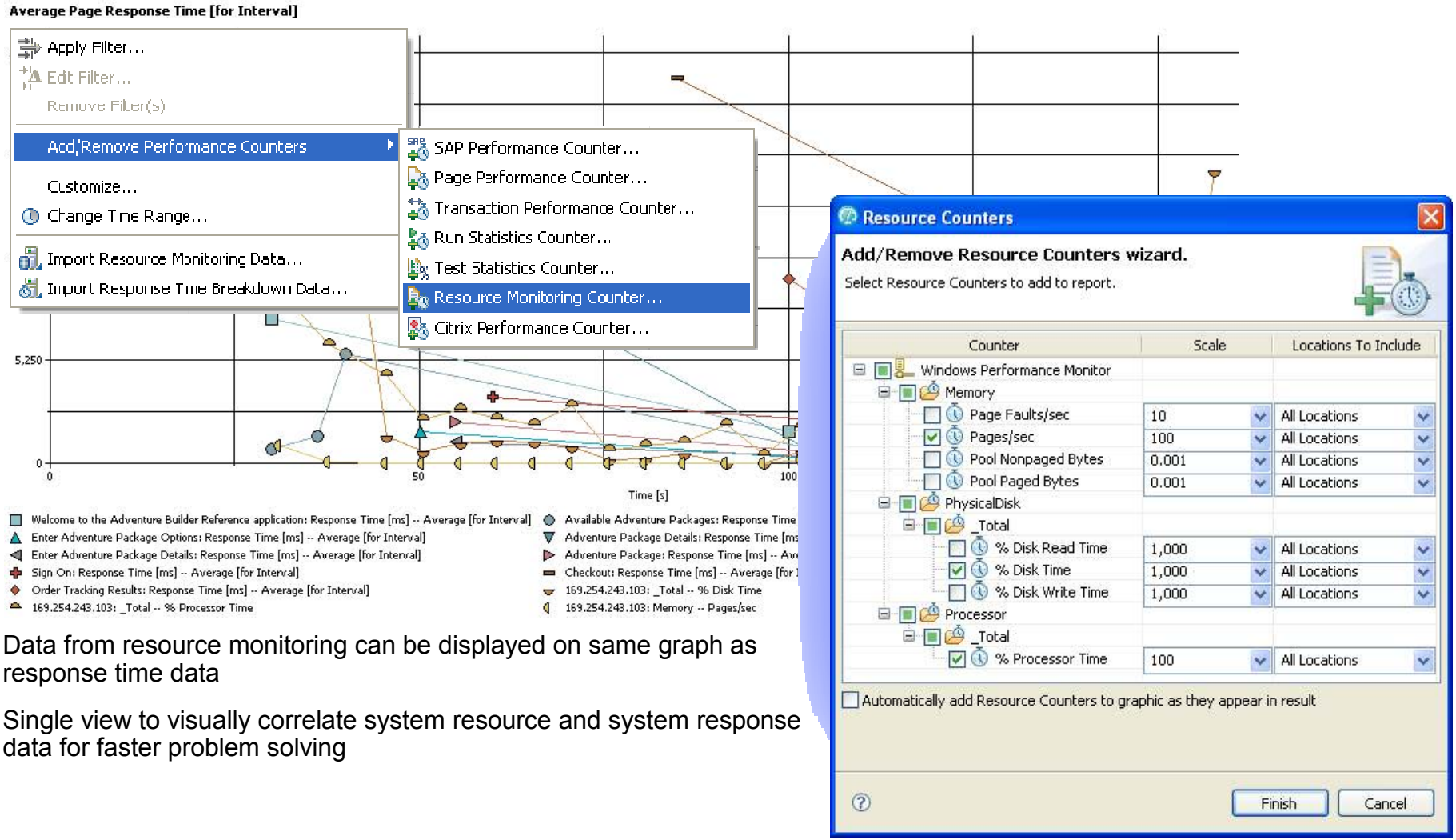
Performance Problem Identification During Test



- Performance Testing finds bottlenecks
  - Next logical question is Why?
  - Root Cause Analysis provides to tools to answer this question

Performance & Resource Statistic Report Overlay

Identifying hardware related performance problems



- Data from resource monitoring can be displayed on same graph as response time data
- Single view to visually correlate system resource and system response data for faster problem solving

Business SLA Reporting

Linking performance results to business objectives

Status Summary

Performance Requirement Status for Run	Failed
Performance Requirements Percent Passed	75

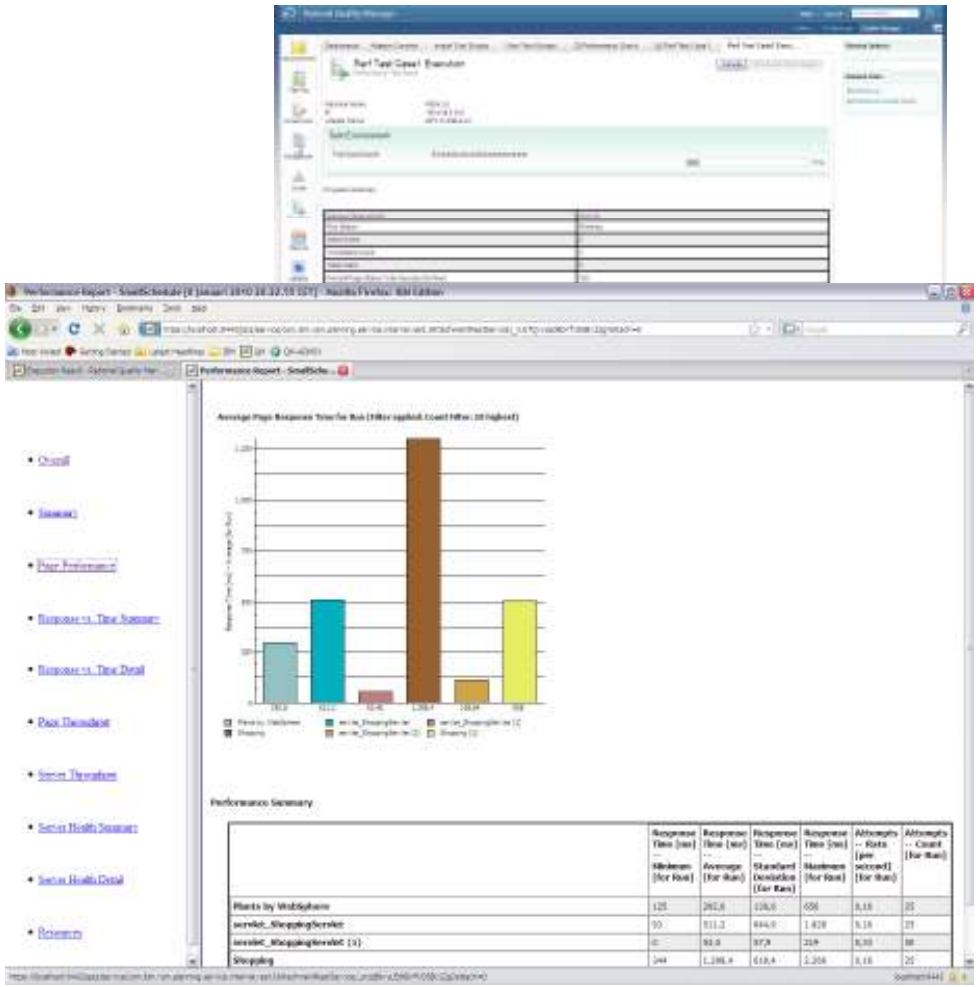
Summary

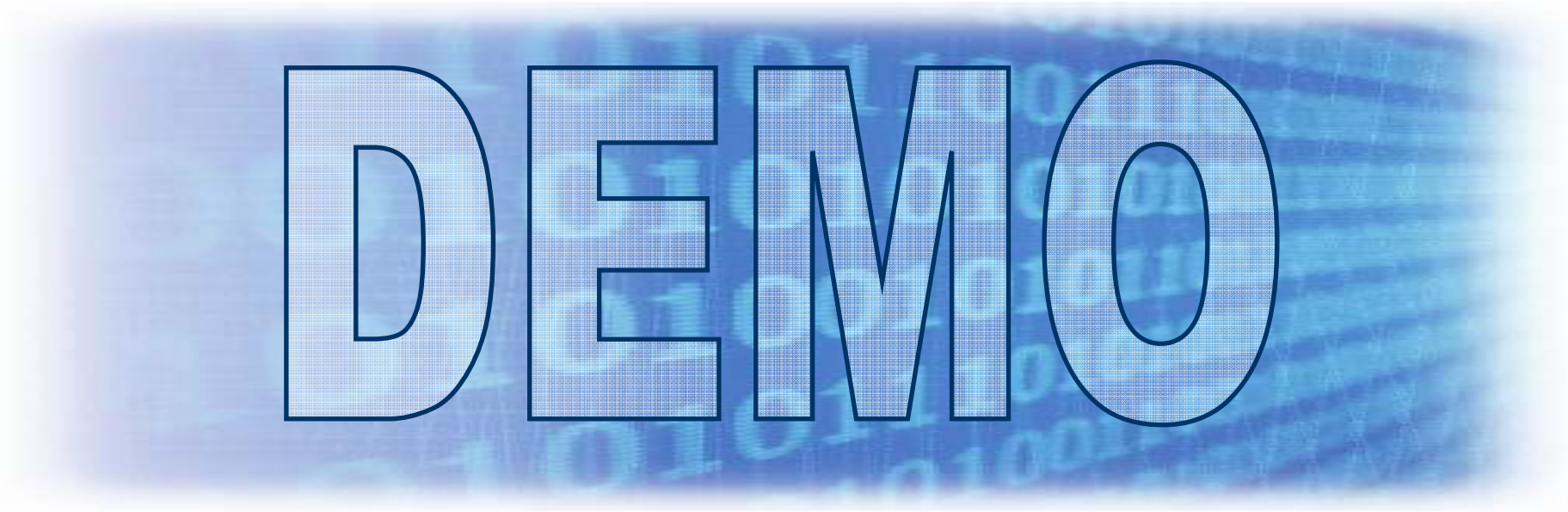
			Performance Requirements -- Specification	Performance Requirements -- Status
HTTP Page	/PlantsByWebSphere_PlantsByWebSphere	Average Response Time for Page [for Run]	<= 3000	Passed
HTTP Request	boomer.rtp.raleigh.ibm.com/PlantsByWebSphere	Average Response Time of Page Request [for Run]	< 1000	Passed
System Resources: localhost	Windows Performance Monitor	% Processor Time (Average for Run)	< 10	Failed
System Resources: localhost	Windows Performance Monitor	% Processor Time (Max for Run)	< 70	Passed

- Define detailed performance requirements in Rational Performance Tester
- Communicate results against performance criteria
- Results automatically rolled up and reported against user-defined SLA
- Results and reports are passed to RQM for wide visibility

Run Performance Test from Rational Quality Manager

- Utilize any RQM browser to start Performance Test Case
- Utilizing power of RQM
  - Schedule daily at 01:00
- Follow progress while executing
- Results are communicated back to RQM



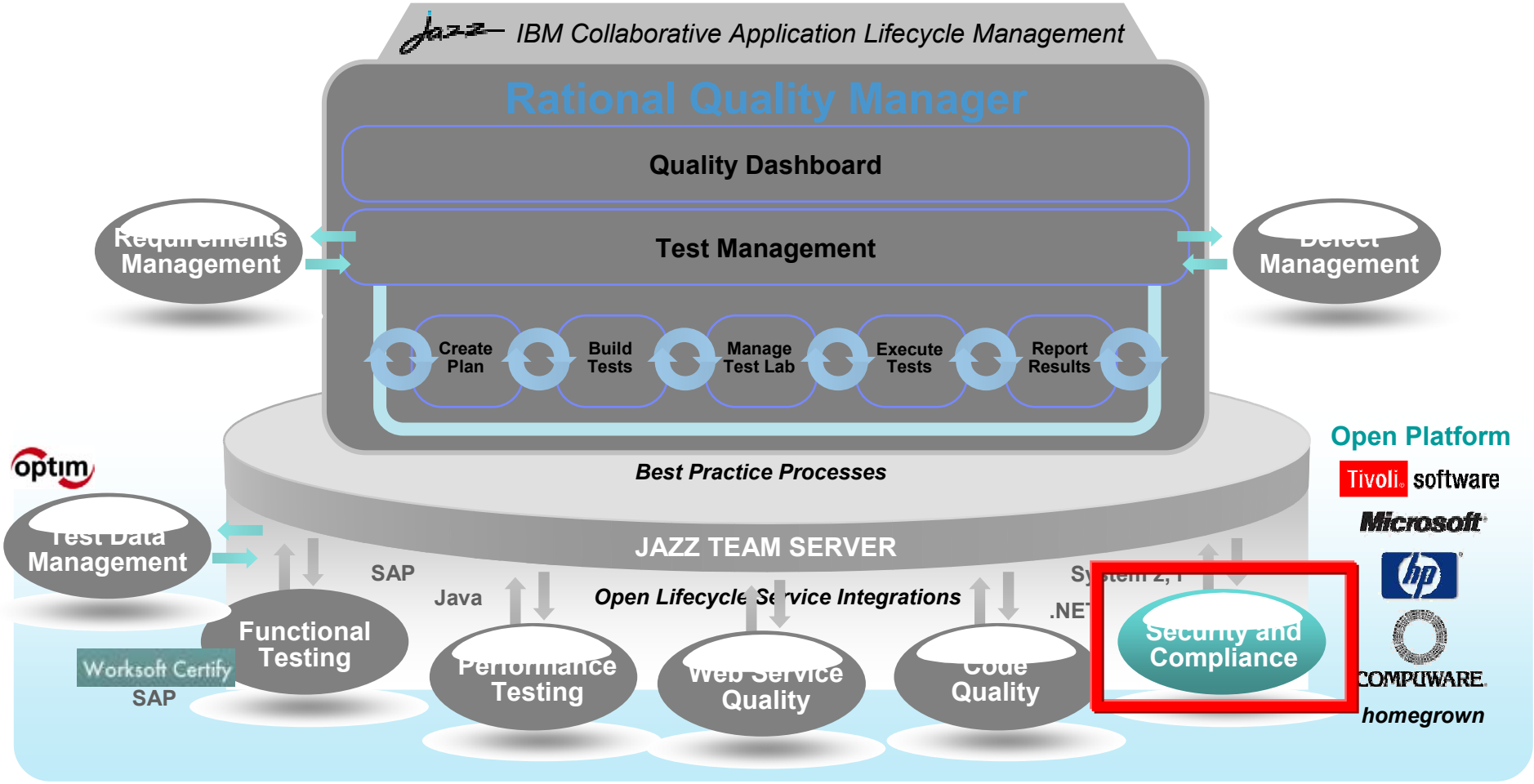


What You'll See:

- Rational Performance Tester

Security Testing
Rational Appscan Family

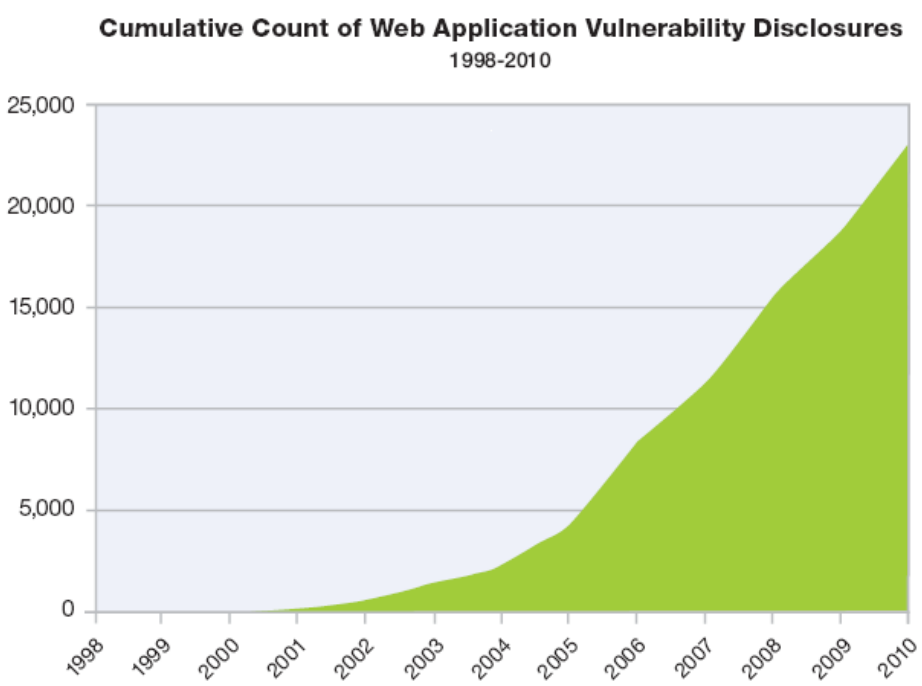
Centralized test management offering allowing full lifecycle support across all types of testing and platforms

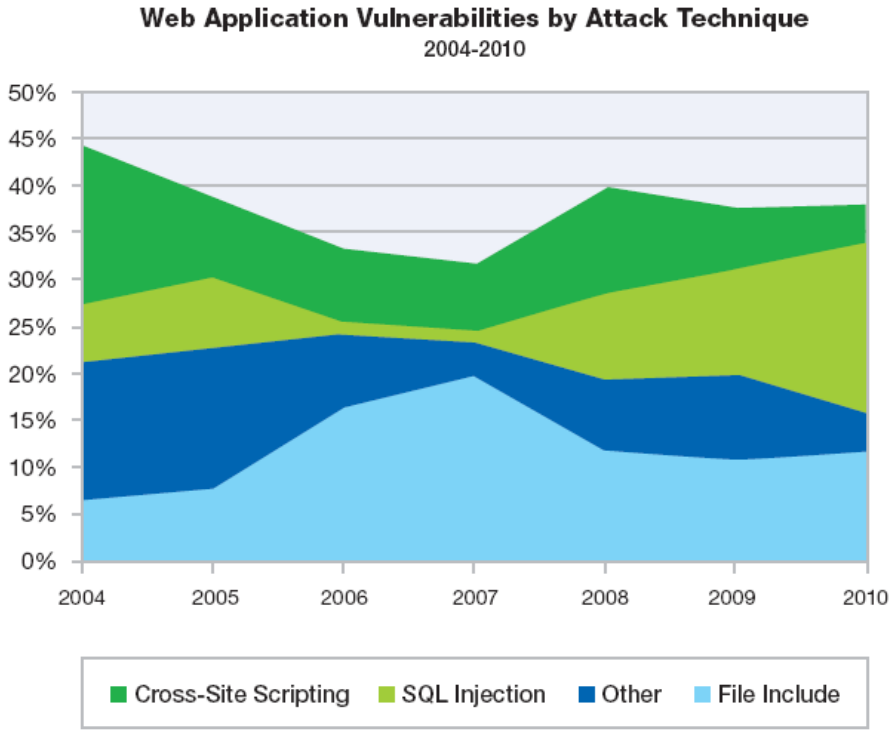


Application security challenges: vulnerabilities

Web application vulnerabilities dominate enterprise threat landscape

- **49%** of all vulnerabilities are in web applications*
- Cross-Site Scripting & SQL injection vulnerabilities continue to dominate





* IBM X-Force 2010 Trend & Risk Report

Application security challenges: security-development disconnect fails to prevent vulnerabilities in production applications

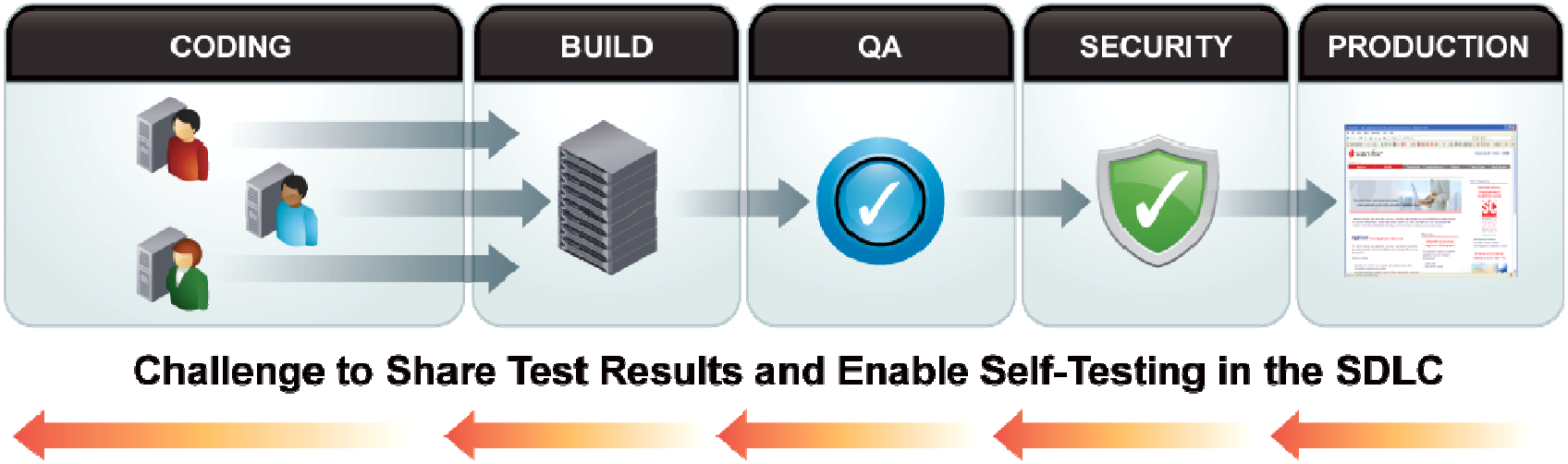
▪ Developers Lack Security Insights

(or Incentives to Address Security)

- Mandate to deliver functionality on-time and on-budget – but not to develop secure applications
- Developers rarely educated in secure code practices
- Product innovation drives development of increasingly complicated applications

▪ Security Team = SDLC Bottleneck

- Security tests executed just before launch
  - Adds time and cost to fix vulnerabilities late in the process
- Growing number of web applications but small security staff
  - Most enterprises scan ~10% of all applications
- Continuous monitoring of production apps limited or non-existent
  - Unidentified vulnerabilities & risk



Security testing within the application life cycle

Prevention...

Ignore the issue until...

- High cost
- High pain
- High disruption

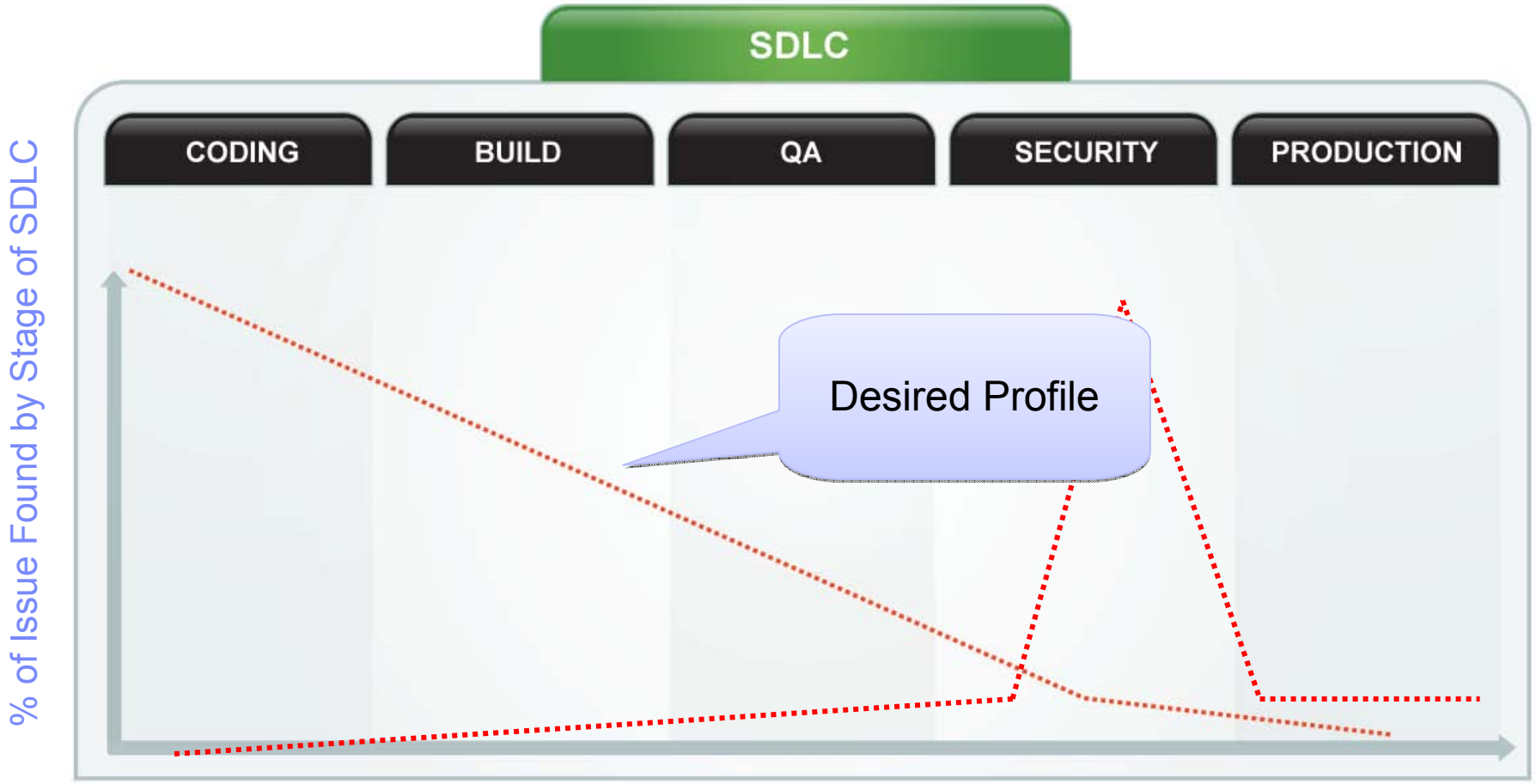
Or This?

A little bit every day

- Low cost
- Low pain
- Low disruption

This?

Security testing within the application life cycle



Make applications secure, by design

Cycle of secure application development

- **Design**
- Consider security requirements of the application & apply threat models
- Issues such as required controls and best practices are documented on par with functional requirements
- Secure code libraries maintained for reusable secure code

- **Development**
- Create work items that map to security requirements
- Use secure code libraries
- Software is checked during coding for:
  - Implementation error vulnerabilities
  - Compliance with security requirements

- **Build & Test**
- Map test plan to security requirements
- Testing begins for errors and compliance with security requirements across the entire application
- Applications are also tested for exploitability in deployment scenario

- **Deployment**
- Configure infrastructure for application policies
- Deploy applications into production

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

Cost is a significant driver

80% of development costs are spent identifying and correcting defects!*

During the
CODING phase
$80/defect

During the
BUILD phase
$240/defect

During the
QA/TESTING
phase
$960/defect

Once **released**
as a product
$7,600/defect
+
Law suits, loss of customer trust, damage to brand

*National Institute of Standards & Technology
Source: GBS Industry standard study
Defect cost derived in assuming it takes 8 hrs to find, fix and repair a defect when found in code and unit test.
Defect FFR cost for other phases calculated by using the multiplier on a blended rate of $80/hr.

Solution requirements: advanced security testing + collaboration & governance through application lifecycle

Advanced Security Assessments

Dynamic Analysis

- Analysis of a running/ deployed application
- Key requirements
  - Threat coverage: WASC, OWASP Top 10, etc
  - Web Services/ SOA
  - Web 2.0 & Rich Internet Applications

Static Analysis

- Scanning source code for security issues
- Key requirements
  - Application/language support
  - Ease of use for non-security users (developers and build managers)

Runtime & Hybrid Analysis

- Glass box testing with runtime analysis
- Automated correlation of static & dynamic results
- Key requirements
  - Precise & Actionable results
  - Broad threat coverage

Collaboration & Governance in Application Lifecycle

Security testing, shared results, assign ownership

Track corrections and integrate with development systems

Solution Requirements: Static, Dynamic and Runtime Analysis

	Static Analysis (White Box testing)	**Dynamic Analysis** (Black Box testing)	**Runtime Analysis** (Glass Box testing)
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	Similar to black box to scan running web applications with an agent installed on the application
Assessment techniques	Uses "taint analysis" and pattern matching techniques to locate issues	Tampering of HTTP messages to locate application and infrastructure layer issues	Agent monitors application performance during a black box scan for expanding threat coverage and greater detail
Role in application development lifecycle	**Development:** Scan code and work remediation from IDE **Build:** Scan nightly or weekly build to highlight defects for developers to correct **Security:** Define & customize security best practices for developers; Execute pre-production scans and audits	**Build:** Scan as part of build acceptance tests before releasing build to testing team **Test:** Execute security test scripts as part of quality plan **Security:** Define test scripts for quality plan; Execute pre-production scans and audits	**Build:** Provides added layer of vulnerability detail that assists developers with security de-bugging **Security:** Expands threat coverage for hard-to-identify vulnerabilities (including all OWASP Top 10)
Results & Output	Results are presented by line of code, source to sink functions flow	Results are presented as HTTP messages (exploit requests)	Results are presented as a combination of HTTP messages (exploit requests) and the line of code

Application Security: Where do I start?

- **First time conducting in-house application security assessments**
- **Most clients start with dynamic testing**
  - Dynamic analysis (black box testing) allows security groups to assess application risk in both development and production apps
  - Easy to roll out & automate work previously done with outsourced penetration testing
  - Select a solution that combines ease of use, advanced security analysis and results that can be shared outside of security
- **Application security testing confined to security team**
- **For deployments led and executed only by security teams, start with dynamic and later consider static (white box)**
  - Dynamic analysis is executed against compiled applications in lab environments, so security teams can control & execute the application security program
  - Select a solution that allows you to share results with development, cover all of your applications in both development and production, and later scale program with static analysis
- **Development & security teams integrate security testing in the SDLC**
- **Most clients evolve to this level of application security program with various use cases of dynamic and static analysis that fit their development processes**
  - Developers execute static analysis from their IDE or at least access static results from IDE
  - Build: Static analysis of each build and dynamic analysis before releasing build
  - Test: Dynamic testing included in test plan and executed from testing tools
  - Security: Conduct advanced dynamic and static testing before launch (benefit from early testing that eliminates the common security defects like SQL Injection and Cross-Site Scripting
  - Select a solution that delivers governance and collaboration while empowering non-security users

IBM AppScan: Advanced research drives precise security testing that integrates with application development lifecycle

Legacy of Security Innovation

Advanced testing technologies

- Dynamic Analysis (black box); IBM holds the original patent for dynamic web app security scans (US6584569)
- Static Analysis (white box)
- Runtime Analysis (glass box); patent filed 2008
- JavaScript Security Analyzer (static scans of client-side JavaScript)

Broad application support

- Web applications
- Packaged applications (SAP)
- Legacy applications (COBOL)

Broad technology coverage

- Web 2.0 and Rich Internet Applications
- Web Services/ SOA/SOAP

Governance and Collaboration in Application Development Lifecycle

Code

- Scan code, manage work items and remediate vulnerabilities from the IDE

Build

- Integrate security testing as a natural extension of build extension testing
- Find & fix defects before releasing a build

Test

- Include security testing in quality plan
- Execute basic security test scripts from quality management platform

Security

- Build security test scripts for non-security experts
- Focus pre-production audits on most advanced threats
- Manage test policies and scan permissions
- Collaborate with development to triage findings and assign ownership

AppScan Standard: **Desktop solution combines advanced security testing, broad technology coverage and ease of use**

Web Application Assessments for Pen-Testers and Security Practitioners

Dynamic Analysis (black box)

- **Covers all relevant OWASP & WASC TCv2 threat classes**
  - SQL Injection
  - Cross-Site Scripting
  - HTTP Response Splitting
  - OS Commanding
  - LDAP Injection
  - XPath Injection
  - Buffer Overflows
  - ***1000s more***
- **Web 2.0 and Rich Internet Applications**
  - JavaScript & Ajax
  - Adobe Flash & Flex
- **Malware analysis**
  - Scan site with malware analysis from IBM X-Force Security Research
- **Web Services/ SOA**
  - SOAP/XML parser issues (External entities, XML blowup, etc.)
  - Application-layer issues
  - Infrastructure issues

Hybrid Technology

- **Runtime Analysis (glass box testing)**
  - Expanded threat coverage with less configuration
  - Precise results (line of code) assist remediation
- **JavaScript Security Analyzer**
  - Static taint analysis of client-side JavaScript

Ease of Use

- **Configure & test**
  - Scan Expert provides recommended settings based on your apps
- **Details & guidance to correct the vulnerability**
  - Explanation of threat and recommended fix
- **Integrate with Defect Tracking Systems**
  - Rational® ClearQuest
  - HP Quality Center
- **Compliance & Reporting**
  - 40+ compliance reports
  - Executive-level summaries
  - Guidance for development

AppScan Enterprise

- **AppScan Enterprise: Application Security Governance & Risk Management**

Governance

- **Scale security testing**
  - Assess 1000s of apps
  - Engage more testers
  - Integrate testing in SDLC
- **Control**
  - Scan permission
  - Test policies & templates
  - User roles & access control
  - Processes & best practices
- **Measure and improve**
  - KPIs
  - Trending

Collaboration

- **Manage security issue resolution**
  - Multi-level reporting
  - Issue classification
  - Integration with defect tracking systems
- **Traceability**
  - Security requirements
  - Development tasks
  - QA test cases

Risk Management

- **Visibility of risk and compliance**
  - High-level view of application security risk
  - View of non-compliance issues
- **Security intelligence**
  - Metrics
  - Correlation of findings
- **Mitigate risk**
  - Virtual WAF patches*
  - Fixing security code errors

Application Security Analysis

Dynamic **Static** **Runtime**

AppScan Enterprise: **Security testing and visibility throughout the SDLC for enterprise-wide application risk management**

Information Security	▪ Schedule and automate assessments ▪ Manage test policies and scan permissions ▪ Collaborate with development and QA by publish findings for remediation ▪ Build protection strategies based on known vulnerabilities
Development & Build Automation	▪ Analyze source code for security issues in applications, projects or files from IDE or automatically trigger scans in Build system ▪ Remediate vulnerabilities with details and recommended fixes available in IDE ▪ Execute source code scans Execute dynamic test of compiled applications to identify and remediate issues before passing build to QA
Quality Assurance	▪ Create security test plans & test scripts in Rational Quality Manager ▪ Manage open issues via defect tracking systems
Management	▪ Enterprise-view of application security risk ▪ Trending and reporting with key performance indicators
Compliance Officers	▪ Review compliance reports ▪ Audit vulnerability resolution

AppScan Enterprise + AppScan Source: **Static analysis (white box) security & quality testing in the collaborative application lifecycle**

Source Code Analysis for Security Testing in Development & Build Automation

Broad Application Support

- **Out of the Box for Security Testing**
  - Java
  - JSP
  - C
  - C++
  - Classic ASP (VB6)
  - COBOL
  - SAP ABAP*
  - .NET
    - C#
    - VB.NET
    - ASP.NET
  - PHP
  - HTML
  - Perl
  - ColdFusion
  - Client-Side JavaScript
  - Server-Side JavaScript
  - VBScript
  - PL/SQL
  - T-SQL

Code Quality Static Analysis

- Identify code-level quality defects within IDE
- Automate code quality analysis as part of the build process for centralized software code scanning
- Key Performance Indicators (KPIs) to help developers learn best practices
- Languages: Java, C, C++

Application Lifecycle Integrations

- **Develop**
  - IDE plug-ins to remediate identified issues (*Source for Remediation*)
  - Options to scan code locally from IDE (*Source for Developer*)
- **Build**
  - Automatically trigger security scans with each build (*Source for Automation*)
  - Review results from IDE or Security user & create work items for remediation
- **Security**
  - *Source for Security* power user creates SAST scans executed from IDE or in build automation
  - Executes advanced scans in pre-production security audits

* Requires Virtual Forge CodeProfiler for AppScan Source Edition

IBM Rational Appscan portfolio summary

Rational AppScan offering	Description
AppScan Enterprise Edition	• Enterprise platform for managing application security and risk management • Identify application risk with advanced security testing • Mitigate risk by collaborating with developers to remediate security vulnerabilities • Measure, monitor and drive risk reduction with reporting, issue tracking, KPIs and trending • Empower security teams to drive security testing throughout the software development life cycle (SDLC) • Collaborate with developers to remediate security vulnerabilities • Integrate with web-application firewalls to provide custom tuning based on actual vulnerabilities • Plan and execute dynamic (black box) tests against applications in development and production • Integrates with Rational Quality Manager software for QA teams to use in test scripts, and can conduct security checks within their familiar testing environments
AppScan Source Edition	• Adds source code analysis to Rational AppScan Enterprise Edition to identify the latest security threats with static (white box) analysis • Enables quick analysis and recommended corrections, all within the IDE • Automated security testing within build environments
AppScan Standard Edition	• Desktop application for security analysts and penetration testers • Advanced security testing based primarily on dynamic (black box) analysis, but also includes static analysis for client-side JavaScript • Glass-box testing with run-time analysis that applies an internal agent to monitor application behavior during a dynamic test, provide more accurate test results and identify specific lines of code • Coverage of the latest rich-Internet applications and web technologies (web services, SOAP, Flash, Ajax and more • Designed for ease of use
AppScan Tester Edition	• Server and web interface solution designed for QA teams to integrate security testing into existing quality management processes • Integrates with Rational Quality Manager for QA teams to use in test scripts, and can conduct security checks within their familiar testing environments
AppScan Policy Tester	• Online compliance solution to assess quality, privacy and accessibility-compliance issues for corporate web properties

Rational Solution for SAP
SAP Connector, Worksoft, VirtualFor

Rational Solution for SAP
SAP Connector, Worksoft, VirtualFor

Change is critical to Business Innovation

There are 3 key drivers for change events...

Business Drivers

- Changed Business Processes
- M&A and Reorganizations
- New Products, Vendors, …

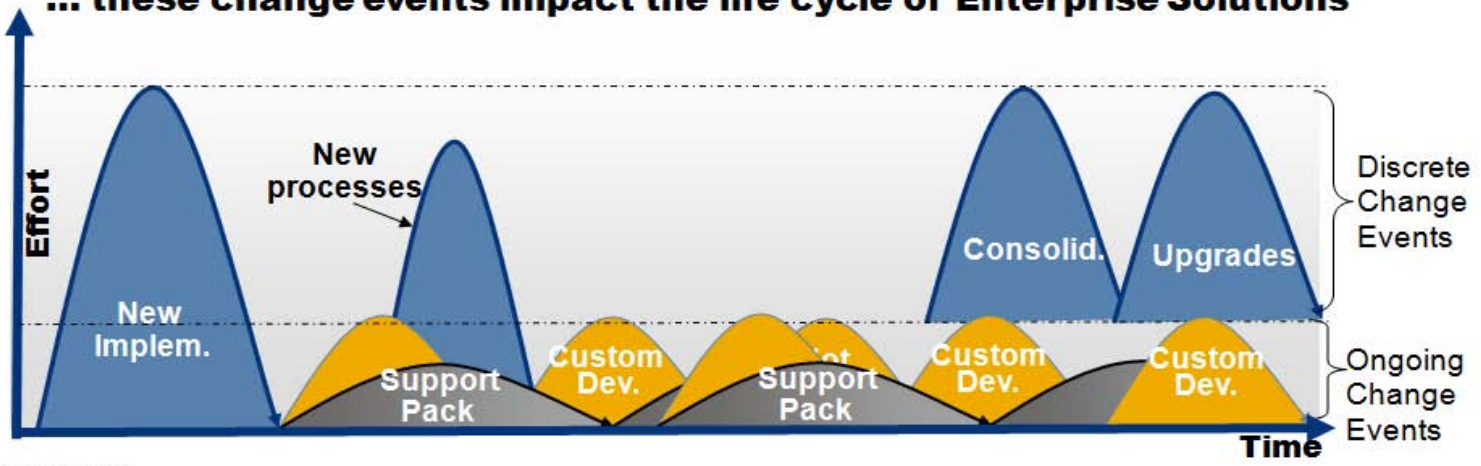
Compliance Drivers

- Governance
- Risk Management
- Compliance
- Examples: SOX, SEC, FDA, Basel2

Technology Drivers

- Maintenance
- New functionality
- Upgrades
- (Global) Rollouts

… these change events impact the life cycle of Enterprise Solutions

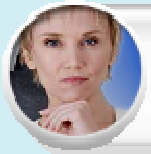


Change in the SAP Environment and Beyond

Customer Pain Points Around Managing Solution Integrity

How Do I Align Business Requirements With IT Requirements?

How Can I Manage My SAP and non-SAP projects in a unified way?



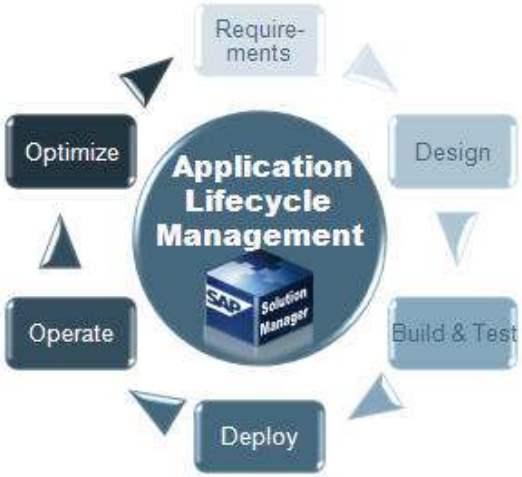
Can I Verify That My Business Processes Have Been Deployed & Tested?

Can I Coordinate IT Changes Across My Heterogeneous Systems & Business Processes?

Silos of people, process, and projects

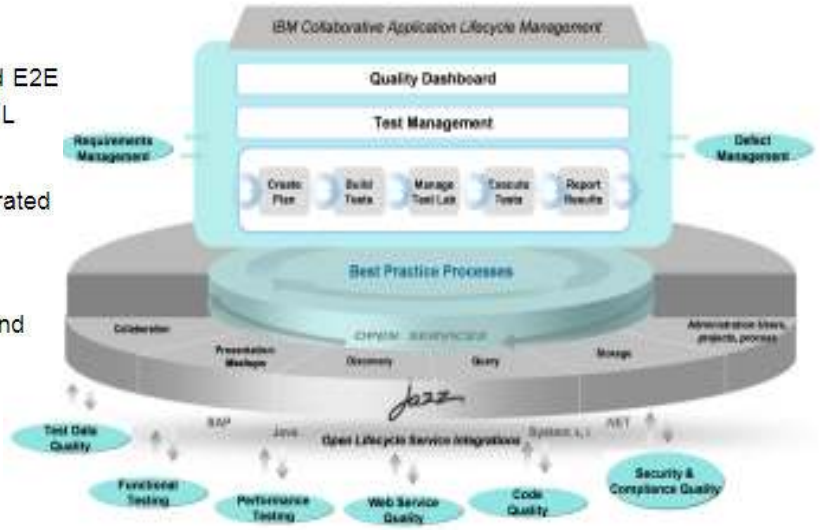
Application Lifecycle Management with SAP Solution Manager and IBM Rational software

SAP Solution Manager



- **Reduces risk and lowers TCO** through standardized and integrated E2E Solution Operations according to ITIL
- **Optimizes business benefits** through a comprehensive and integrated quality management process
- **Leverages built-in innovation** by making innovation a non-event and protecting existing investments

IBM Rational



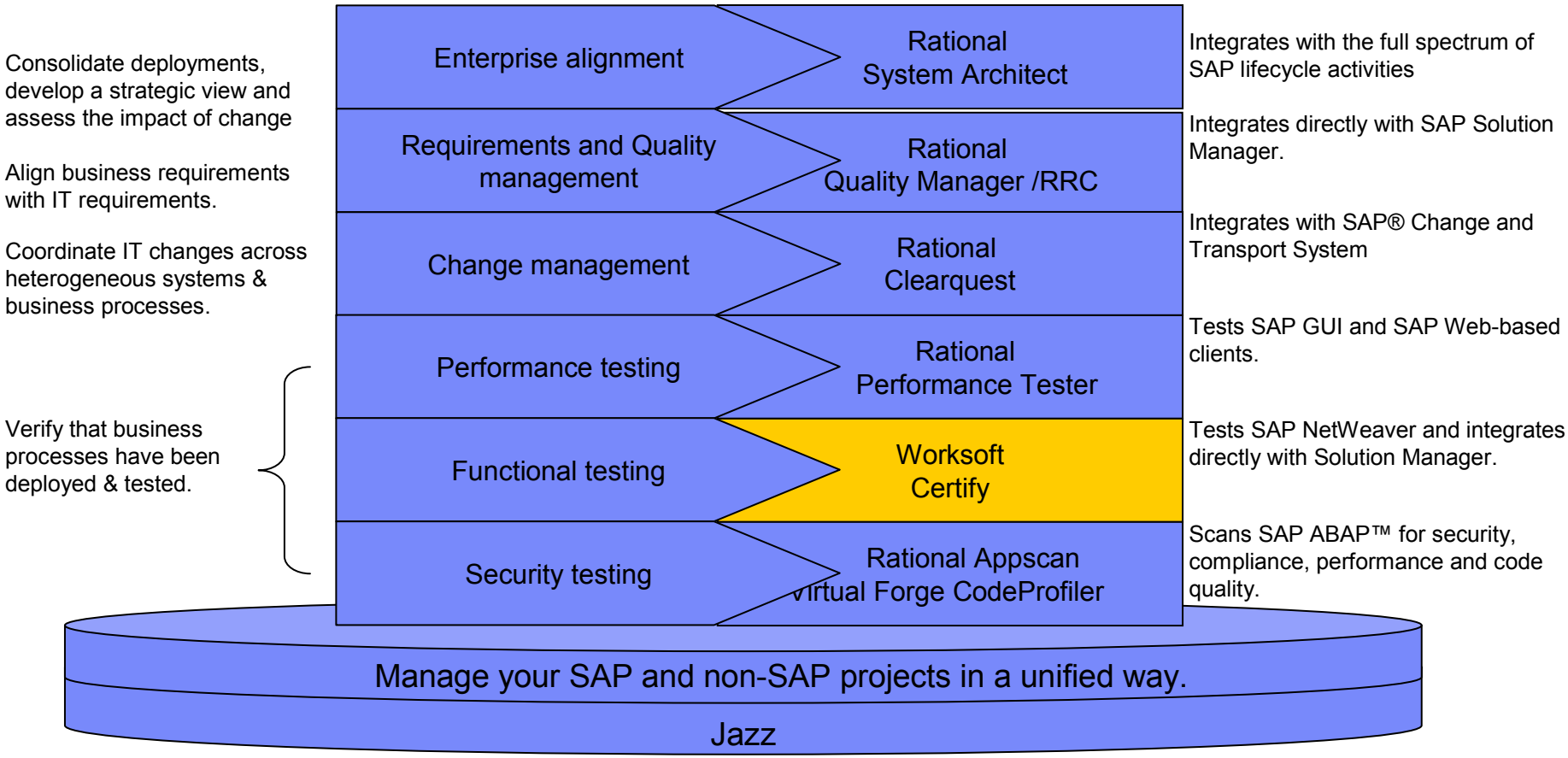
End-to-end Quality Management is a key element of effective Application Lifecycle Management to help lower costs and improve deployment success

SAP and IBM Rational are collaborating on an integrated Lifecycle Management solution to support our joint customer needs by starting with an end-to-end Quality Management offering.

Rational solutions for SAP

Plan a business transformation | Implement a new package solution | Add ALM to an existing packaged solution

At the heart of the organization



"Our collaboration with IBM Rational brings together the best of our combined application lifecycle management market leadership and can help customers reduce costs, manage change, and improve quality across the enterprise applications lifecycle."

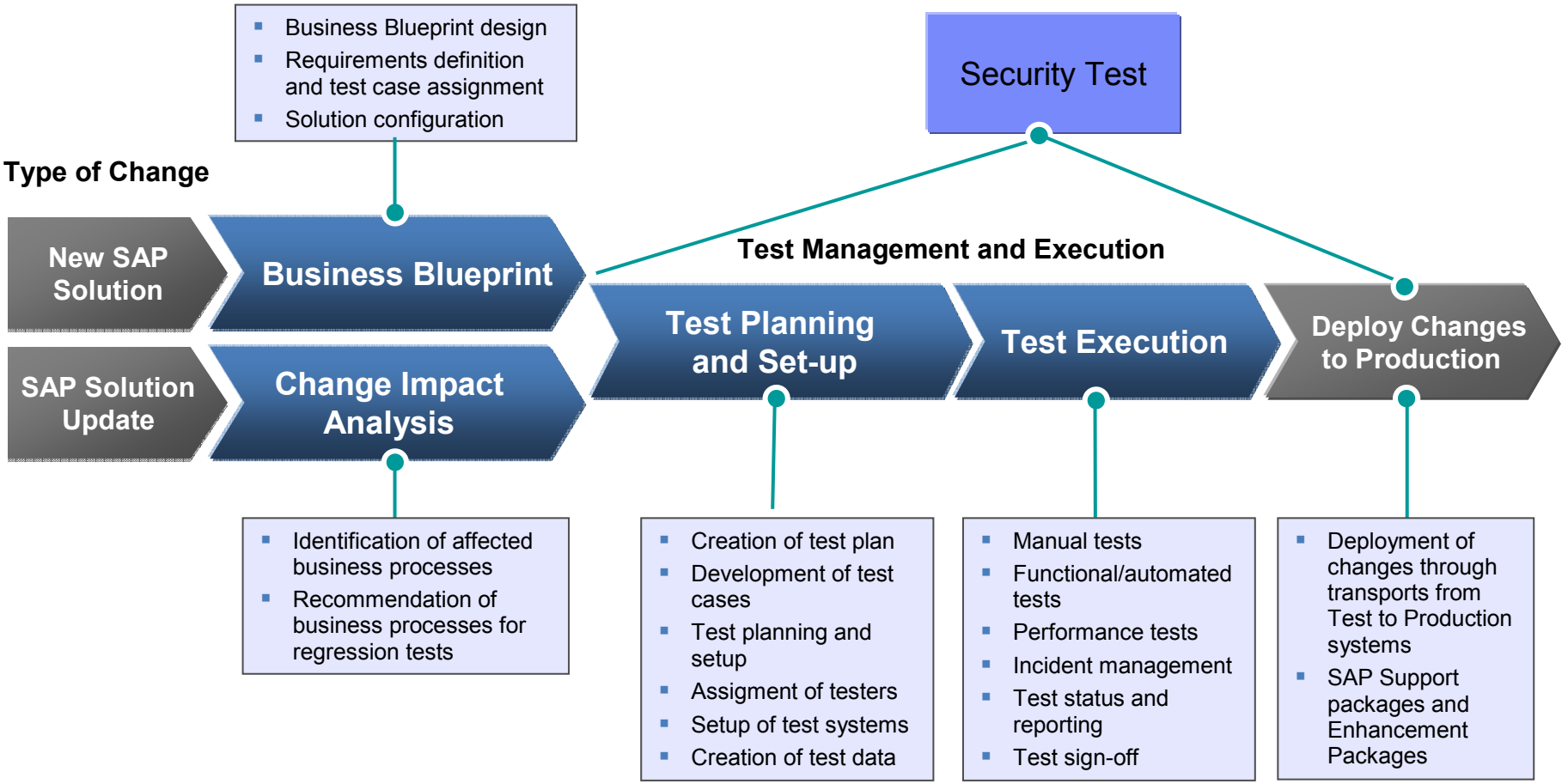
Dr. Uwe Hommel Executive Vice President and Head of SAP Global Active Support, SAP AG

Rational Solutions For SAP

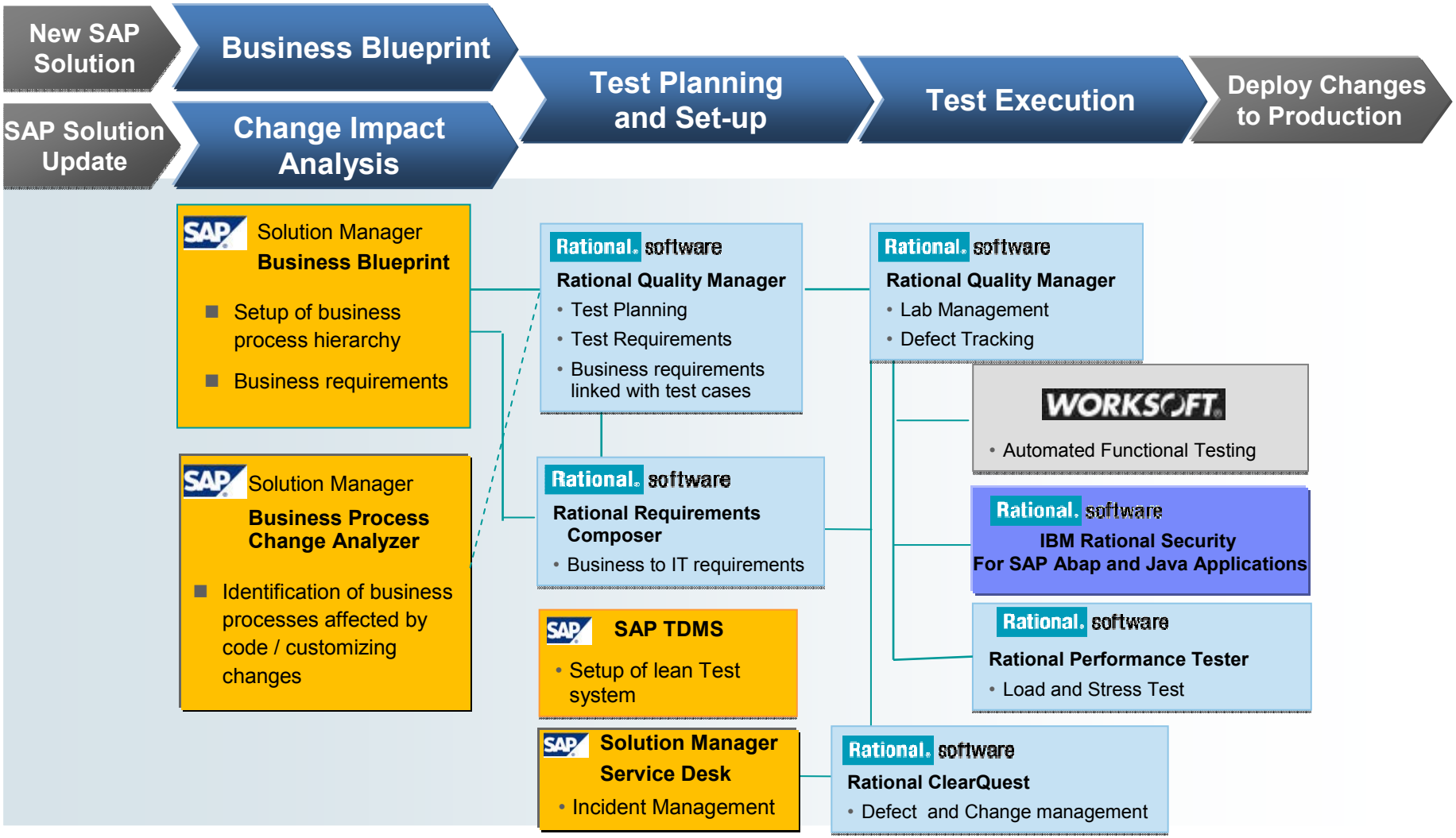
- Overview
- Rational Quality Solution
- Worksoft Solution
- SAP Connector
- Security

Quality Management for SAP-centric Solutions

End-to-end integration testing of SAP solutions



End-to-end Quality and Test Management with SAP Solution Manager and IBM Rational software



IBM Rational Requirements Composer

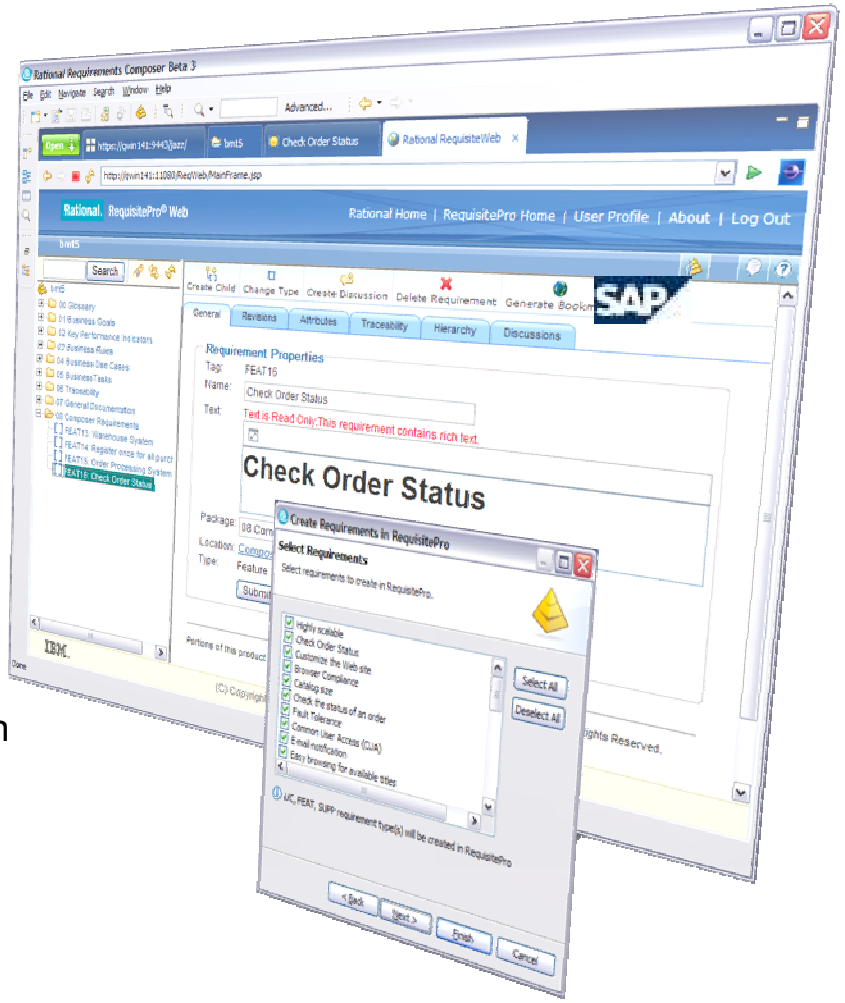
Capture and integrate requirements for SAP and non-SAP projects

Work smarter through integrated requirements

- Capture and manage SAP business requirements and align associated development and test activities
- Enable a single, holistic view of requirements across enterprise projects and stakeholders
- Leverage the Web 2.0 interface to foster collaborative software delivery and coordination among distributed teams

Reduce project risk and improve responsiveness

- Leverage a rich requirements authoring environment and governance capabilities to manage business goals, rules, dependencies and priorities
- Link test cases with requirements through integration with IBM Rational Quality Manager
- Integration with IBM Rational ClearQuest & IBM Rational Team Concert allows traceability of requirements implementation activities through the software delivery lifecycle



IBM Rational Quality Manager

A central hub for business-driven software quality

Mitigate business risk with collaboration

- Stakeholder and team coordination reduces mistakes
- Risk identification and management leads to educated prioritization decisions
- Test traceability linked to business requirements (including link with SAP solution blueprint) improves customer satisfaction

Improve operational efficiency with automation

- Running tests earlier leads to reduced repair costs
- Running more tests in less time improves coverage
- Reducing manual labour leads to fewer testing errors
- Lab configuration automation improves efficiency and asset utilization

Make confident decisions with effortless reporting

- Real-time dashboards enable proactive risk management
- Customizable reports facilitate ongoing process improvement
- Transfer test results back to SAP Solution Manager for consolidated project reporting

Rational Appscan & Virtual Forge CodeProfiler for Rational AppScan Source Edition

Java Applications

- Dynamic (black box) analysis
  - *Rational AppScan Standard*
  - *Rational AppScan Enterprise*
- Static (white box) analysis
  - *AppScan Source*
- IBM Rational AppScan
  - Recognized leader of dynamic (DAST) and static (SAST) security testing solutions that integrate into the software development life cycle
  - Integrates security testing into AppScan platform for complete application lifecycle management

ABAP Applications

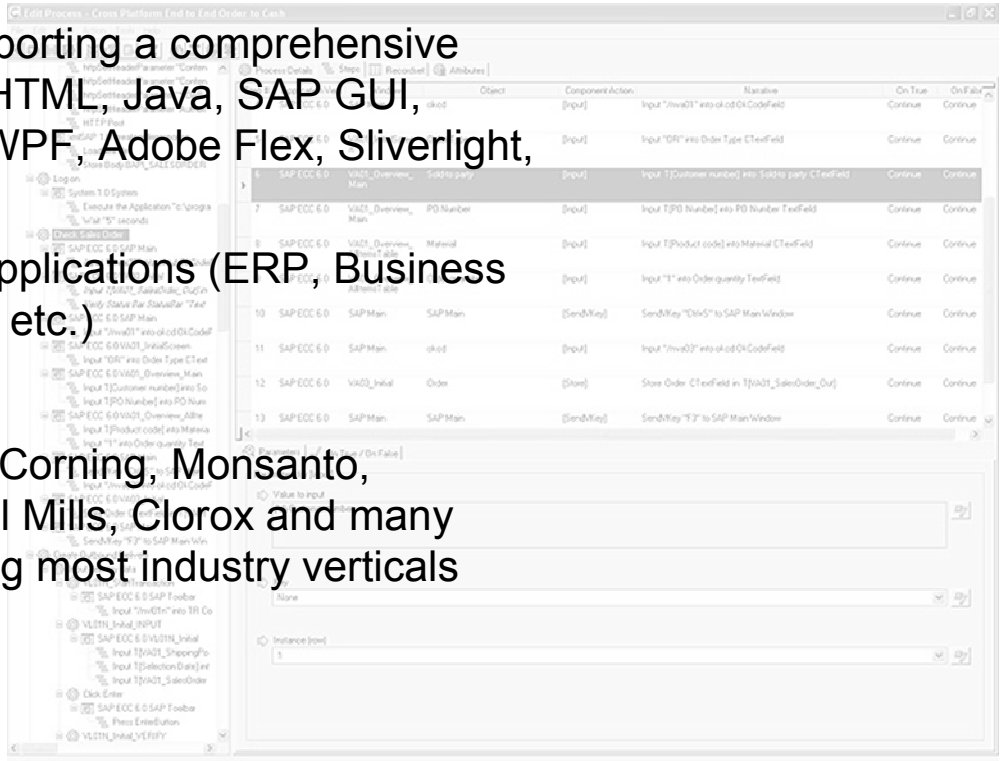
CodeProfiler for Rational AppScan Source Edition identifies security vulnerabilities in SAP ABAP applications and enables enterprises to eliminate SAP application risk with advanced static (white box) security testing of ABAP source code.

- Identify and remediate security vulnerabilities in your SAP applications by analyzing ABAP source code to expose security defects with static (white box) analysis
- Empower developers to write secure ABAP applications by integrating security scanning into the ABAP Workbench and SAP user interface
- Enforce service level agreements for security for applications and code developed by consultants and third parties
- Drive remediation efforts with recommended code fixes and triage results in AppScan Source Edition for a single view of all static analysis testing and results
- Manage SAP security as part of your enterprise application risk management program by integrating with Rational **AppScan Enterprise**

Worksoft Certify®

Comprehensive Automated Functional Testing

- Cross platform, script-less automation supporting a comprehensive suite of underlying technologies including HTML, Java, SAP GUI, Adobe Dynamic Forms, Mainframe, .Net, WPF, Adobe Flex, Sliverlight, Unix, Win 32, Microsoft Office, etc.
- SAP optimized solution supports all SAP applications (ERP, Business Suite 7, Portals, Web Dynpro Applications, etc.)



Global Fortune 500 Customer Base

- Successfully deployed at eBay, Intel, Dow Corning, Monsanto, Panasonic, Fujitsu, Under Armour, General Mills, Clorox and many other multi-national companies representing most industry verticals

Faster Time to Value

- Easy to learn, easy to use
- Comprehensive pre-built, re-usable SAP test content
- Documentation & automation in one step

SAP® Certified
Integration with SAP NetWeaver®

Ready for IBM. | Rational. software

IBM Rational Performance Tester

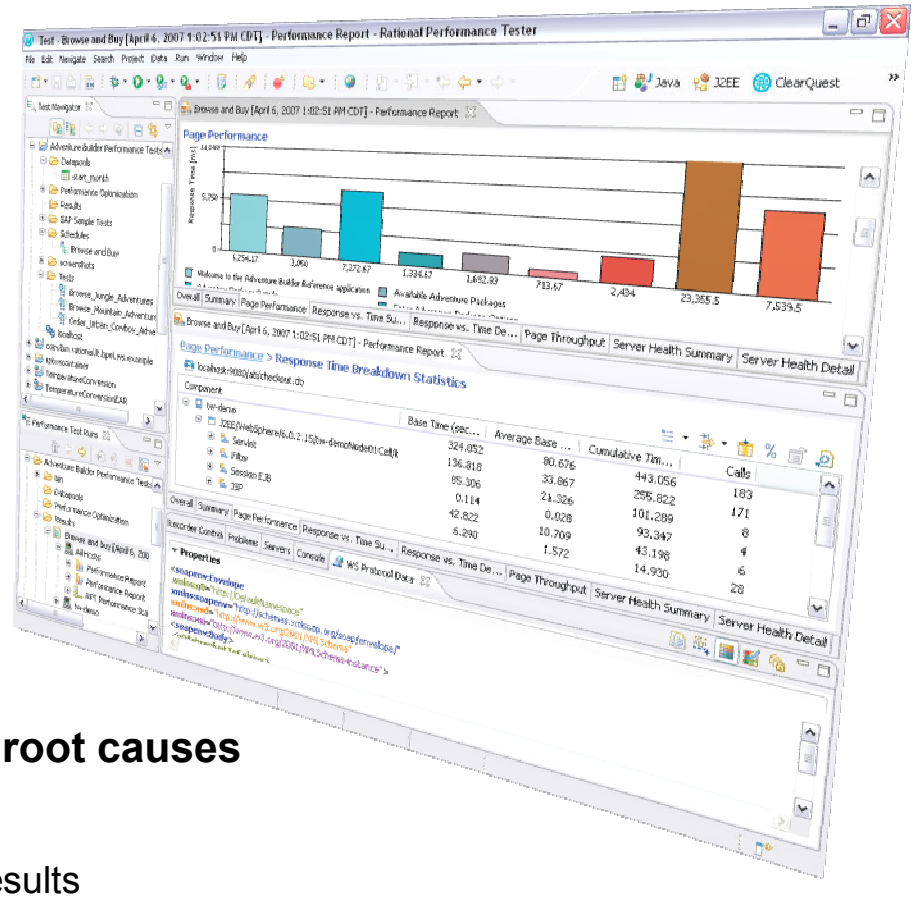
Powerful workload creation combined with ease of use

Speed test creation

- Easily capture all SAP transactions in SAPGUI or NetWeaver Portal through SAP recorder
- Reduce test development effort with automatic data correlation engine significantly

Simplify test execution

- Easily read and edit tests with visual representation of SAP activity
- Create and deploy realistic user workloads that exercise the key business transactions
- Leverage automated data variation and synchronization



Quickly identify performance problems and root causes

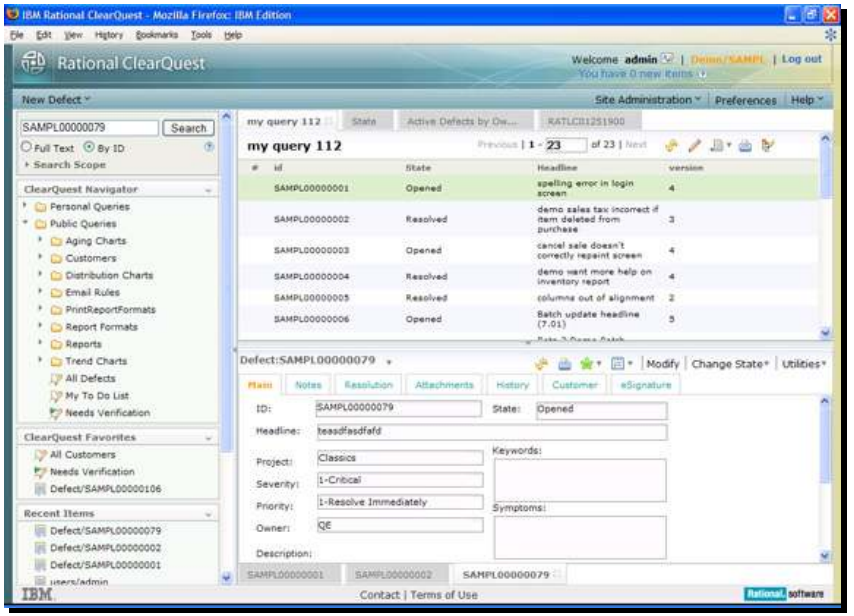
- Test and report against desired business SLAs
- Quickly pinpoint bottlenecks through graphical results representation
- Automatically identify the location and root cause of performance problems in hardware and software

IBM Rational ClearQuest

Integrated change and defect management for SAP and non-SAP projects

Reduce project risk improve project governance

- Leverage a set of best practice templates to facilitate governance of SAP implementation activities including enhancement requests, project deliverables, change requests and known defects
- Extend and customize SAP schemas with electronic signature and audit trail capabilities to automate regulatory and/or internal record keeping requirements



Accelerate project delivery and improve productivity

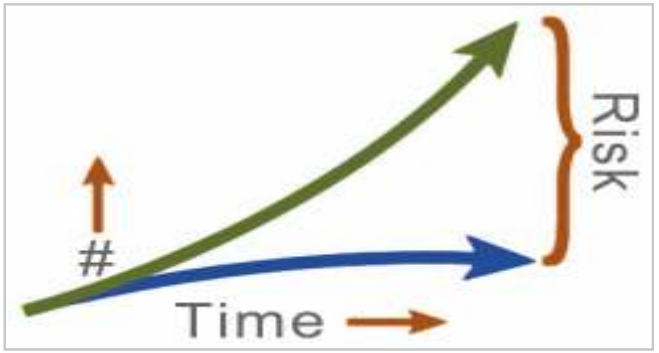
- Automate project defect/incident management through integration with IBM Rational Quality Manager and SAP Solution Manager
- Leverage flexible, customizable and automated workflows that provide repeatable, enforceable and predictable processes
- Enable secure and role-based stakeholder collaboration across project environments and globally outsourced and internally distributed teams

The Business Case for Automation

- Speed to Market is Top Priority
  - Limited / Fixed Resources
  - Manual Testing can't keep pace
  - Many changes go untested
  - No reusability means decreasing coverage
  - Time, turnover cause loss of knowledge
  - Backlog grows
  - Business agility suffers

- Traditional Methods are Too Costly
  - Script-based solutions require many customizations and an army of consultants/programmers
  - Writing code to test code is not the right answer

Risk Increases Over Time

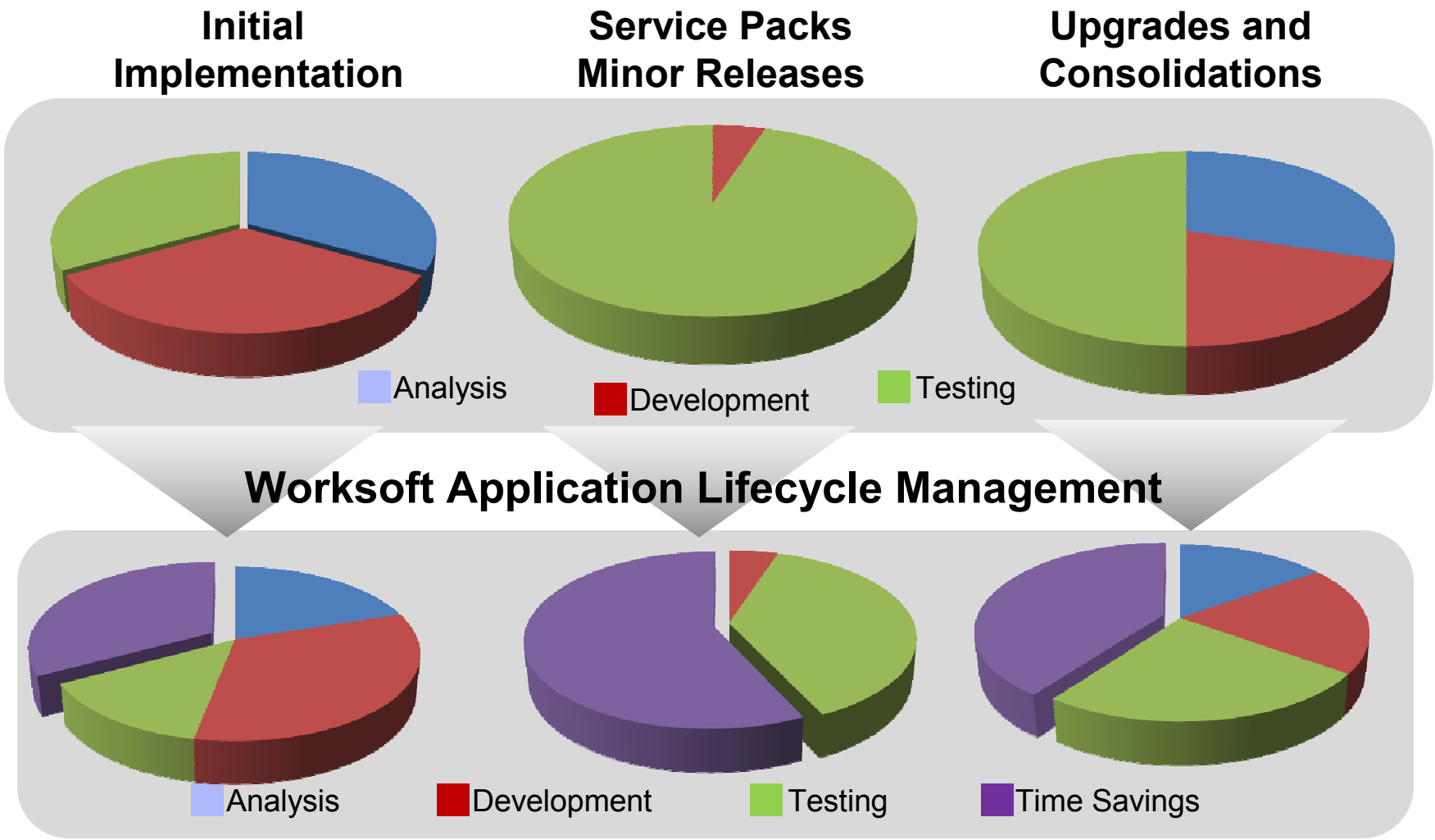


Business processes
Test resources

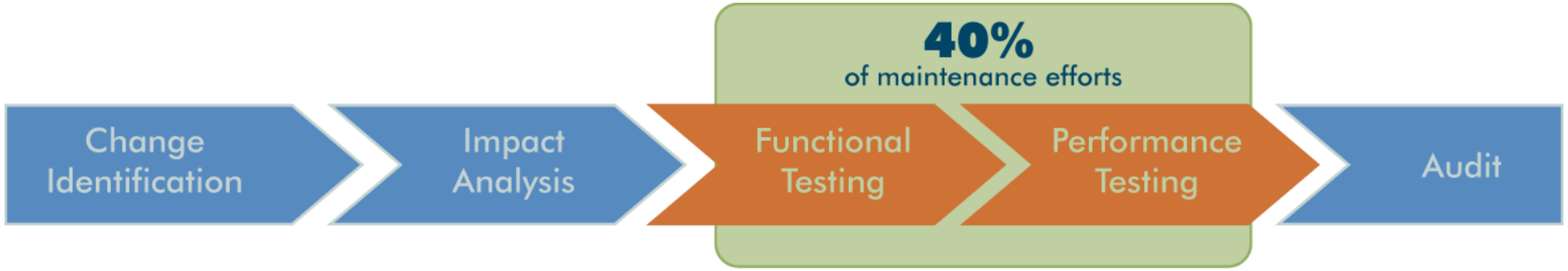
Automation is the Right Answer

- Reduced time and effort
- Expanded coverage
- More flexibility

Typical SAP Maintenance Efforts



Enterprise Change Management
The High Cost of Manual Testing



- Identify source of change
- Identify type of change

- Determine impact of change
- Calculate risk of change
- Recommend areas for validation

- Test case design
- Test case development
- Test case execution
- Test case management
- Test data management

- Load testing
- Stress testing
- Test data management

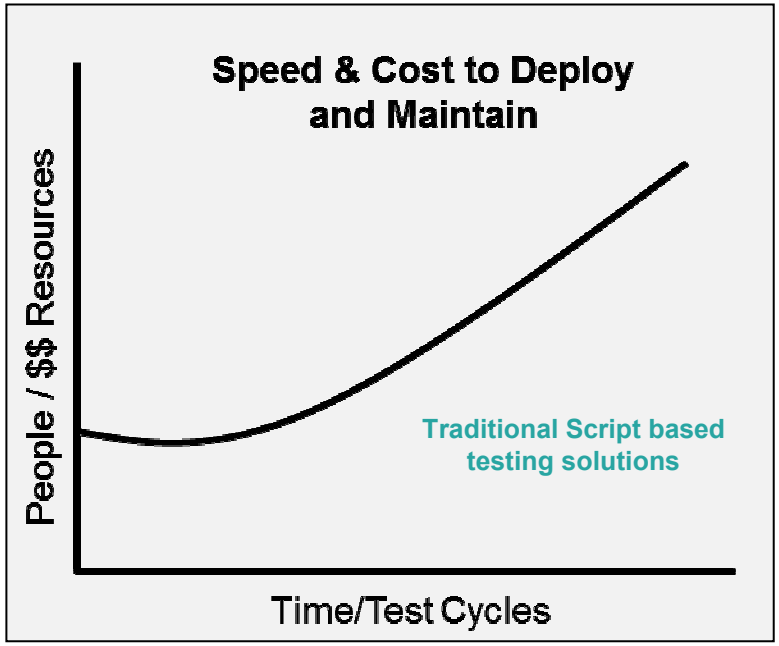
- Document process
- Report on progress - dashboard
- Capture auditable events – audit trail

Typical lifecycle process required for deploying application changes is people and time intensive

= 2 weeks to 6 months!

Challenges with Traditional Test Automation

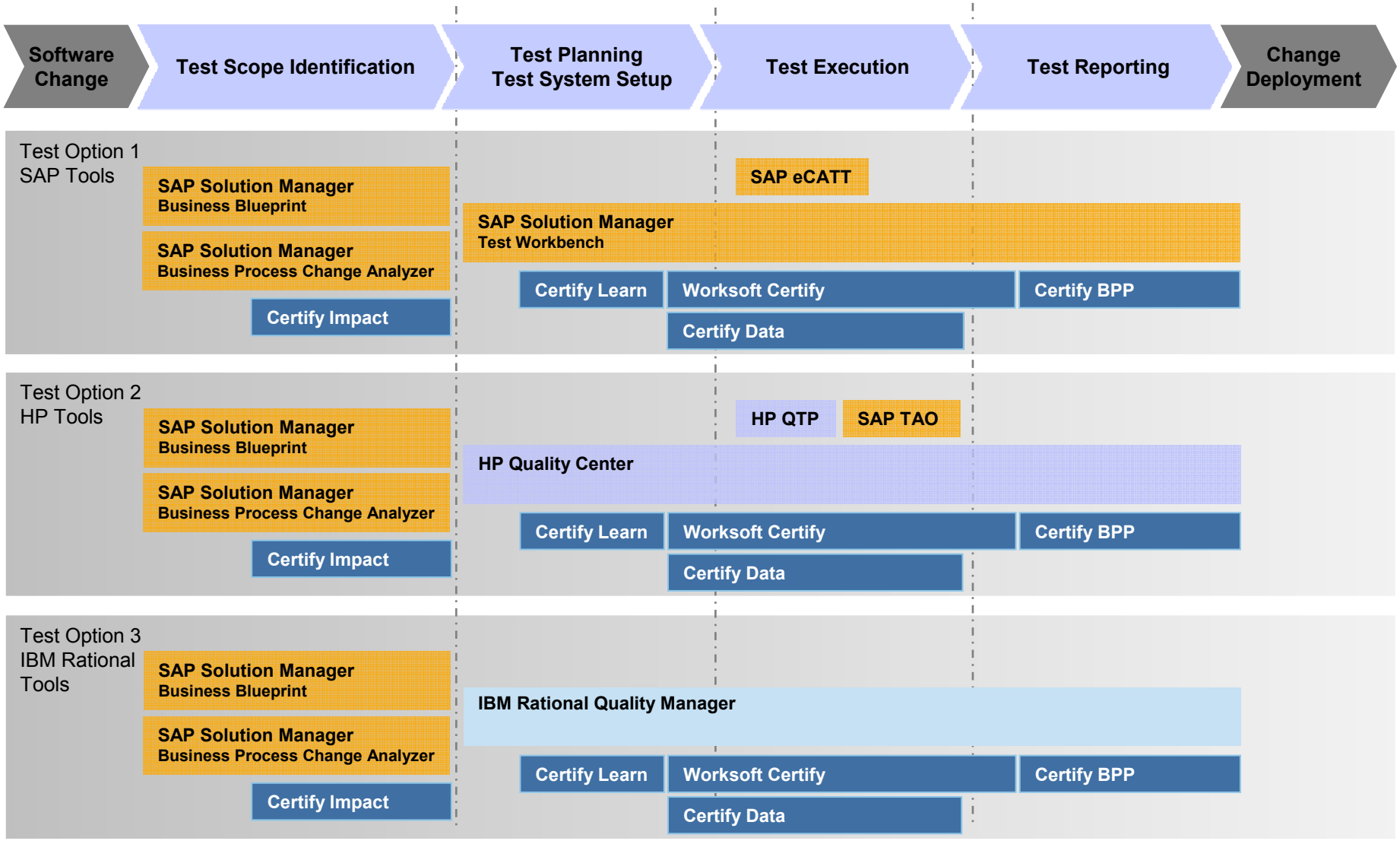
- 90%+ of SAP application testing is manual
  - Scripting tools have been around for 20 years
- Costs too much, takes too long
  - Need special skills to implement
- Test maintenance can't keep up
  - Too much code
  - Pace of change is too fast
- Manual testing is the bottleneck to accelerating deployment



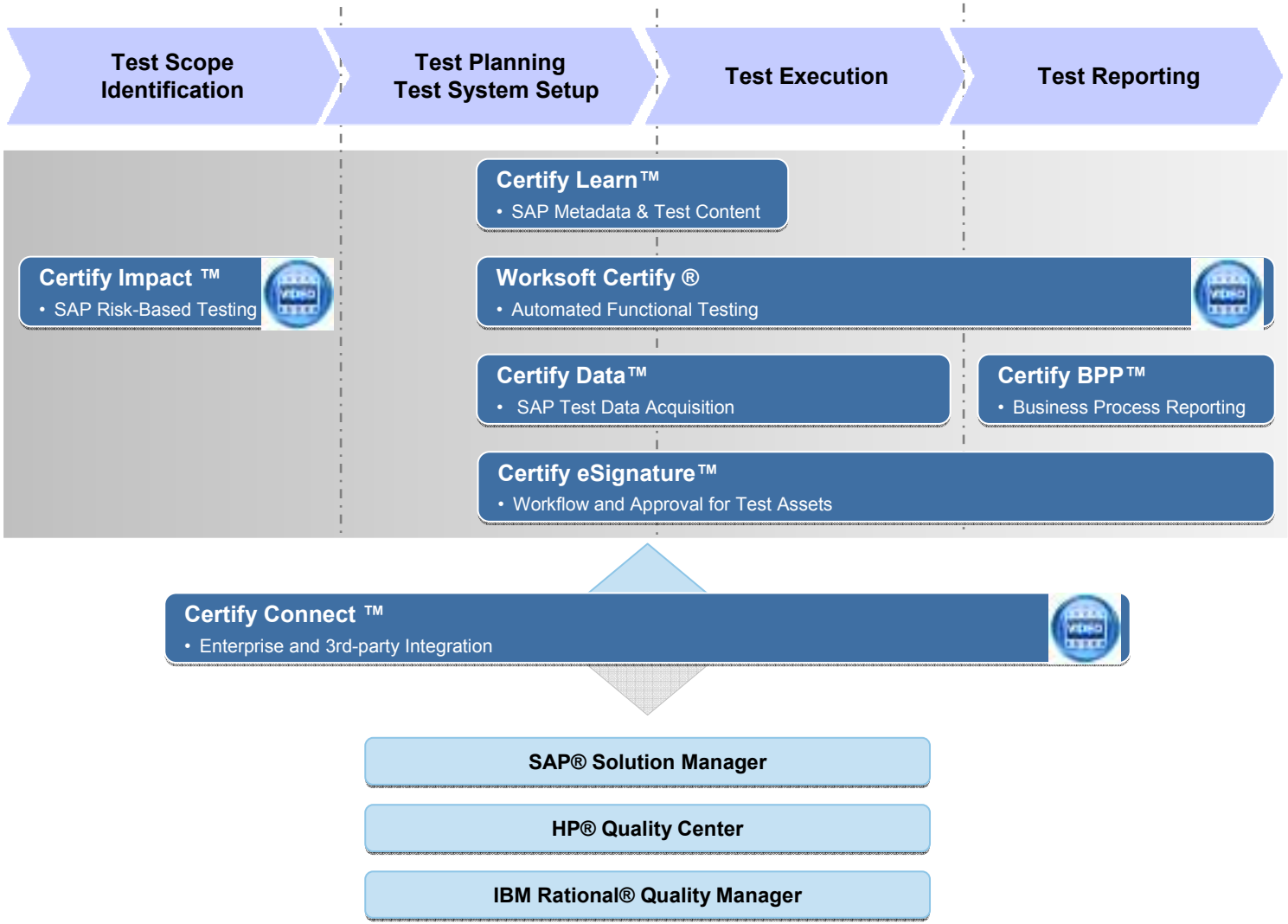
Worksoft changes the rules on testing

- **Faster Test Development**
  - No script code is written, generated, or maintained
  - Build the test in the time it would take to document the test case
  - Extract test data directly from SAP as the test is being written
  - Focus testing efforts by inspecting SAP transports and identifying the correct tests to run and any “gaps” in test coverage
  - Process and test documentation (BPPs, audit, training, test results) generated automatically
- **End to End Business Process Validation, One User Interface**
  - Manage, maintain and execute with one application
  - Test business processes end-to-end, across multiple applications and platforms
  - Significantly fewer assets to manage (250 ETE processes vs. 3000 manual tests)
- **Fast, Easy Test Maintenance**
  - Instantly identify and update tests for changes within applications
  - Low Cost, minimal resource requirements
- **Faster Time to Value**
  - Pre-built SAP test content
  - Easy to learn, easy to use by Business Analyst/SME
  - Easily obtain and manage SAP test data across environments
  - Document and automate in one step
  - Start test development ahead of code delivery to QA
  - Lights Out Testing…Increase test coverage of business processes to near 100%

Supported SAP Testing Tool Suites



Worksoft Product Suite



Worksoft Product Suite – Spanning the Lifecycle

- Certify LiveLearn & Certify SAP Test Content
  - Automates the process of loading all SAP objects into Worksoft Certify allowing SAP teams to define and execute more automated tests in a shorter period of time
  - Includes comprehensive set of pre-built, re-usable SAP test content
- Certify Connect
  - Automates the connection of Worksoft Certify to enterprise or 3rd-party applications
  - Provides bi-directional integration with all 3 leading SAP Test Management solutions:
    - SAP Solution Manager
    - IBM Rational Quality Manager
    - HP Quality Center
- Certify Impact
  - Automated risk-based testing solution that can identify, recommend, and execute end-to-end business process tests based on a set of proposed SAP application changes and their relationship to critical business processes (leveraging SAP BPCA)
  - Also identifies Certify tests that need to be updated, as well as a “gap” report listing SAP changes that do not have a corresponding Certify test

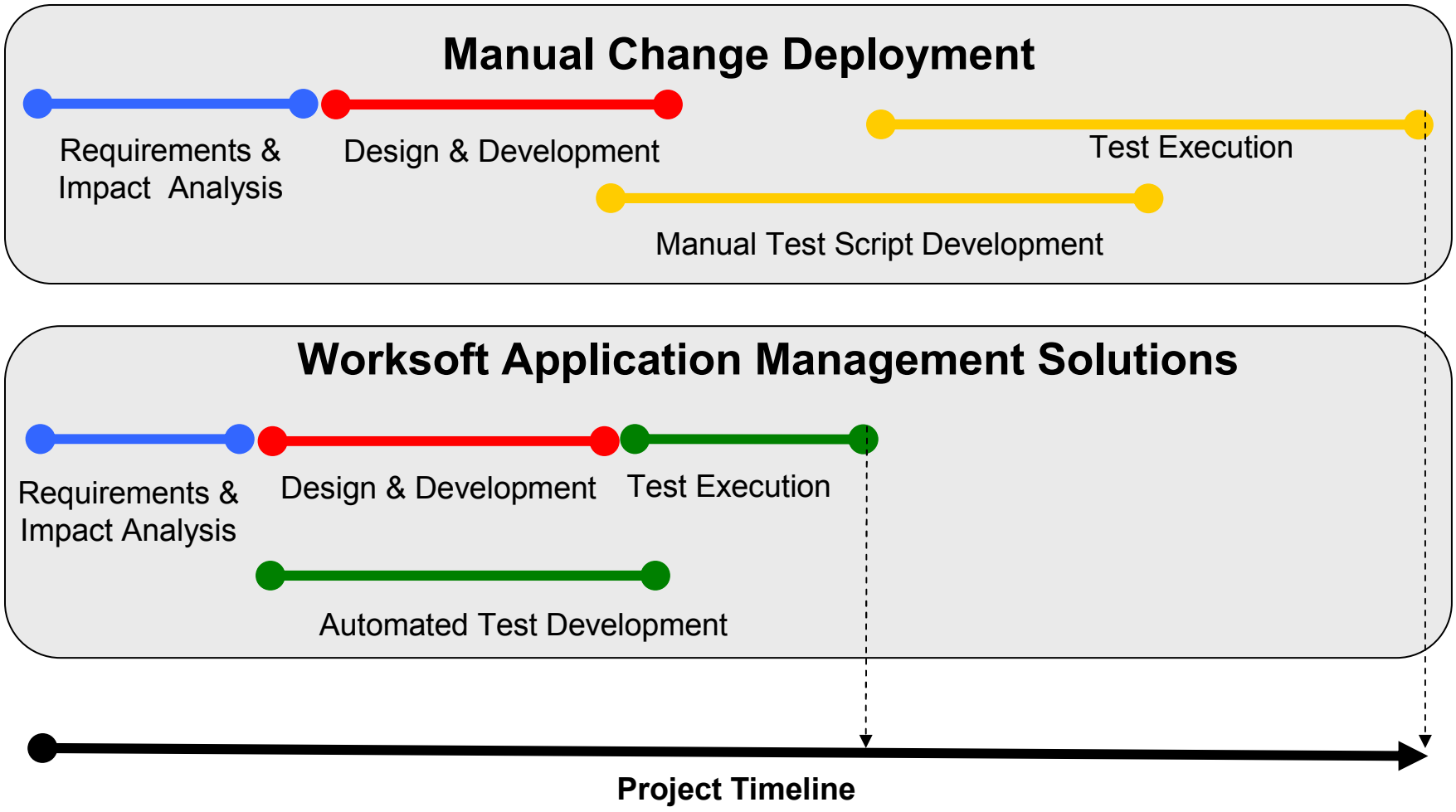
Worksoft Product Suite – Spanning the Lifecycle

- Certify Data
  - Automates the extraction of SAP data and storage in the corresponding Worksoft Certify test process as the test steps are constructed
  - Minimizes the initial and recurring costs of test data acquisition and validation
    - Initial: During test construction, Certify Data extracts test data directly from the application under test
    - Recurring: Existing test data may be maintained and refreshed from within Certify
- Certify Business Process Procedures (BPP)
  - Automates the generation of comprehensive documentation for quality assurance, training and compliance from the results of a Certify test process execution
  - Documentation is based on templates that can be customized to meet specific requirements
- Certify E - Signature
  - Automates the workflow and approval for Worksoft Certify assets including test requirements, test processes and test results
  - Records and preserves test assets and generates documentation to aid in compliance

Worksoft Certify for SAP

- **Simple**
  - All solutions designed with non-technical, business users in mind
  - Create automation by simply interacting with the application
- **Script Less**
  - Object/Action paradigm fundamentally different from other automation solutions
  - No scripts are ever generated, stored or required to be used
  - No programming skills are required to create or maintain automated tests
- **Reusable**
  - Automated tests can be reused from one customer to another saving hundreds of man hours
  - Lights out regression tests can be used over and over again and can be run on a daily/weekly basis to ensure core business processes are functioning correctly
- **Maintainable**
  - Simple single click maintenance by comparing maps from a new version with an old version of the application and automatically identifying/updating affected test steps
- **Versatile**
  - Support provided for almost any kind of software technology used to build the enterprise/portal application

Certify Speeds Development Reducing Project Timelines



BPP Documentation is real time, and auto generated

It Can Be Done

- **Dow Corning** [Link to Case Study]
  - Experts being lost to rotation, retirement
  - Automation enabled knowledge capture, transfer to others
  - 75% processes automated in 90 days
  - 250 processes executed every night
- **Large Airframe Manufacturer**
  - 8 man months saved in first project
  - Refreshed user knowledge of business processes
  - Fewer emergency transports due to increased coverage

Extensive Customer Experience

Green Hat Overview

Green Hat delivers better software quality with its unique virtualization and integration testing capabilities

Organization	▪ Founded in 1996, Green Hat has a built a strong expertise in integration testing and virtualization for testing ▪ Jointly headquartered in London, England and Wilmington, DE
Capabilities	▪ **Makes automated testing simple** for complex systems relying on Cloud, web services, messaging, SOA, ESB, BPM, SAP, etc. ▪ Enables **earlier and more parallel, continuous testing** across the development lifecycle
Technology	▪ **Mature** technology (version 5.3) ▪ **Superior** architecture for scalable enterprise deployments ▪ **Broad** support for middleware and applications ▪ **Ready for IBM Rational** software
Customers	▪ 80+ customers ▪ **Marquee** financial, healthcare, government, and Telco customers ▪ Used by several GSIs, including IBM Global Services

Green Hat customers include…

The combination of IBM and Green Hat solutions delivers first class quality management and agile development capabilities

IBM is a leader in…

- Application lifecycle management
- Quality management
- Scaling agile development

Green Hat is a leader in…

- Integration testing and application virtualization

Quality Management Capabilities

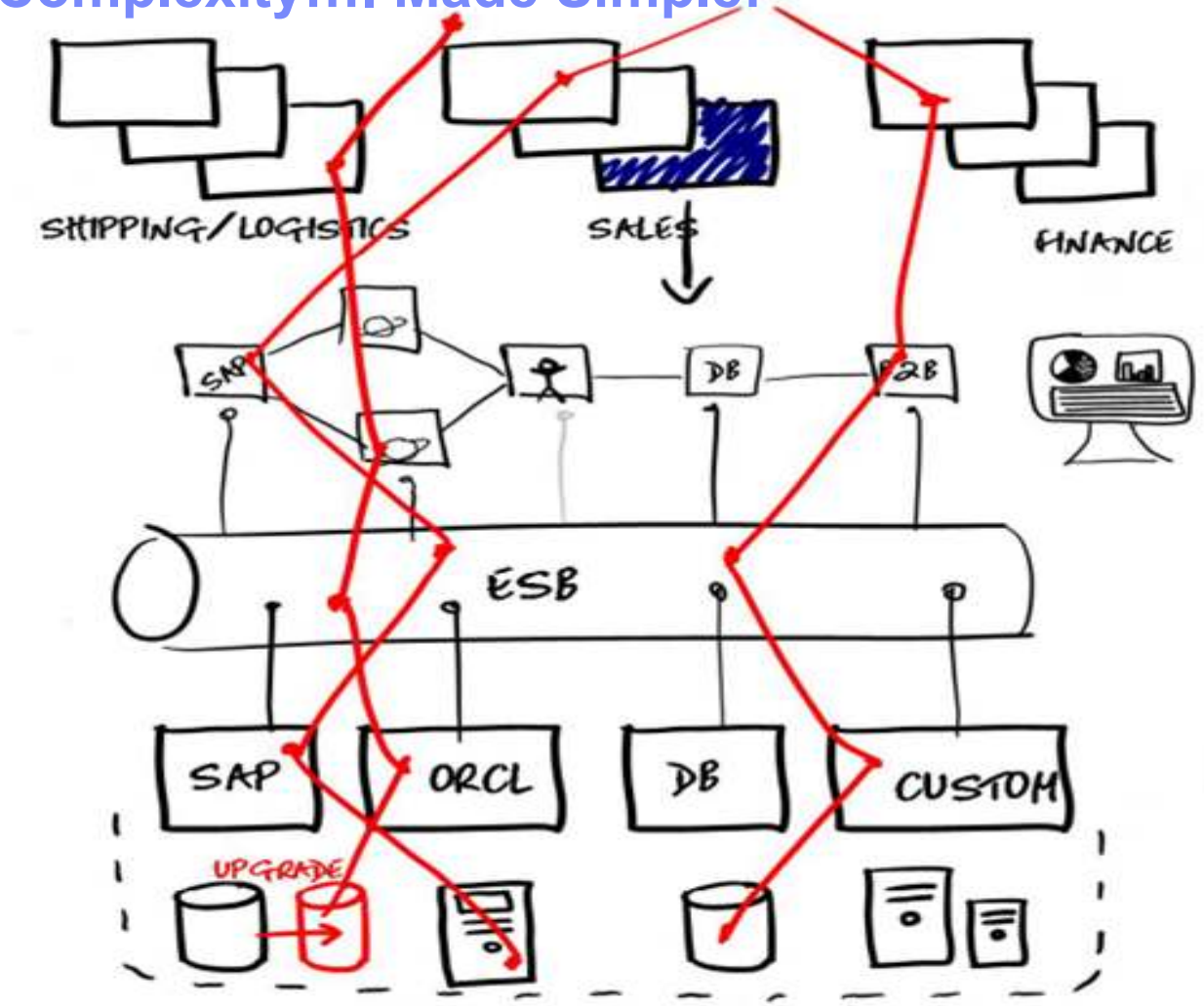
Dynamic Planning | Risk Management | Test Management | Analytics

Coverage Analysis | Defect Resolution

Agile Development Capabilities

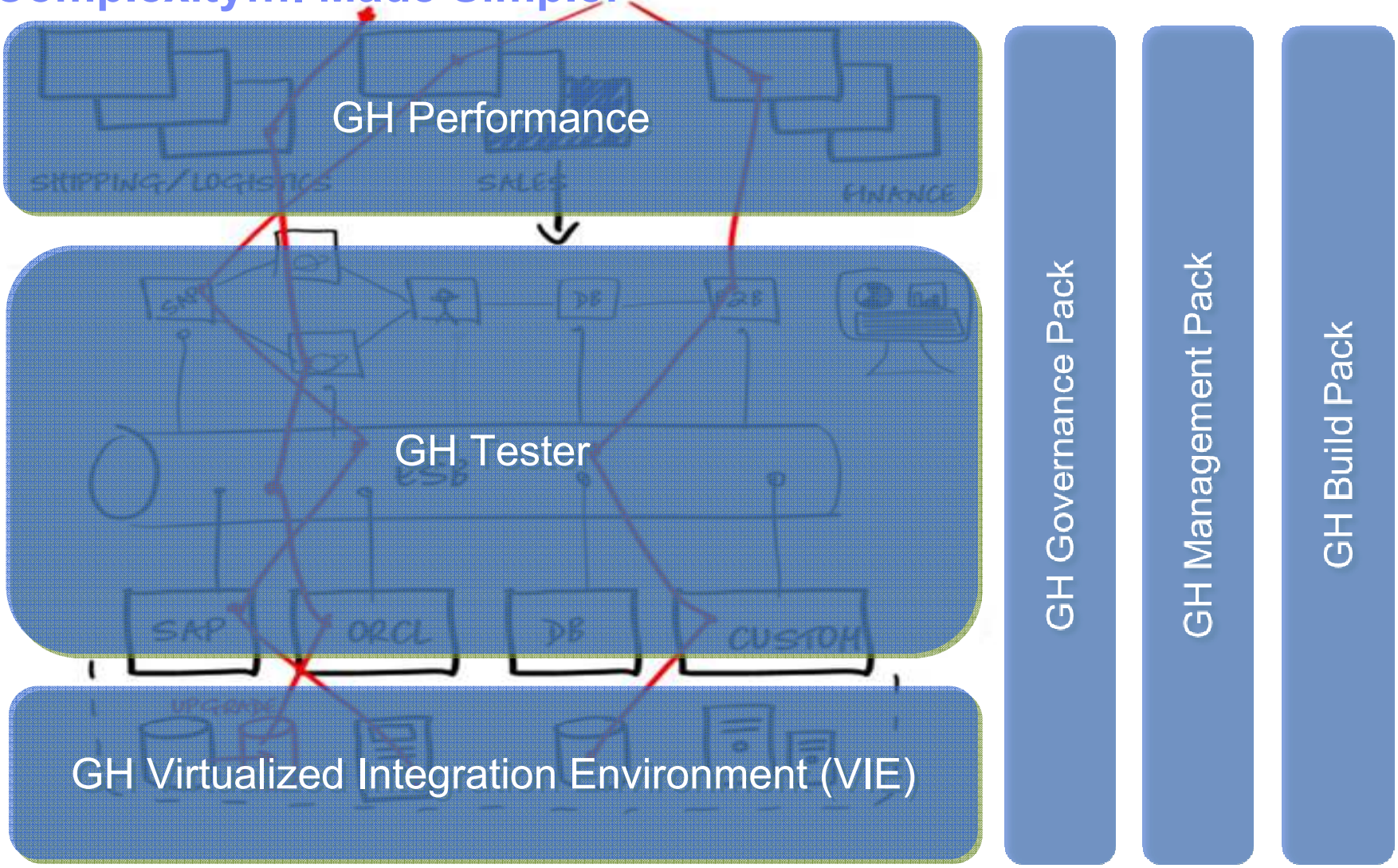
Analytics | Collaboration | Visibility | Process | Integration Testing First

Complexity…. Made Simple!



- Multiple layers of technology
- Multiple vendor platforms
- Complex transactions
- Complex dependencies
- Multiple stakeholders

Complexity…. Made Simple!

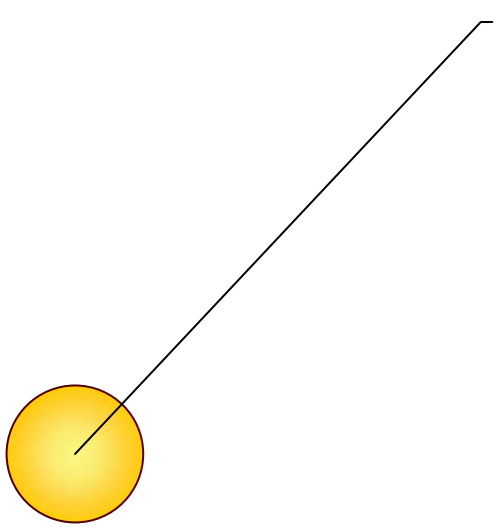


Technologies: 70+

Messaging Protocols	SOA and ESB	Message Formats
ActiveMQ BEA Tuxedo Email (SMTP, IMAP) Files FTP/S HTTP/S JMS (JBOSS et al) IBM WebSphere MQ JBoss MQ SAP IDoc, BAPI, RFC & XI/PI Software AG's IB & IS Solace Sonic MQ FIX TCP TIBCO Rendezvous, Smart Sockets & EMS Custom	CentraSite Oracle Fusion SCA Domain Software AG IS, BPMS Sonic ESB TIBCO ActiveMatrix UDDI Web Services WebSphere RR WSDL **Other Technologies** BPM Databases Log Files GUI	.Net Objects Bytes COBOL Copybook ebXML EDI FIX Fixed Width HL7 IATA Java Objects MIME OAG SOAP Software AG Broker Docs SWIFT TIBCO ActiveEnterprise XML (DTD, XSD, WSDL) Custom

Functional Test Automation

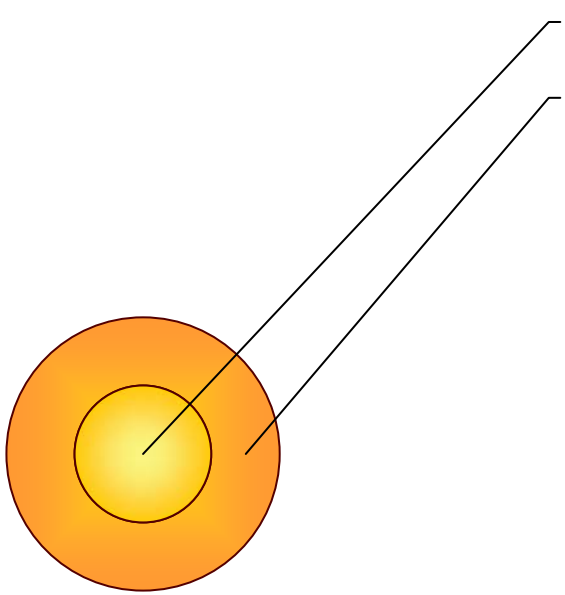
Build and Send Payloads 1

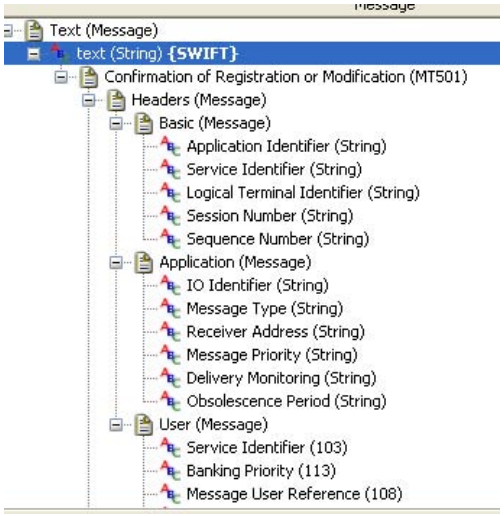


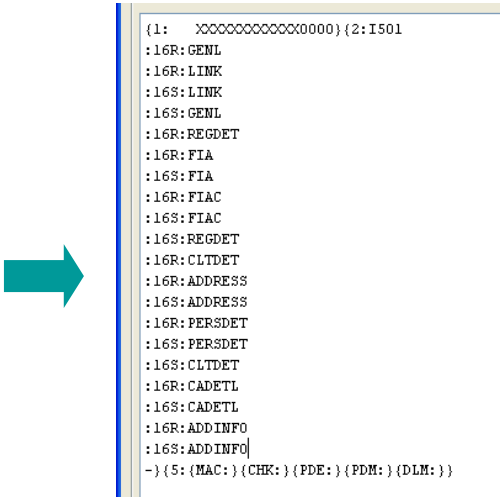
e.g. SWIFT, FIX, OFAC, Custom

```
{1:   XXXXXXXXXXXX0000}{2:I501            }{3:{103:}{113:}{108:}{119:}{115:}}{4:
:16R:GENL
:16R:LINK
:16S:LINK
:16S:GENL
:16R:REGDET
:16R:FIA
:16S:FIA
:16R:FIAC
:16S:FIAC
:16S:REGDET
:16R:CLTDET
:16R:ADDRESS
:16S:ADDRESS
:16R:PERSDET
:16S:PERSDET
:16S:CLTDET
:16R:CADETL
:16S:CADETL
:16R:ADDINFO
:16S:ADDINFO
-}{5:{MAC:}{CHK:}{PDE:}{PDM:}{DLM:}}
```

Build and Send Payloads 2

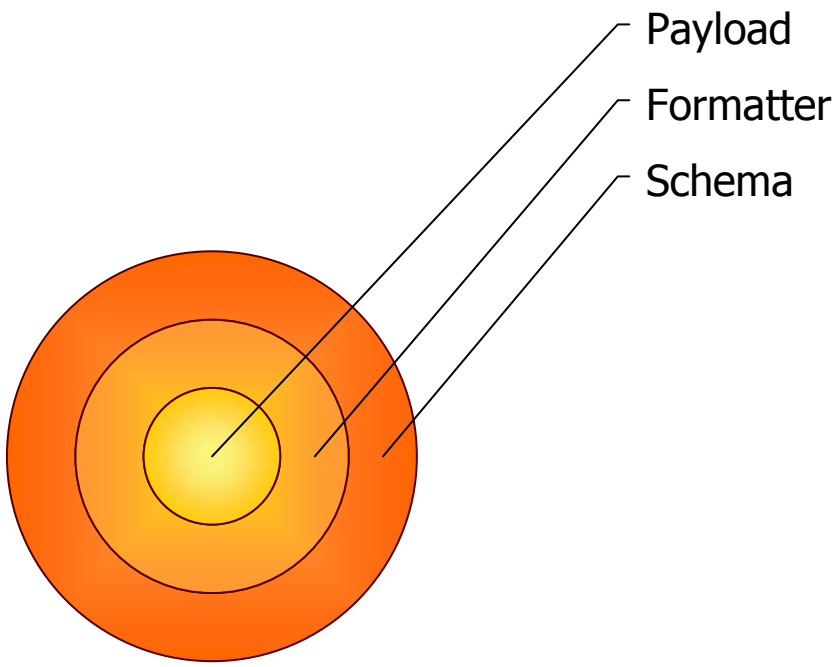






```
{1:   XXXXXXXXXXX0000}{2:I501
:16R:GENL
:16R:LINK
:16S:LINK
:16S:GENL
:16R:REGDET
:16R:FIA
:16S:FIA
:16R:FIAC
:16S:FIAC
:16S:REGDET
:16R:CLTDET
:16R:ADDRESS
:16S:ADDRESS
:16R:PERSDET
:16S:PERSDET
:16S:CLTDET
:16R:CADETL
:16S:CADETL
:16R:ADDINFO
:16S:ADDINFO
-}{5:{MAC:}{CHK:}{PDE:}{PDM:}{DLM:}}
```

Build and Send Payloads 3



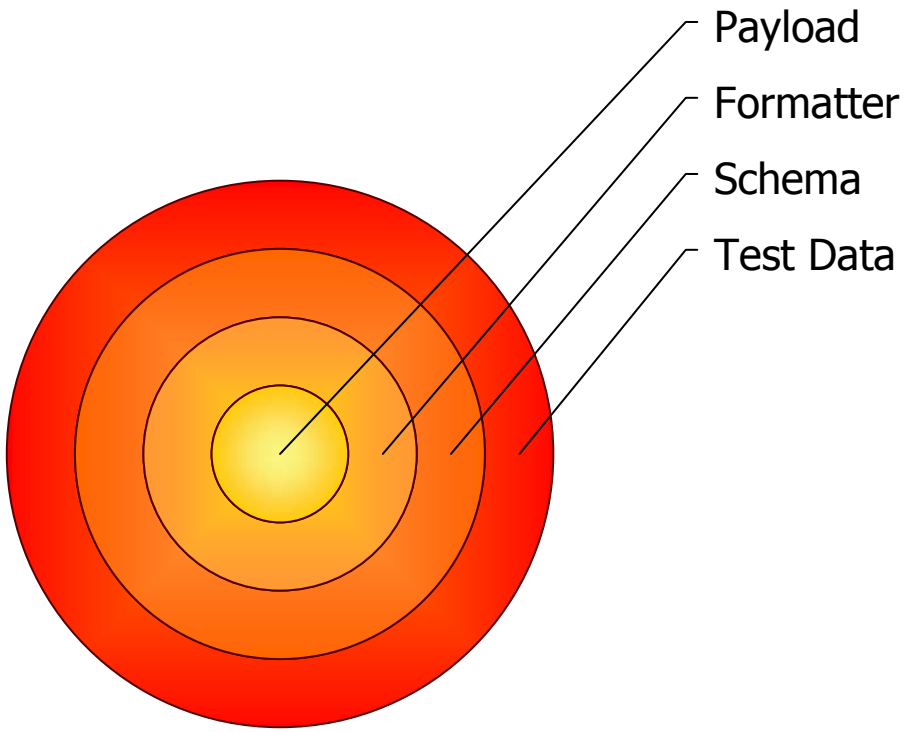
SOAP	XML
SWIFT	COBOL Copybook
DTD	XSD
WSDL	HL7
EDI	IATA
Text	MIME
Byte Array	Java Objects
OAG	SAP BAP/RFC
FIX	.Net Objects

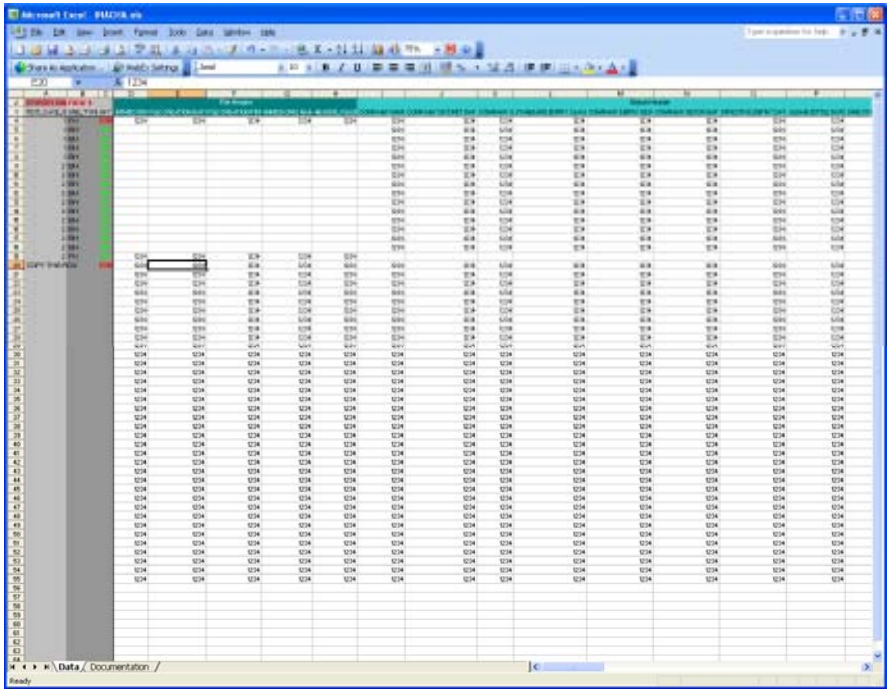
TIBCO ActiveEnterprise

webMethods IB and IS Documents

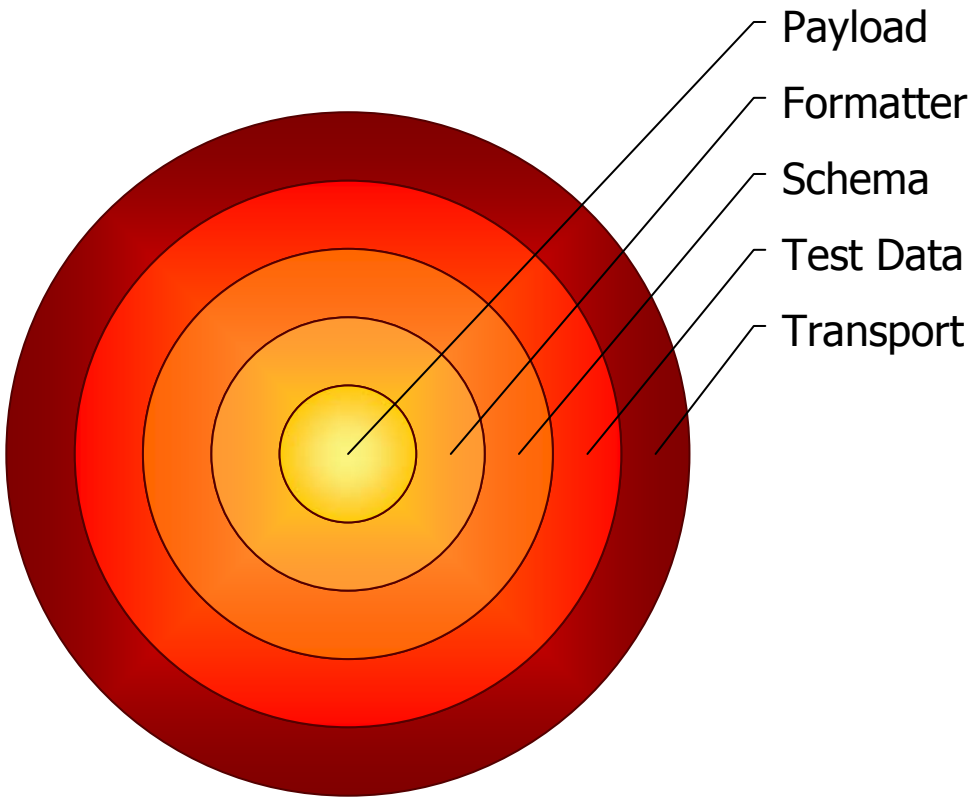
Custom…

Build and Send Payloads 4

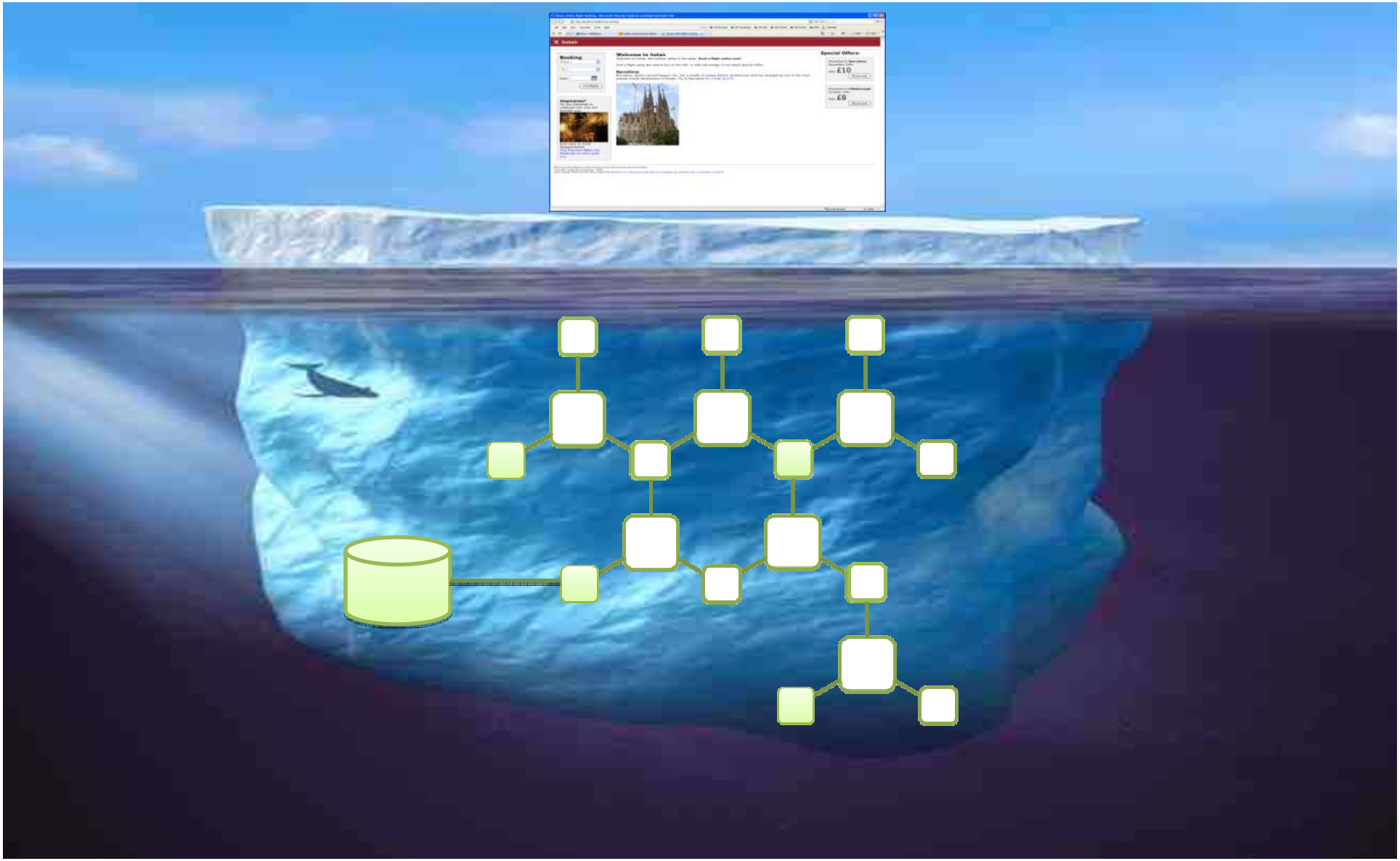


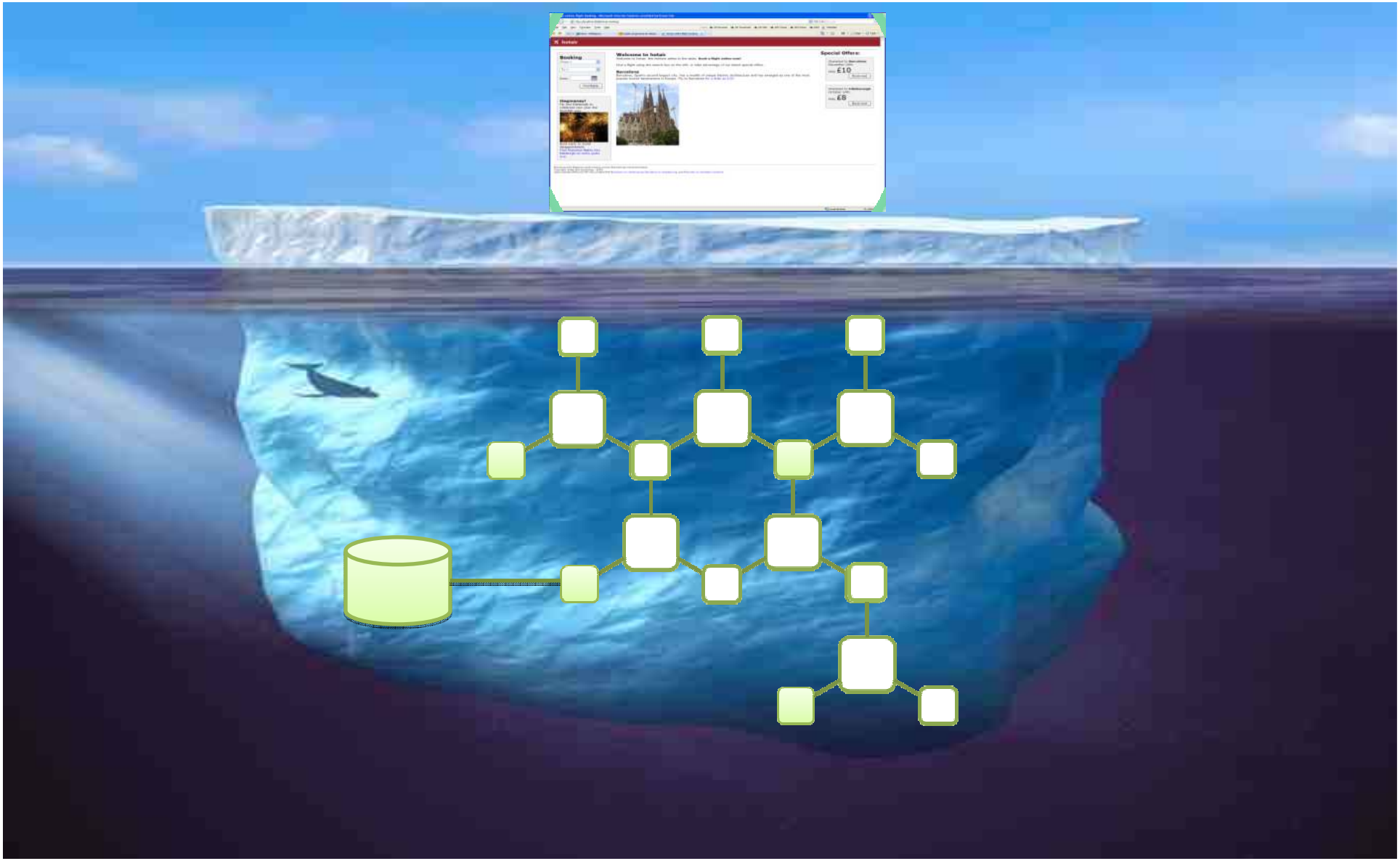


Build and Send Payloads 5

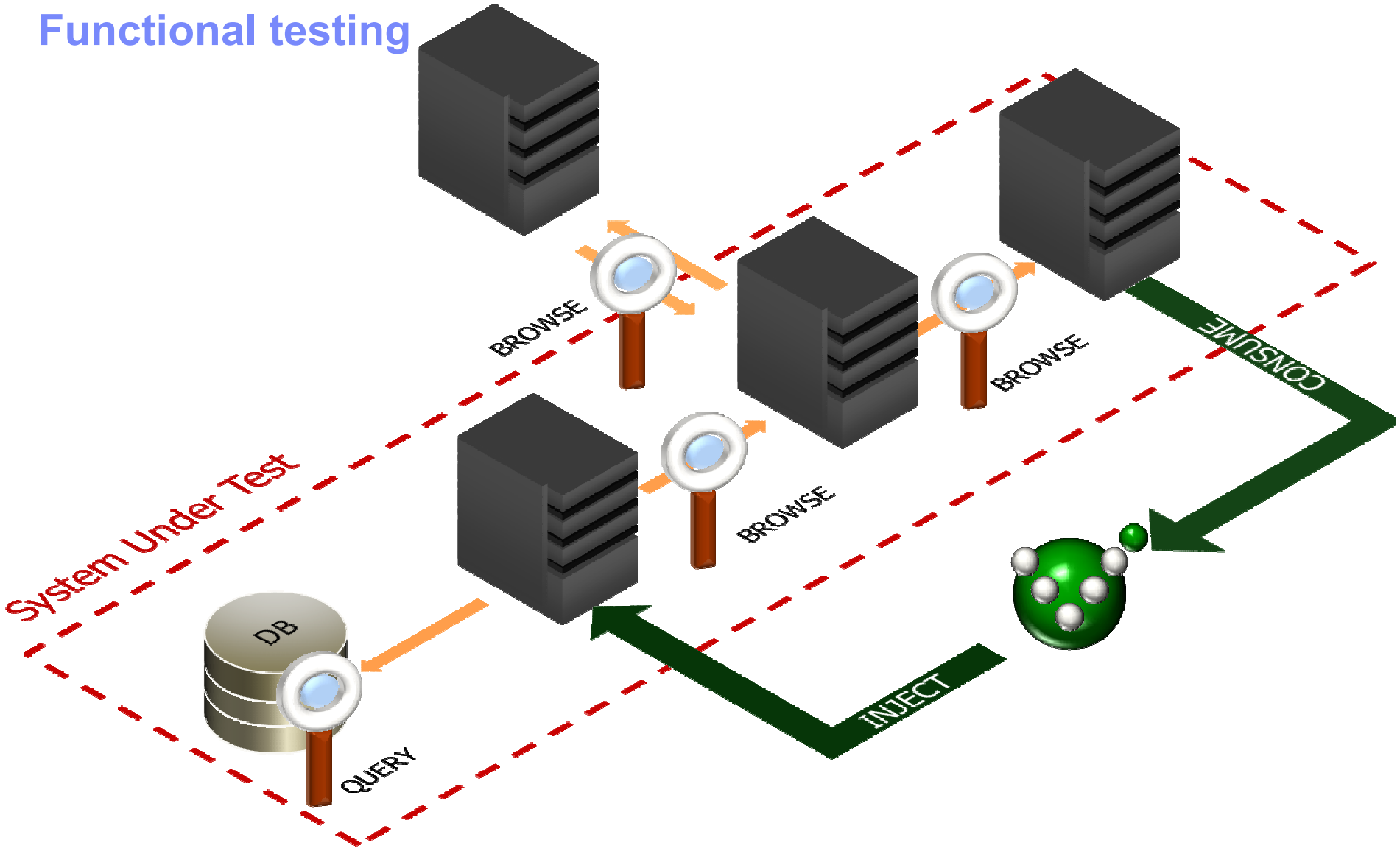


JMS	webMethods
HTTP/S	FTP
NDM	Oracle/BEA
JDBC	Sonic MQ
TCP/UDP	PL/SQL
Flat Files	Shell commands
TIBCO EMS	TIBCO iProcess
TIBCO Rendezvous	
IBM WebSphere MQ	
Custom Transports…	





Functional testing



Simple Virtualization

Blockers for e2e testing

"Too costly to setup a test message feed for test"

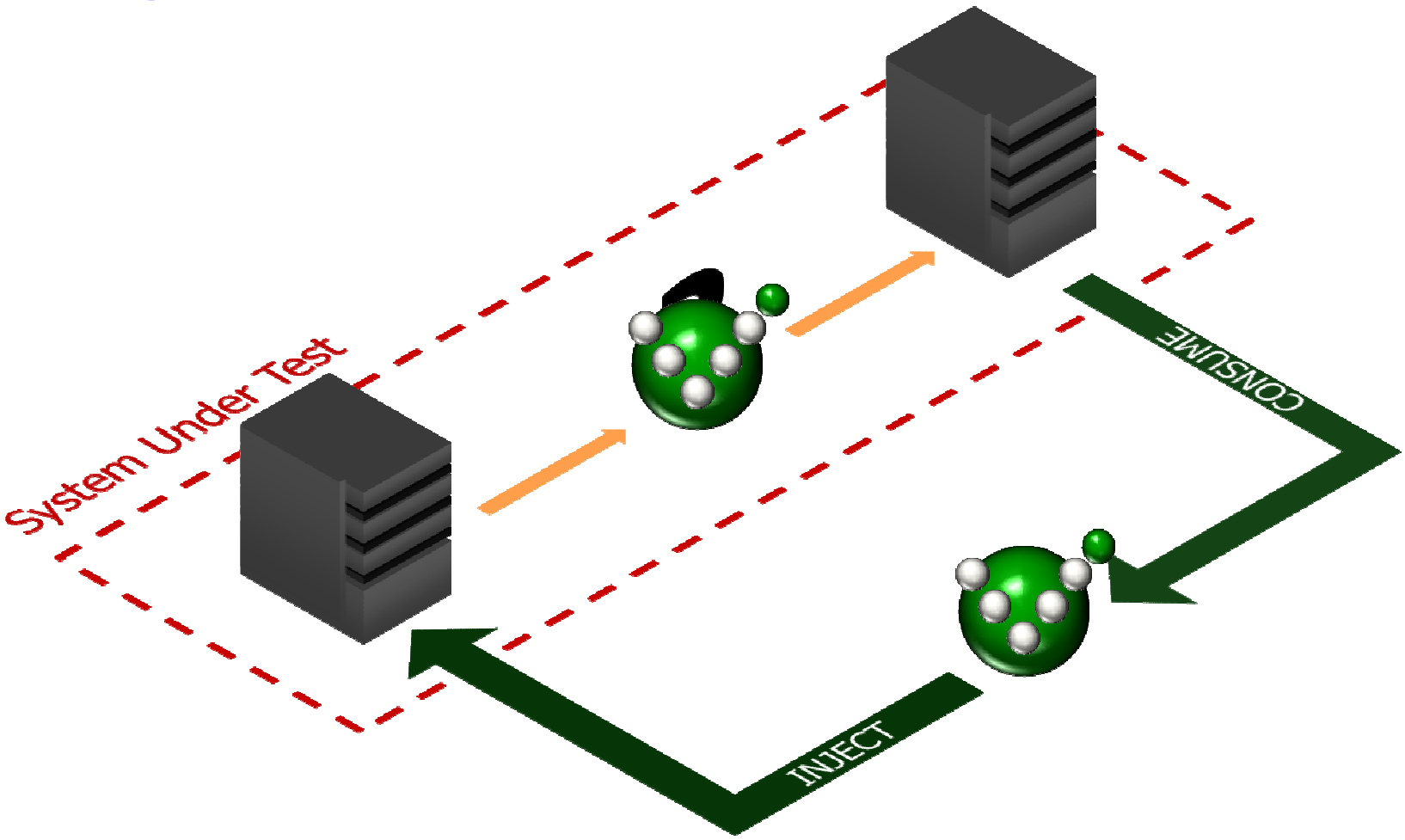
"We're still waiting for them to deploy"

"We don't have anything to test against"

"The environment will be another 3 weeks"

"The dev team hasn't started work on it yet"

Stubbing



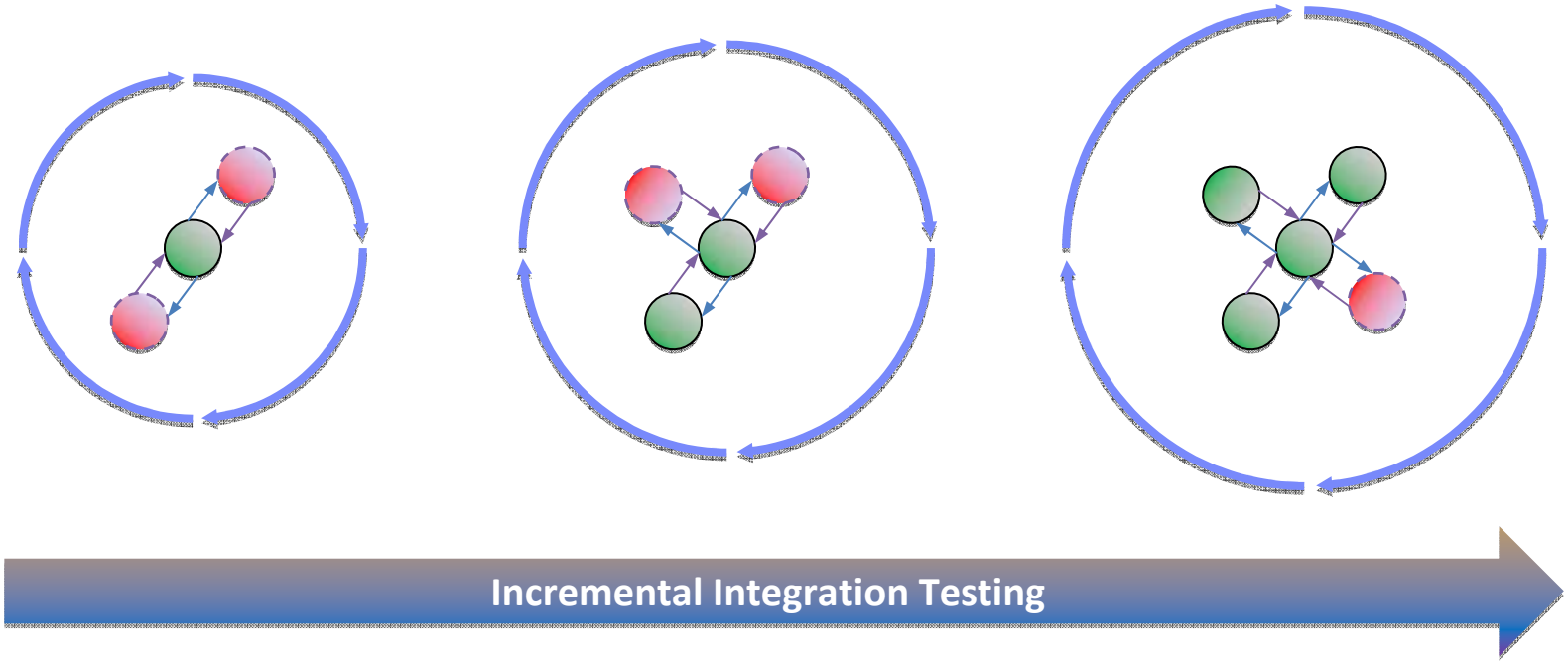
Incremental Integration Testing

Units are introduced into the continuous integration cycle in a prioritised, controlled fashion.

Units not yet built can be simulated and tested against..

Actual Component

Simulated Component

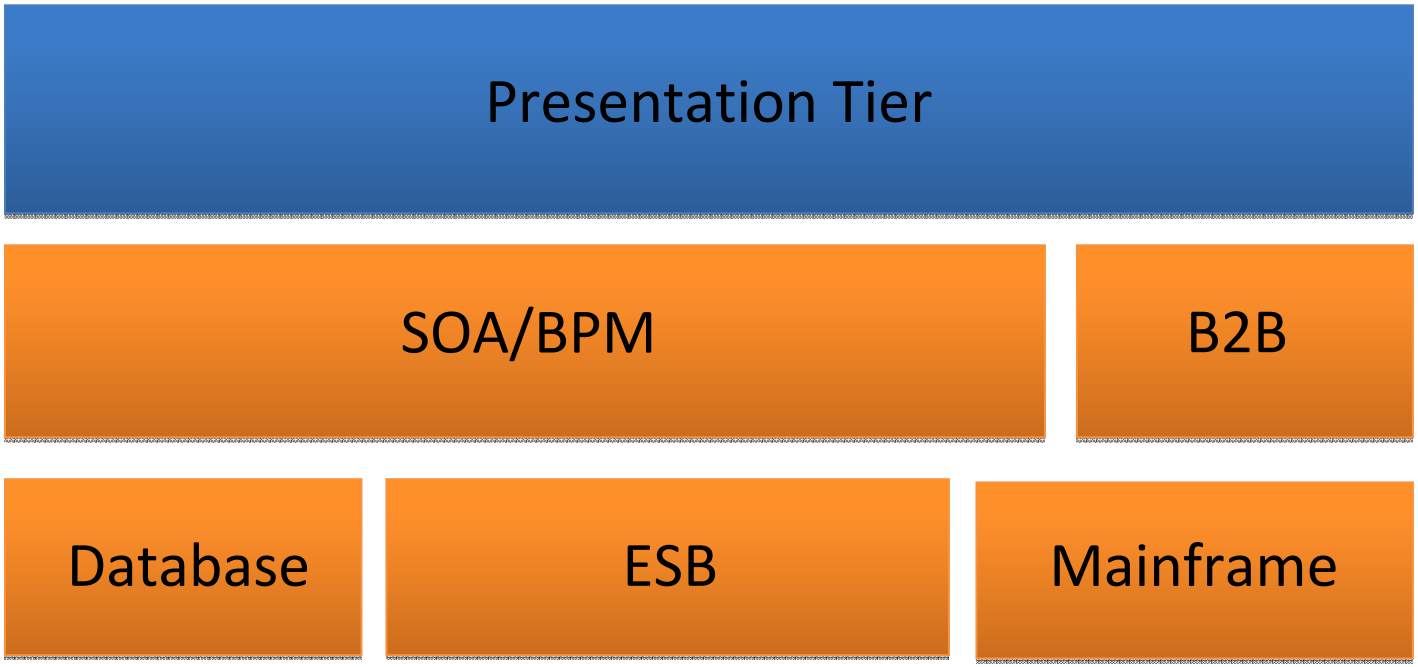


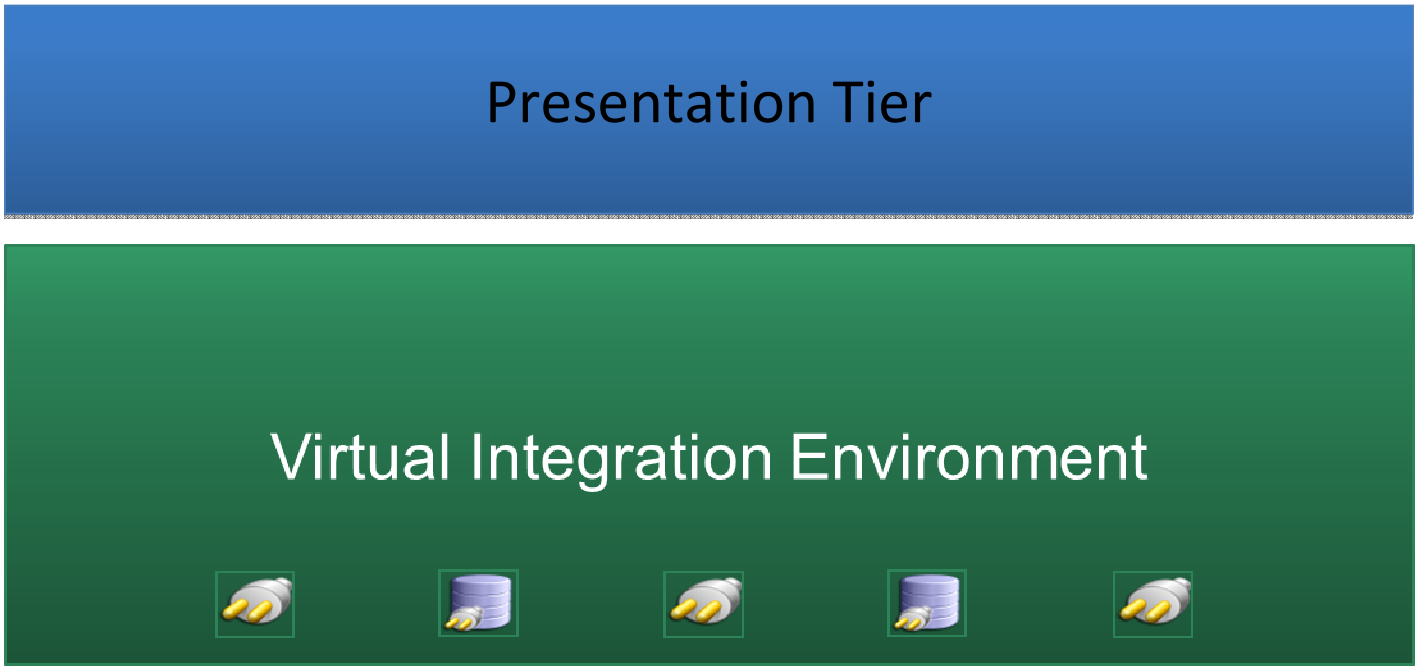
Complex/Orchestrated Virtualization

What is VIE?

"VIE is a means of virtualising a complete application domain and dependent services that are understandable and can be built without relying on teams of developers for coding"

- Tools for capture/analysis of the business domain
- Tools for automated generation of virtual applications
  - Extensible & code free (Stateful, Stateless, Behavioural, Model Driven)
- Management interfaces to enable remote administration
- Repository to enable reuse and collaboration





- Virtualise entire set of applications
- Users unaware of virtualisation

Virtual Applications Evolve as Project Matures

Problem	How VIE Solves It
In a Green Field, no systems exist at the beginning	Virtual Applications and their models can be built from interface definitions and fleshed out with whatever logic is required
Inconsistent access to resources in non-Green Field projects	Virtual Applications can be built easily from recordings of working systems
Business partner has no test system	Virtual Application replaces business partner
Don't want to modify system to facilitate recording/stubbing	VIE has a battery of techniques to circumvent this problem
Need to model *intended* as well as existing functionality	Virtual Applications can be built in conjunction with business rules

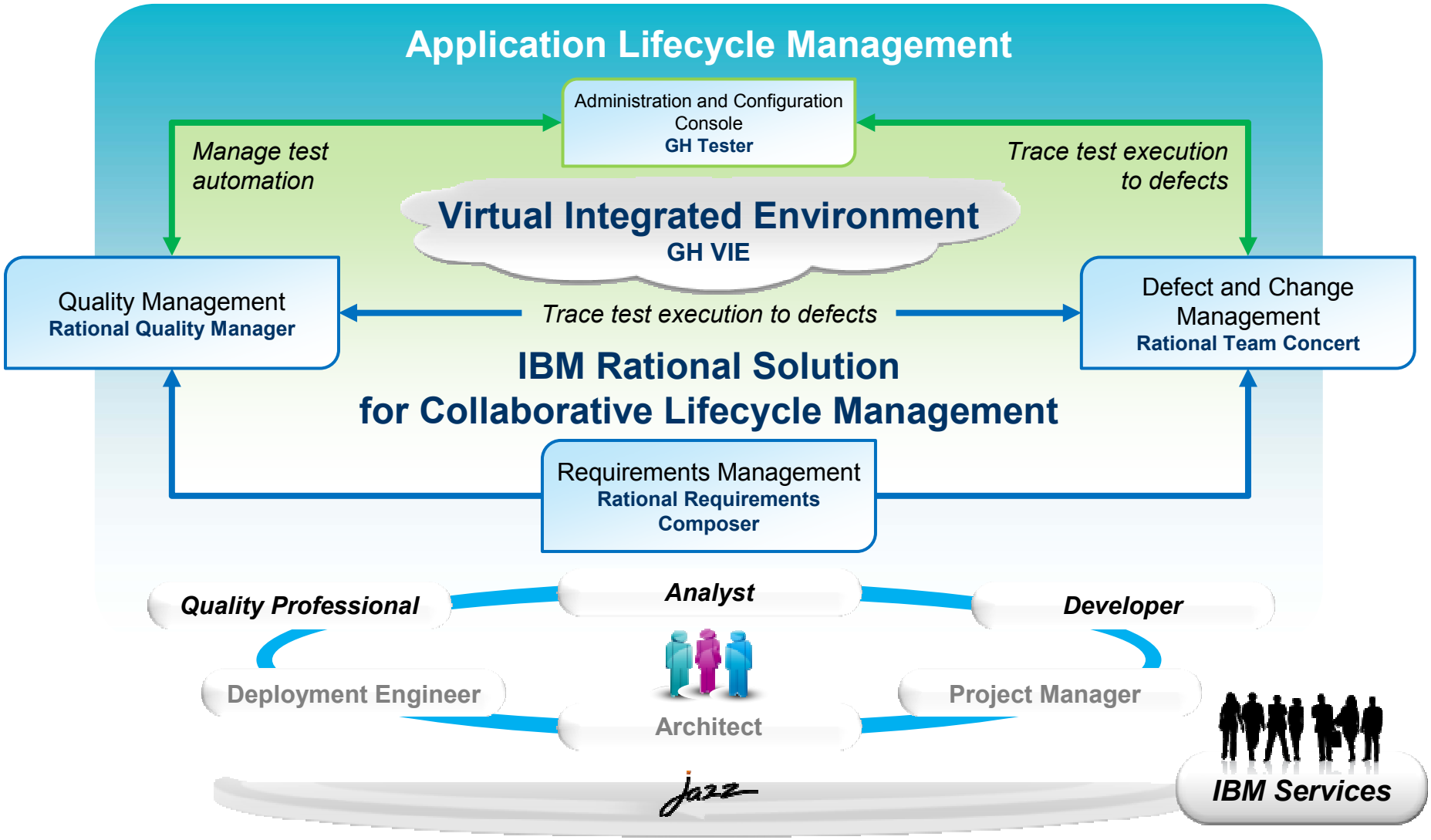
VIE Benefits…

No need to modify application code	Variety of tools to enable creation of virtual service eg recording	Frees projects from external constraints eg databases
Simplifies management of Stub execution	Enables traceability over where stubs are running	Provides audit of who ran stubs and when
Enables simple versioning of stubs	Removes delays caused by late delivery of external interfaces	Speeds up development of stubs for testers and developers
Stubs can be reused and embellished to support different test cycles	Allows control over datasets being used across an environment	Enables erroneous data to be played back into systems
Removes the need for costly 3rd party interface leasing	Let's developers get on with developing code not stubs	Puts testers back in control and removes dependency on other teams
Easily simulate "+1" changes to interfaces and environments	Enables multi interface, complex and stateful simulation easily	**Limits risk and lowers integration issues when going into production**

In Simple Terms, if you remember nothing else!

- Greater change frequency requires more regular releases, which means more testing
- So more organizations interested in reducing the cost and complexity of testing, and testing earlier in the lifecycle
- This means doing integration testing earlier, when only some of the components are available
- It also means testing at a different layer, more API and service testing than traditional UI test automation
- The offerings acquired from Green Hat address this problem from two directions:
  - More testing means more environments and more dependencies. **GH VIE** addresses this problem by providing a new way to deal with test environment, creating lightweight virtual counterparts of the components the system under test connects to, allowing test environments to be smaller, faster, cheaper and in the control of the individual developer or tester
  - When it comes to testing at a different layer, **GH Tester** is a flexible, extensible tool with out-of-the-box support for dozens of different technologies used behind the UI in the modern enterprise, be it SOAP, JMS or obscure file formats from mainframes sent over MQ 15 years ago, the same users can test all of these technologies earlier than they would with other traditional products

Test virtualization capabilities from Green Hat extend IBM Application Lifecycle Management



Conclusion


Developing Software
Should
Be This Fun
Rational® software

Da dove iniziare?

2009 Quality Achievements

✔ Verification best practices, recommendations and test methodologies model in place. Improvements in Testers participation in requirements and design phase, Test Planning, Documentation, Tracking, Execution, Closing, Best practices, Tooling, Automation, resolved test team issues

✔ Regular monthly meetings with Test leads and testers, to share tools, best practices, productivity and quality improvements. XXXXi Test Corner Wiki and Lotus Connections Community in place to share knowledge.

✔ Consultancy to test teams to drive systematic improvement in product and solution quality

✔ Interface with other XXXXX Verification organizations/teams (i.e. QSE, etc.) to identify and contribute to best practices/methodologies

✔ Established links with Rational Quality management products and planned workshop on RQM

✔ Participation in XXXXi Test Architect Board activities to work towards consistency and continued improvement across XXXXi test teams

✔ Ongoing coaching and monitoring to testers and test leads to support their professional development and career plans and to improve Quality

Measuring success

- Operational metrics
  - Show progress of implementation
- Business impact metrics
  - Show value of test automation
  - Defect arrival rates over time
  - Number of times a test case is executed before it passes
  - Time spent on execution of tests and analysis of results
  - Defects found by customers
  - % of test cases automated
  - % of time spent on test execution and analysis
  - Number of runs of a test suite
  - Number of environments covered per iteration
  - Number of assets contributed for reuse and reused

Next Steps – Learn and Explore

- ibm.com/rational
  - Full Product Information
  - Product demonstrations
  - Product datasheets
- ibm.com/developerWorks
  - Download Evaluation Software
  - User Forums
  - Product tutorials
  - Tech Notes, etc…

Additional resources

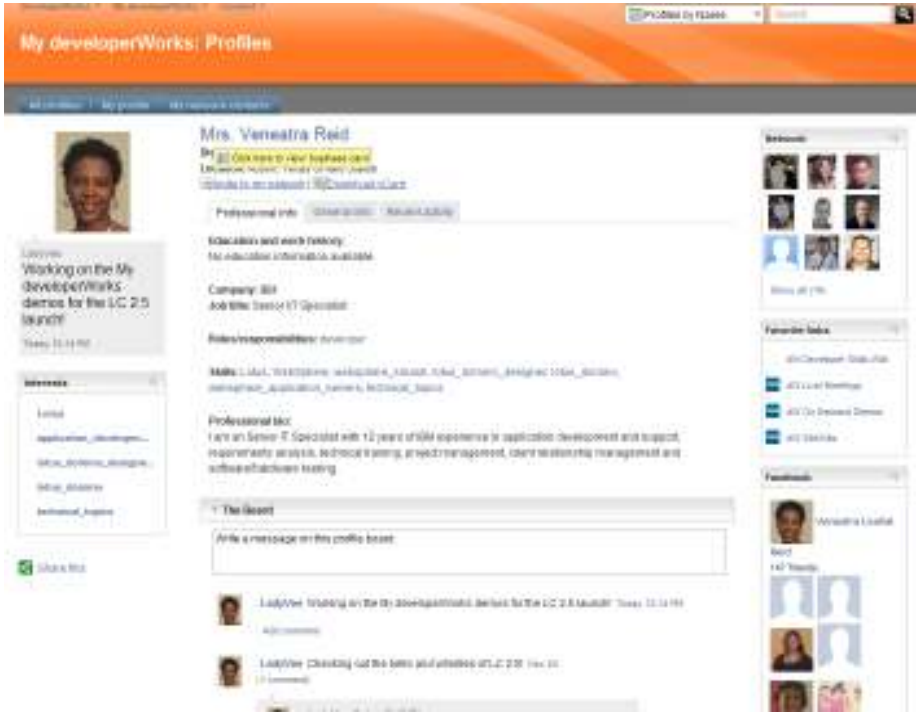
- Find out more about Rational Quality Manager
  - http://www.ibm.com/software/awdtools/rqm/
- Download the Rational Quality Manager Trial
  - Rational Quality Manager Trial on DeveloperWorks
- Learning resources - Webcasts/Telcons/Podcasts.
  - Ensuring Lifecycle Quality through RQM integration capabilities
  - Reduce the Cost of Quality Solution Sheet
  - Quality Driven Software Delivery
- Blog with us
  - Rational Quality Manager Blog
  - Rational Tester Blog
- Facebook - Rational Quality Manager
- Twitter - The Rational Tester
- Videos and quick demos (IBM TV, YouTube)
  - Rational Quality Manager in Three Minutes
  - IBM Rational Collaborative ALM Demo
  - Rational Quality Manager Common Reporting

My developerWorks: “The Geekiest Social Network”

Join My developerWorks today

Continue the conversation with the speaker, classmates, and a network of 8 million IT professionals around the world.

- ✓ **Build relationships** with technical professionals who have similar interests.
- ✓ **Collaborate** to find ideal solutions to your tough technical questions.
- ✓ **Learn about additional resources** to deepen your skills.
- ✓ Already an expert? **Have your voice heard!**



ibm.com/developerworks/mydeveloperworks

Ready to Get Started?

Join the [Customize with your group name] group today!

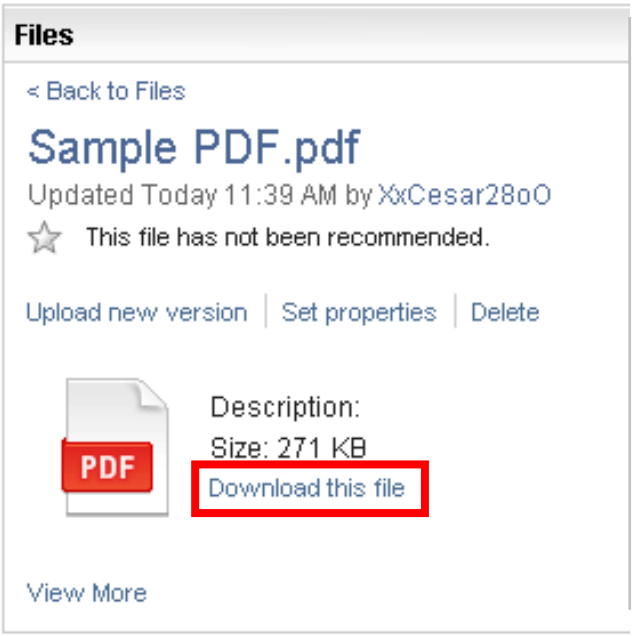
Search for the group name under my profile [Your name]

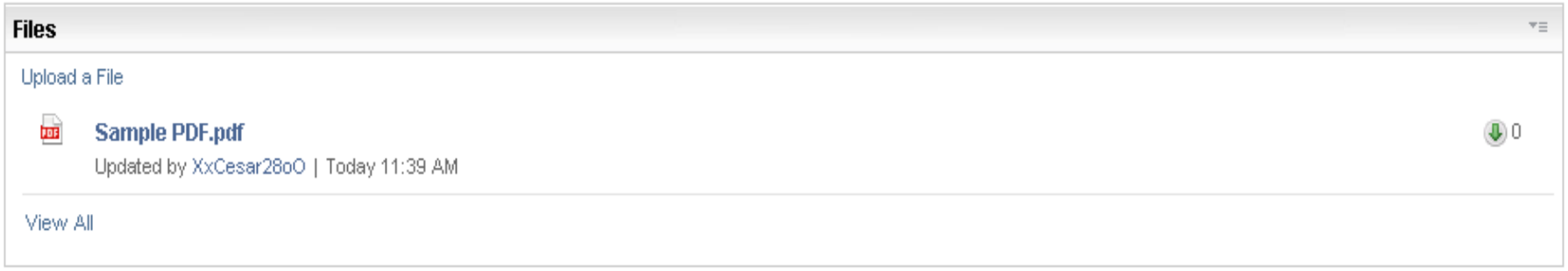
Access PDF files and other briefing information

Download files from a group in My developerWorks.

Go to the group's main page and look for the "Files" widget (Below).

Click on the file to download it.





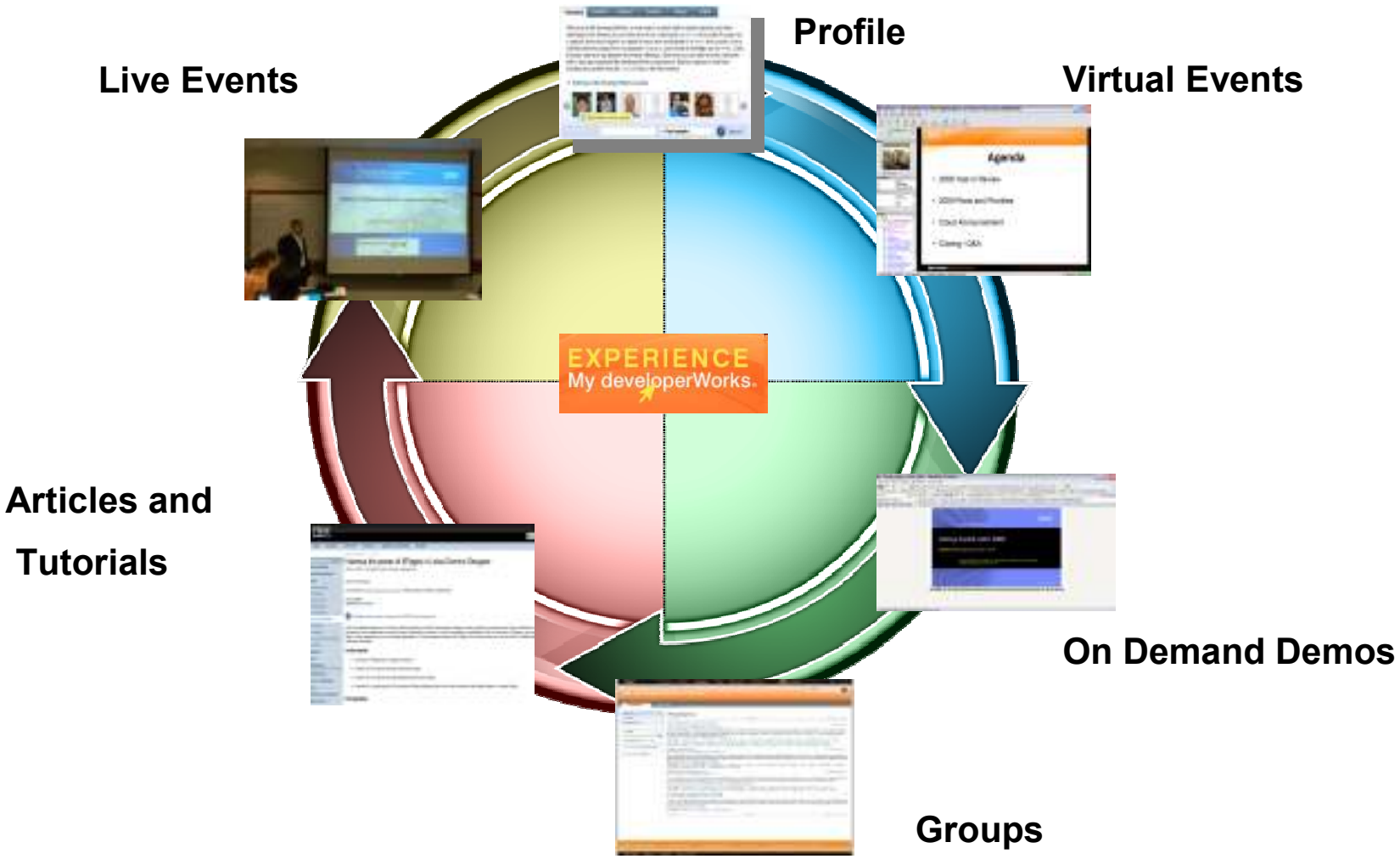
ibm.com/developerworks/mydeveloperworks

Get the detailed scoop on My developerWorks here:

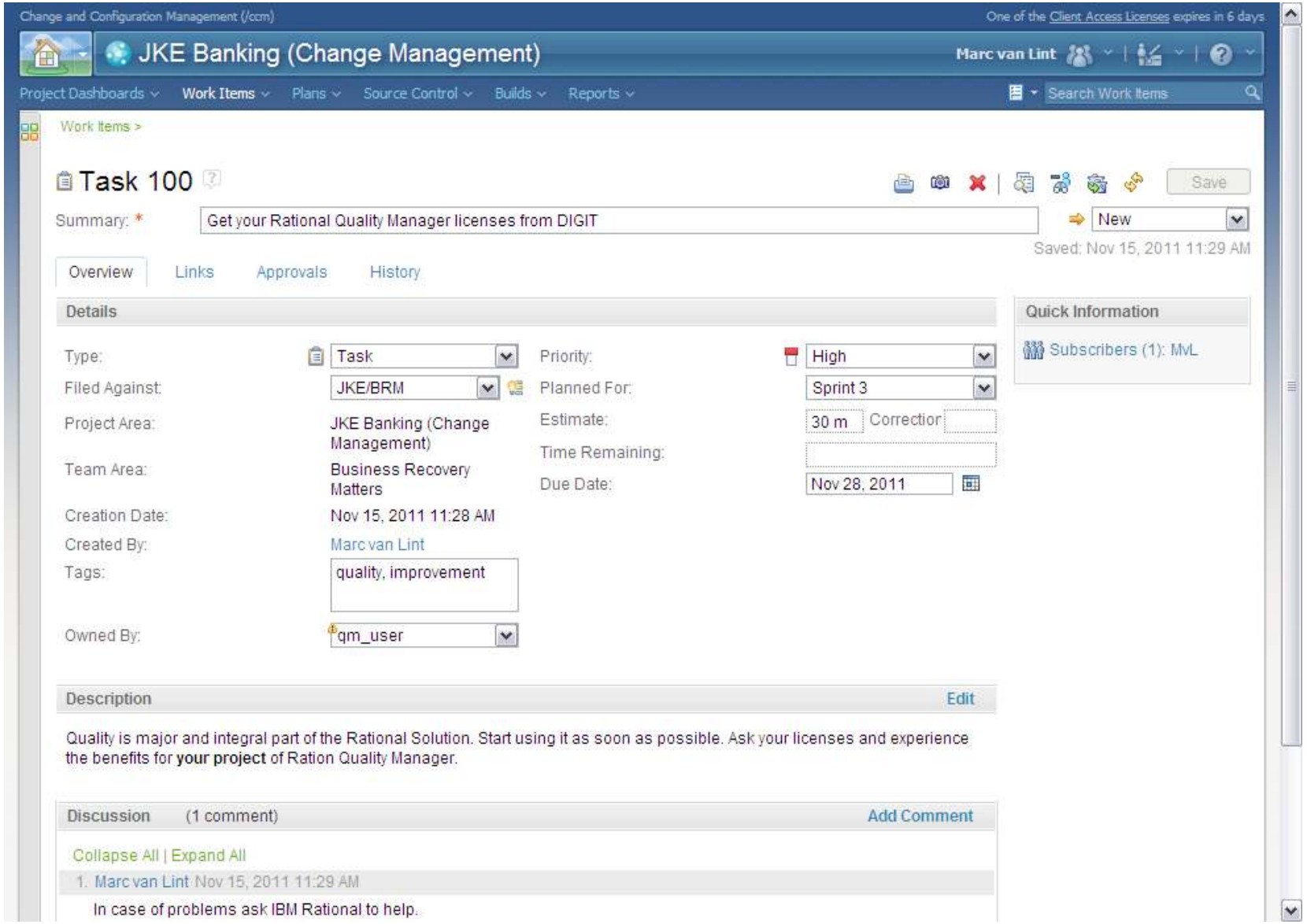
http://www.ibm.com/developerworks/web/library/j-mydeveloperworks-intro/index.html

ibm.com/developerworks

Continue the Conversation



HIGH PRIO WORK ITEM!



www.ibm/software/rational

BACKUP