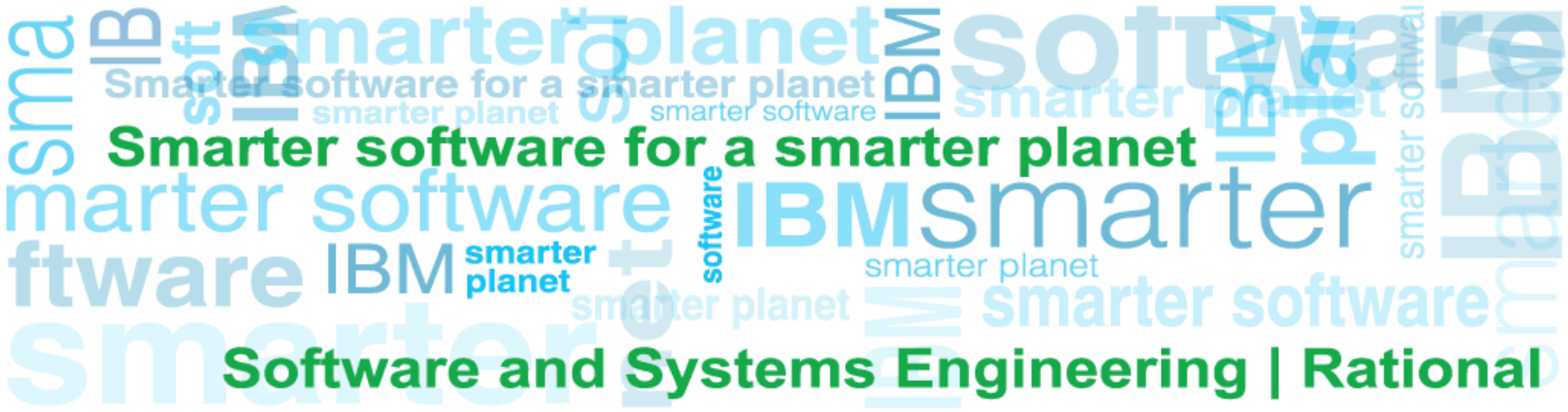
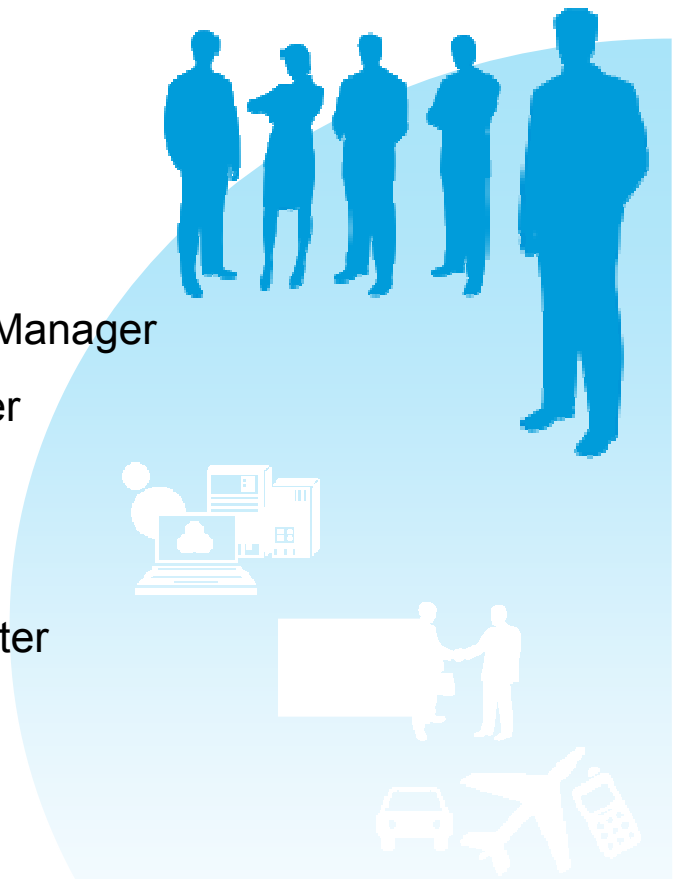
Testing and Quality

Key highlights of a broad expanding portfolio



© 2011 IBM Corporation

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

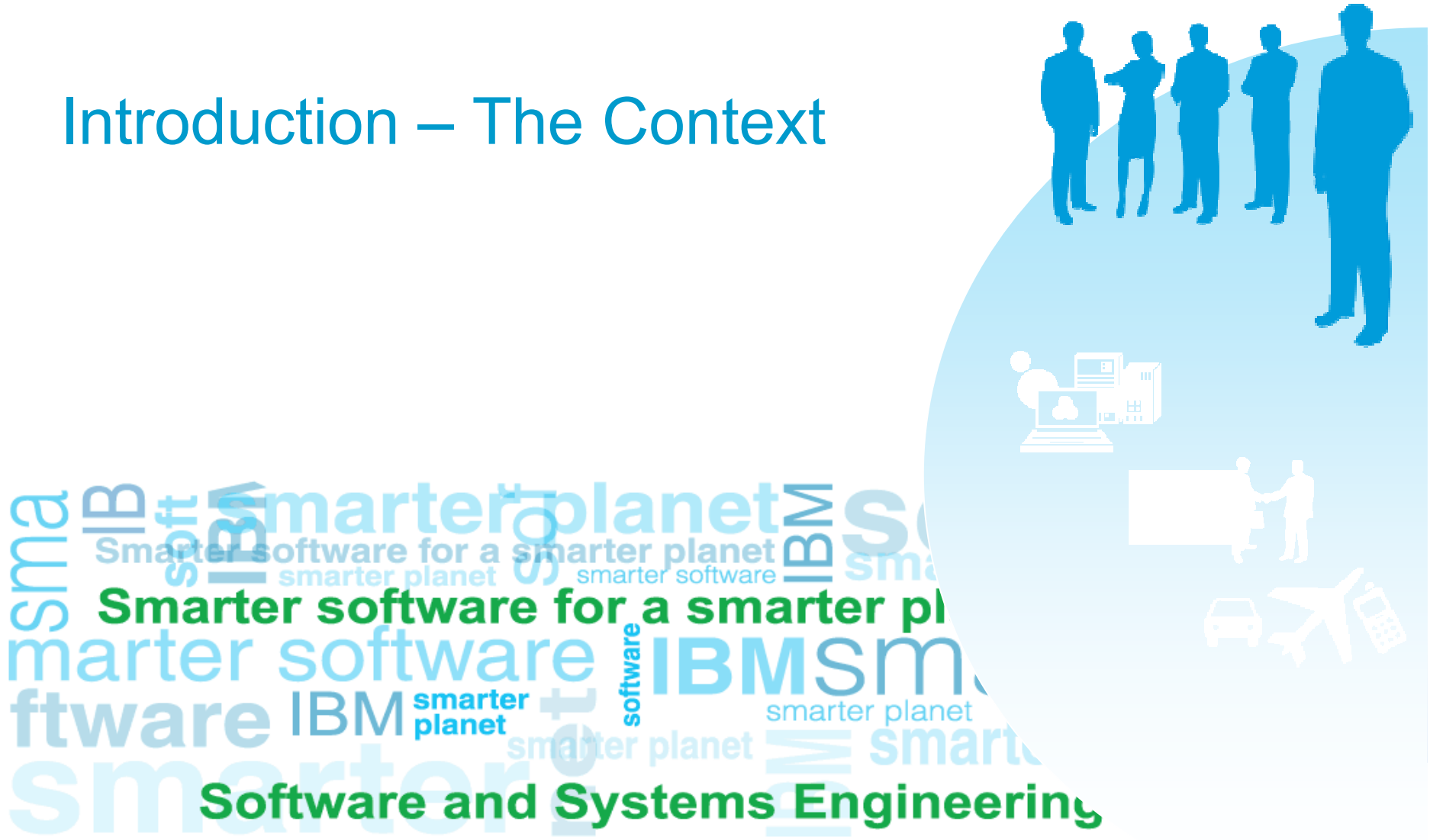
© 2011 IBM Corporation

Tester's dream…

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

© 2011 IBM Corporation

Introduction – The Context



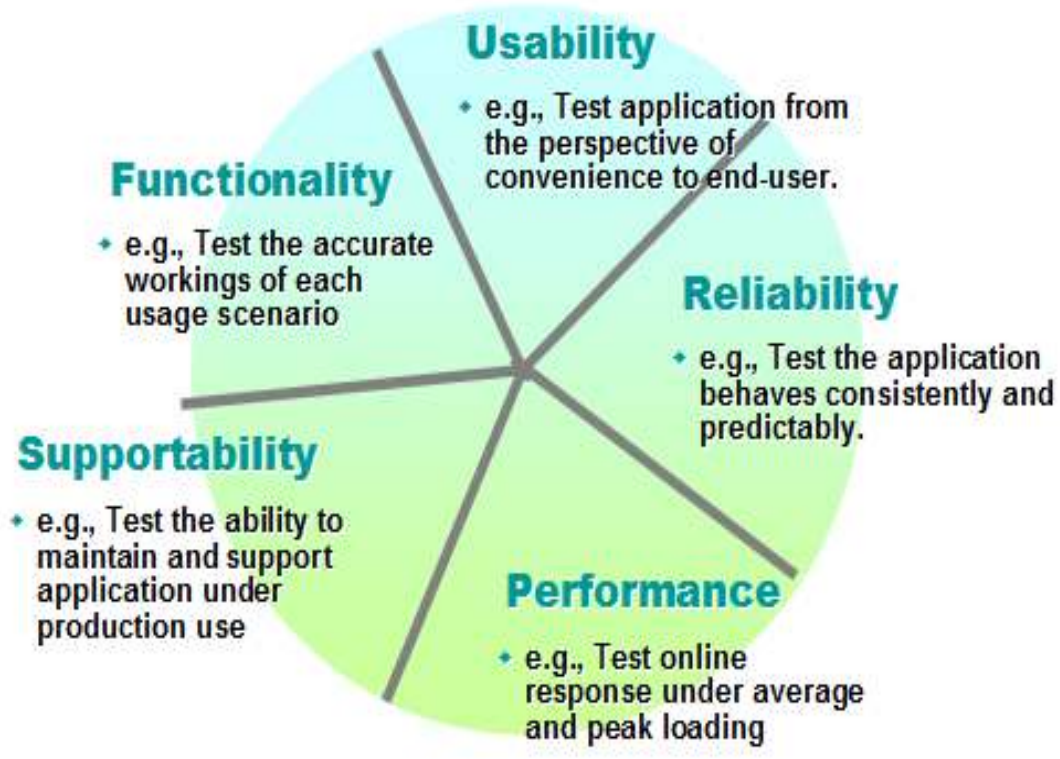
© 2011 IBM Corporation

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

© 2011 IBM Corporation

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

© 2011 IBM Corporation

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.

© 2011 IBM Corporation

Where to start?

© 2011 IBM Corporation

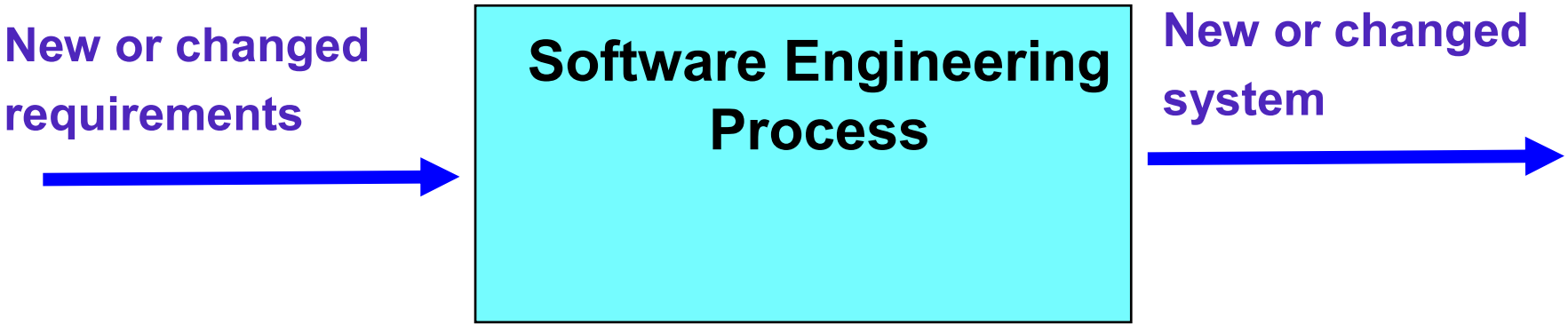
Tools, Process, or Something Else?

- 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

© 2011 IBM Corporation

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

© 2011 IBM Corporation

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

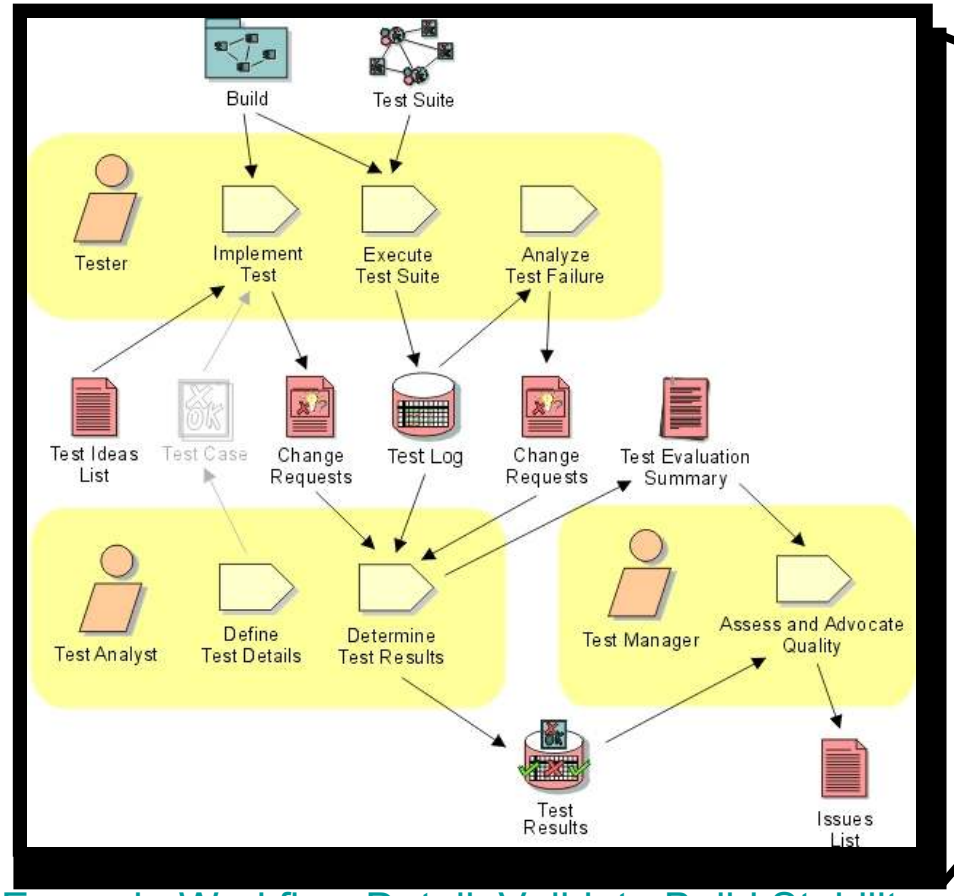
Example Practices (Teams practice…):
•Iterative Development
•Test-Driven Development
•Continuous Integration
•Requirements Management

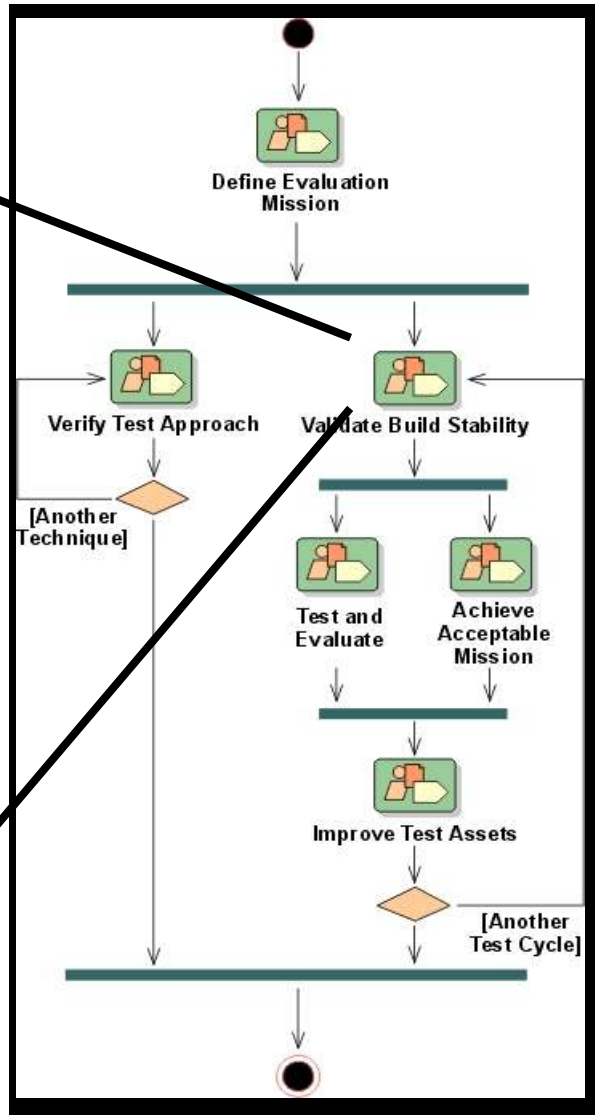
Results:

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

© 2011 IBM Corporation

Testing Workflow from Rational Unified Process





Example Workflow Detail: Validate Build Stability

© 2011 IBM Corporation

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?

© 2011 IBM Corporation

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"

© 2011 IBM Corporation

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

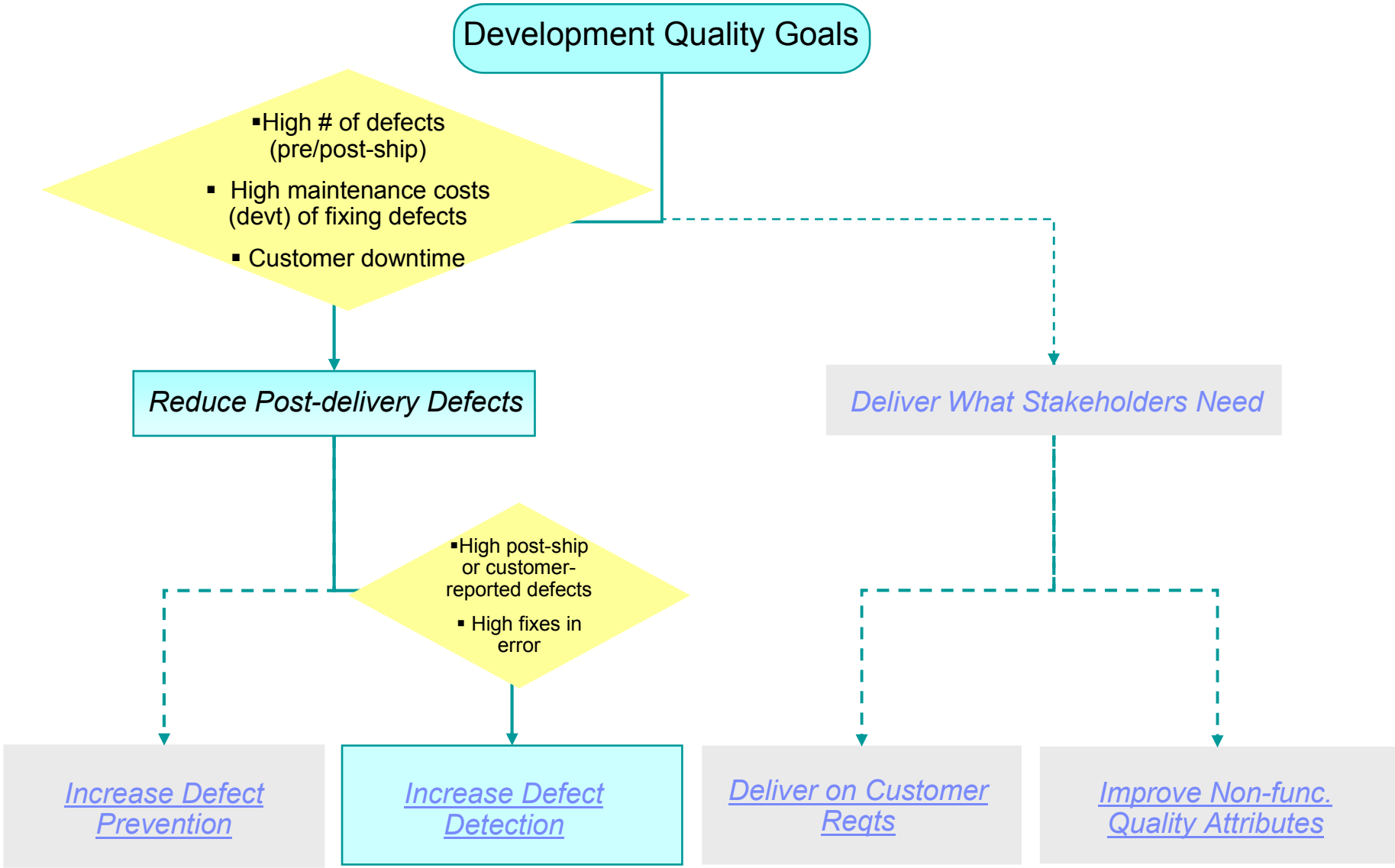
Improve Non-functional Quality Attributes

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

© 2011 IBM Corporation

Quality Traceability Tree: Decision Criteria



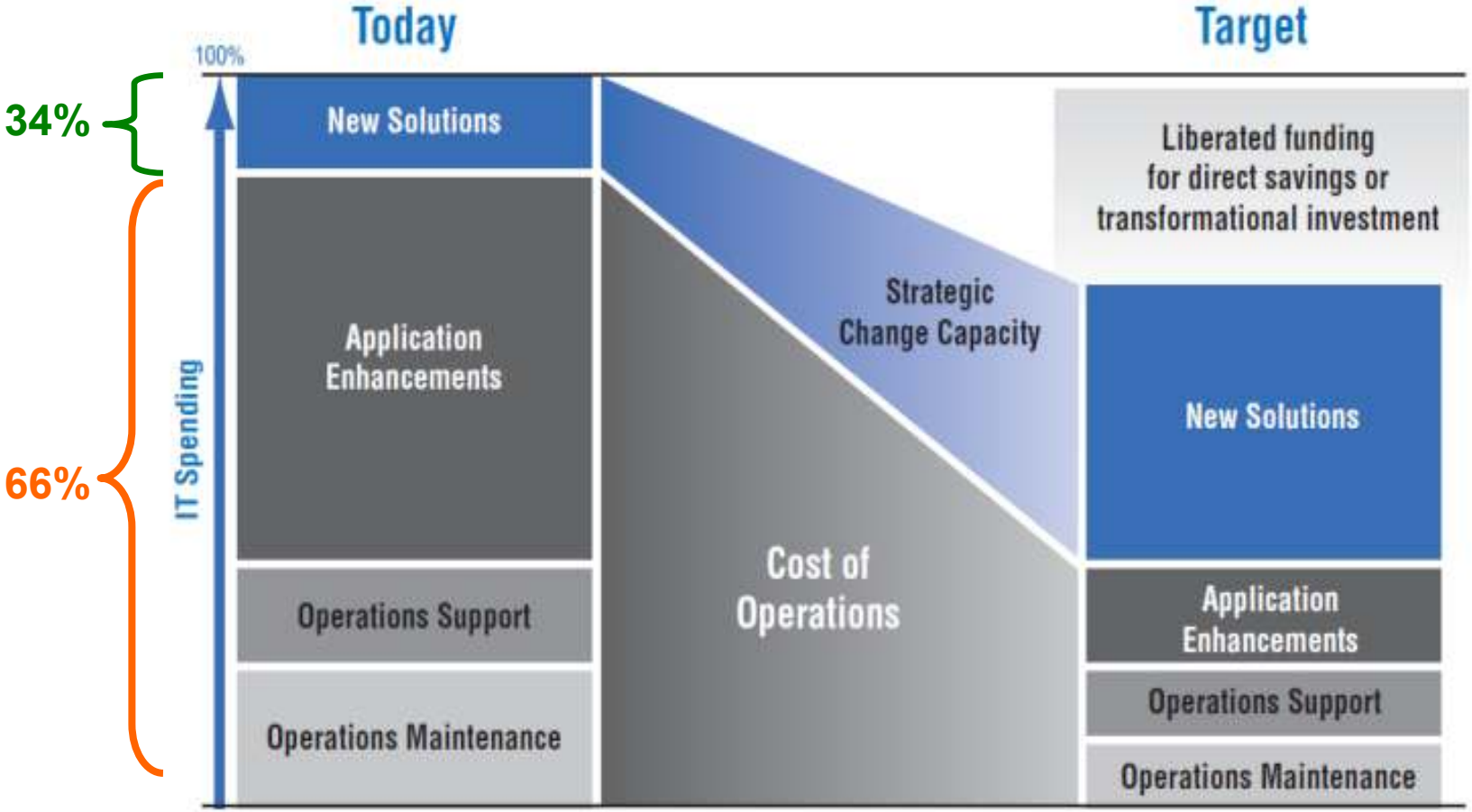
© 2011 IBM Corporation

IBM Rational Quality Management Strategy

© 2011 IBM Corporation

Business and IT Agility: Balancing Resources to Support Business Innovation

Balance IT investments to focus on new solutions.



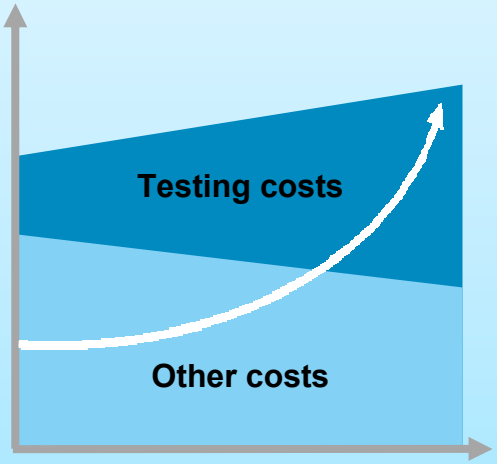
- Forrester estimates that ongoing operations and maintenance consume ***66%*** of IT budgets
- While new projects and software initiatives represent only ***34%***

© 2011 IBM Corporation

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

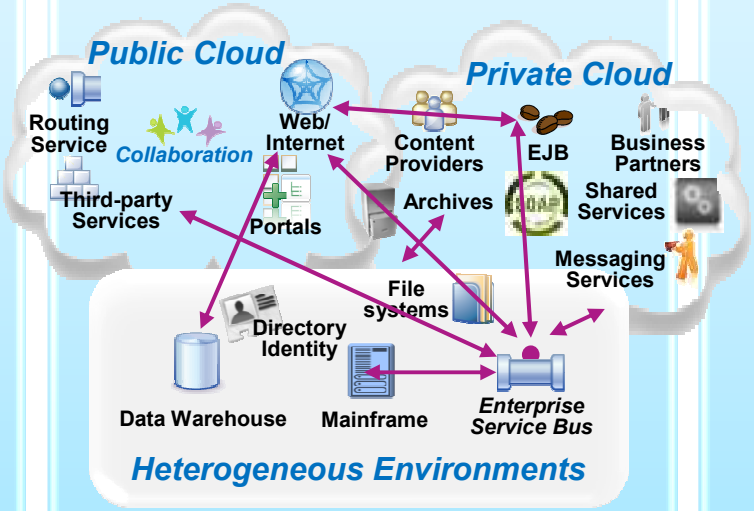
An estimated 60 - 80 percent of the cost of software development is in rework*

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

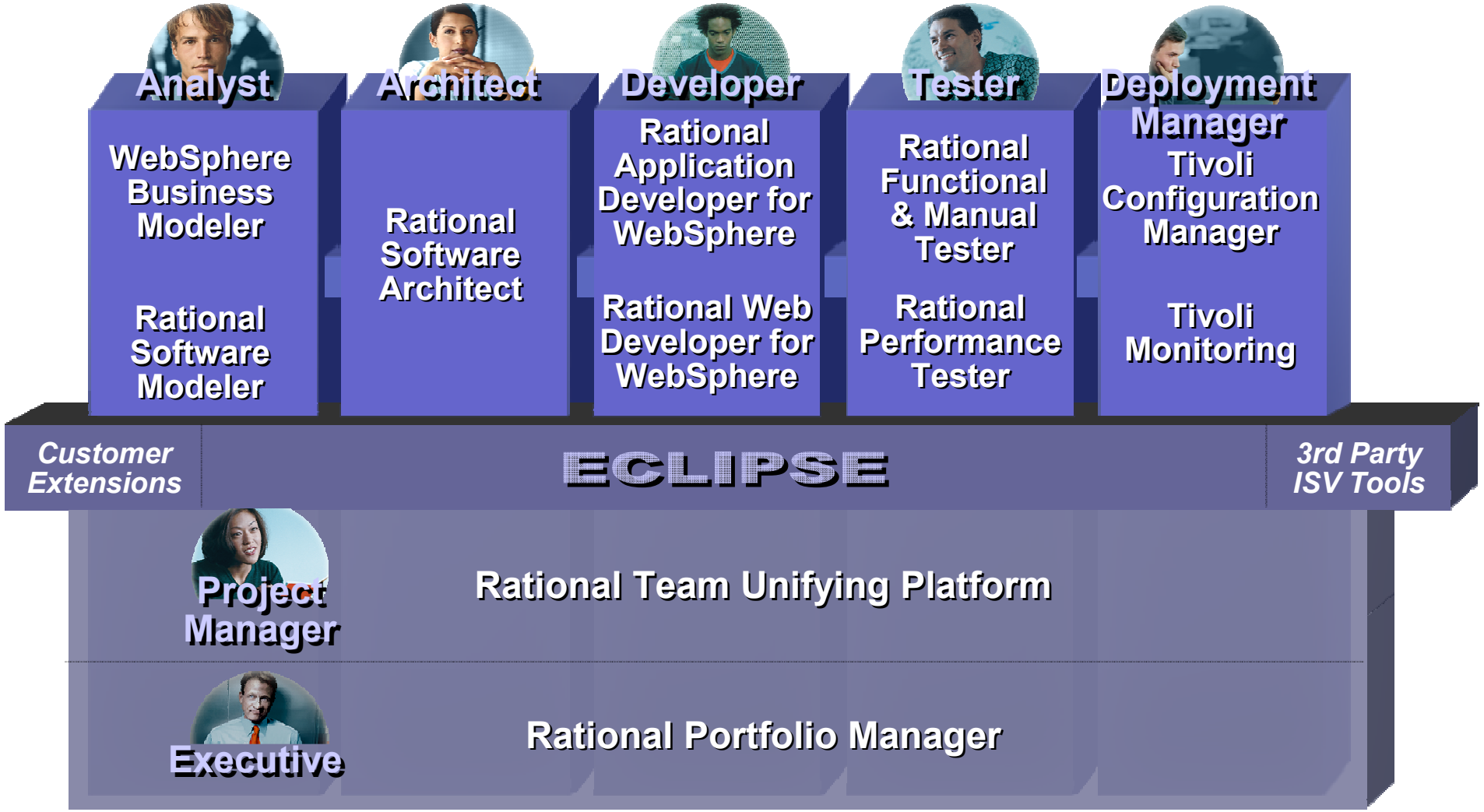
Balancing Quality and Speed

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

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

© 2011 IBM Corporation

We are moving From …



© 2011 IBM Corporation

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.

© 2011 IBM Corporation

…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

© 2011 IBM Corporation

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

Integrate

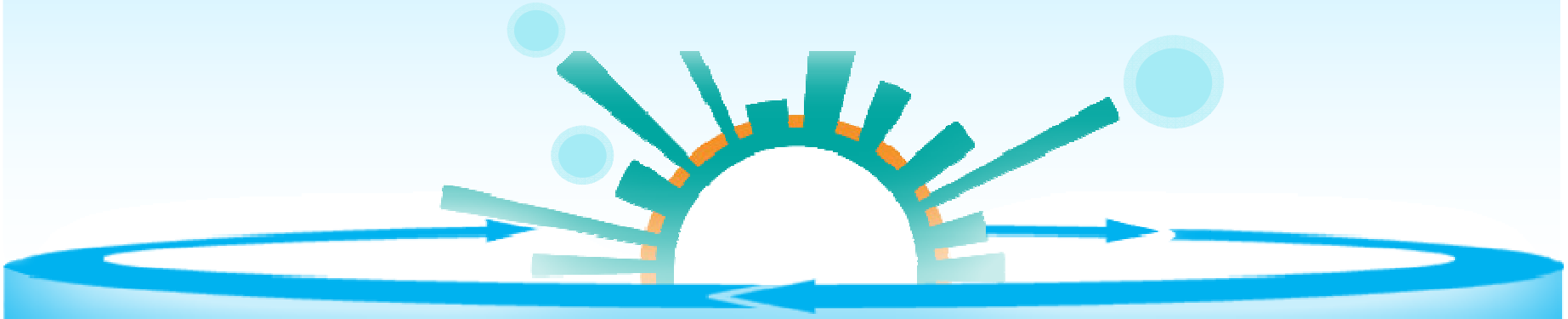
Connecting process and information
Software, data, and tools

Collaborate

Unifying teams
Projects, and organizational cultures

Optimize

Simplifying governance
Plans, scope, and measures



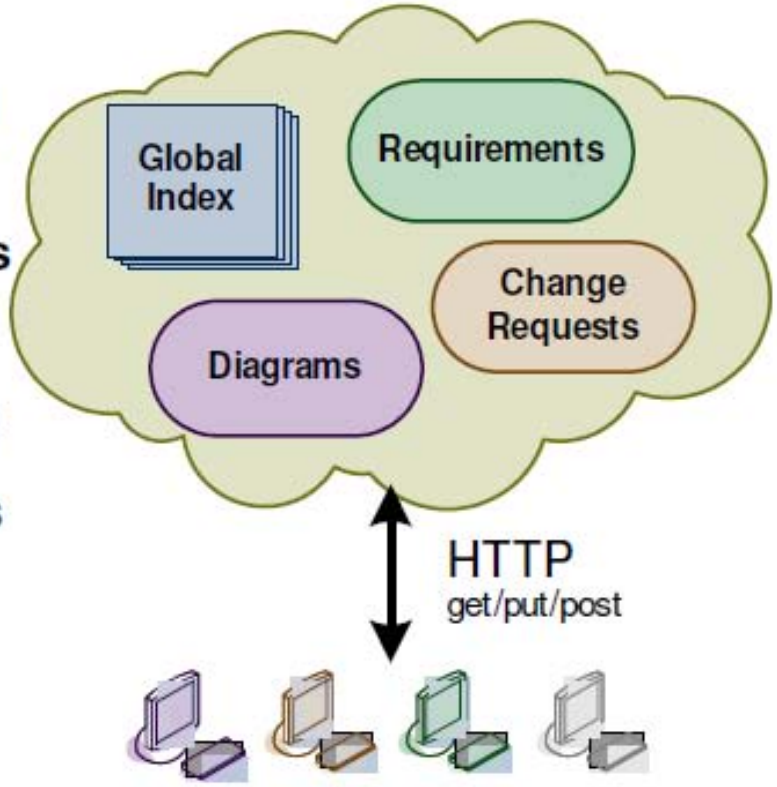
Realized benefits:

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

© 2011 IBM Corporation

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)

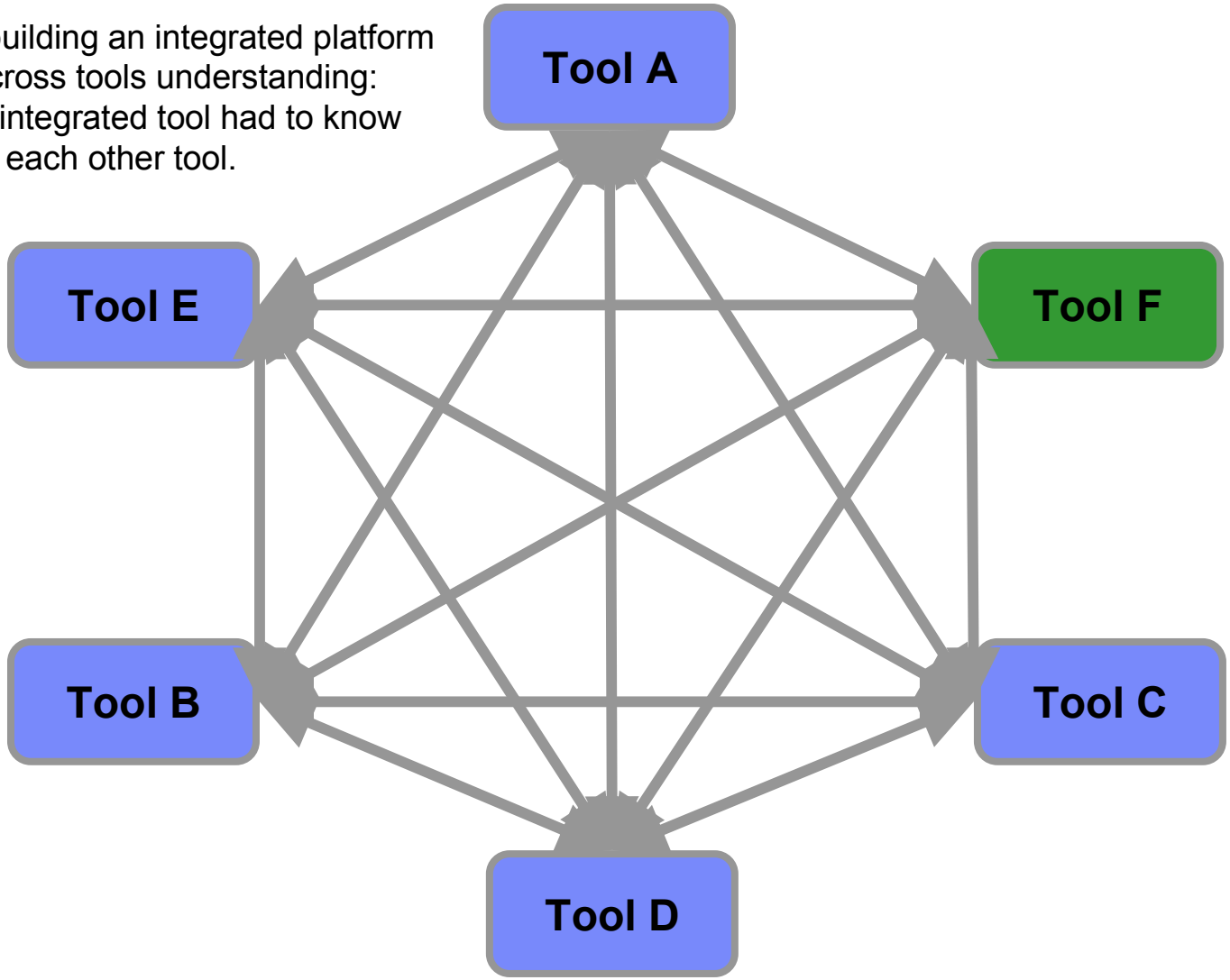


© 2011 IBM Corporation

Tools collaboration generally means exponential complexity!

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

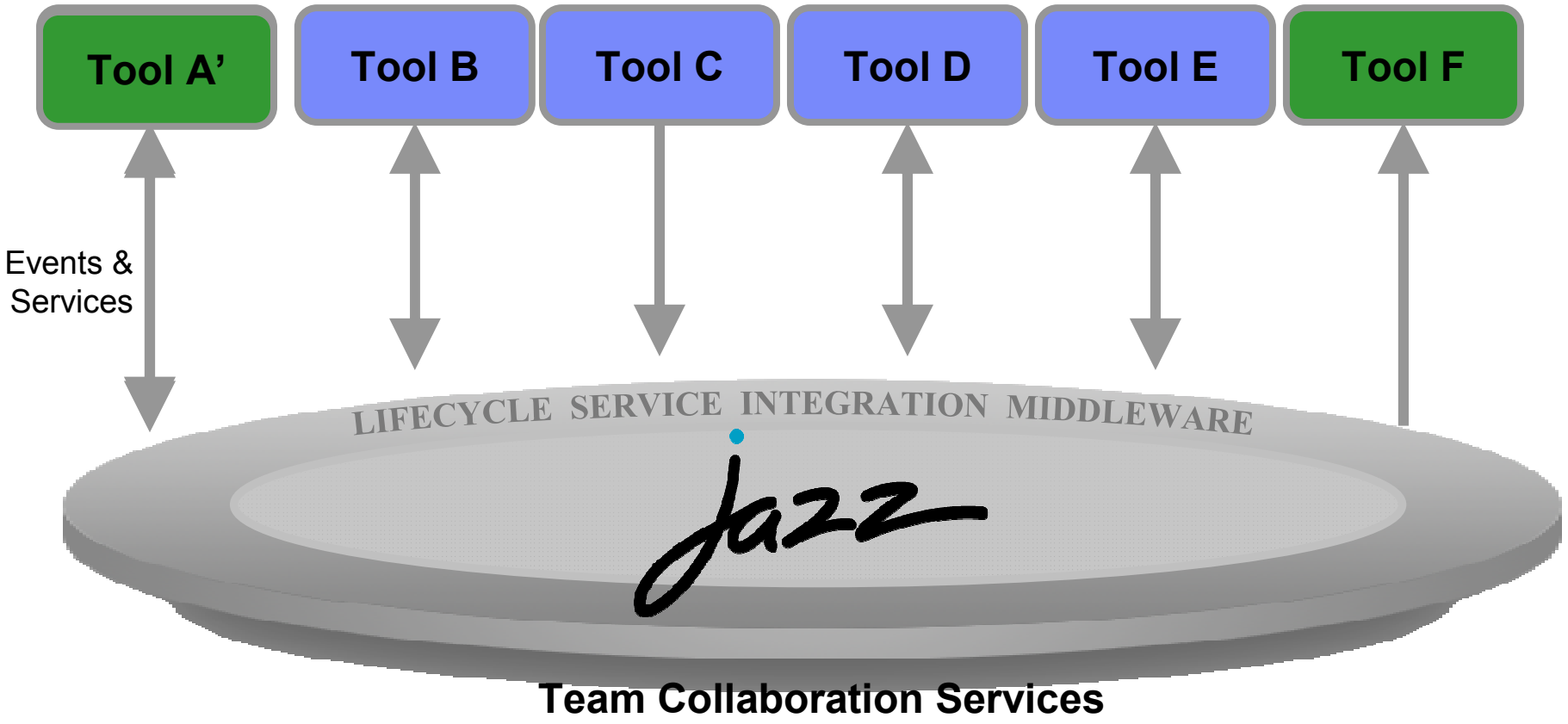
Each integrated tool had to know about each other tool.



© 2011 IBM Corporation

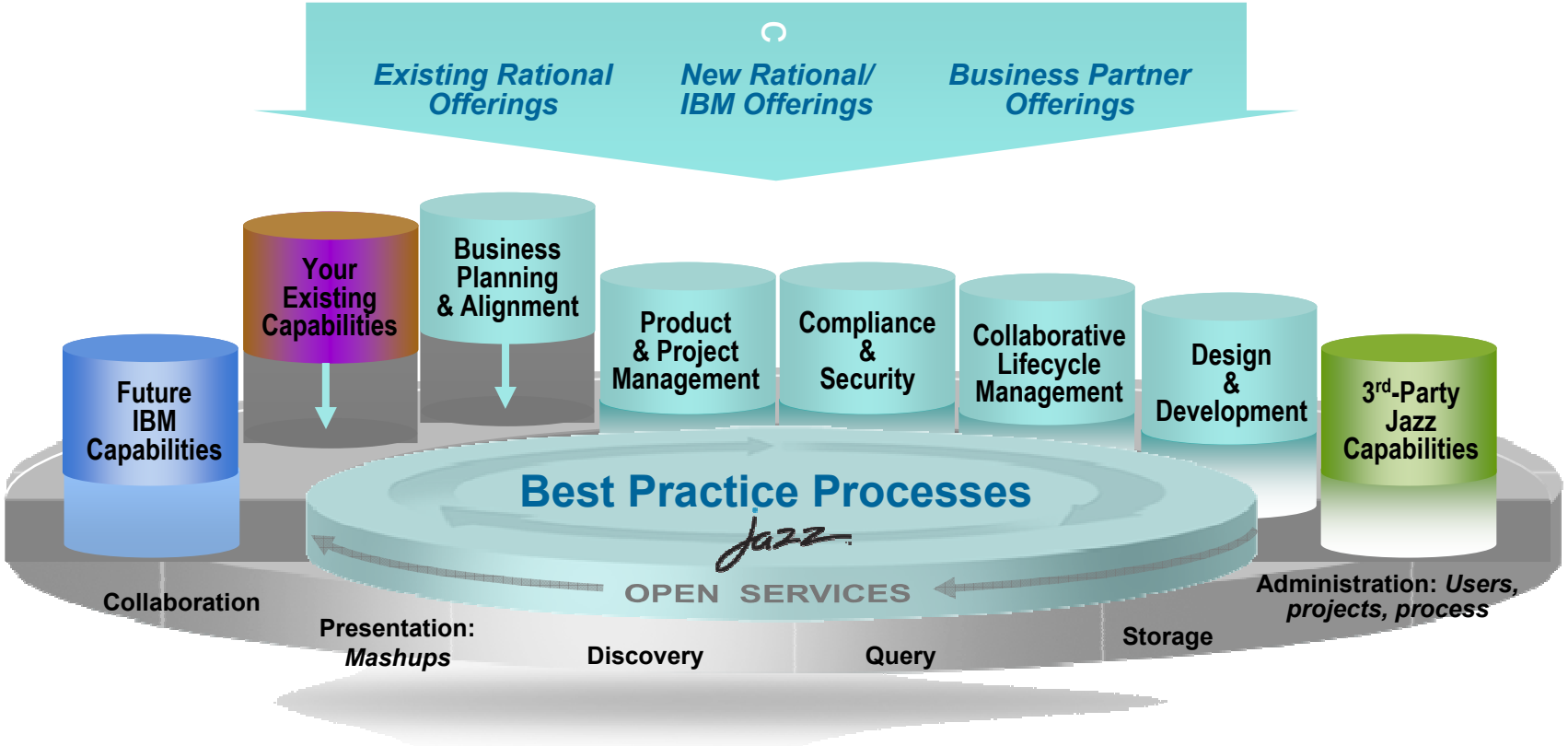
Tools collaboration based on middleware services

- 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



© 2011 IBM Corporation

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

© 2011 IBM Corporation

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

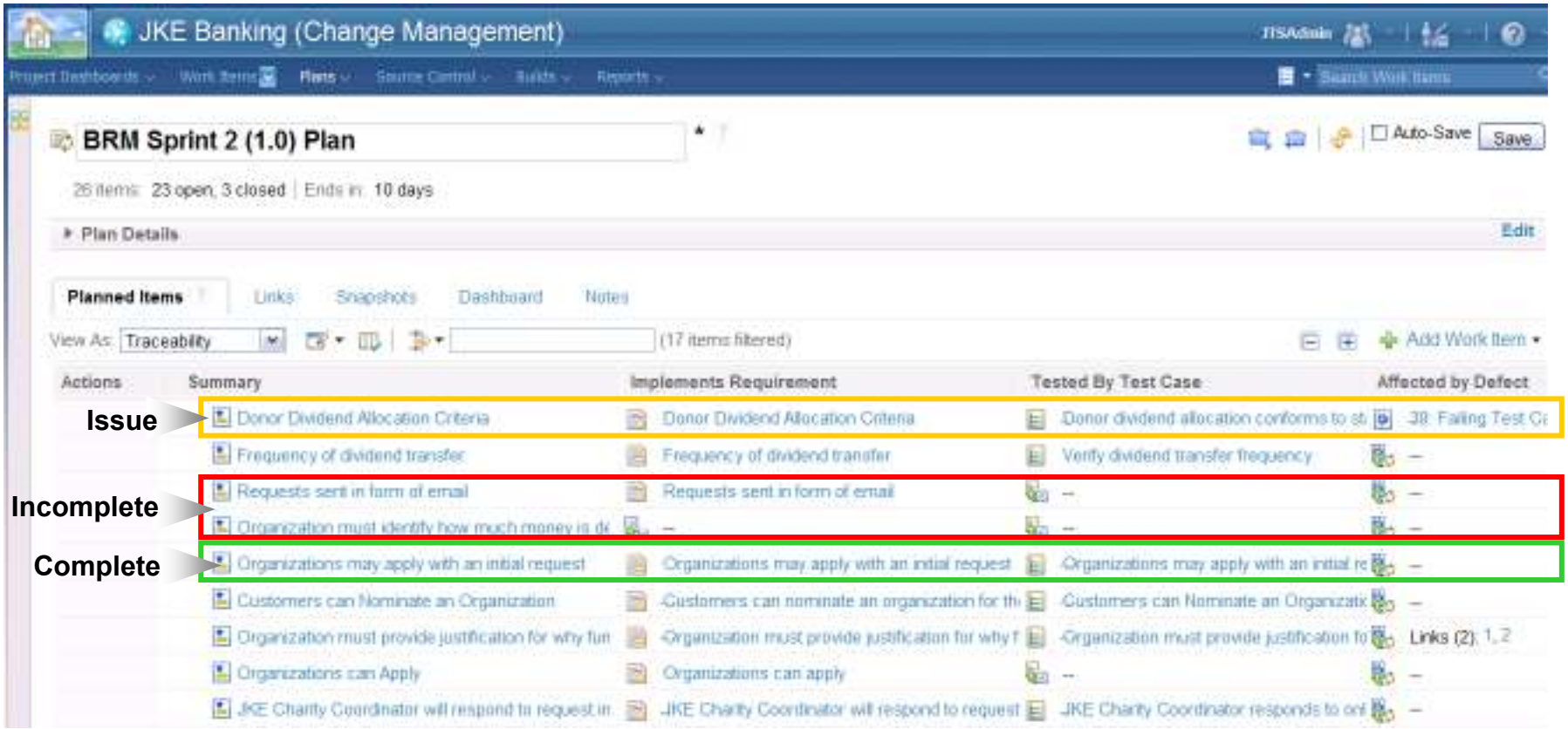
Rational Requirements Composer	Rational Team Concert	Rational Quality Manager
Requirements Management	Planning, Change, Configuration & Build Management	Quality Management

*extensions

© 2011 IBM Corporation

Team leads improve release quality & predictability

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



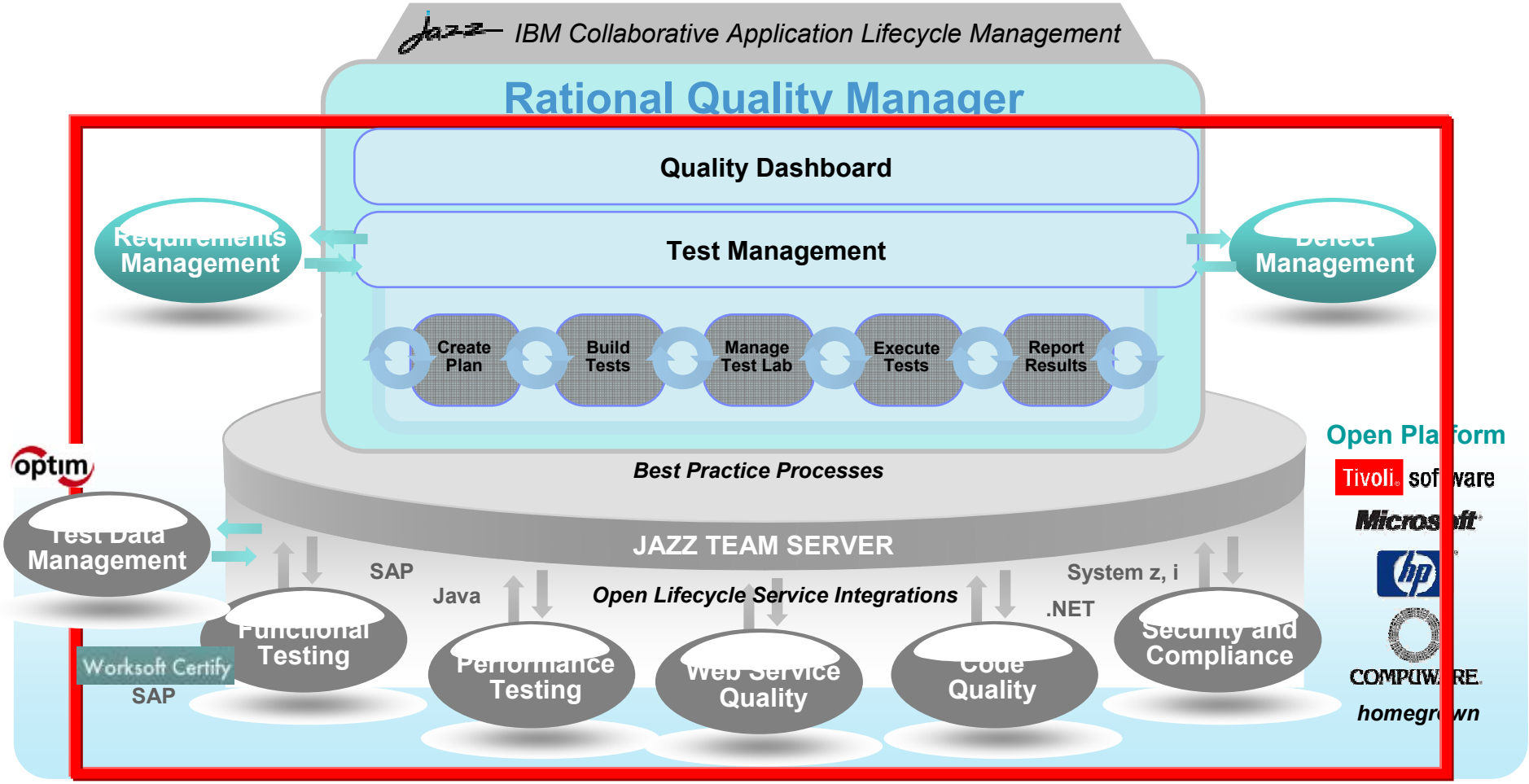
© 2011 IBM Corporation

Enterprise Quality Management
Rational Quality Manager

© 2011 IBM Corporation

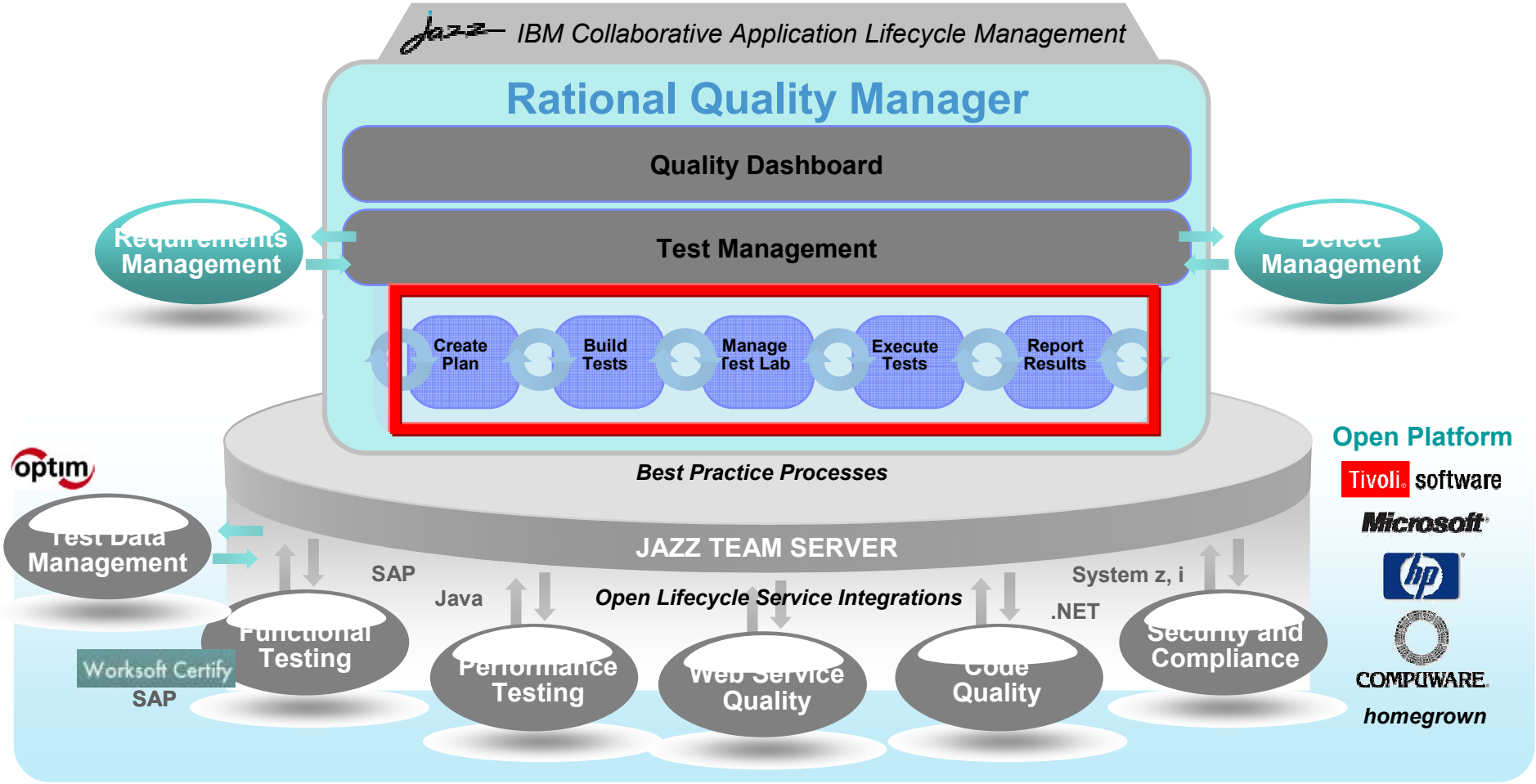
Today's High Level QM Segment Architecture

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



© 2011 IBM Corporation

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



© 2011 IBM Corporation

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

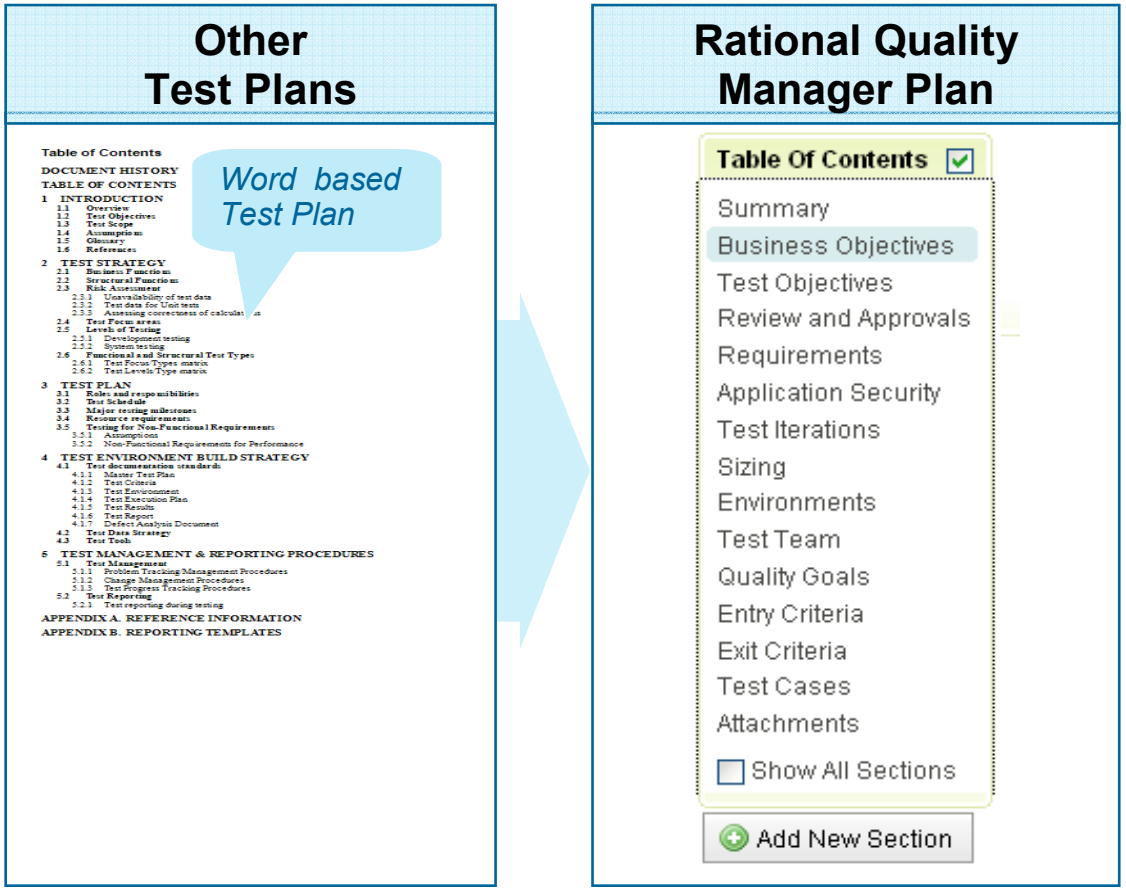
Make confident decisions with effortless reporting

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

© 2011 IBM Corporation

Comprehensive dynamic planning and updates

Process flow, not artifacts drives team activities

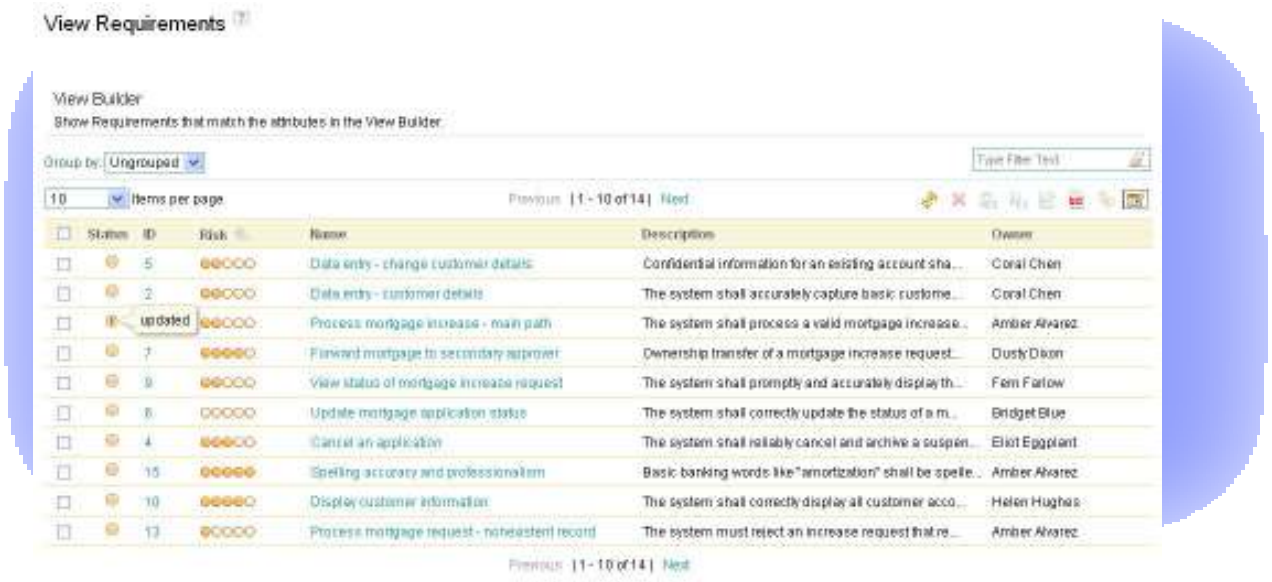


- Live dynamic documentation
- Defines test process and strategy
- Defines responsibilities
- Activity based versus hierarchy
- Business level reporting against quality objectives

© 2011 IBM Corporation

Requirements driven testing

Knowing what to test



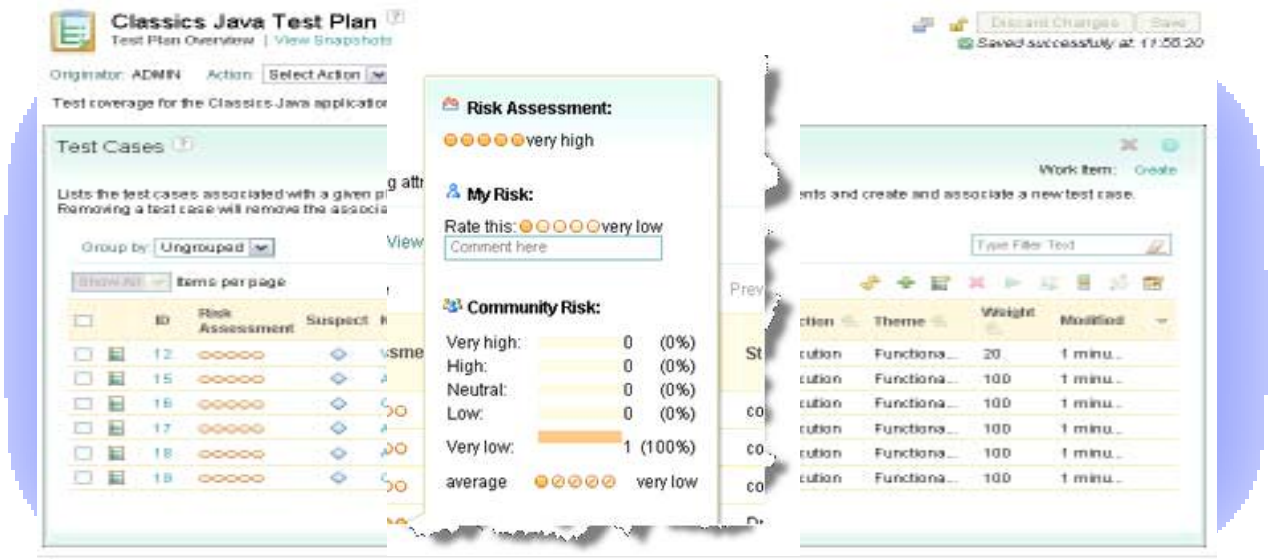
- **Requirements tracking built into the test management tooling**
- **Customizable attributes enable you to track what is important to your team**
- **Real-time impact analysis of requirements changes**
- **Traceability of test results to user needs**

Know you are testing the right things

© 2011 IBM Corporation

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

© 2011 IBM Corporation

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

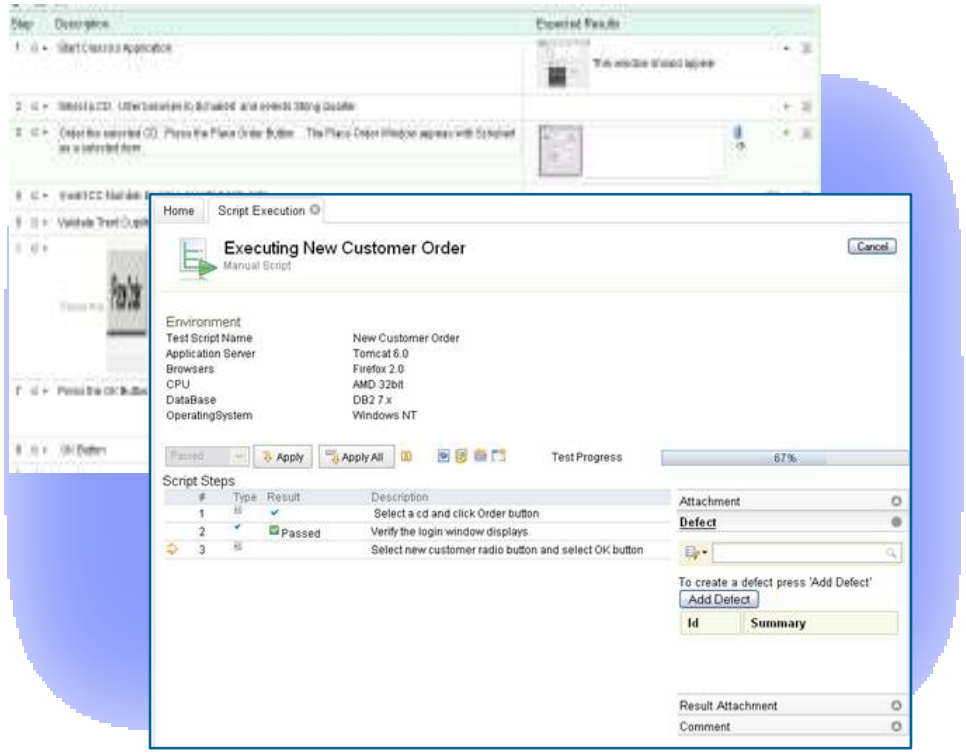
Test Platform Environment Management

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

Test the right cases instead of everything
Plan optimal execution

© 2011 IBM Corporation

Integrated Manual test authoring and execution



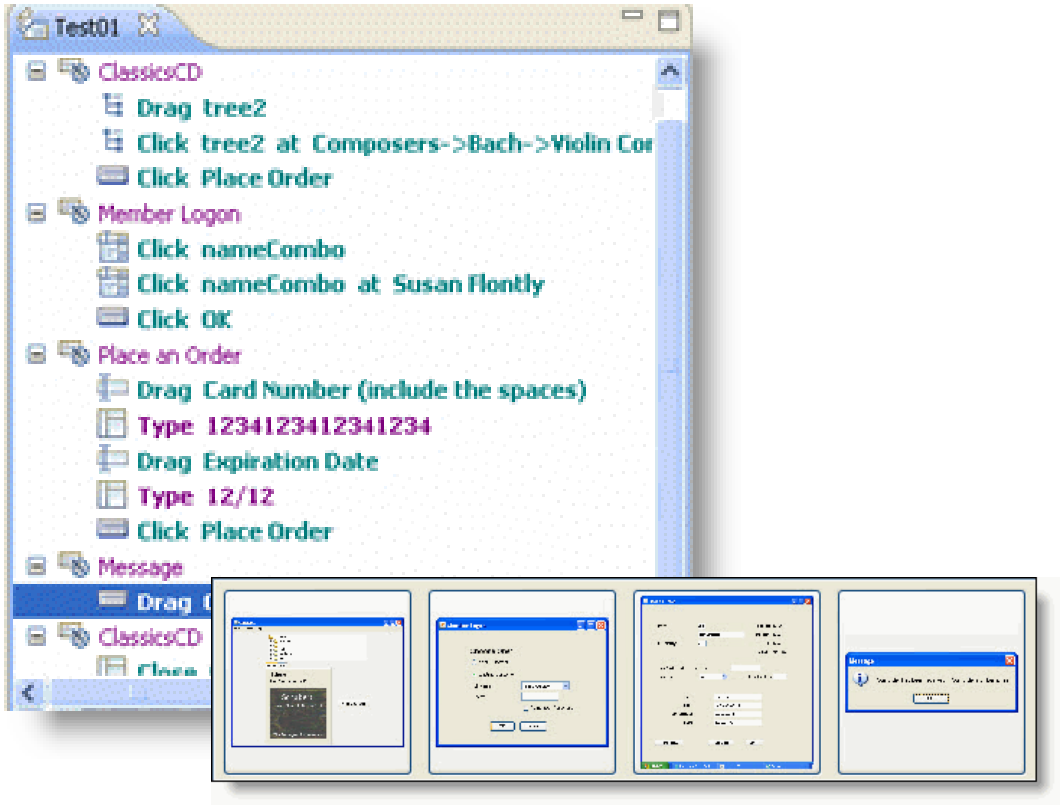
- Manual test author and execute
  - Step by step capture and execution of manual tests
  - Assisted data entry
  - Keyword support for integrated manual and automated testing
  - Rich defect capture during execution, including screenshot and attachments
  - Simple intuitive interface for quick test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

Maximizing efficiency of manual testing

© 2011 IBM Corporation

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



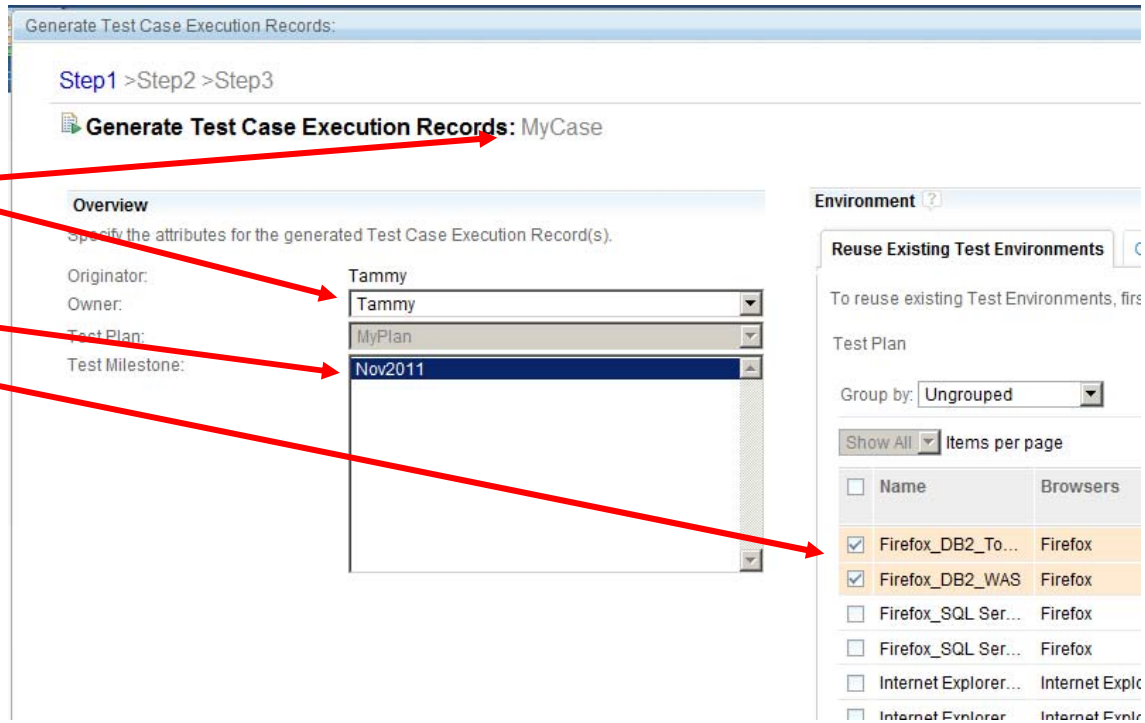
Accelerate test execution and deepen test coverage through automated test execution

© 2011 IBM Corporation

My Favorite: Test Execution Record

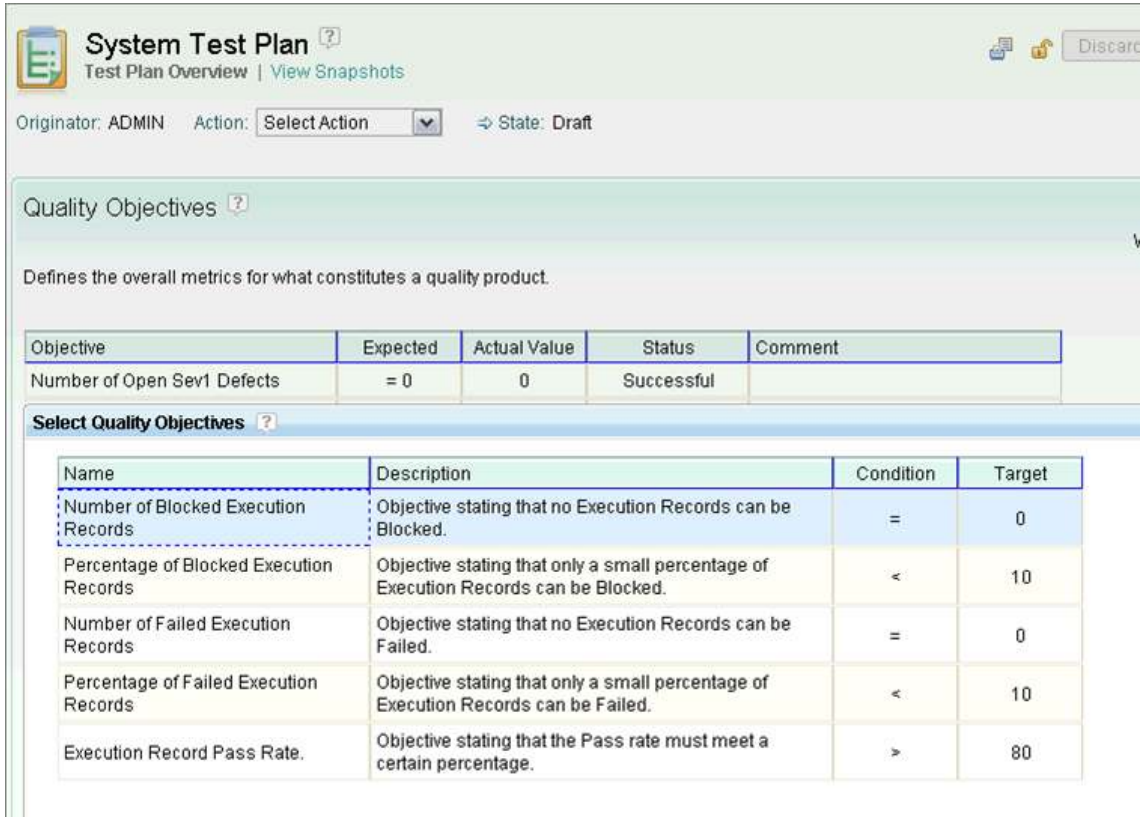
‘Test order’

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



© 2011 IBM Corporation

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

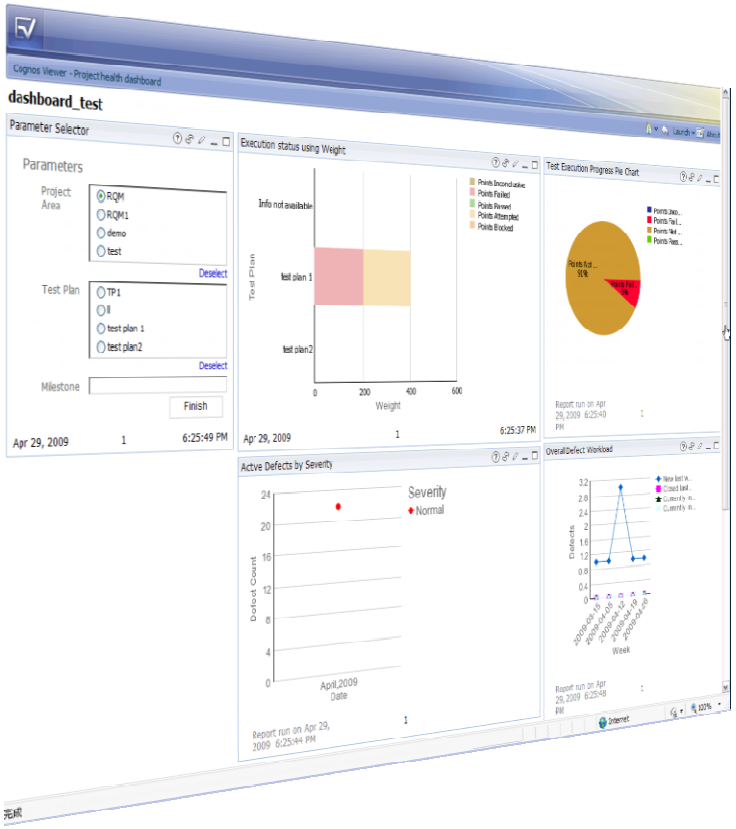
- Assessing status
  - Standard Objectives
  - Reuse across Test Plans
  - All working toward same objectives
  - Measures against business objectives

Drive continuous and measured improvement

© 2011 IBM Corporation

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

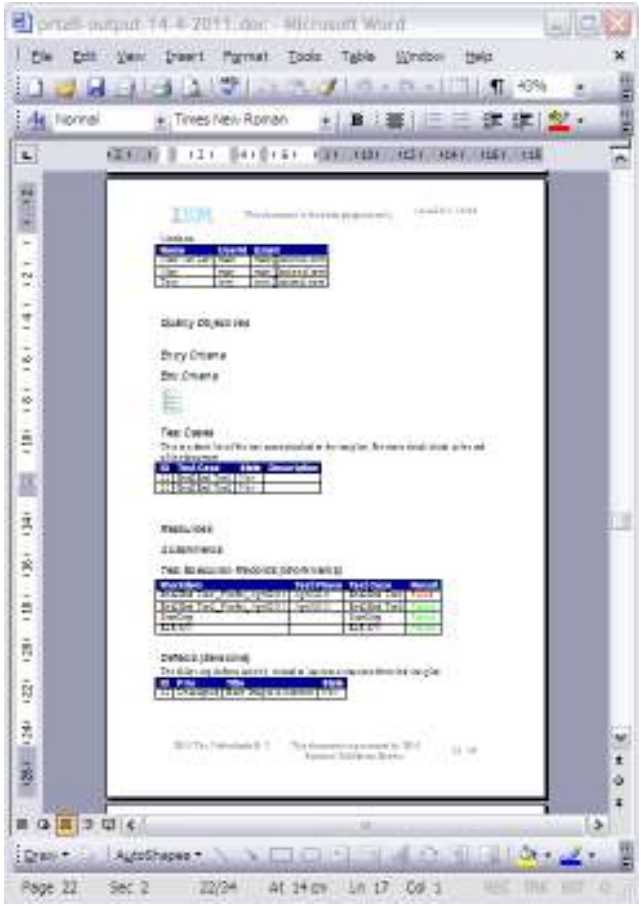
© 2011 IBM Corporation

Reporting with IBM Rational

© 2011 IBM Corporation

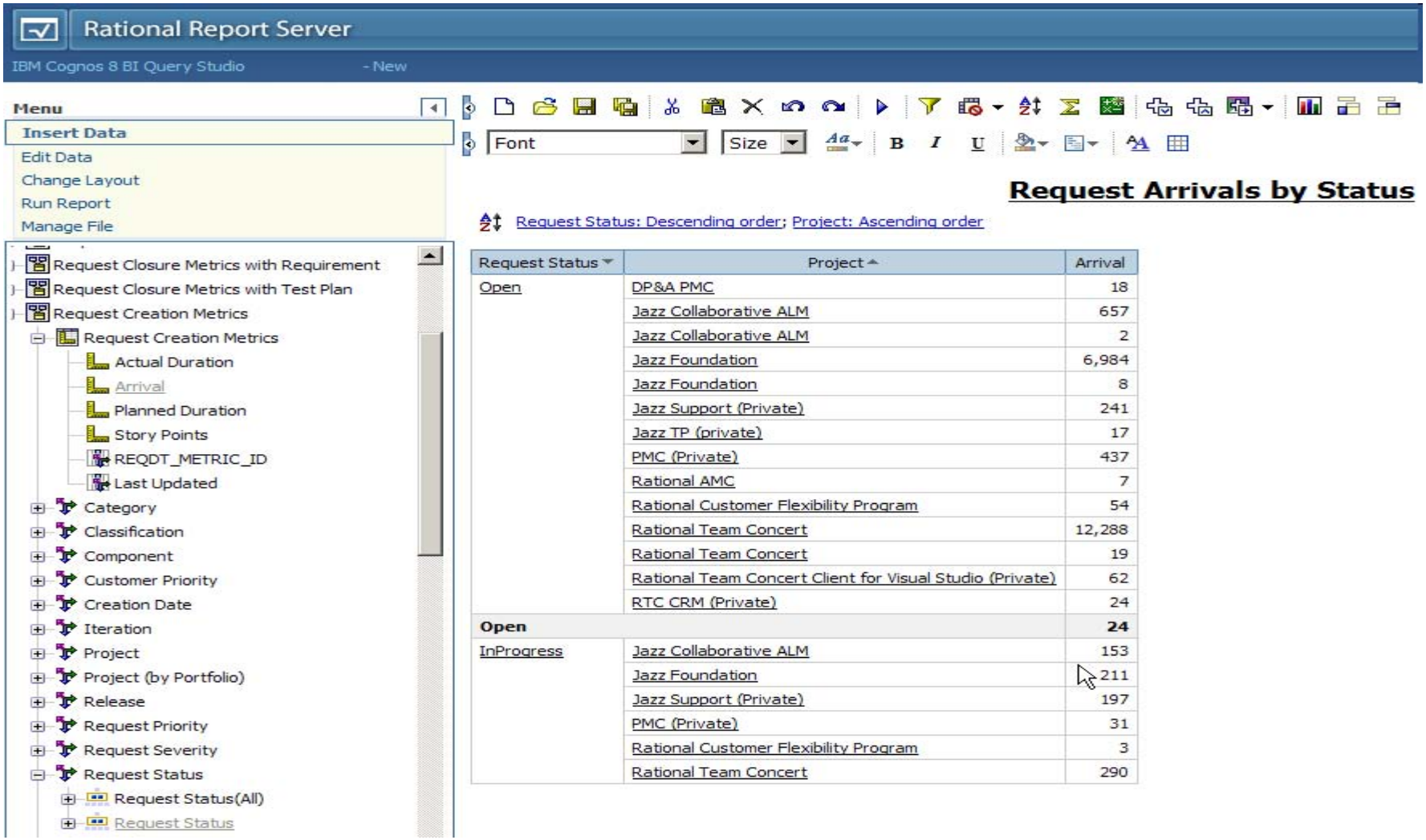
Generating Documents – Rational Publishing Engine

- 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



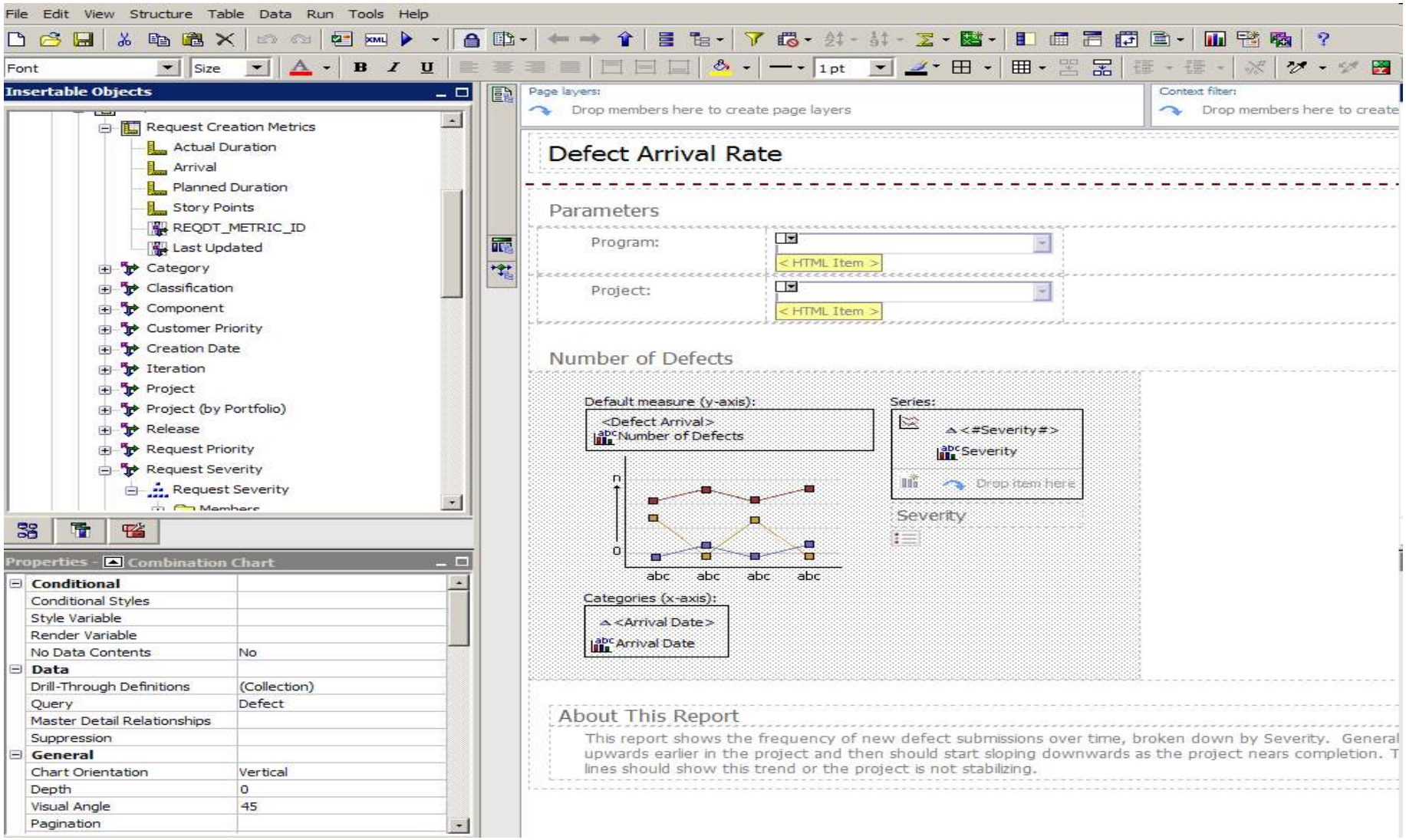
© 2011 IBM Corporation

Rational Reporting for Developer Intelligence – Query Studio



© 2011 IBM Corporation

Rational Reporting for Developer Intelligence - Report Studio

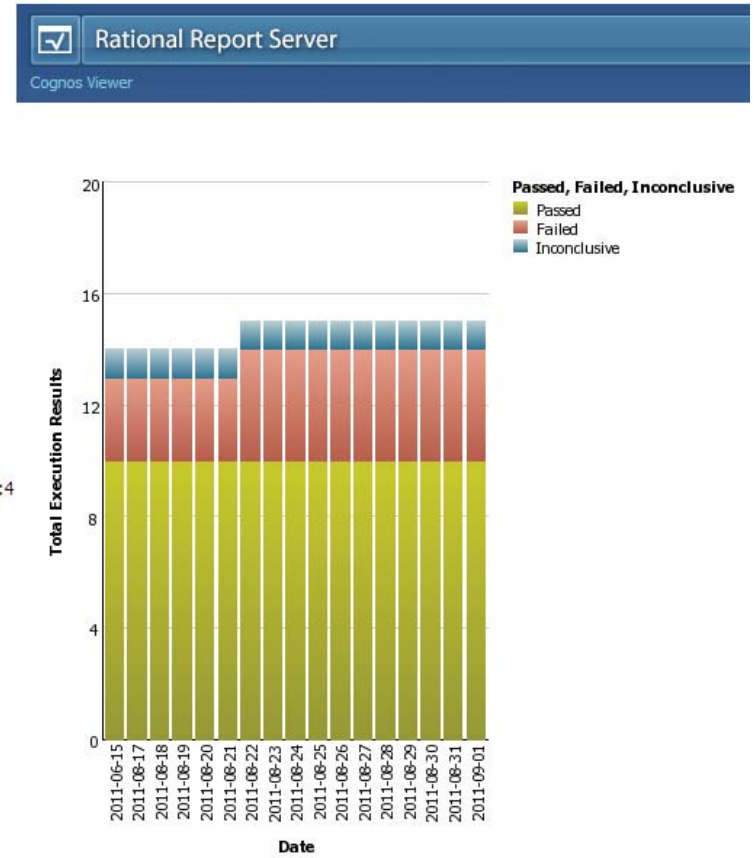


© 2011 IBM Corporation

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

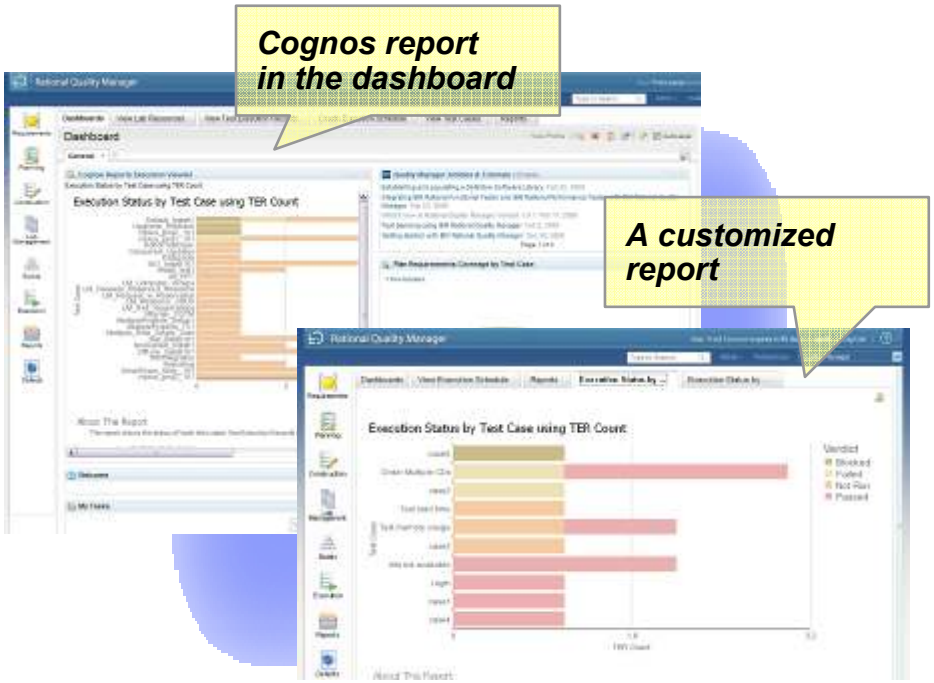


© 2011 IBM Corporation

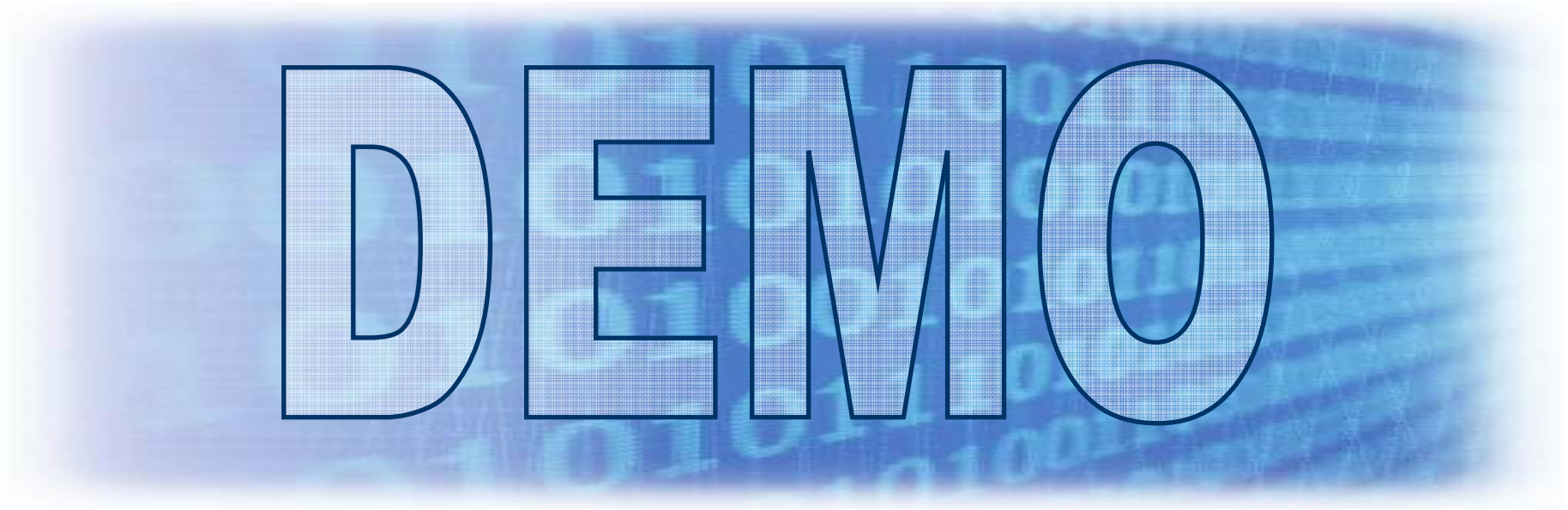
Make informed decisions and proactively change with real-time analysis and actionable reporting

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



© 2011 IBM Corporation



What You'll See:

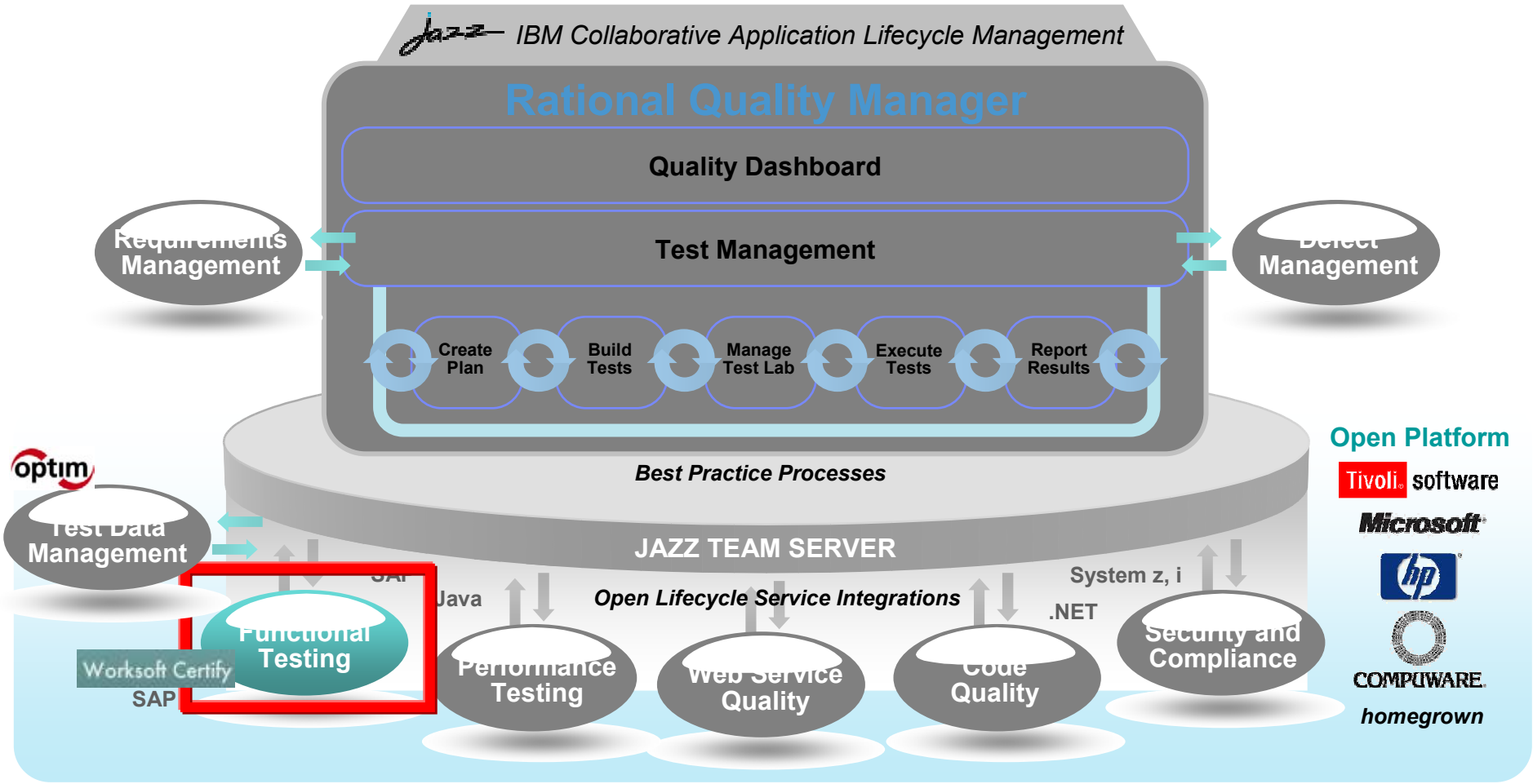
- Rational Quality Manager

© 2011 IBM Corporation

Automated Testing
IBM Rational Functional Tester

© 2011 IBM Corporation

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



© 2011 IBM Corporation

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

© 2011 IBM Corporation

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

© 2011 IBM Corporation

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

© 2011 IBM Corporation

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

© 2011 IBM Corporation

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

© 2011 IBM Corporation

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

© 2011 IBM Corporation

A Fairy Tale

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

© 2011 IBM Corporation

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.

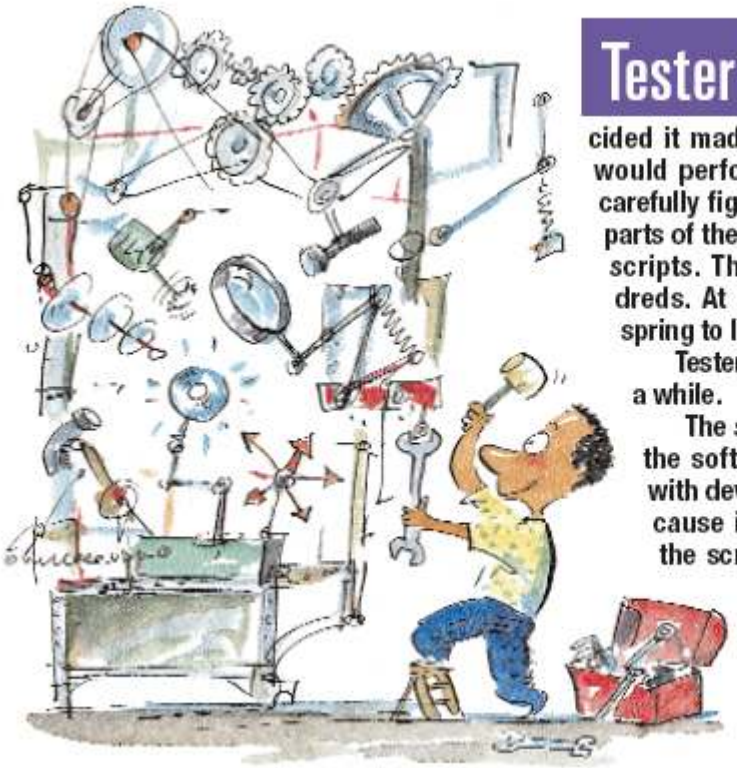
Tester 1 started hands-on testing immediately, and found some nice bugs. The development team happily fixed these bugs, and gave Tester 1 a fresh version of the software to test. More testing, more bugs, more fixes.

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.

© 2011 IBM Corporation



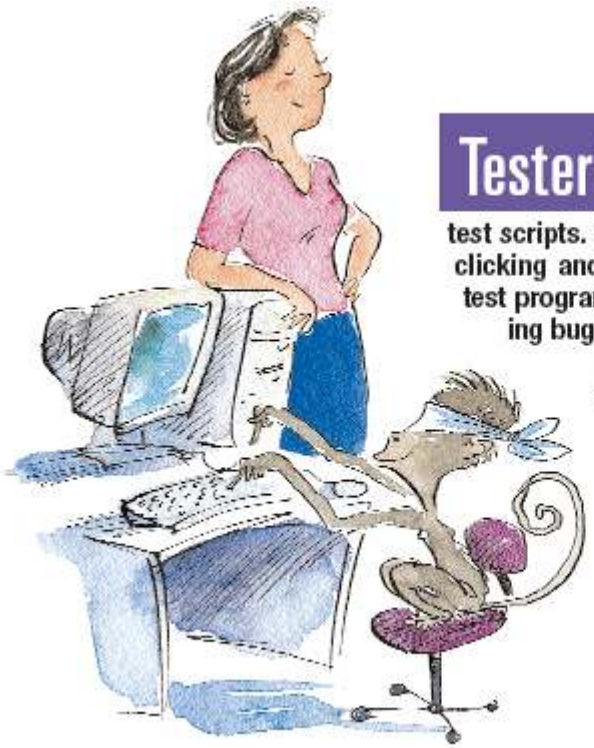
Tester 2 started testing by hand, but soon decided it made more sense to create test scripts that would perform the keystrokes automatically. After carefully figuring out tests that would exercise useful parts of the software, Tester 2 recorded the actions in scripts. These scripts soon numbered in the 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.

© 2011 IBM Corporation



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.

© 2011 IBM Corporation

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.

© 2011 IBM Corporation

Maximize your investment in test automation

With IBM Rational Functional Tester

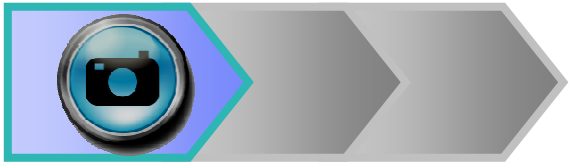
- Achieve success quickly and minimize maintenance
  - Simplified natural language scripting with Storyboard testing
  - Eclipse based or Visual Studio .net
  - Easy to learn
  - Maximize reuse
- Complete test coverage
  - Supports testing for Java, Web, Visual Basic .Net, SAP, Siebel, Web 2.0, Power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

© 2011 IBM Corporation

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

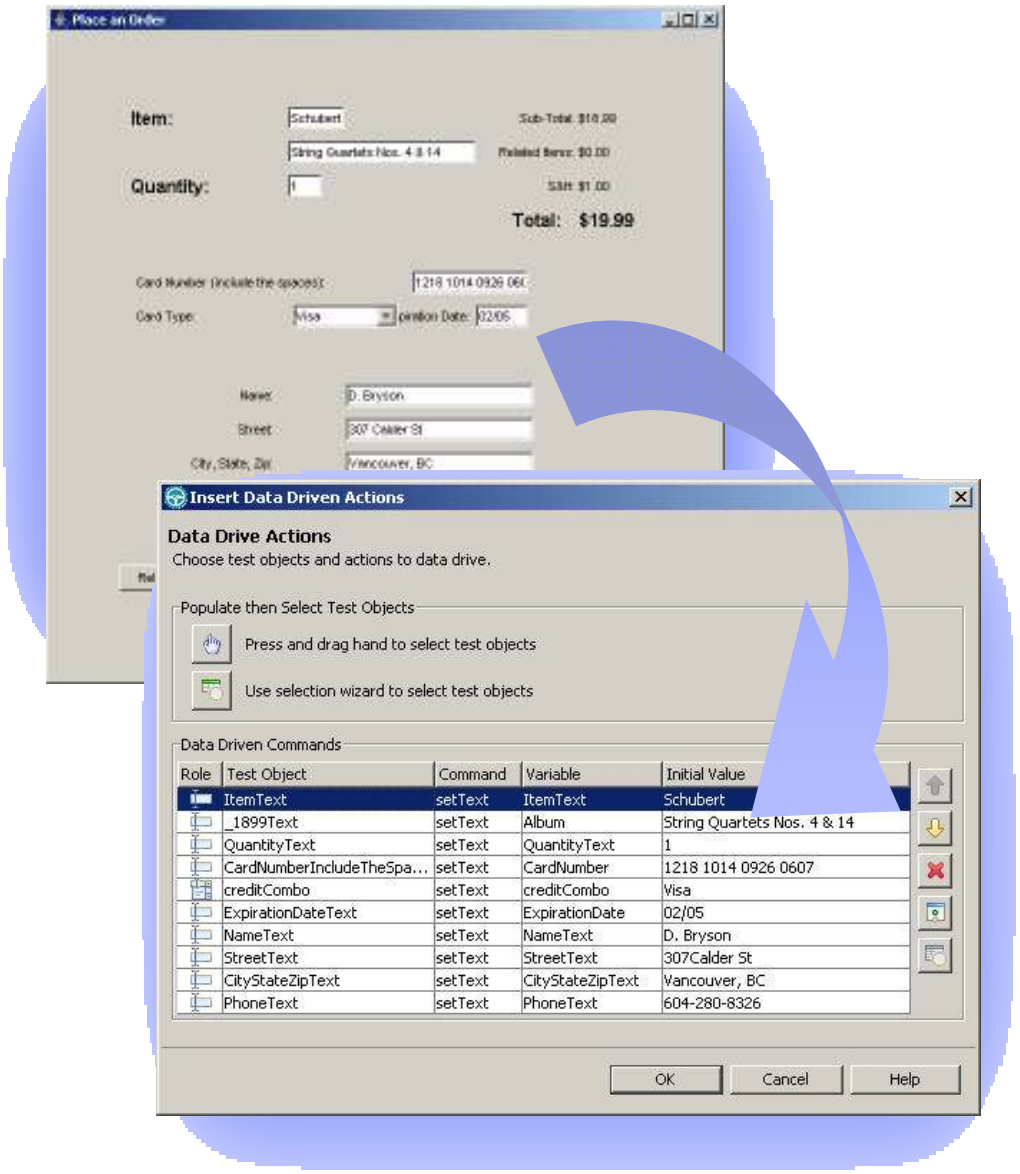
- Tool mentors and process advisors accelerate training
- Broad environment support
- Create data driven tests without coding
- Static data and properties verification
- Dynamic data validation without coding

© 2011 IBM Corporation

Recording Scripts

Data Driven Testing

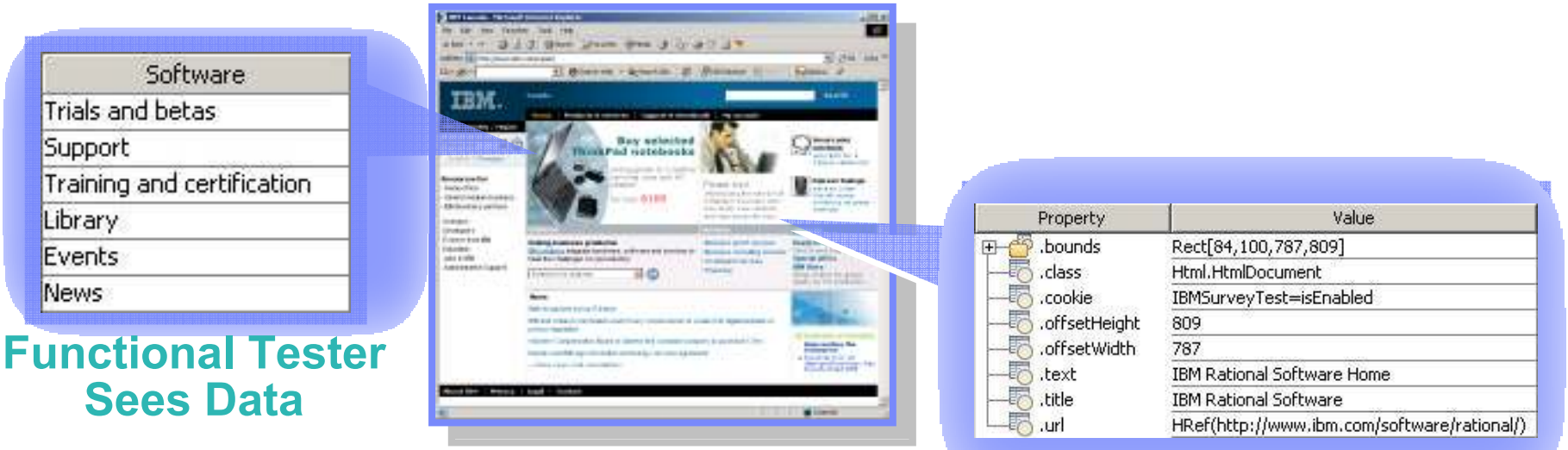
- **Data Driven Testing**
  - Separates test data from test script
  - Enables a single script to run multiple tests by using multiple data sets
- **Wizard driven process**
  - No programming involved
  - Import data from external sources



© 2011 IBM Corporation

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

© 2011 IBM Corporation

Recording Scripts

Validating Dynamic Data

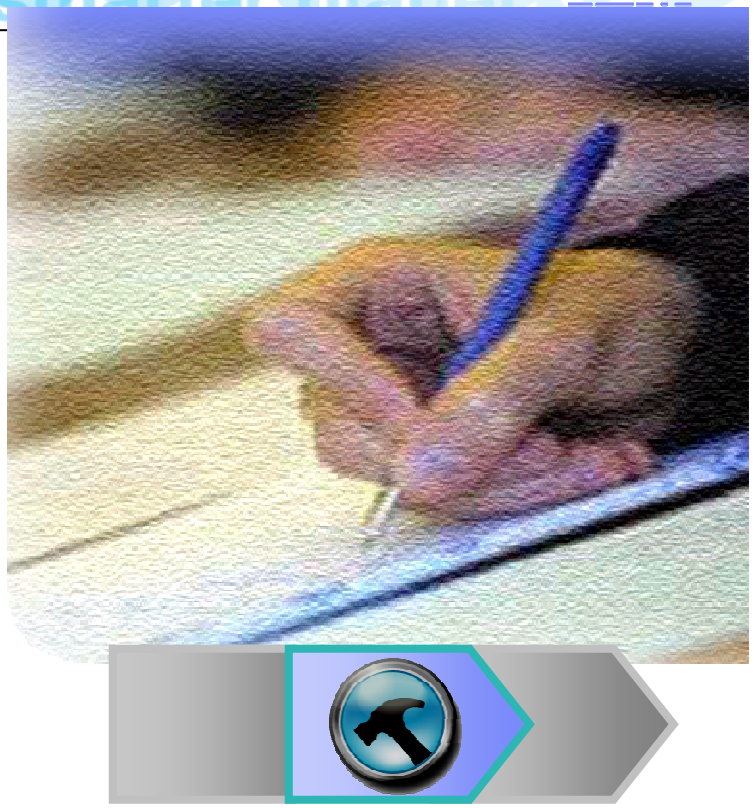
- Use pattern matching technique to verify dynamic data and create robust tests
- E.g. Instead of validating “Order ID 230”, validate “Order ID ###”
  or Order ID 2##, etc.
- This allows for a wide variety of acceptable responses as well as restrictions on acceptable responses when validating the application’s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

© 2011 IBM Corporation

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

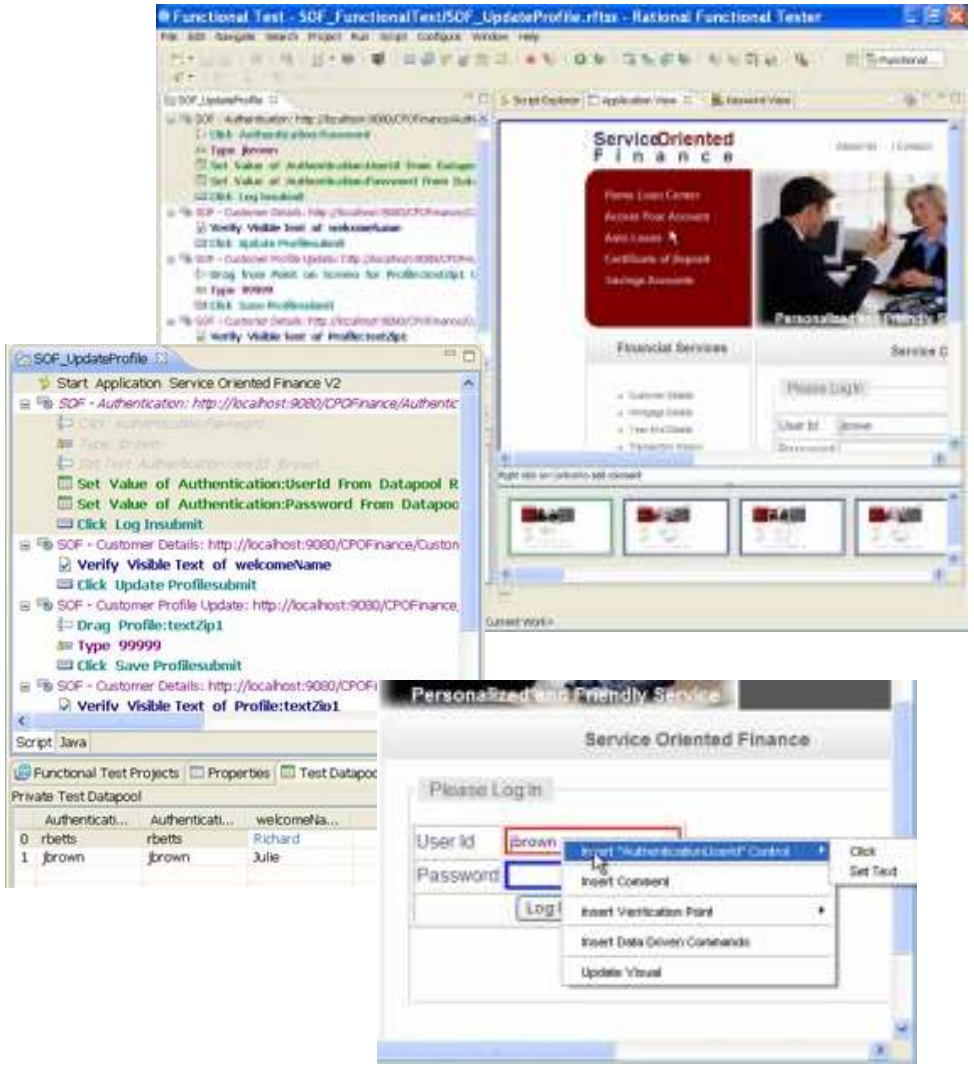
- Pure Java and VB.Net provides flexibility
- Professional debugger
- Central object map to minimize rework
- RTC-ready for version control
- Dynamic data validations without coding

© 2011 IBM Corporation

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

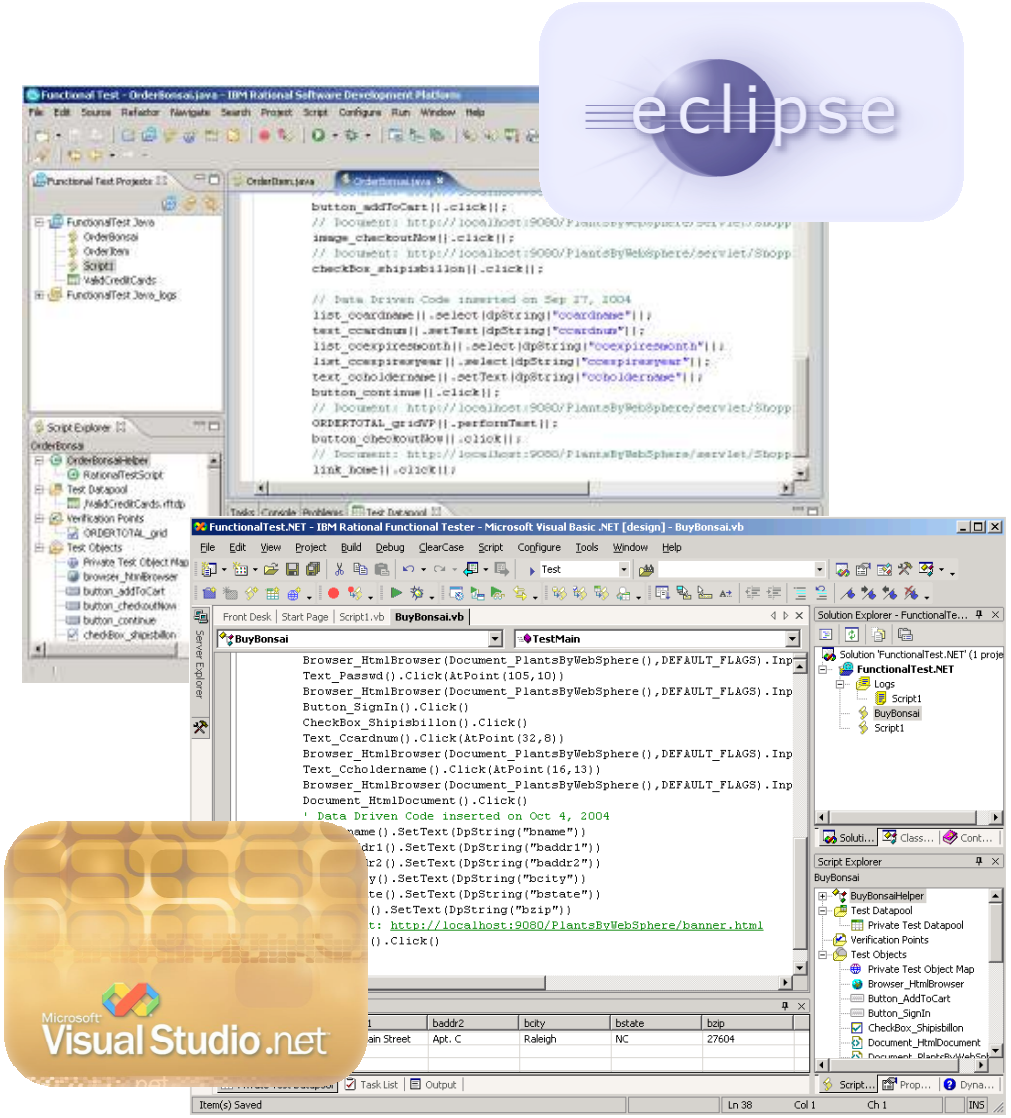


© 2011 IBM Corporation

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…

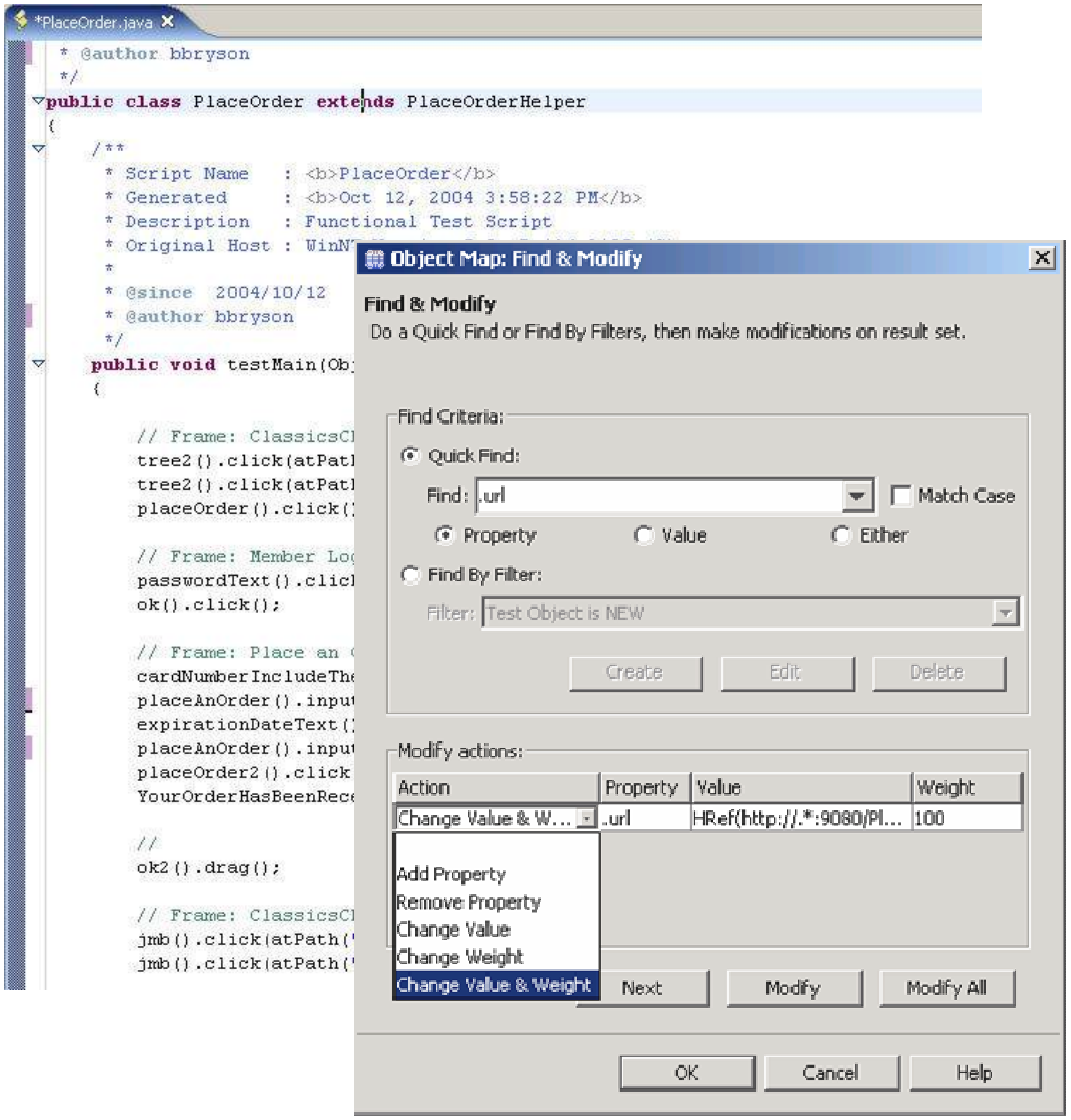


© 2011 IBM Corporation

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

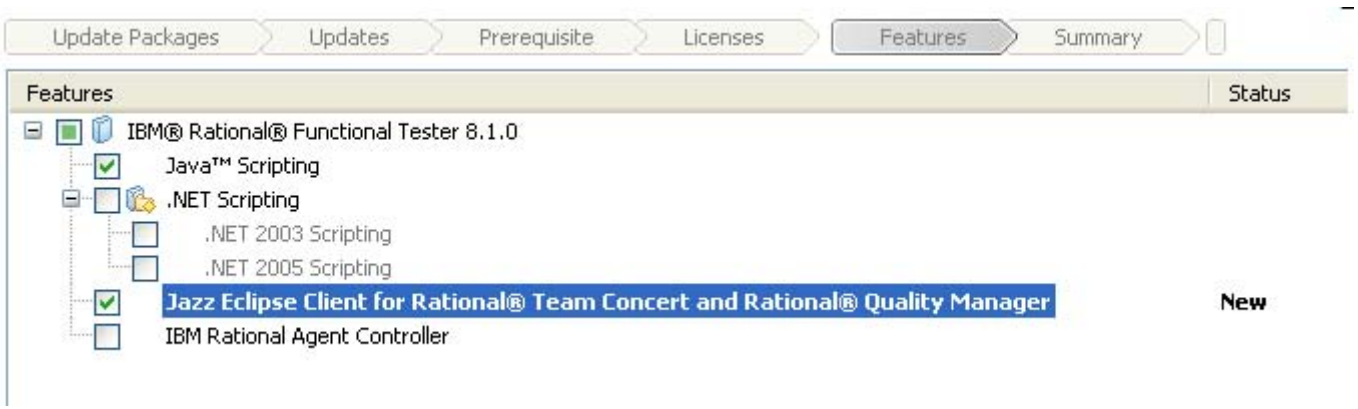


© 2011 IBM Corporation

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



© 2011 IBM Corporation

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

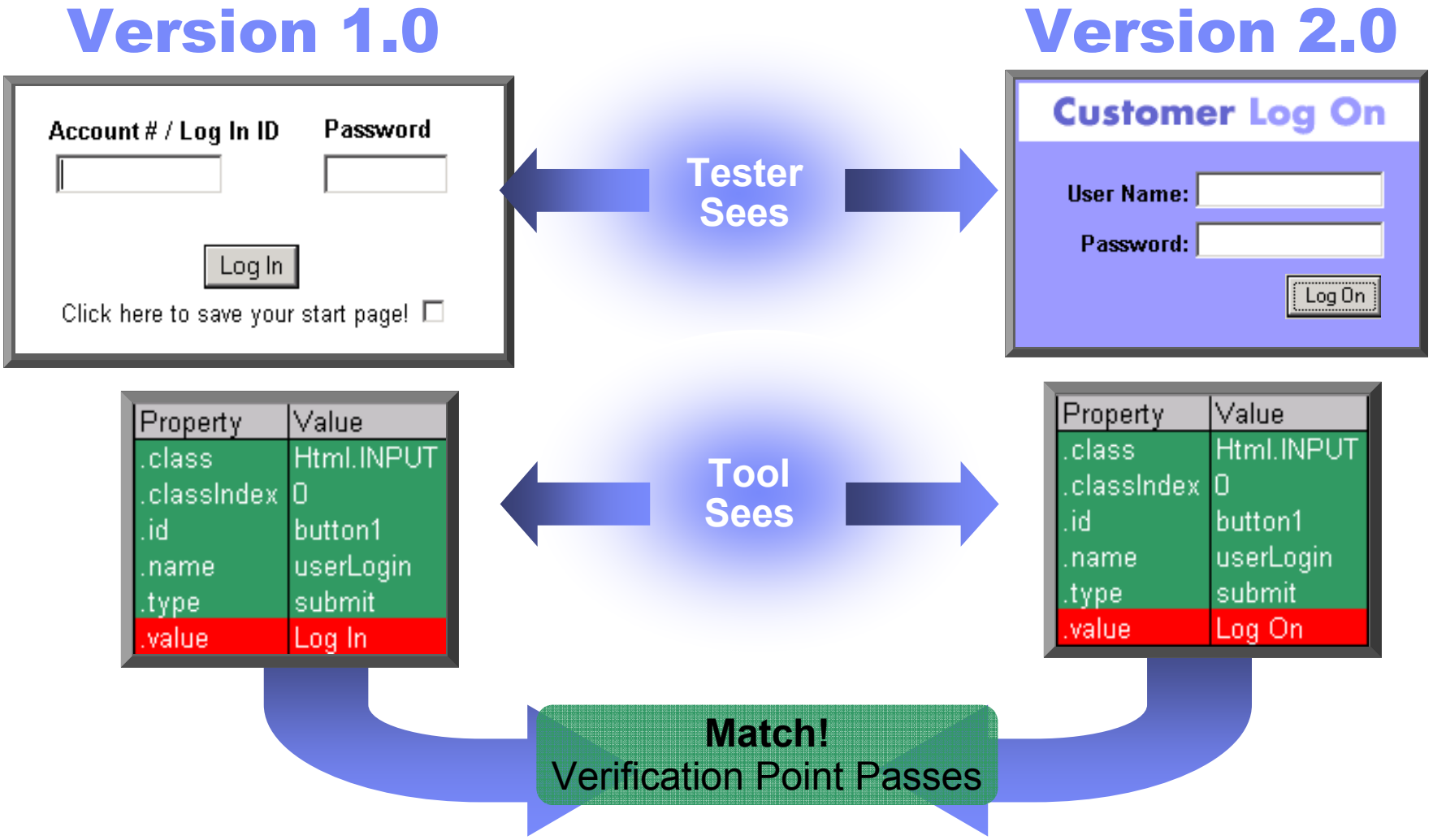
Functional Tester Highlights

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

© 2011 IBM Corporation

Reduce Test Script Maintenance

Reliable Playback with Script Assure



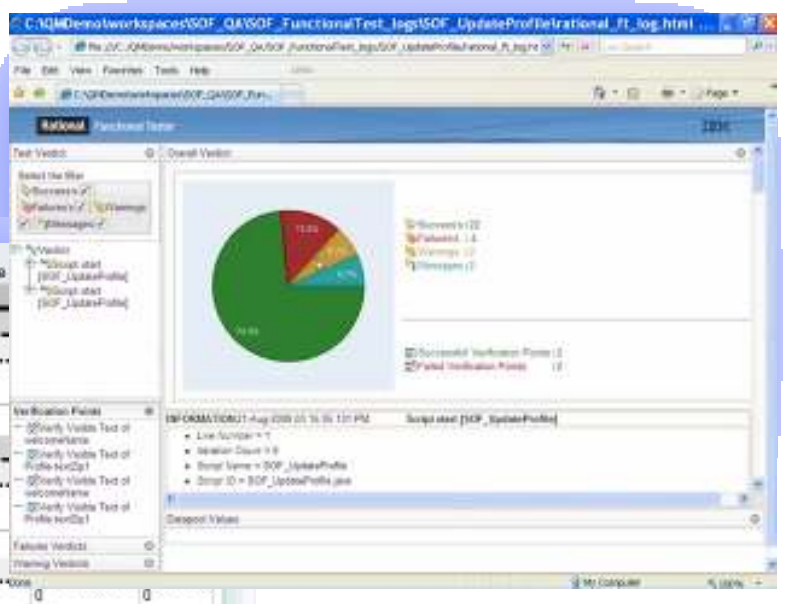
© 2011 IBM Corporation

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



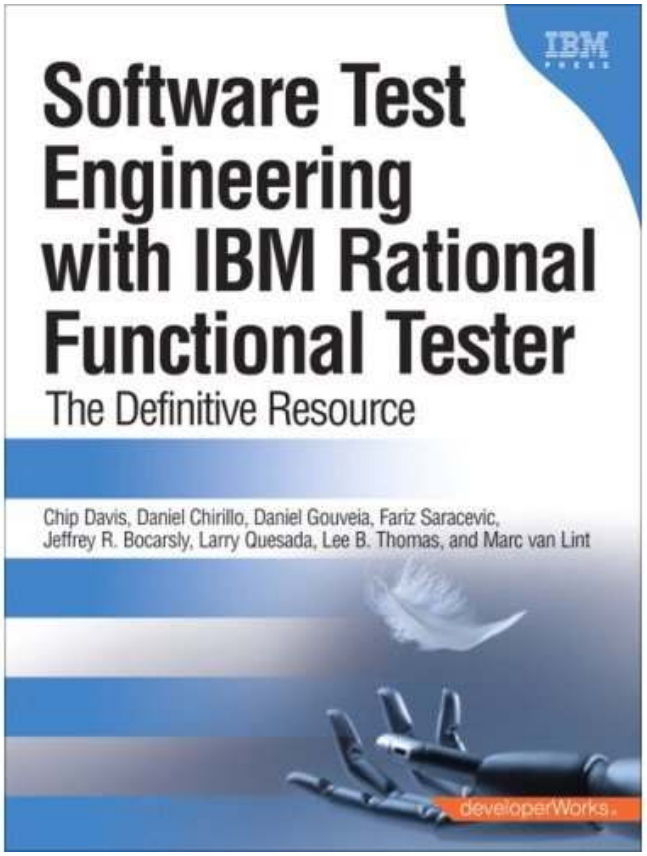
Rational Quality Manager

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

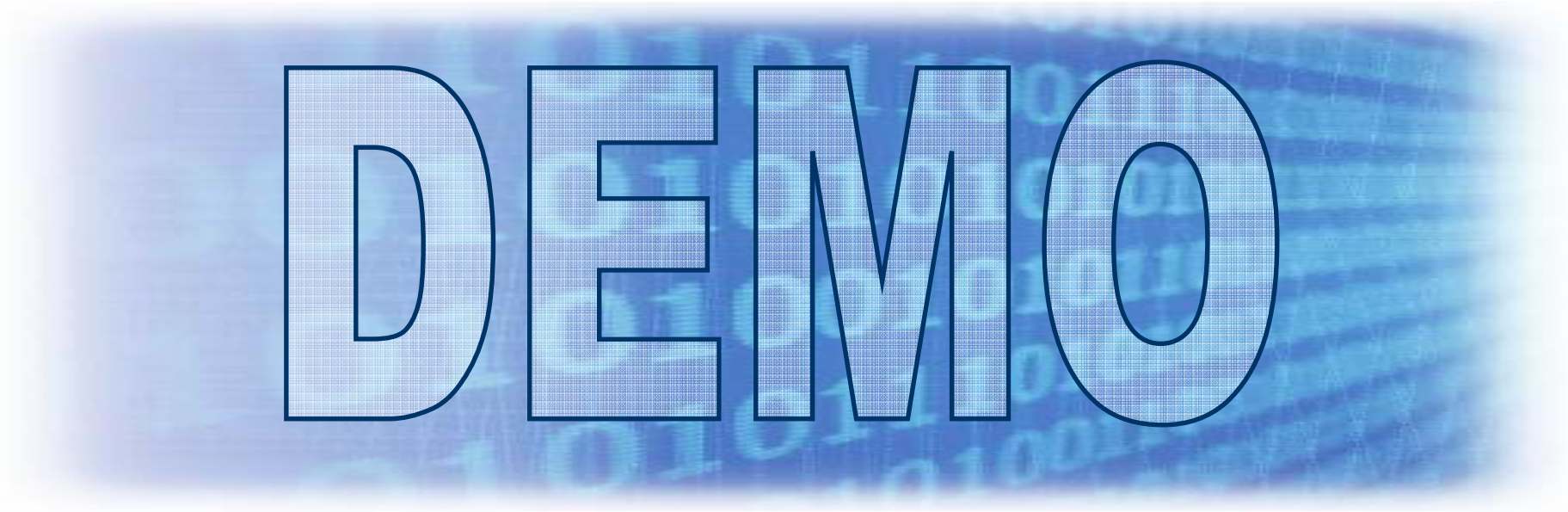
© 2011 IBM Corporation

Supporting resources

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



© 2011 IBM Corporation



What You'll See:

- Rational Functional Tester

© 2011 IBM Corporation

© 2011 IBM Corporation