



IBM Rational Software Conference 2009  
As Real as It Gets!



# Quality Management Keynote

Real Teams, Real Insights, Real Results

Kamala Parvathanathan  
Senior Manager, IBM Rational

**Rational.** software

# Quality Management Keynote

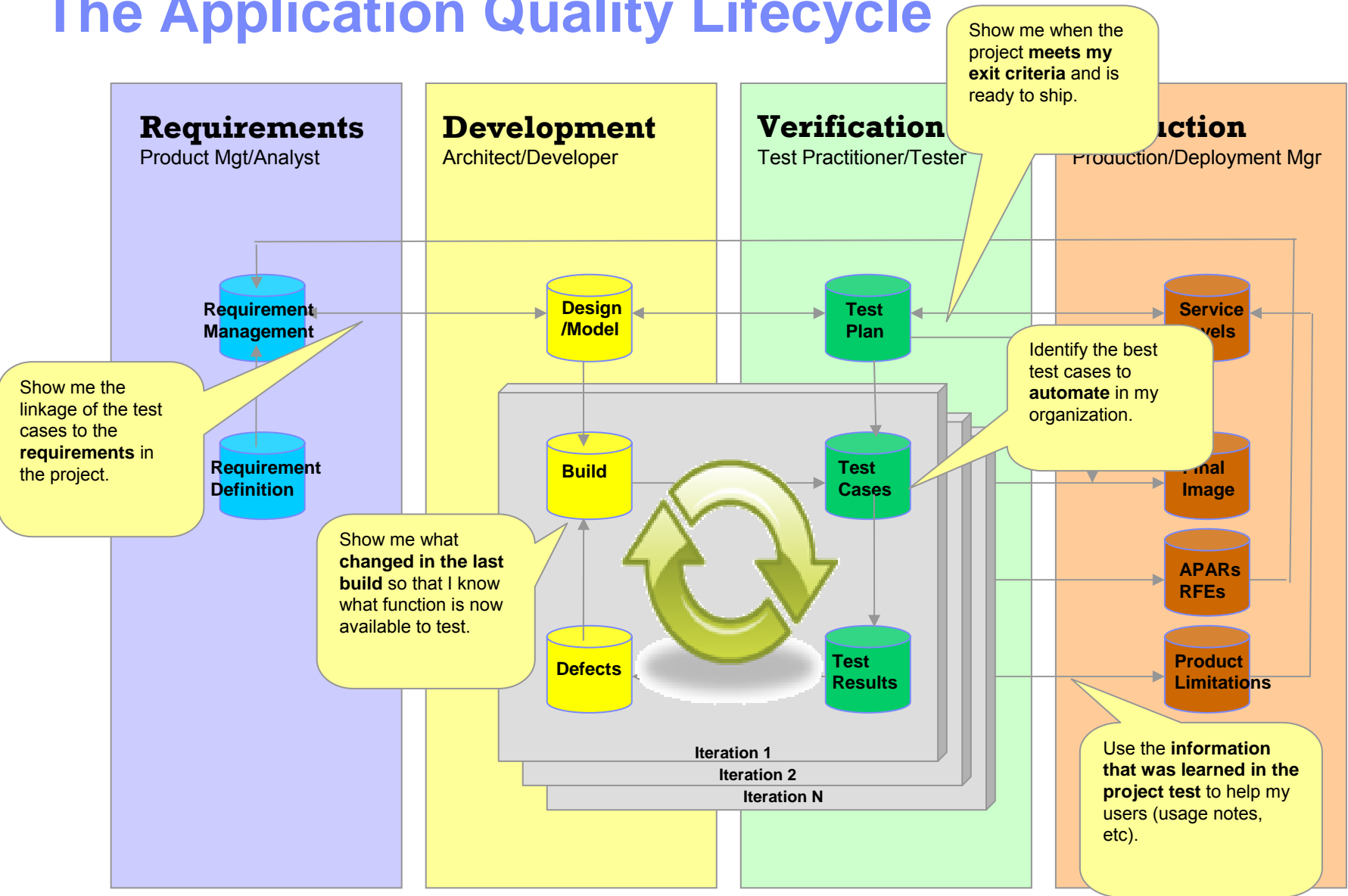
- Real Challenges for real Teams
- Real Insights
- Real Results
- Wrap Up



## Today's Challenges to Software Quality

- Enterprise pressure to **reduce cost**
- Competitive demand to **deliver faster**
- Worldwide geographical **dispersed development**
- **Unpredictable/unknown use** of application
- Increasingly **complex software** ecosystems
- New requirements for **government standards**
- Lack of **domain knowledge/content**
- Inability to **search/reuse test assets**
- Lack of leveraging **previous test experiences**
- Unclear **business goals** for test criteria
- Limited use of **real customer data** in verification
- Lack of published **open standards** for vendor integration

# The Application Quality Lifecycle



# Quality Management Keynote

- Real Challenges for real Teams
- Real Insights
- Real Results
- Wrap Up



fig. 2.



fig. 3.



fig. 4.



fig. 5.



	1858	1860
Balance Scale	30	25.00
Case	10	1.00
Cotton Wool	50	12.00
3rd	20	1.00
4th	10	1.00
Capac	10	2.00
Mercury	10	1.00
Oil	10	1.00
Tea	10	1.00
Sugar	10	1.00
Toler. Set. 4th	10	1.00
" " " " 1st	10	1.00
Cast. Iron	10	1.00
3rd	10	1.00
4th	10	1.00
Case	10	1.00
Winding	10	1.00
Minute	10	1.00
Clock (1st cut)	10	1.00
Cast. Iron	10	1.00

(G. E. DeLong)

# The increasing costs of fixing a defect

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



**\$80/defect**

During the requirements phase



**\$240/defect**

During the design phase



**\$960/defect**

During the QA/Testing phase



**\$7,600/defect**

Once released as a product

**Traditional QA Testing**  
25 – 30 % delivery time in testing  
Poor upstream quality yields rework  
Compressed schedules make it worse

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  
Defect FFR cost for other phases calculated by using the multiplier on a blended rate of \$80/hr.

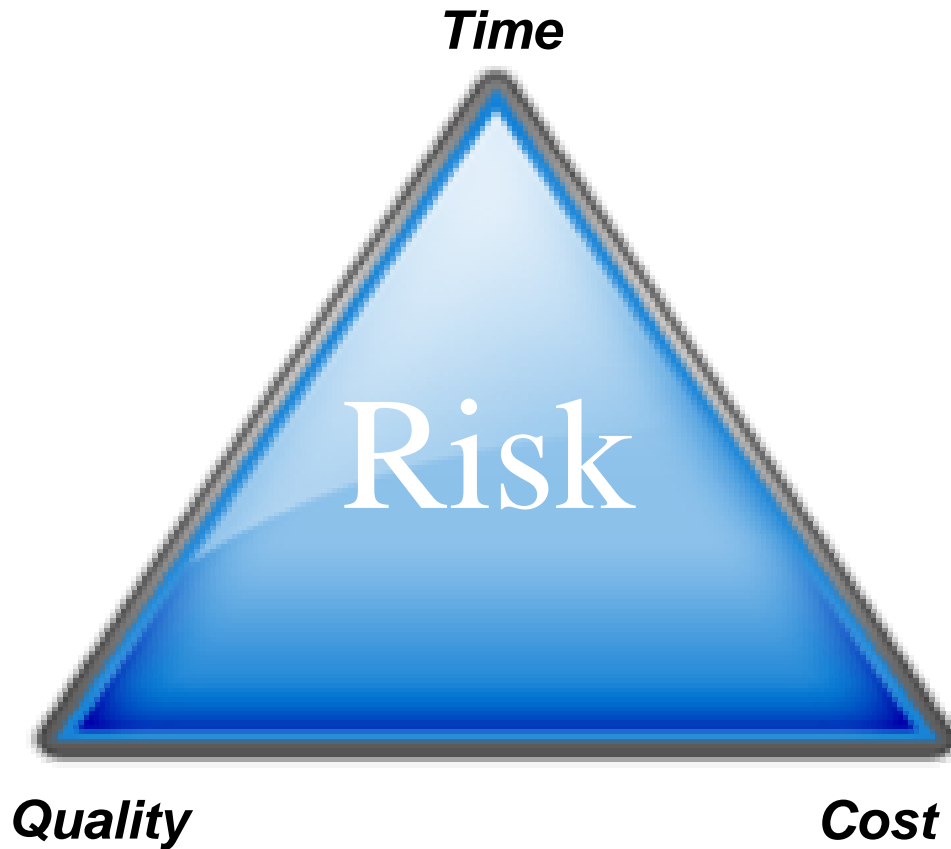
# Rain sensing wiper system design failure

*Individual Systems Worked, But Failed When Integrated*

- Windshield provided by local supplier
  - ▶ Incompatible with the operation range of the sensor
  - ▶ *Cars were sent to customers with non-functioning wiper system*
- Initial diagnostics designated software as culprit for malfunction
  - ▶ Mechanics couldn't test software behavior
  - ▶ *Failure was not of individual components, but in the interaction at a system level*



# Elements of Risk: Time, Quality, Cost



**Each of the vertices is  
an embodiment of risk:**

**Time Risk  
Cost Risk  
Quality Risk**

**Risk cannot be removed**

**But it can be understood  
and controlled**

---

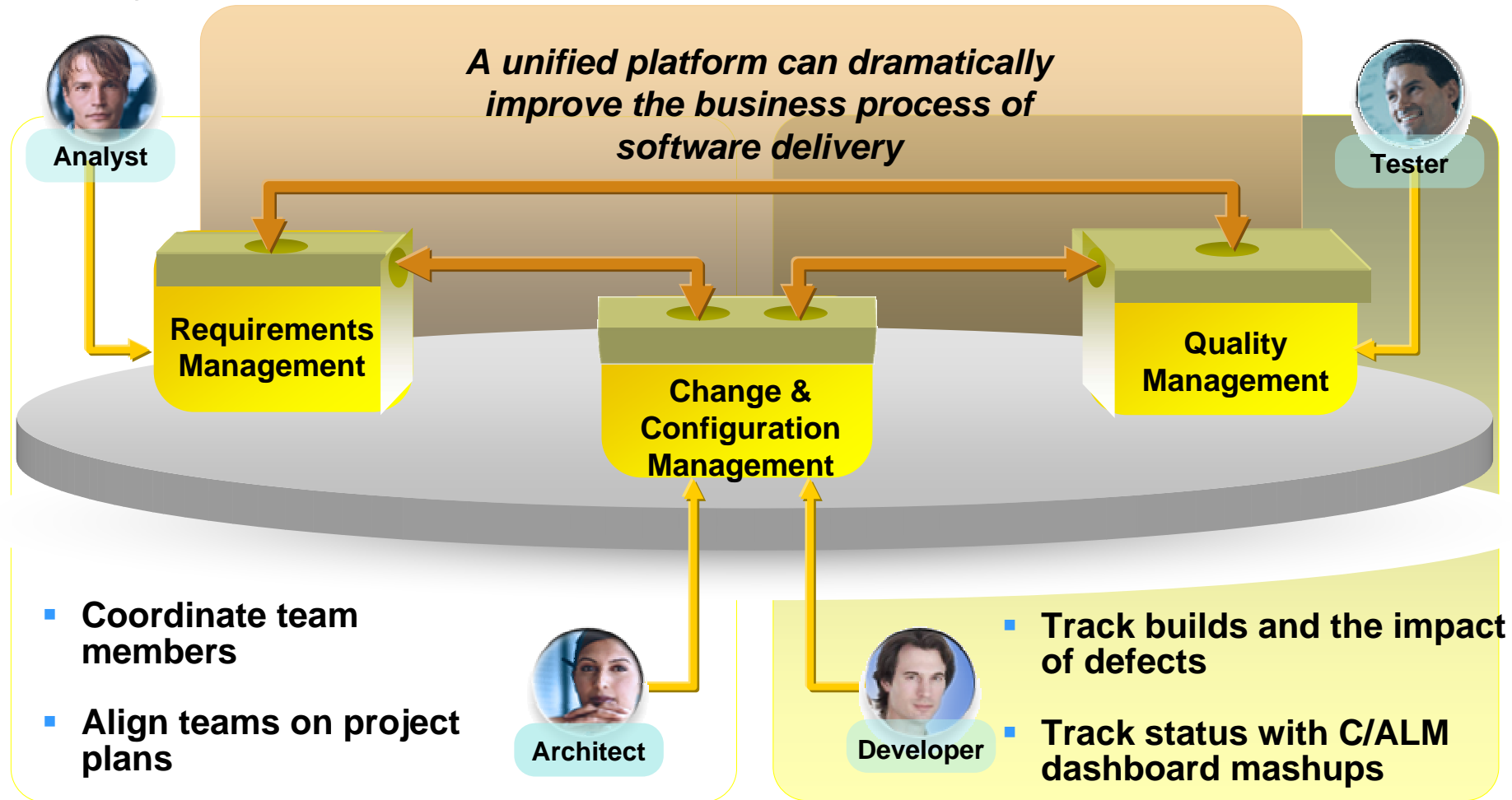
*With IBM Rational Quality Management...*





# Unified platform for software delivery

*Moving beyond individual tools*



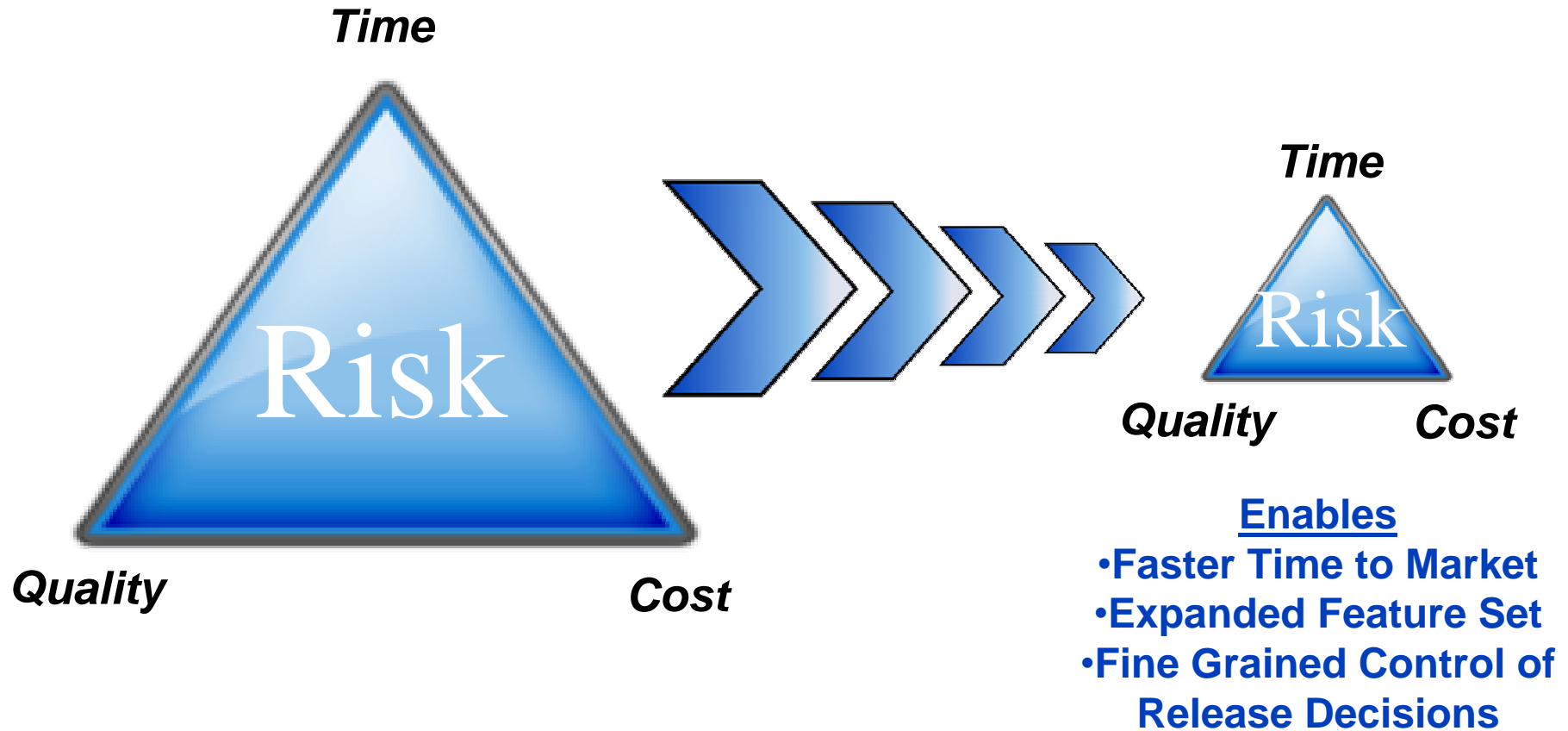
# Risk Management through Process Improvement



*Process Improvements and Tooling efficiencies **Reduce your Risk and Cost***



# Risk Management through Process Improvement



*With IBM Rational Quality Management...*

# Quality Management Keynote

- Real Challenges for real Teams
- Real Insights
- Real Results
- Wrap Up



fig. 2.



fig. 3.



fig. 4.



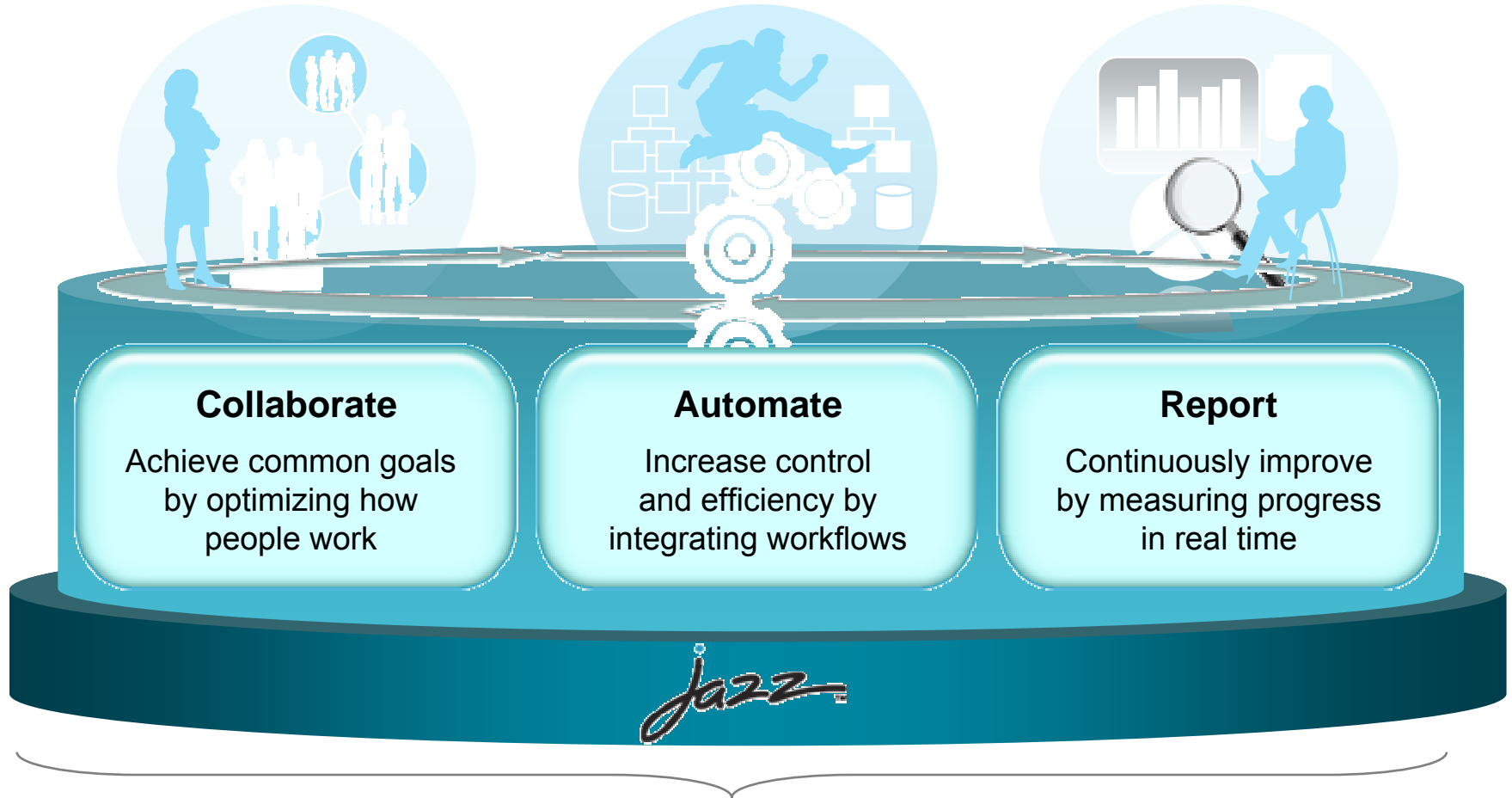
fig. 5.



Balance	Weight	Value
Balance	30	1.200
Balance Wheel	60	1.200
3rd	30	1.100
4th	30	1.100
Cyclo	12	1.100
Mercury	60	1.100
5th	60	1.100
6th	60	1.100
7th	60	1.100
8th	60	1.100
9th	60	1.100
10th	60	1.100
11th	60	1.100
12th	60	1.100
13th	60	1.100
14th	60	1.100
15th	60	1.100
16th	60	1.100
17th	60	1.100
18th	60	1.100
19th	60	1.100
20th	60	1.100
21st	60	1.100
22nd	60	1.100
23rd	60	1.100
24th	60	1.100
25th	60	1.100
26th	60	1.100
27th	60	1.100
28th	60	1.100
29th	60	1.100
30th	60	1.100

(G. P. Delong)

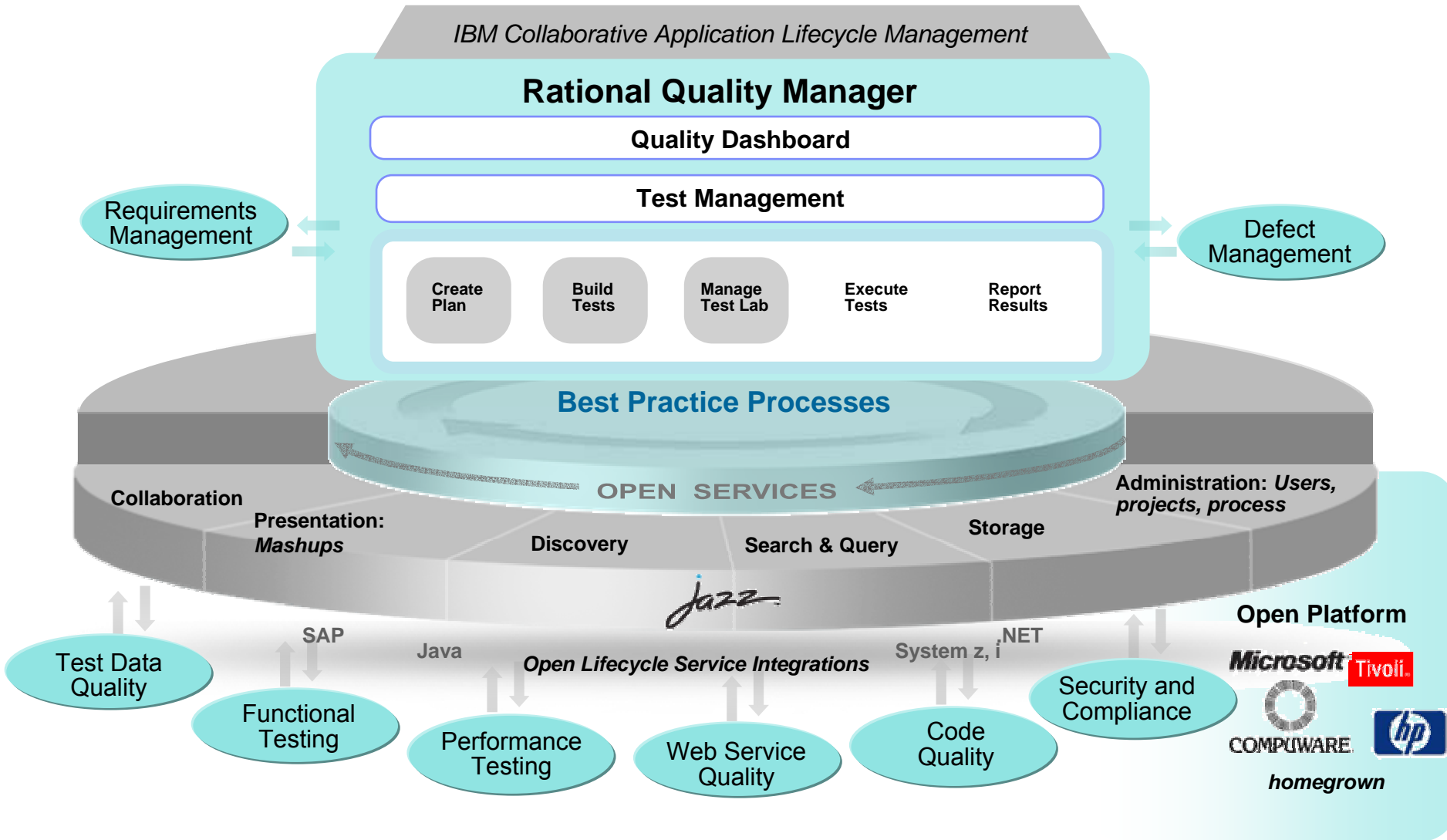
# Quality Management Imperatives



**Improve knowledge and practice maturity with an environment that develops individual and team talent.**



# Quality Management Hub



## Review 2008

- ✓ Rational Quality Manager/Rational Test Lab Manager 1.0
- ✓ Rational Functional Tester 8.0
- ✓ Rational Performance Tester/ Rational Service Tester 8.0

## Review 2009

- ✓ Rational Quality Manager/Rational Test Lab Manager 2.0
- ✓ Rational Functional Tester 8.1
- ✓ Rational Performance Tester/ Rational Service Tester 8.1



## Rational Quality Manager 2.0

### ■ We heard you -- feedback concerning 1.0/1.0.1 of RQM

- ▶ Usability
- ▶ Better searching capabilities
- ▶ Needing to have a faster ROI
- ▶ Integrations
- ▶ Custom Reporting
- ▶ Performance



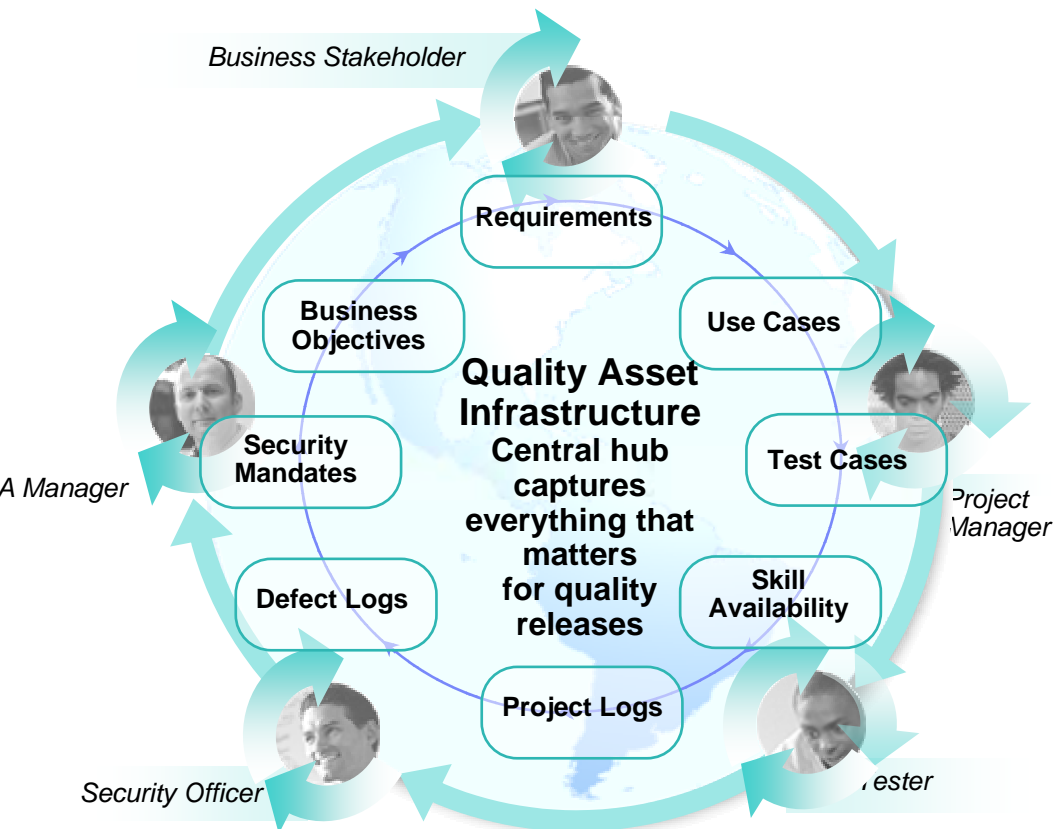
### ■ Problems to solve

- ▶ Two weeks chopped off my test cycle, need help prioritizing
- ▶ Tracking test progress
- ▶ Understanding retesting



# Comprehensive rich test plan

*A quality contract for the entire software delivery team*



- **Collect and track all test data**
  - ▶ Central location for business objectives requirements, resources, platform and exit criteria to name a few
- **Defined Responsibilities**
  - ▶ Individual sections are assigned to team members to clearly establish ownership
- **Goal Oriented**
  - ▶ Formalized and documented exit criteria
- **Extensible**
  - ▶ Add sections, import custom data
- **Keep track of changes**
  - ▶ Snapshot version control to track plan history throughout the life of the project

# Comprehensive dynamic planning and updates

*Process flow, not artifacts drives team activities*

## Other Test Plans

Table of Contents  
DOCUMENT HISTORY  
TABLE OF CONTENTS

**1 INTRODUCTION**

1.1 Overview  
1.2 Test Objectives  
1.3 Test Scope  
1.4 Assumptions  
1.5 Glossary  
1.6 References

**2 TEST STRATEGY**

2.1 Business Function  
2.2 Structural Function  
2.3 Risk Assessment  
2.3.1 Unavailability of test data  
2.3.2 Test data for Unit team  
2.3.3 Assessing correctness of calculation  
2.4 Test Focus areas  
2.5 Levels of Testing  
2.5.1 Development testing  
2.5.2 System testing  
2.6 Functional and Structural Test Types  
2.6.1 Test Focus Type matrix  
2.6.2 Test Levels Type matrix

**3 TEST PLAN**

3.1 Roles and responsibilities  
3.2 Test Schedule  
3.3 Major testing milestones  
3.4 Resource requirements  
3.5 Testing for Non-Functional Requirements

**4 TEST**

4.1  
4.1.1  
4.1.2  
4.1.3  
4.1.4  
4.1.5  
4.2  
4.2.1  
4.2.2  
4.2.3  
4.2.4  
4.2.5  
4.2.6  
4.2.7  
4.2.8  
4.2.9  
4.2.10  
4.2.11  
4.2.12  
4.2.13  
4.2.14  
4.2.15  
4.2.16  
4.2.17  
4.2.18  
4.2.19  
4.2.20  
4.2.21  
4.2.22  
4.2.23  
4.2.24  
4.2.25  
4.2.26  
4.2.27  
4.2.28  
4.2.29  
4.2.30  
4.2.31  
4.2.32  
4.2.33  
4.2.34  
4.2.35  
4.2.36  
4.2.37  
4.2.38  
4.2.39  
4.2.40  
4.2.41  
4.2.42  
4.2.43  
4.2.44  
4.2.45  
4.2.46  
4.2.47  
4.2.48  
4.2.49  
4.2.50  
4.2.51  
4.2.52  
4.2.53  
4.2.54  
4.2.55  
4.2.56  
4.2.57  
4.2.58  
4.2.59  
4.2.60  
4.2.61  
4.2.62  
4.2.63  
4.2.64  
4.2.65  
4.2.66  
4.2.67  
4.2.68  
4.2.69  
4.2.70  
4.2.71  
4.2.72  
4.2.73  
4.2.74  
4.2.75  
4.2.76  
4.2.77  
4.2.78  
4.2.79  
4.2.80  
4.2.81  
4.2.82  
4.2.83  
4.2.84  
4.2.85  
4.2.86  
4.2.87  
4.2.88  
4.2.89  
4.2.90  
4.2.91  
4.2.92  
4.2.93  
4.2.94  
4.2.95  
4.2.96  
4.2.97  
4.2.98  
4.2.99  
4.2.100

APPEND  
APPEND

Word based Test Plan

**Test Plan Tree**

## Rational Quality Manager Plan

**Table Of Contents**

Summary

Business Objectives

Test Objectives

Review and Approvals

Requirements

Application Security

Test Iterations

Sizing

Environments

Test Team

Quality Goals

Entry Criteria

Exit Criteria

Test Cases

Attachments

Show All Sections

+ Add New Section

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

# Usability

- Organizing Artifacts
  - User defined Hierarchical Tree Visualization based on artifact relationships
  - More Ability to leverage Categories to limit view of artifacts
- Finding Artifacts
  - Additional Searching/filtering capabilities.
  - Tag cloud for all assets
- Manual Test Improvements
- Catalogue Customizations

## View Test Plans ?

Group by:  View by:

- Project: Paint Shop
  - Test plans
    - Red
      - + Requirements
      - Test Cases
        - Viscosity
          - + Viscosity requirement
          - + Viscosity script
          - Viscosity execution
            - Viscosity result
            - Viscosity defect
        - + Weight
        - + Purity
        - + Brightness
        - + Tone
        - + Saturation
        - + Hue
        - + Color balance
        - + Luminosity
      - + Execution Results
      - + Environments
        - + Pink
        - + White
        - + Black
        - + Orange
        - + Yellow
        - + Chartreuse
        - + Green
        - + Blue
        - + Purple
      - + Test suites
      - + Unassigned
      - + Test lab resources

## But wait, there is more

Group by: **Release** View as: **Explorer** Type Filter Text

Previous Next

Quality Manager	Release 1.1	System Test Plan	Test Cases
<ul style="list-style-type: none"> <li>Release 1.1</li> <li>Unassigned</li> </ul>	<ul style="list-style-type: none"> <li>System Test Plan</li> </ul>	<ul style="list-style-type: none"> <li>Test Cases</li> <li>Requirements</li> </ul>	<ul style="list-style-type: none"> <li>Basic Flow</li> <li>Order CD Basic Flow</li> <li>INVALID CUSTOMER ID OR P...</li> <li>Pre Condition</li> <li>Arrange Shipment</li> <li>SUCCESS</li> <li>Basic Flow</li> <li>TRACK PACKAGES</li> <li>Brief Description</li> <li>TRACK PACKAGES</li> <li>OPTIONS FOR DISPLAY</li> <li>INVALID INFORMATION</li> <li>NO RESPONSE</li> </ul>

- View are dynamically rendered
- Can utilize many ways to view relationships
- Can change the perspective of views (starting point).
  - E.g. View by Release

# Integrated manual test authoring and execution














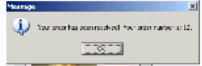


*Maximizing efficiency of manual testing*

Manual Test 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


# Manual Test Improvements

Step	Description	Expected Results
1	Start Classics Application	 This window should appear. 
2	Select a CD. User browses to Schubert and selects String Quarter	
3	Order the selected CD. Press the Place Order Button. The Place Order Window appears with Schubert as a selected item.	 
4	Insert CC Number by using assisted data entry.	 
5	Validate Trent Cuplito is on Name: field using assisted data verification	 
6	 Press the  button. Validate the login screen appears	 
7	Press the OK Button. Validate the order confirmation window appears.	 
8	OK Button	

- **Manual Test Creation**

- Expected results for every step
- Graphic management (thumbnails vs full size)
- Full off-lining of manual test

# Manual Test Improvements

Step	Result	Description	Expected Results	Actual Results
1	Fail	Start Classics Application	 This window should appear.	Login Screen Appeared instead.
2		Select a CD. User browses to Schubert and selects String Quarter		

- **Assistance in manual test execution**

- Inline actual results. Thus, from a log people will see step, expected result, actual result and comments all on screen (and report needs to provide)
- Pause Resume of Manual Test execution
- Manual execution updates the Execution Points based on each step
- Execution Console should show manual tests too
- Ability to easily provide execution comments



# Risk Based Testing

***Risk Management strategy can make or break companies.***

- *Effective Risk Based Testing requires process support.*



- Calculated Risk
  - ▶ Screen hold 10 lbs
  - ▶ Cat weighs 9
  - ▶ High Risk
- ▶ My Risk
  - ▶ Really want that bird
  - ▶ Low Risk
- ▶ Community Risk
  - ▶ What do other cats say?

# Risk Based Testing

**Risk Management strategy can make or break companies.**

- Effective Risk Based Testing requires process support.

Classics Java Test Plan  
Test Plan Overview | View Snapshots

Originator: ADMIN Action: Select Action State: com.ibm.rqm.planning.common.new

Test coverage for the Classics Java application.

Test Cases

Lists the test cases associated with a given plan. You can add and remove associations to test documents and create and associate a new test case. Removing a test case will remove the association to this test plan but not delete the test case.

Group by: Ungrouped Type Filter Text

ID	Risk Assessment	Suspect	Name	State	Category	Function	Theme	Weight	Modified
12	○○○○○	◆	View Report Test Case	com.i...	Reporting	Execution	Functiona...	20	1 minu...
15	○○○○○	◆	Add New Customer	Draft	Web UI	Execution	Functiona...	100	1 minu...
16	○○○○○	◆	Order Multiple CDs	Draft	Web UI	Execution	Functiona...	100	1 minu...
17	○○○○○	◆	Add New Products	Draft	Web UI	Execution	Functiona...	100	1 minu...
18	○○○○○	◆	Administer Orders	Draft	Web UI	Execution	Functiona...	100	1 minu...
19	○○○○○	◆	Order single CD	Draft	Web UI	Execution	Functiona...	100	1 minu...

Requirements  
Test Plans  
Test Cases  
Risk Profiles

Calculated Risk Assessment Collaborative Risk Assessment

**Risk Assessment:**  
○○○○○ very high

**My Risk:**  
Rate this: ○○○○○ very low  
Comment here

**Community Risk:**

Very high:	0	(0%)
High:	0	(0%)
Neutral:	0	(0%)
Low:	0	(0%)
Very low:	1	(100%)
average	○○○○○	very low

- Prioritize efforts where resources are limited
- Drive risk based decisions to meet quality objectives
- Allows for collaboration on risk mitigation strategy
- Documentation of risk related decisions

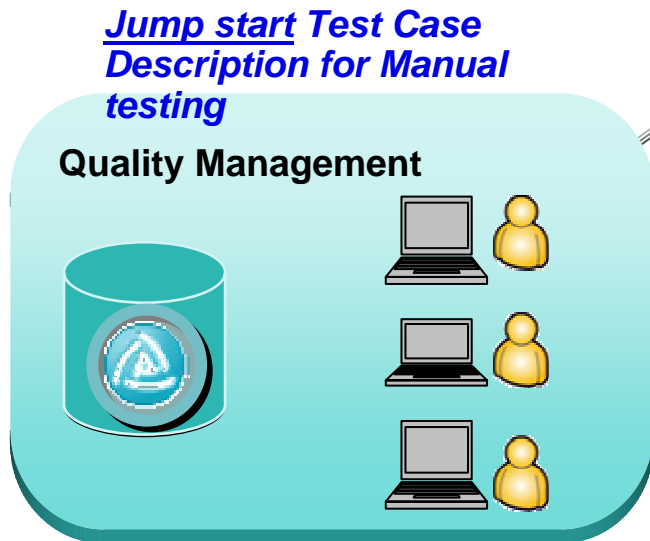
Prioritized Tests to meet business needs



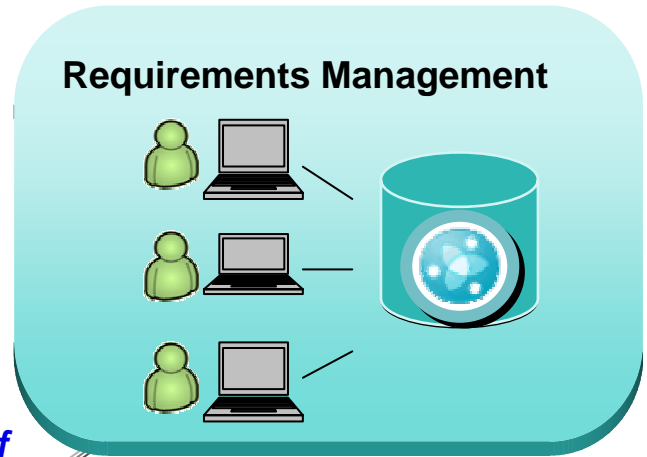
# Productivity Boosters

## Smart Automations...

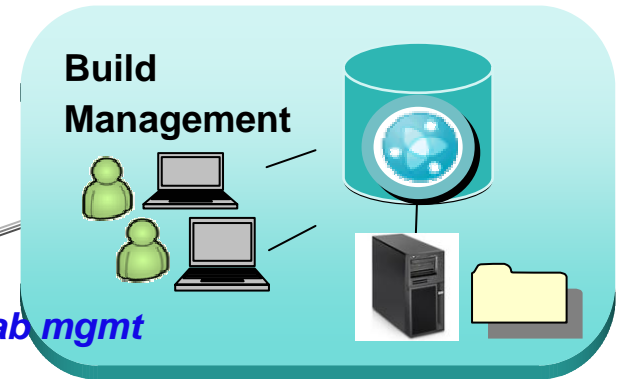
- Increase productivity
- Reduce risk of human error
- Reduce time to quality
- Reduce cost...



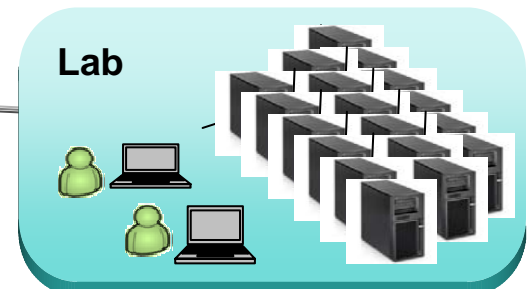
*Jump Start creation of Stubbed test cases from requirements*



*Automated Build record Integration for test and lab mgmt*



*RQM/RTLM Scheduling Automation*



# Organizational Policies

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

**System Test Plan** ?  
 Test Plan Overview | [View Snapshots](#) Discard Changes

Originator: ADMIN Action: Select Action → State: Draft

Quality Objectives ? Work Item:

Defines the overall metrics for what constitutes a quality product.

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

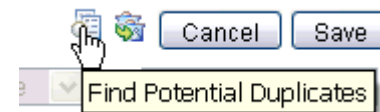
**Select Quality Objectives** ?

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

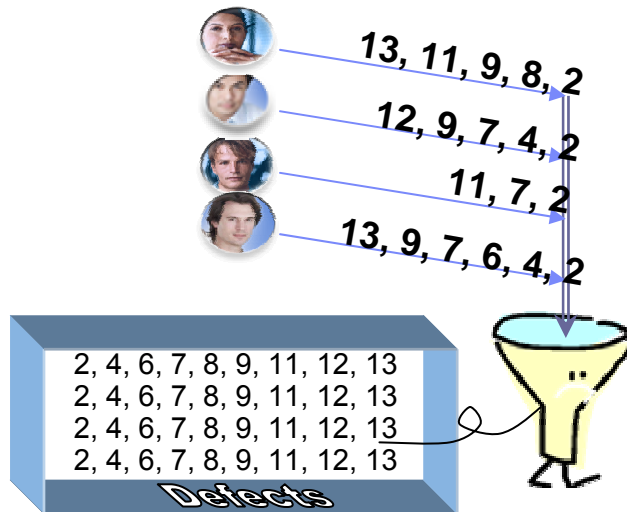
OK Cancel

# Just create defect to ensure in the system. Worry about duplicates later..

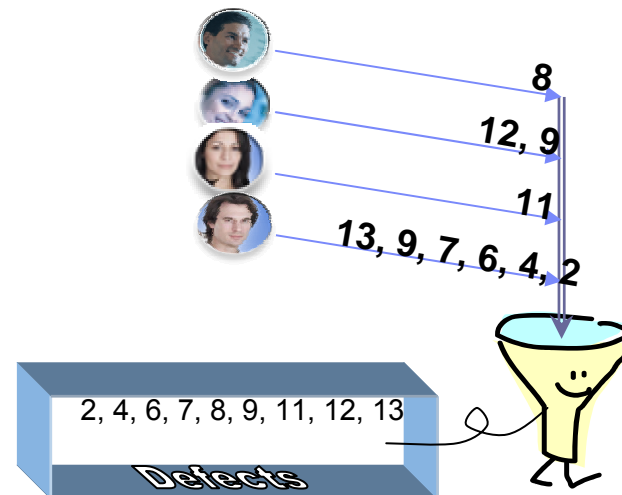
- How long does it take to find and resolve a single duplicate defect?
  - ▶ Are development teams working on same issue unknowingly
  - ▶ Are multiple fixes for the same problem put in different builds
  - ▶ Are testing members validating
- Time is money.....



## Common scenario



## Utopia



# Process-led Automation yields real savings

## Examples of automation capabilities

	Developing repeatable industry test solutions			Advanced Defect Analysis	Developing repeatable test procedures applicable to future projects		Integrating end to end processes		
Asset	Test cases copied	Manual scripts copied	Manual scripts Reuse	Prevent & block duplicate Defects	Baseline & migrate documentation	Baseline artifacts	Leveraging component Reuse	Dynamic updates of test assets	Total
Quantity	343	350	1,393	905	1,365	2023	1029	2,227	9,635
Hours saved	167	175	696	1,755*	683	1,011	515	557	5,558
Value	\$16,690	\$17,514	\$69,633	\$175,452	\$68,254	\$101,125	\$51,459	\$55,673	
Total						\$169,379		\$107,132	\$555,799

**\$175,452**

The biggest hidden QM cost is associated with defect duplications

\* Hours saved assumes an average of 100\$/hr

Source: GBS Test Practices

Average per project saving was \$175,452 in savings, at a rate of 100\$/hr

reality, it often takes much longer.

was calculated on a per asset task and process

Average savings per project - over half a million dollars

# Integrating DOORS with IBM Rational Quality Manager

*Enabling requirements driven testing*



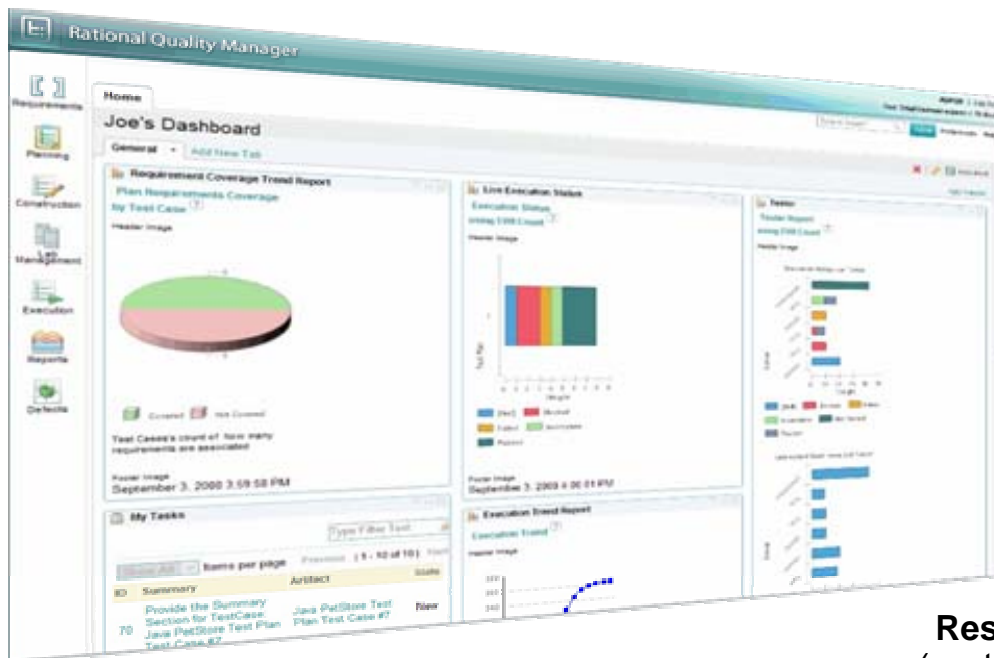
- **Tests based on requirements ensure deliverables meet customer expectations**

- ▶ Plan Tests Early
- ▶ Conduct Tests Early
- ▶ Relate Tests to Requirements
- ▶ Relate Defects to Requirements
- ▶ Measure Progress against Requirements

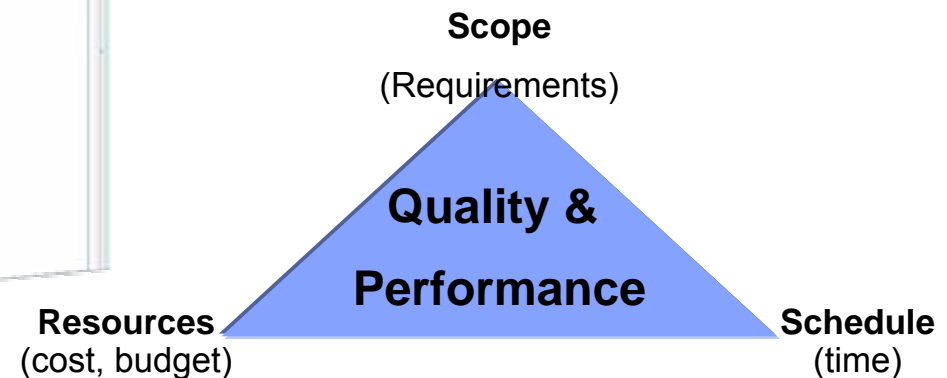
# Closed Loop Analysis & Reporting

## *Reduce time to value, Reduce Risk*

- *Addressing management needs*
  - *Early identification of problem areas*
  - *Standardize reports that communicate ROI (in compliance with MCIF)*
  - *Understanding Test Lab Utilization*

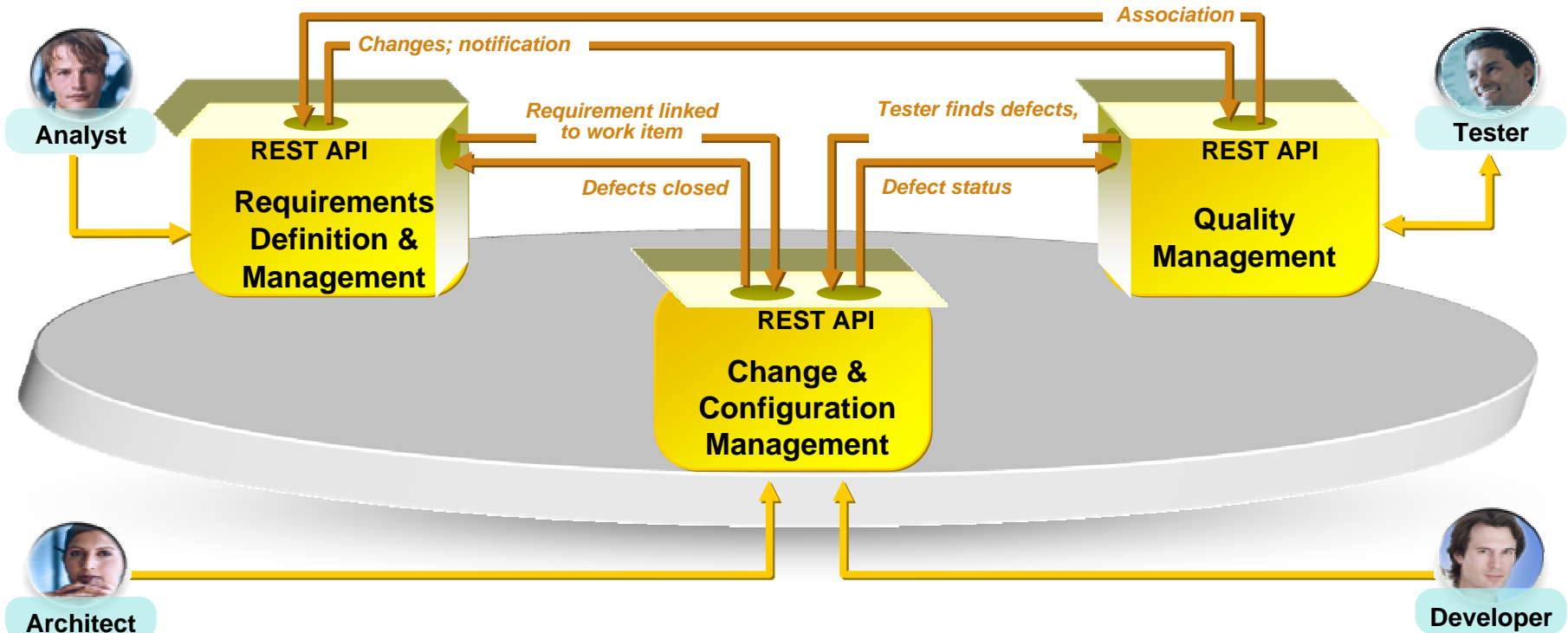


**Increased visibility into  
the quality status  
& actionable reporting**





## Integrations -- Collaborative ALM connects Analysts, Developers and Testers



- **Coordinate analysts, developers and testers**
  - Collaborate in context and track status
- **Align teams on project plans**
  - Which requirements are in the development plan?

- **Track builds and the impact of defects**
  - What build is ready for test?
  - What defects are blocking test from proceeding?
- **Track status with C/ALM dashboard mashups**
  - Transparency across teams and artifacts

# Rational Quality Manager Open Ecosystem Today

**Rational.** software **Tivoli.** software

- Automated Testing
- Rational Functional Tester
  - Rational Performance Tester
  - Rational Service Tester for SOA Quality
  - Rational AppScan Tester Edition
  - Test RealTime
  - Rational Robot
  - Rational Rhapsody

- Requirements
- Rational ReqPro
  - Rational/Telelogic Doors

- Reporting
- Rational Insight

- Builds, WorkItems and Defects
- Rational ClearQuest
  - Rational Team Concert
  - Rational BuildForge

- Provisioning
- Tivoli Provisioning Manager (TPM)
  - Tivoli Service Request Manager
  - TADDM



Supporting bidirectional integration with the Jira change management system



Managing mobile applications testing across a global handset test environment



Quick Test Professional LoadRunner



TMAP Process



Accelerating test cycles with virtual machine management and execution



Enabling greater quality and productivity with automated SOA governance support

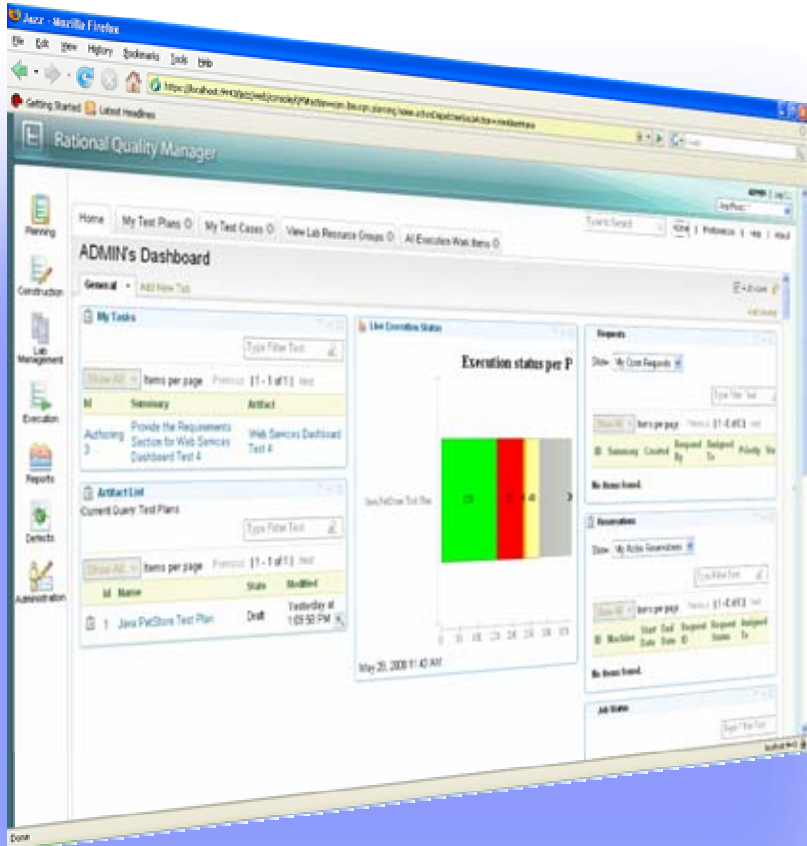


## What else ..... The Grab Bag



- Published API
- Copy Test Assets (Test Plans, Test Cases)
- Support of importing from word/excel
- Scheduled Execution
- Customization of “Catalogue” for test environments
- Bulk Update of Records
- Viewer license capability of WI create/update

# IBM Rational Quality Manager



- **Comprehensive test planning, execution and reporting featuring:**
  - ▶ **Web 2.0 portal architecture**
  - ▶ **Team Collaboration**
  - ▶ **Customizable dashboards**
  - ▶ **Integrated defect tracking**
  - ▶ **Integrated requirements referencing**
  - ▶ **Asset version control**
  - ▶ **Distributed execution**

*Risk Based Testing*

*Dynamic Tree Views*

*Detailed Custom Reporting*

*Jazz Lifecycle Integrations*

*ROI Analysis*

*Productivity Boosters*

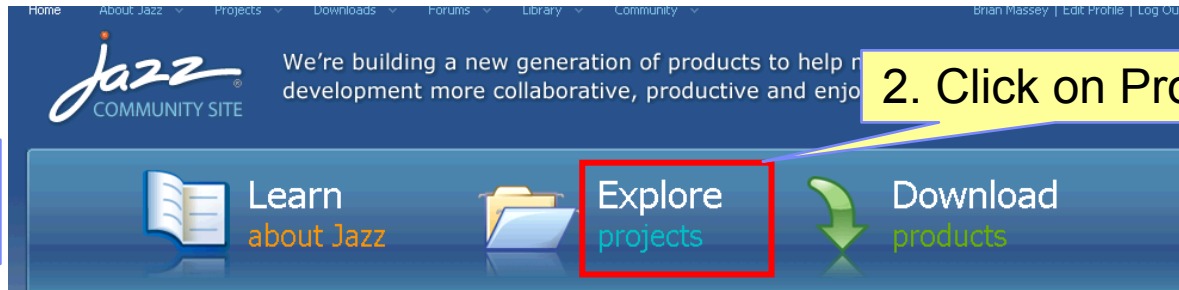
**New in 2.0**

*Open and extensible on*



- ✓ Collaborate in context
- ✓ Right-size governance
- ✓ Day one productivity

# RQM on Jazz.net – Get Involved



**1. Sign-up & sign-in**

Learn more about the capabilities

Want to try it? Log in to download.

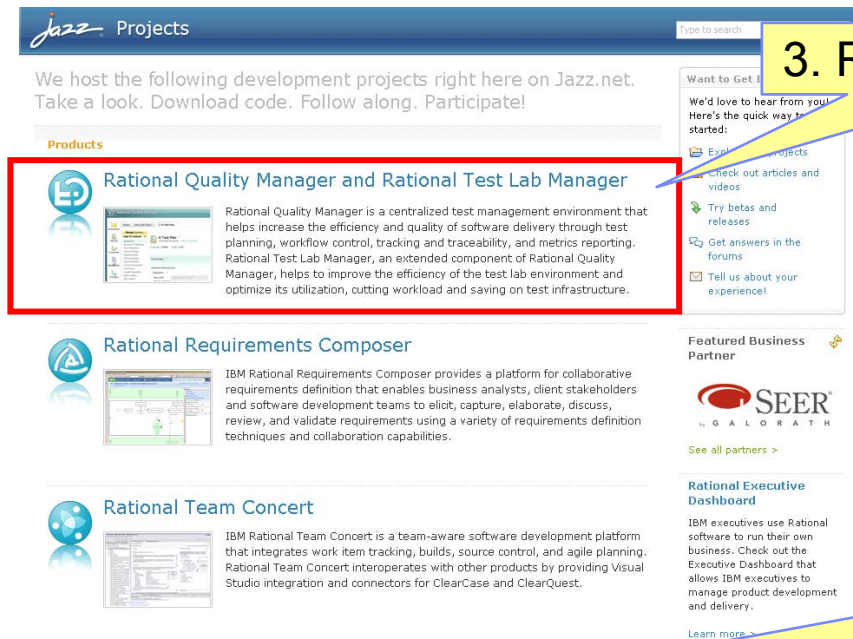
User ID:

Password:

Forgot your User ID or Password?

Not a member yet? [Register now!](#)

Take a look around and read about [Jazz](#) to learn more.



**Participate**

We use our own tools to openly develop this project on jazz.net. Track our progress or get involved:

- [Submit a bug or enhancement request](#)
- [See the project dashboard](#)
- [Explore detailed iteration plans](#)
- [View reports](#)
- [Review release plan](#)

**4. Create Defects, review plans, etc**

# Platform and Technologies

*Delivering functional, service, and performance testing capability across a variety of platforms and technologies*

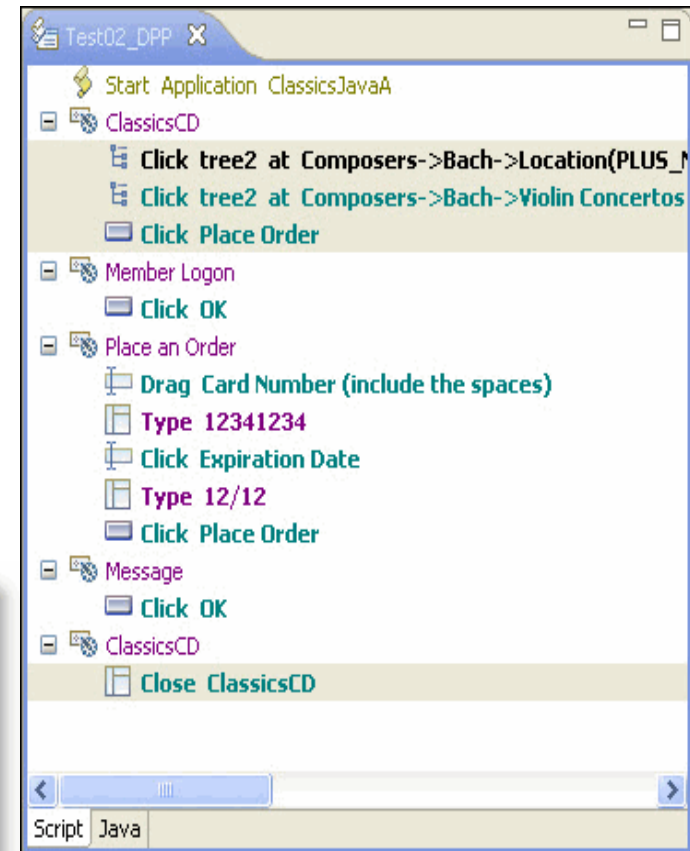


# Storyboard testing with Rational Functional Tester

*Visualize test actions through words and pictures*

## ■ Simplified Test Authoring

- ▶ Edit anywhere: Script, Text, Screen
- ▶ Test flow based on application screenshots
- ▶ Simplified language description of test actions



# Storyboard testing with Rational Functional Tester

The screenshot displays the Rational Functional Tester (RFT) interface. On the left, a 'Script1' tab is active, showing a test script with steps like 'Click tree2 at Composers->Schubert->Location(PLUS\_MINUS)'. Below the script is a 'Properties view' for a 'tree2' control, showing details such as 'Control name: tree2', 'Action: Click Tree Node', and 'atPath: "/>

Script tab

Java tab

Simplified test view

Properties view

Application view

Thumbnails





# Jazz Integration (Rational Quality Manager)

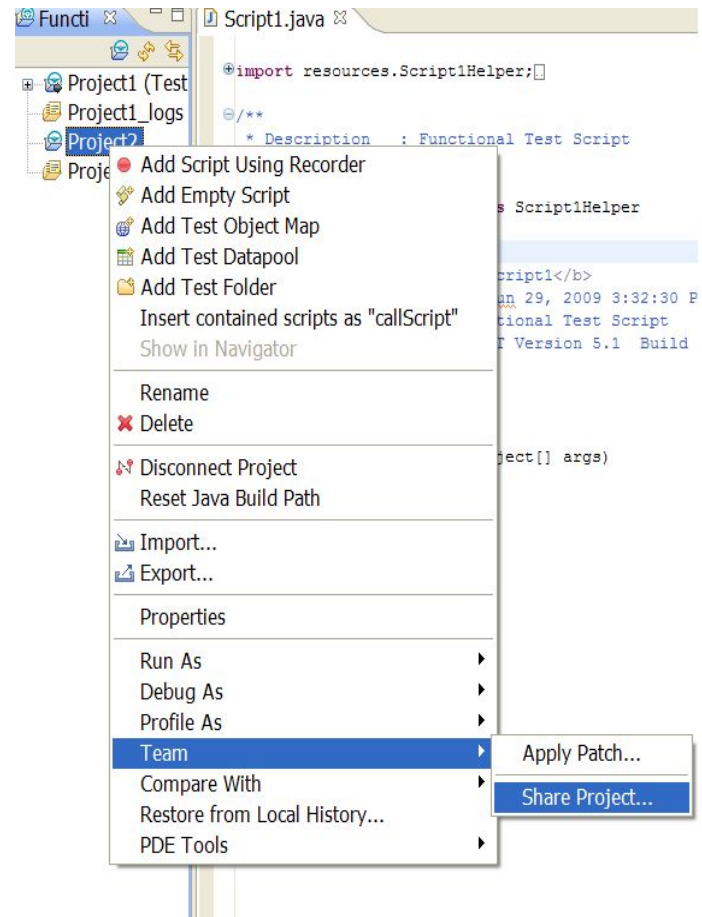
- What's already possible with RFT 8.0
  - Integrates with RQM 1.0
  - View the RQM keywords in RFT
  - Automate the keyword in RFT
  - View the automated keyword in RQM
  - View the automated script in RQM
  - Execute scripts from RQM using the adapter
- Additions in RFT 8.1
  - Integrates with RQM 1.0.1/2.0
  - Multiple Project Area support in the Execution Adapter
    - Adapter is visible only in the project area registered against



# Rational Functional Tester – Rational Team Concert Integration

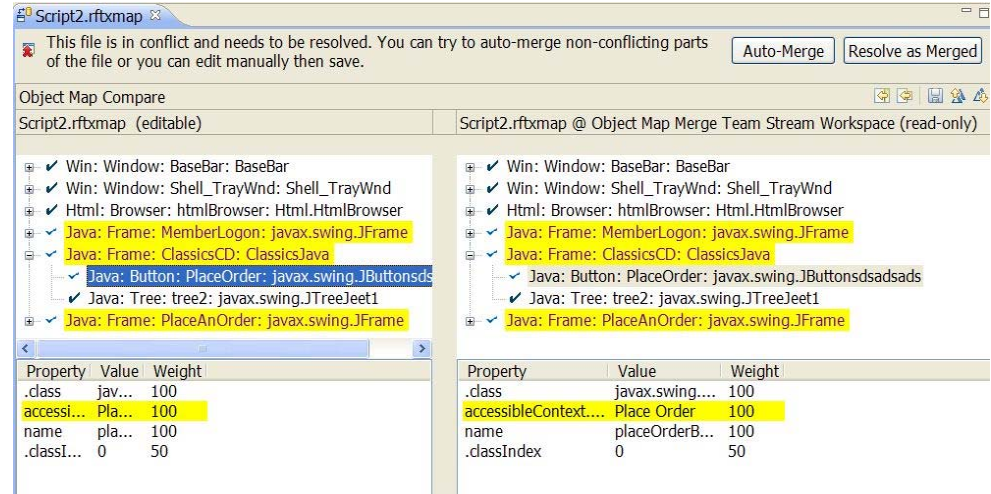
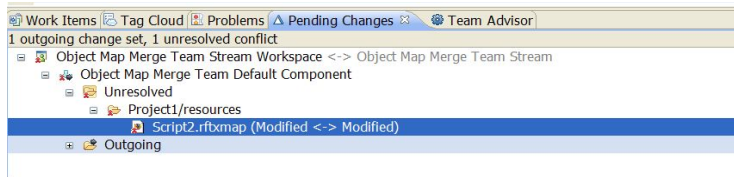
*Leveraging the Jazz foundation for team collaboration*

- Connect to Rational Team Concert/Rational Quality Manager from the Eclipse UI
  - ▶ Access and Update Work Items
  - ▶ Communicate with Instant Messaging
  
- Manage and share assets through RTC SCM integration



# Rational Functional Tester Object Map Merge

- RFT's own Merge Utility for Object Map assets



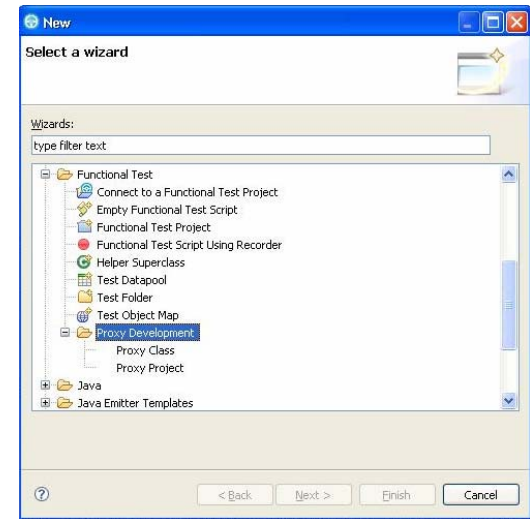
# Extensibility through Proxy SDK

## ■ Proxy Development Wizards

- ▶ Enable partners and power users to develop proxies for custom controls
- ▶ Available in the Eclipse environment
- ▶ Create skeleton/template of the proxy
- ▶ Package and deploy the underlying proxy related files in appropriate directories using export/import facility

## ■ Proxy SDK Documentation

- ▶ Additional complex samples included in documentation
- ▶ Additional samples for VB .NET users



## What else... The Grab Bag

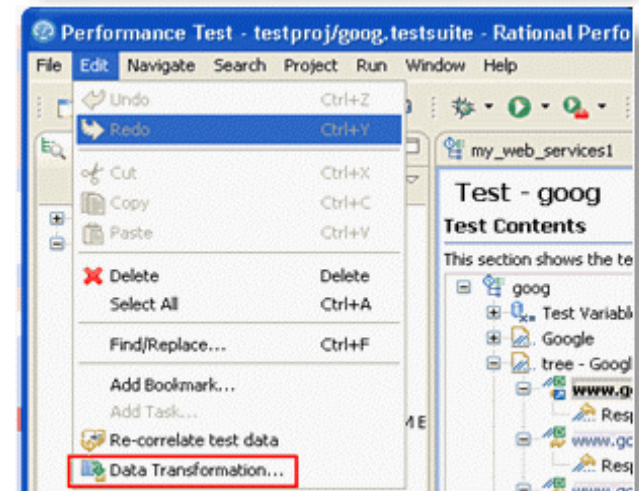
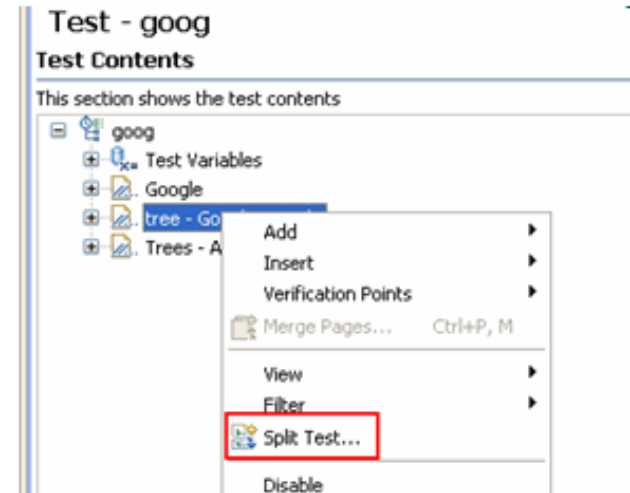
- Internet Explorer 8
  - ▶ Protected Mode
  - ▶ Multi-tab
- Firefox 3
- SAP NetWeaver Enterprise Portal 7.0
  - ▶ Web DynPro
- Dojo 1.1
- Adobe PDF 7.0 and 8.0 Document
- PowerBuilder 11.5
- Flex 3.2
- WPF
- 64 – bit Windows Operating Systems
- RQM Integration
  - ▶ Adapter
  - ▶ Keywords



# Rational Performance Tester

## *Simplifying the test development experience*

- Test Modularity with test variable support
- Binary Editor view supporting UTF8, EBCDIC, GB 18030
- Replay of http live-browser view
- Custom/Conditional color pallets for reports



# Rational Performance Tester

## *Encoded Data support for Web 2.0 applications*

The typical scenario ...



- Encoding prevents the user from finding or modifying test data points
- Users must create their own code and methods for dealing with obscured data (such as binary data)

With IBM Rational Performance Tester ...

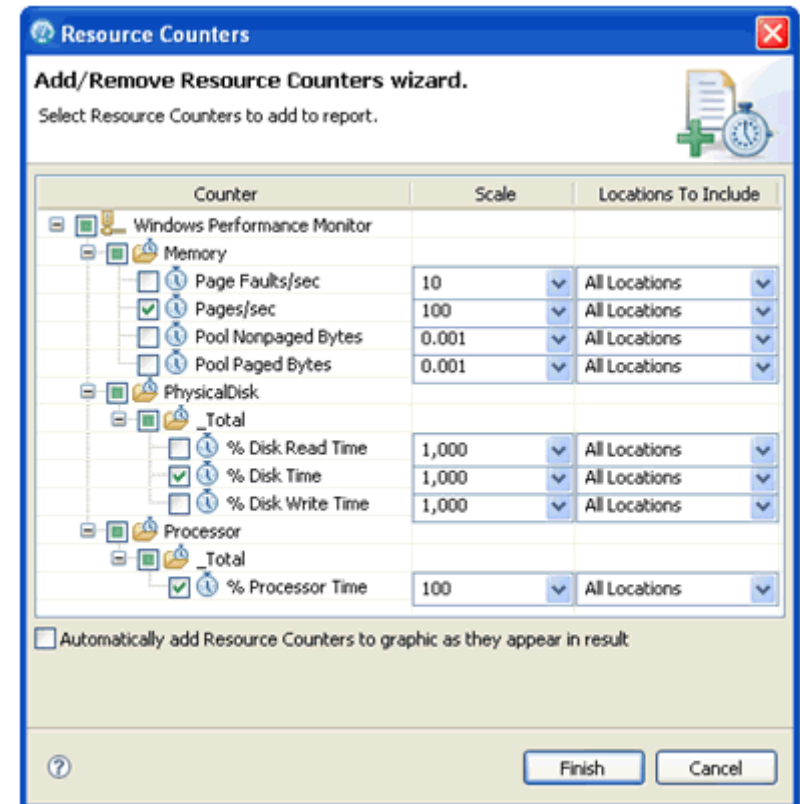


- Data transformations allows user to “see” into encoded data for editing, data variation, and data correlation
- Built-in transformations for Java Serialized Objects and Binary data
- Extensible to accommodate custom formats

# Rational Performance Tester

*Accelerate problem identification with agent-less resource monitoring*

- Agent-less monitoring for
  - ▶ **\*NEW\*** WebSphere and JBoss application servers
  - ▶ Windows Perfmon
  - ▶ Unix rstatd
  - ▶ IBM Tivoli Monitoring
- Aggregated (per-run) counters for resource monitoring
- Overlay counters on performance reports

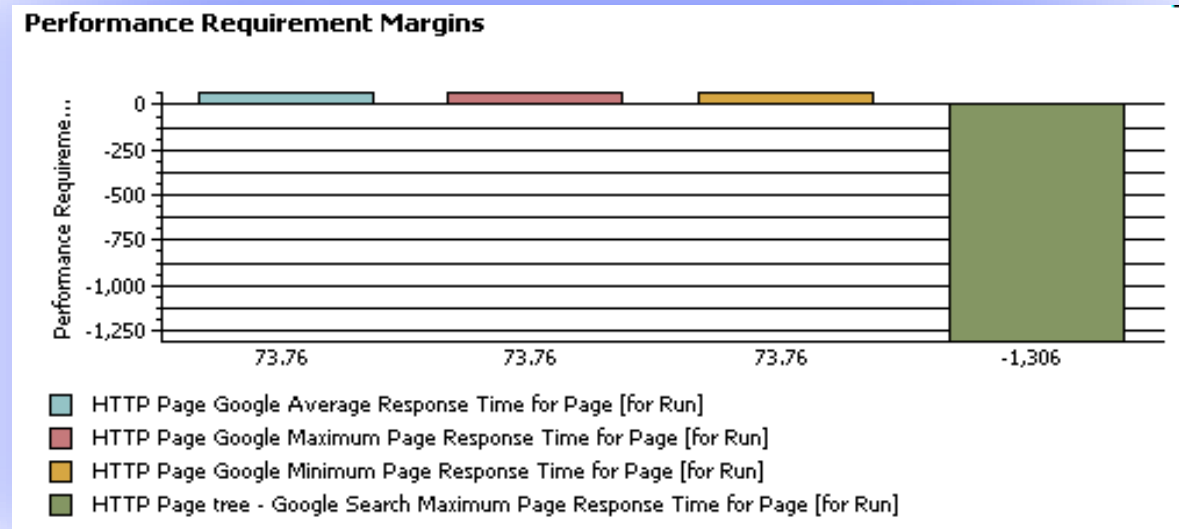
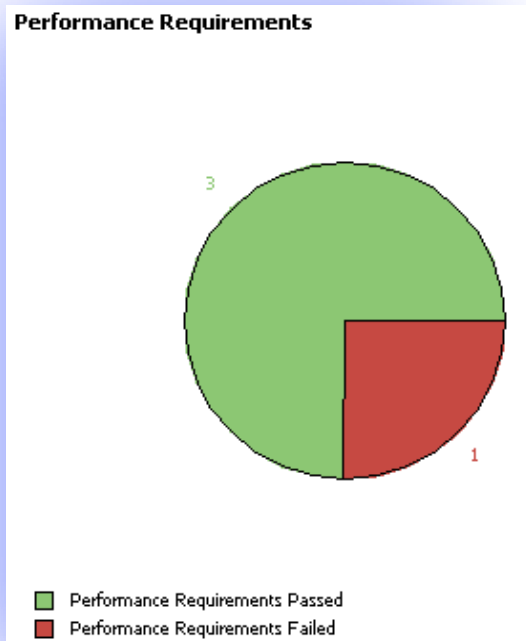




# Rational Performance Tester: Service Level Agreement Reporting

## *Linking performance results to business objectives*

- New reports summarize test run against performance requirements
- “Supplemental” requirements can be defined and reported separately
- Concept of “margin” helps visualize the quality of the results



# Rational Service Tester for SOA Quality

*Simplifying service testing*

## Service Tester for SOA Quality and SOA Extension enhancements

- ▶ Support for DIME attachments
- ▶ Support for Kerberos and NTLM Authenticated proxies
- ▶ Support for HEAD method
- ▶ XML fragment reference/substitution

The screenshot displays the Rational Service Tester interface. At the top, a call definition is shown: `* createAccount("John", "Doe", "200 Ocean St...", "33902", "Miami Beach")`. Below this, a workflow is visible with buttons for **Edit Data**, **Invoke**, and **View Response**. The main area shows a **Message** tab with various options like **Attachments**, **Transport**, **Security for Request**, and **Security for Response**. A **Form** tab is active, displaying a **Forms-based data** entry screen for the `createAccount` operation. The form includes input fields for `FirstName` (John), `LastName` (Doe), `StreetAddress` (200 Ocean Street), `ZipCode` (33902), and `City` (Miami Beach). A **Generic Service Client** section is also visible, providing instructions on how to invoke a service call, such as adding a WSDL file or an endpoint.

**Test creation wizard**

**Forms-based data**

**Generic Service Client**

**Common client for service calls**

To invoke a service call, perform one of the following tasks:

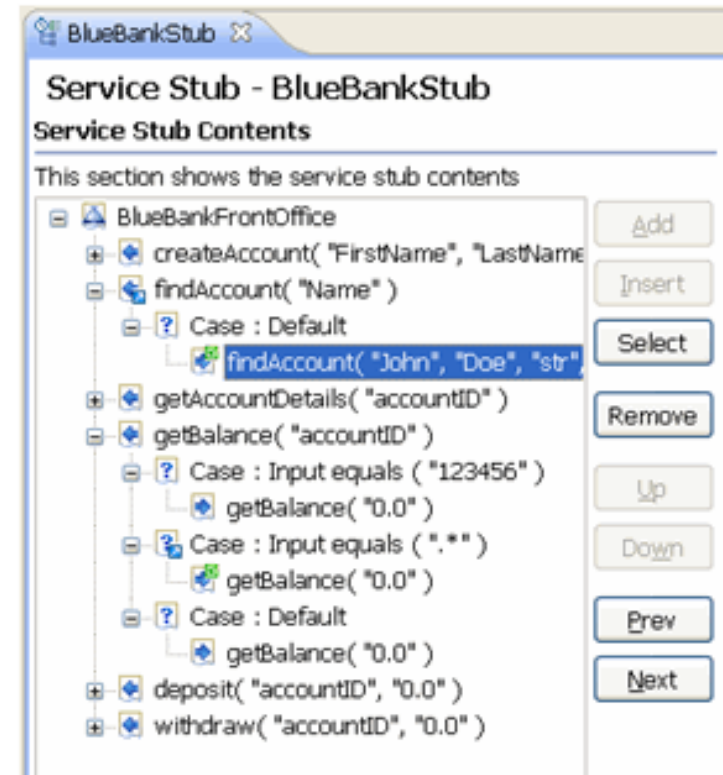
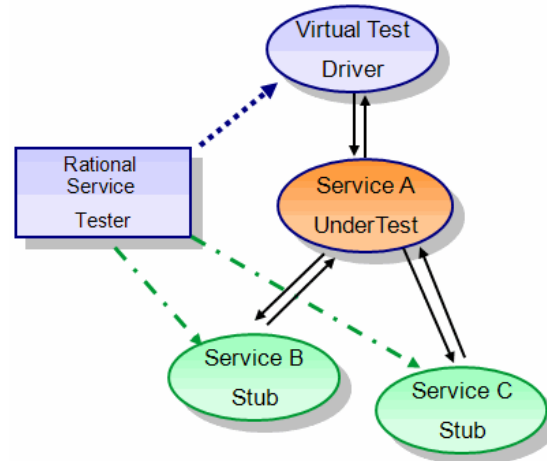
- ▶ add a **WSDL** file for a Web service call and then select the port of the call,
- ▶ add an endpoint for an **HTTP**, a **JMS** or a **WebSphere MQ** call.

You can also select an existing call in the Call Library or Call History sections.

# Rational Service Tester for SOA Quality

*Enabling agile and incremental testing with web service stubbing*

- Automatically create service stubs from WSDL
- Visually edit stub behavior
- Automatically deploy stubs for testing



# The Road Ahead

*A peek into the Quality Management Labs*

## ■ Research & Development

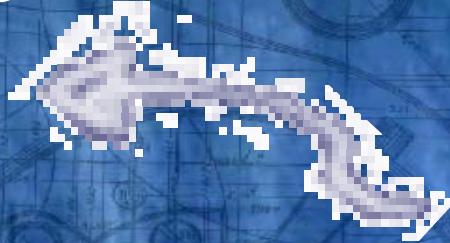
- ▶ SAP Integrations
- ▶ Oracle Solution integrations
- ▶ Centralized Data Management
- ▶ Data Integrity

## ■ Cloud Computing



# Quality Management Keynote

- Real Challenges for real Teams
- Real Results
- Real Insights
- Wrap Up



*fig. 2.*



*fig. 3.*



*fig. 4.*



*fig. 5.*



Material	Quantity	Price	Total
Balance	100	1.50	150.00
Material	50	1.50	75.00
Cost of Material	50	1.50	75.00
3rd	50	1.50	75.00
4th	50	1.50	75.00
Cash	100	1.50	150.00
Material	50	1.50	75.00
3rd	50	1.50	75.00
4th	50	1.50	75.00
Total	100	1.50	150.00
Cash	100	1.50	150.00
3rd	50	1.50	75.00
4th	50	1.50	75.00
Cash	100	1.50	150.00
Material	50	1.50	75.00
3rd	50	1.50	75.00
4th	50	1.50	75.00
Total	100	1.50	150.00

*(G. E. Delany)*

## MORE! MORE! MORE!

- Tons of Content:
  - Deep Dive Sessions
  - Birds of a Feather Sessions
  - Hands on Workshops
- Demo booths
- Customer Quality Initiatives and Scenario Inspection Booth

But Wait...  
**THERE'S  
MORE!**

AS SEEN AT  
**RSC**