



IBM Rational Software Conference 2009
As Real as It Gets!



Quality Management

Real Teams, Real Insights, Real Results

Richard Crisp
Director, Requirements and Quality Management

Rational. software

Quality Management

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



fig. 3.

fig. 4.

fig. 5.

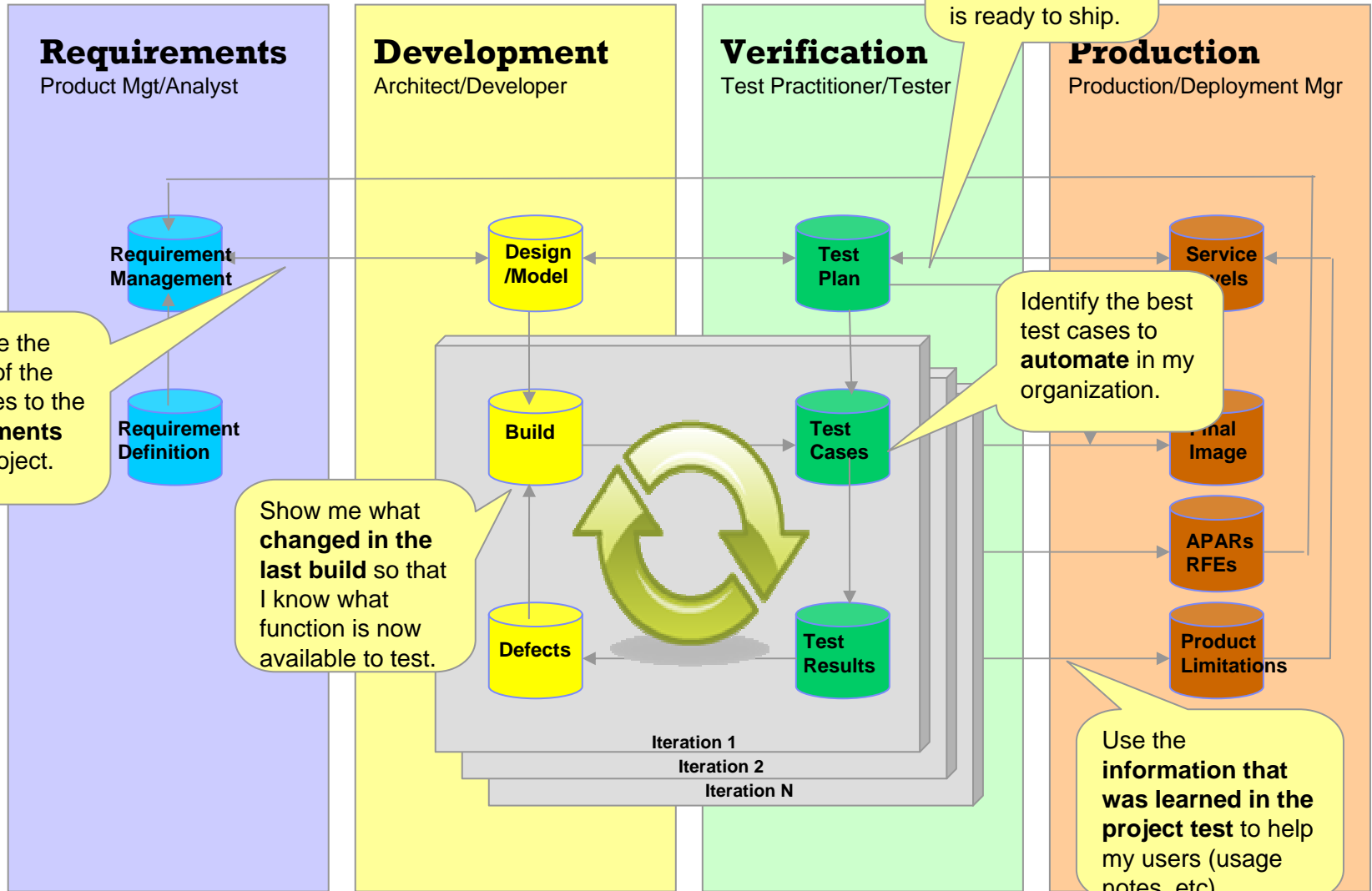
Material	Quantity	Price	Total
Balance Sheet	100	1.00	100.00
Capital Stock	50	2.00	100.00
3rd	20	5.00	100.00
4th	10	10.00	100.00
Cash	10	10.00	100.00
Notes	10	10.00	100.00
Debt	10	10.00	100.00
Profit	10	10.00	100.00
Interest	10	10.00	100.00
Income	10	10.00	100.00
Expenses	10	10.00	100.00
Net Income	10	10.00	100.00
Dividends	10	10.00	100.00
Reserves	10	10.00	100.00
Retained Earnings	10	10.00	100.00
Equity	10	10.00	100.00
Liabilities	10	10.00	100.00
Assets	10	10.00	100.00

(G. P. Delany)

Today's Challenges to Software Quality

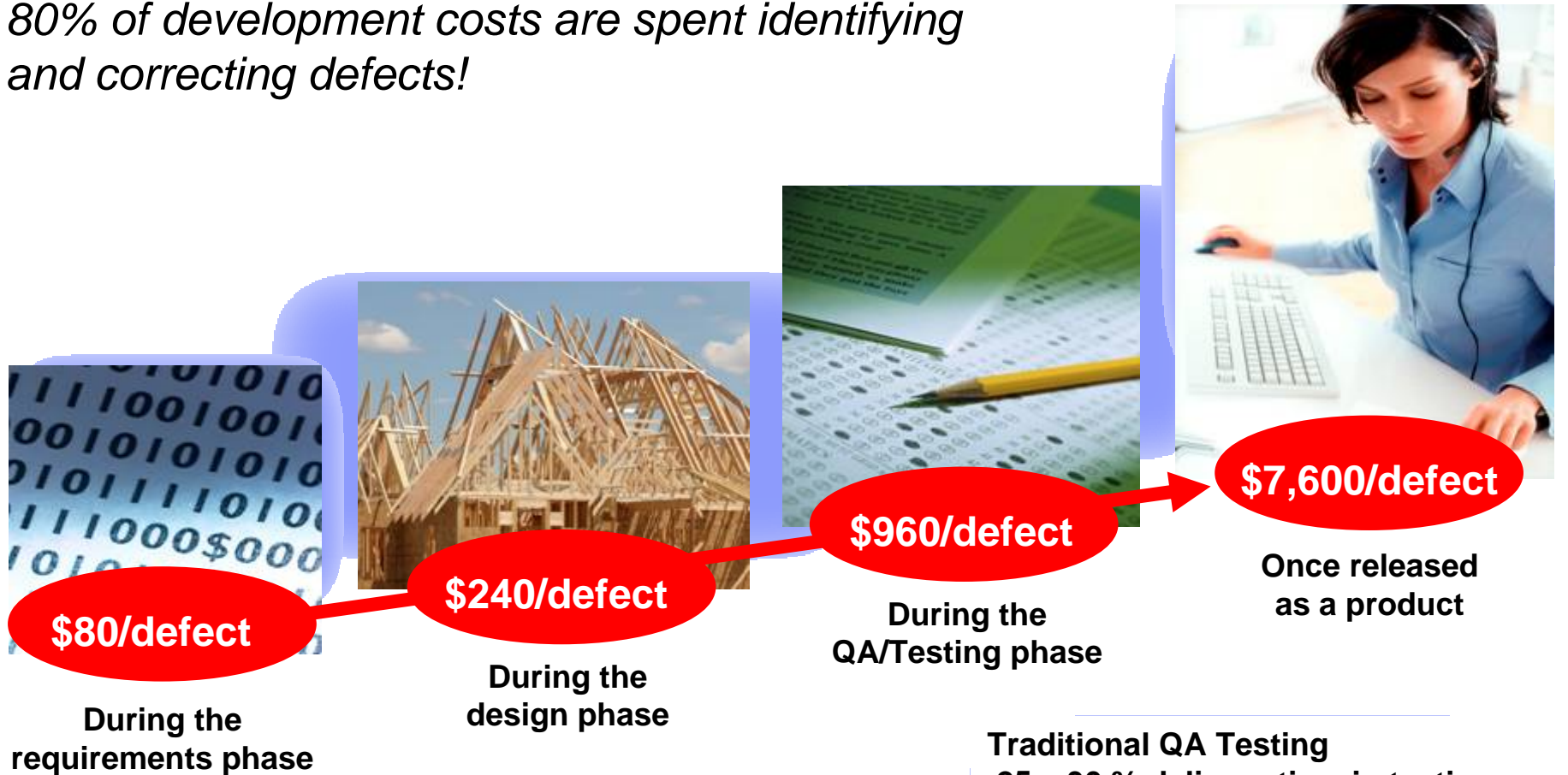
- Enterprise pressure to **reduce cost**
- Competitive demand to **deliver faster**
- Worldwide geographical **dispersed development**
- **Unpredictable/unknown use** of application
- Increasingly **complex software** ecosystems
- New requirements for **government standards**
- Lack of **domain knowledge/content**
- 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



The increasing costs of fixing a defect

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



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

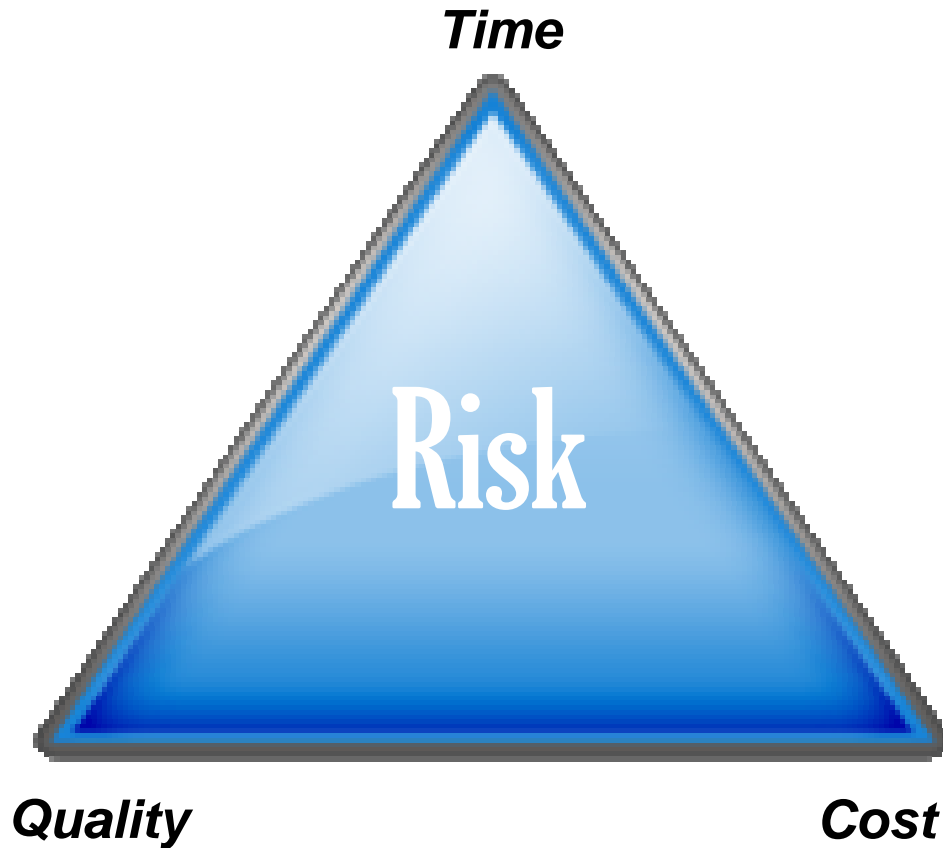
Rain sensing wiper system design failure

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**

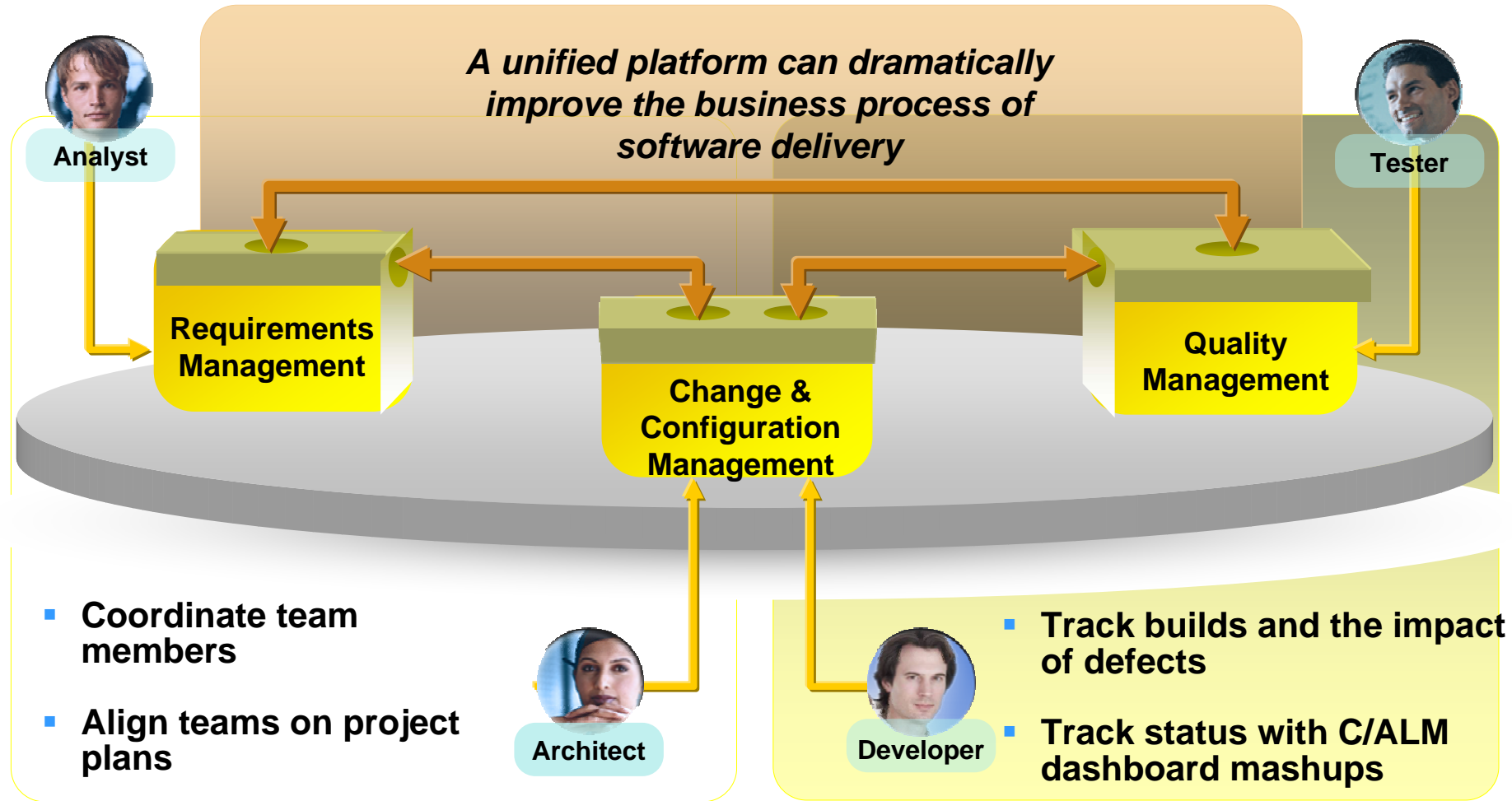
Risk cannot be removed

**But it can be understood
and controlled**

With IBM Rational Quality Management...

Unified platform for software delivery

Moving beyond individual tools

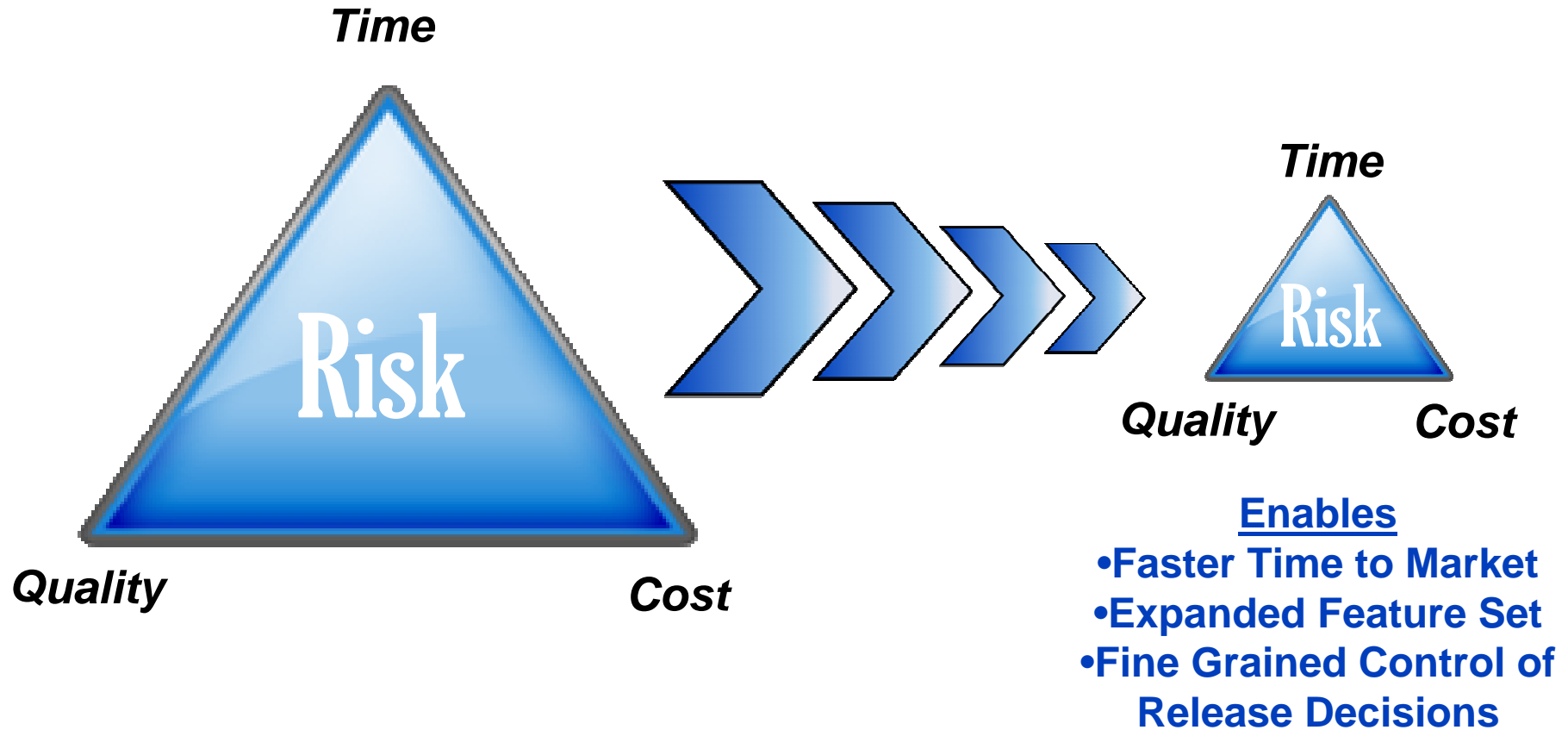


Risk Management through Process Improvement



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

Risk Management through Process Improvement



With IBM Rational Quality Management...

Quality Management

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



fig. 2.

fig. 3.

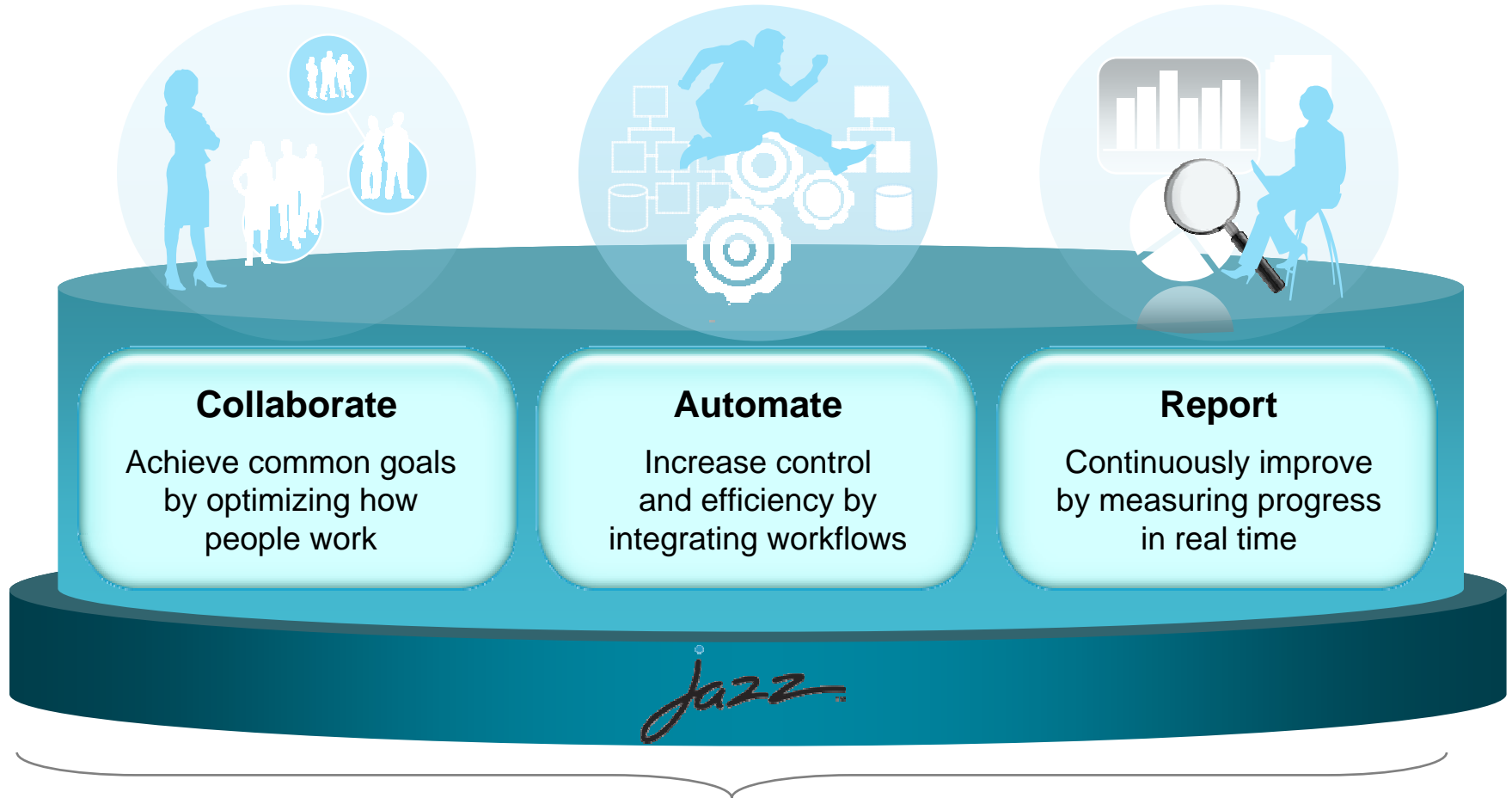
fig. 4.

fig. 5.

	Time	Value
Balance Sheet	30	1.500
Capital Stock	50	2.500
3rd	20	1.000
2nd	10	500
1st	5	250
Capital	100	5.000
Reserve	50	2.500
Profit	10	500
Loss	5	250
Interest	10	500
Other	10	500
Total	200	10.000
Capital	100	5.000
2nd	10	500
1st	5	250
Profit	10	500
Loss	5	250
Interest	10	500
Other	10	500
Total	200	10.000

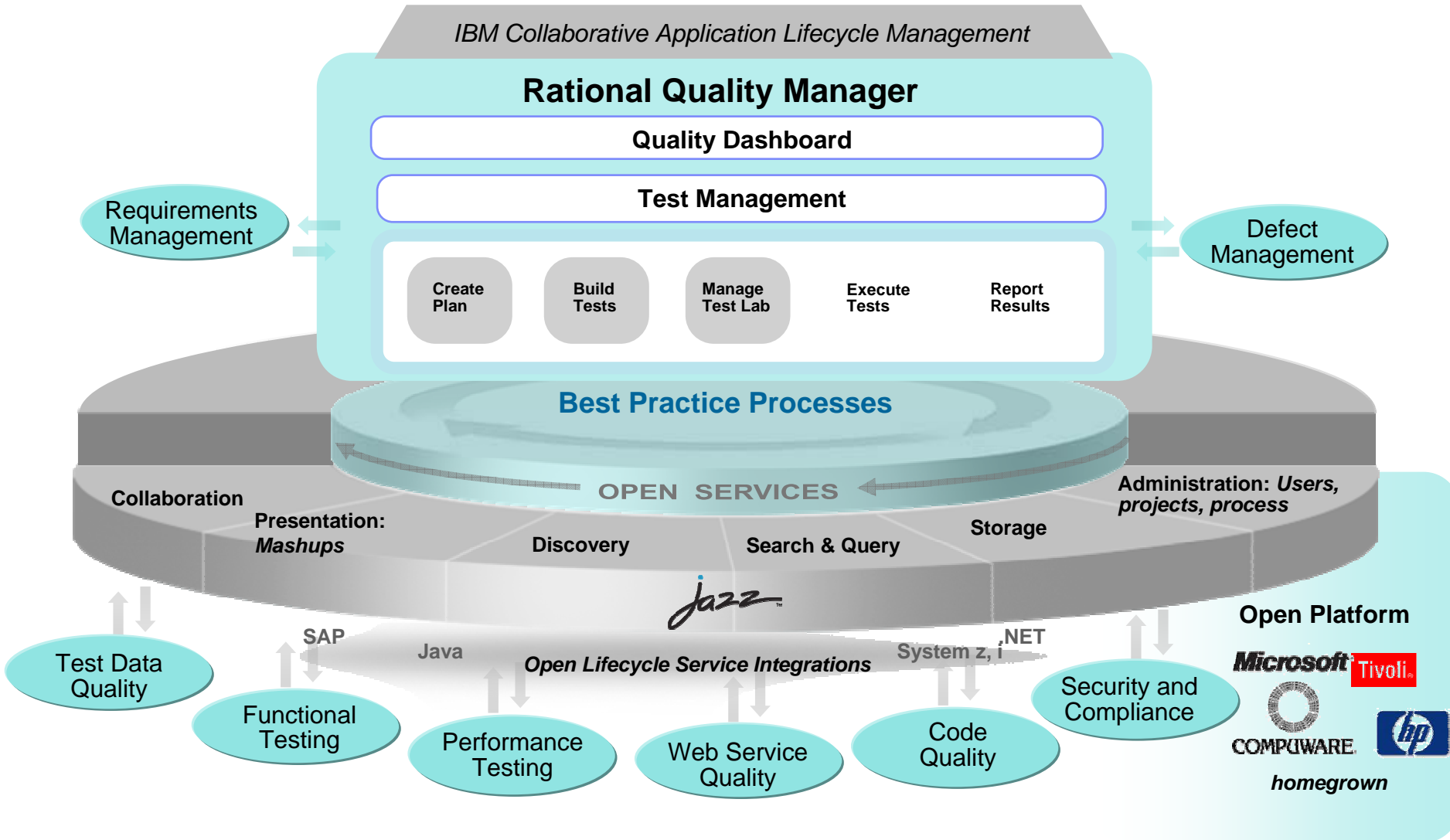
(G. P. Delany)

Quality Management Imperatives



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

Quality Management Hub



Rational Quality Manager 2.0

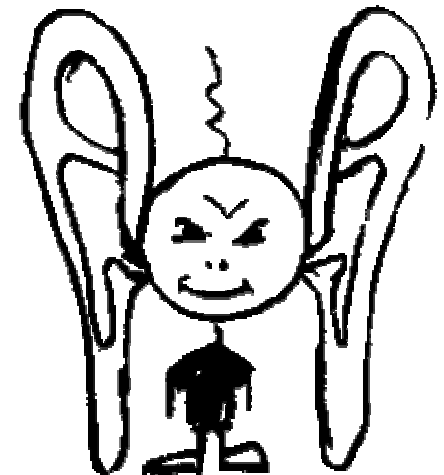
■ We heard you -- feedback concerning 1.0/1.0.1 of RQM

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



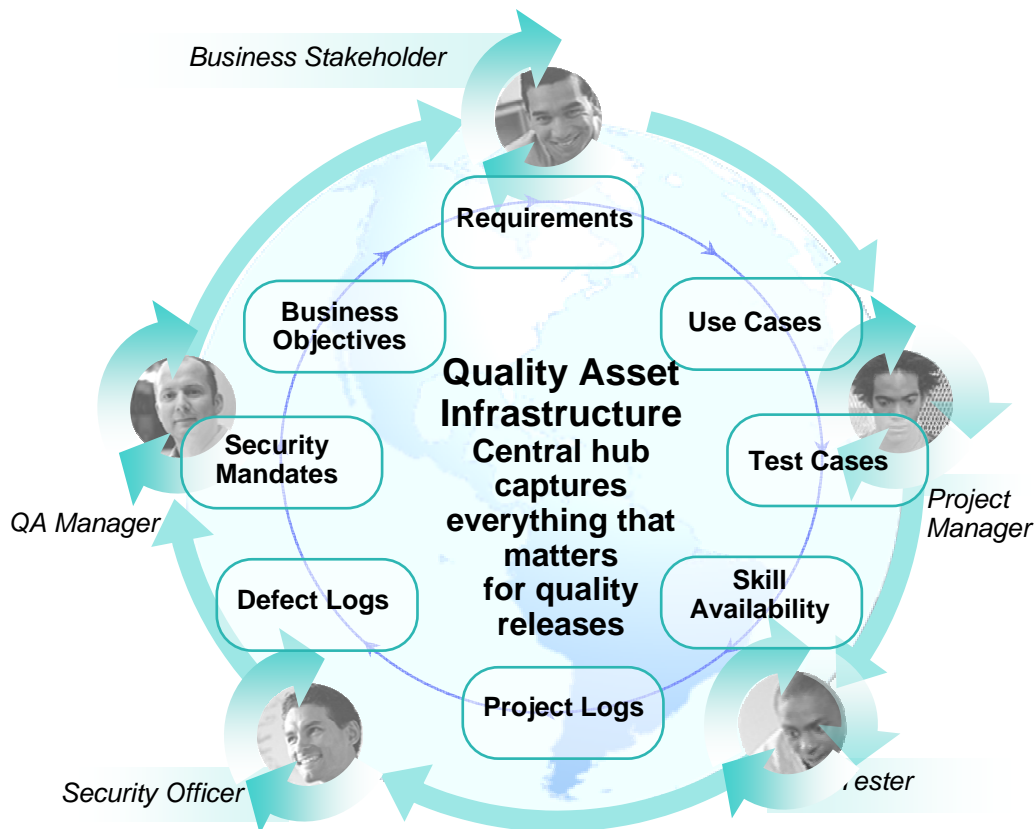
■ Problems to solve

- ▶ 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

Rational Quality Manager Plan

- ✓ 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

Integrated manual test authoring and execution














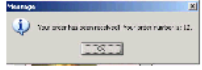


Maximizing efficiency of manual testing

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

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.

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

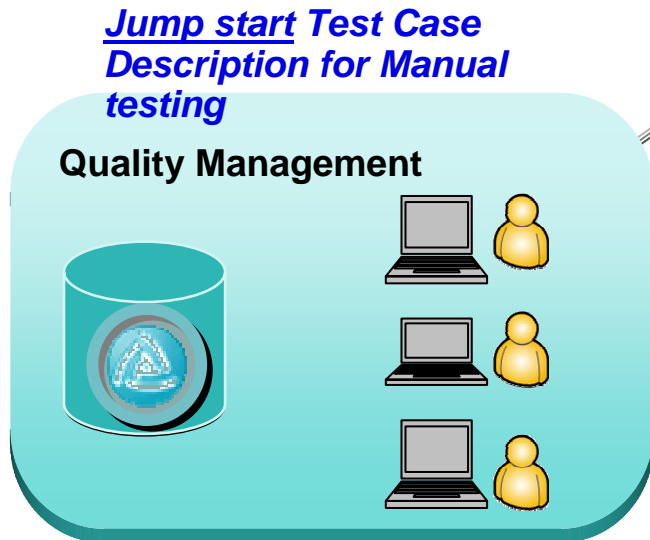
- Prioritize efforts where resources are limited
- Drive risk based decisions to meet quality objectives
- Allows for collaboration on risk mitigation strategy
- Documentation of risk related decisions

Prioritized Tests to meet business needs

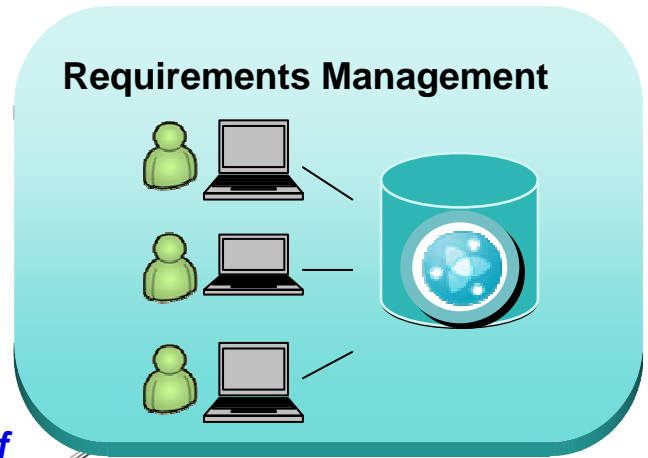
Productivity Boosters

Smart Automations...

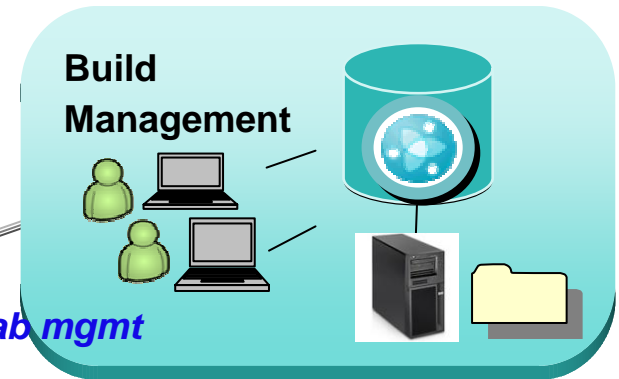
- Increase productivity
- Reduce risk of human error
- Reduce time to quality
- Reduce cost...



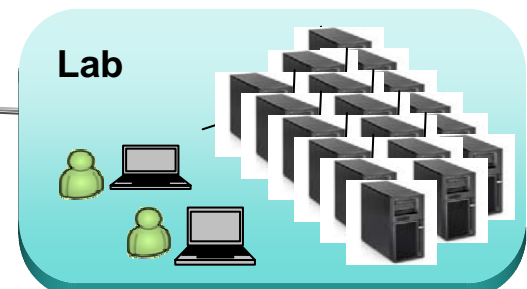
Jump Start creation of Stubbed test cases from requirements



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	

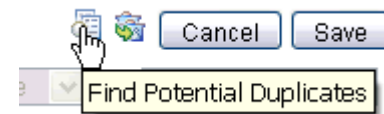
Select Quality Objectives ?

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

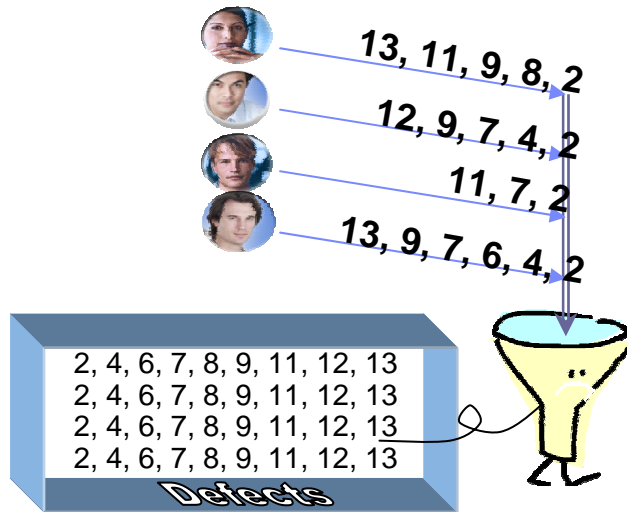
OK Cancel

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

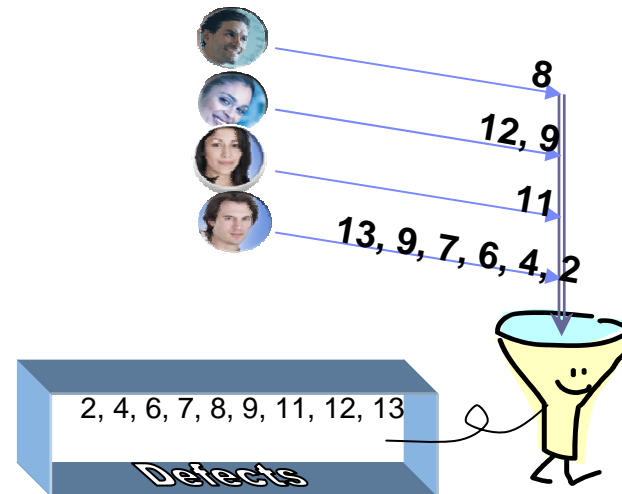
- How long does it take to find and resolve a single duplicate defect?
 - ▶ Are development teams working on same issue unknowingly
 - ▶ Are multiple fixes for the same problem put in different builds
 - ▶ Are testing members validating
- Time is money.....



Common scenario



Utopia

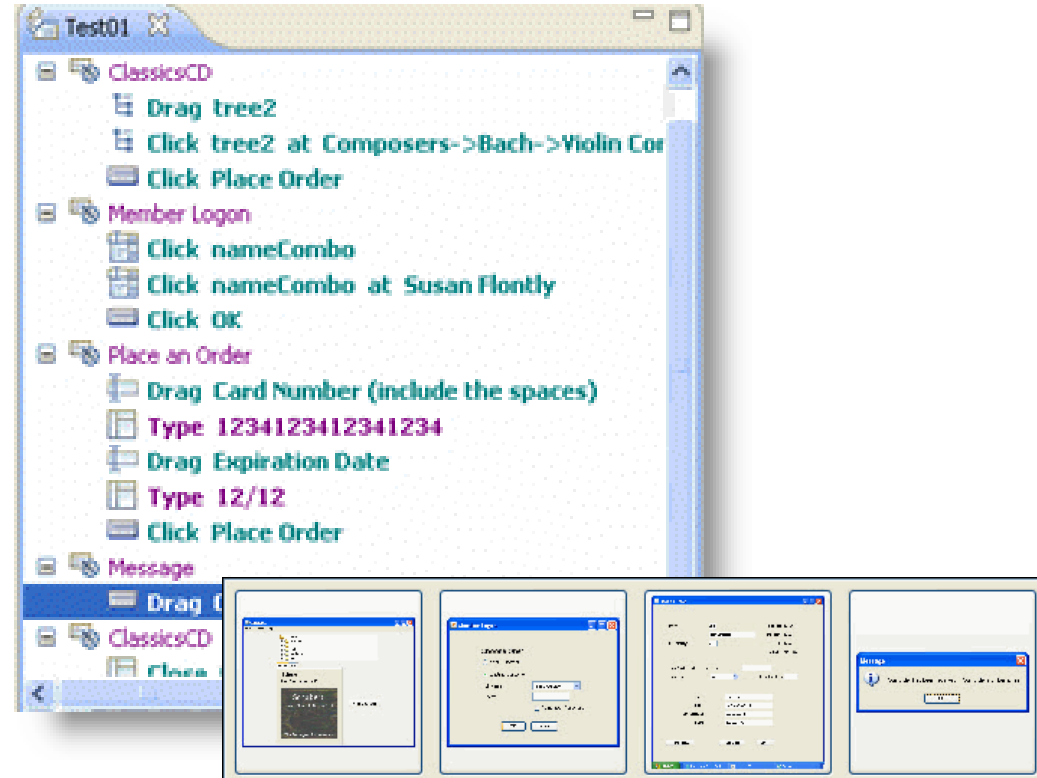


Integrated Functional and Regression test execution

Challenge: Testing can't keep pace with rapidly changing software

Solution: Automated 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



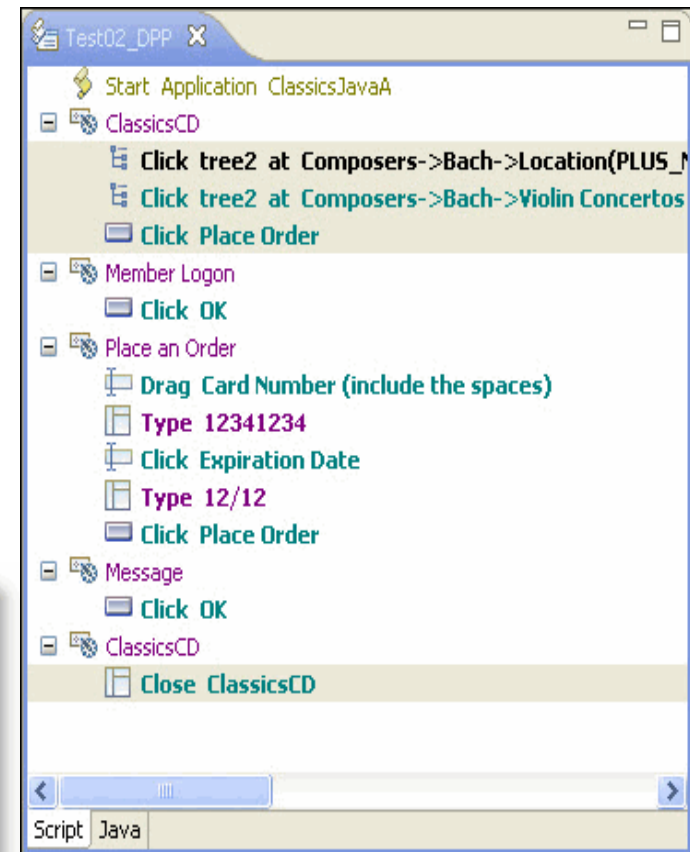
Accelerate test execution and deepen test coverage through automated test execution

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, the 'Script1' tab is active, showing a test script with steps such as 'Click tree2 at Composers->Schubert->Location(PLUS_MINUS)', 'Click tree2 at Composers->Schubert->Symphonies Nos. 5 & 9', and 'Click Place Order'. A 'Simplified test view' callout points to this script. Below the script, the 'Properties view' shows details for a 'tree2' control, including its 'Action' (Click Tree Node) and 'atPath' ('Composers->Schubert->Symphonies Nos. 5 & 9'). A 'Script tab' callout points to the script editor, and a 'Java tab' callout points to the Properties view. The main area shows the 'Application view' of the 'ClassicsCD' application, which includes a tree view of composers (Schubert, Haydn, Bach, Beethoven, Mozart) and an album detail for Schubert's 'String Quartets Nos. 4 & 14'. A 'Place Order' button is visible. A 'Thumbnail' callout points to a row of five small application view thumbnails at the bottom. The interface also includes a 'Script Explorer' and 'Keyword View' at the top.

Script tab

Java tab

Simplified test view

Properties view

Application view

Thumbnail

Integrated Performance test execution

Challenge: Application performance and scalability limitations present high risks

Solution: Automated performance verification before systems go live

1. Capture and track Business SLAs in your test plans
2. Create realistic user workloads that exercise the key business transactions
3. Deploy the environment and schedule execution of your workload
4. Identify the root cause of performance problems
5. Communicate results and areas of non-compliance to the stakeholders

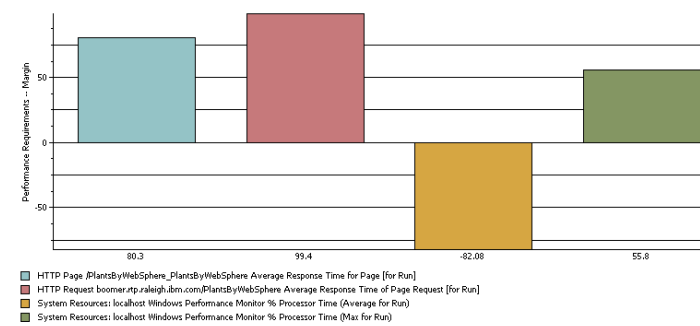
Status Summary

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

Performance Requirement Margins



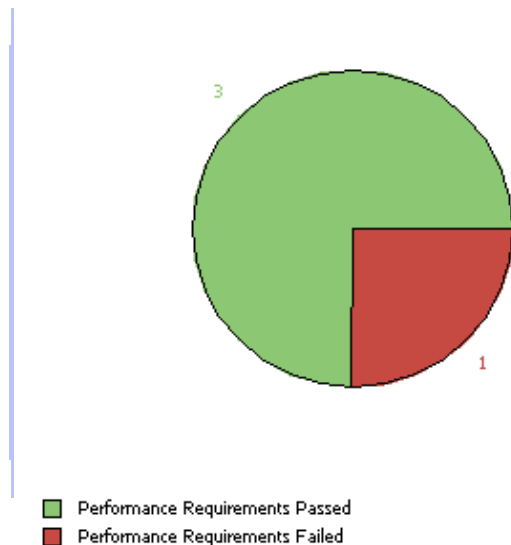
Comprehensive test planning, authoring, and analysis to identify and manage the risk of application performance failures

Rational Performance Tester: Service Level Agreement Reporting

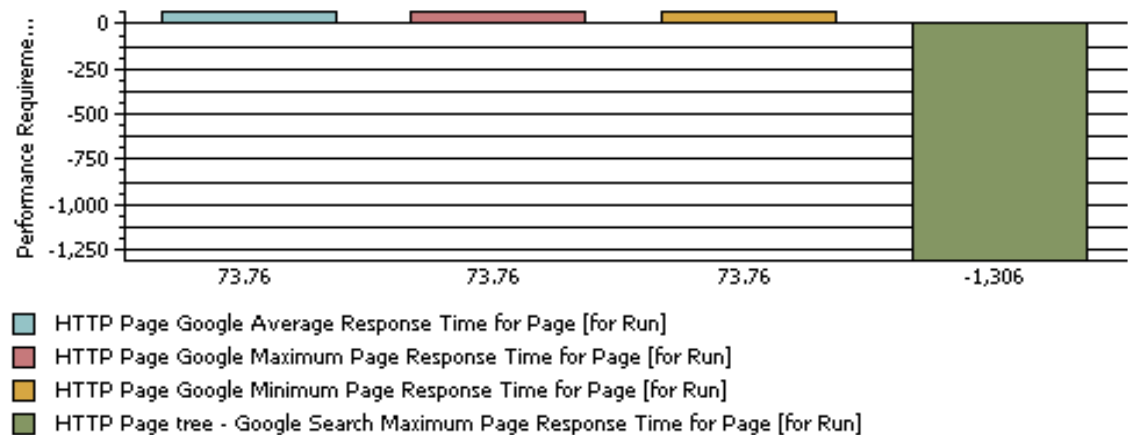
Linking performance results to business objectives

- 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

Performance Requirements



Performance Requirement Margins



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



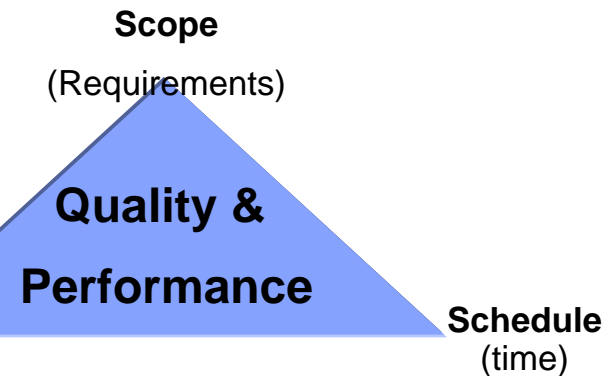
- **Tests based on requirements ensure deliverables meet customer expectations**
 - ▶ Plan Tests Early
 - ▶ Conduct Tests Early
 - ▶ Relate Tests to Requirements
 - ▶ Relate Defects to Requirements
 - ▶ Measure Progress against Requirements

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



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

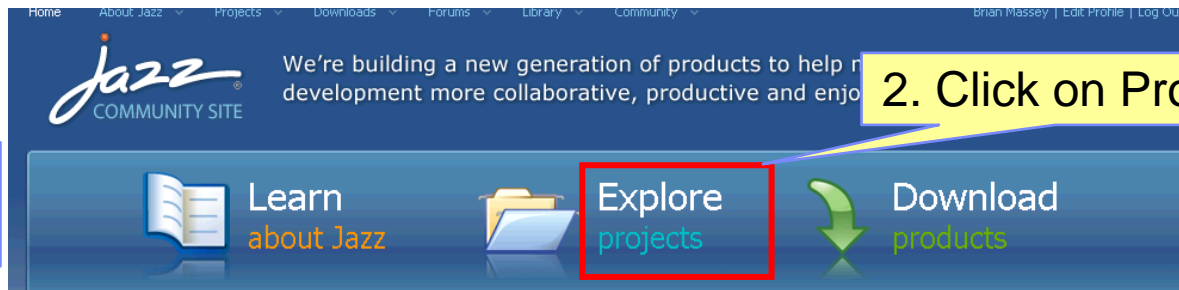


Accelerating test cycles with virtual machine management and execution



Enabling greater quality and productivity with automated SOA governance support

RQM on Jazz.net – Get Involved



1. Sign-up & sign-in

Learn more about the capabilities

Want to try it? Log in to download.

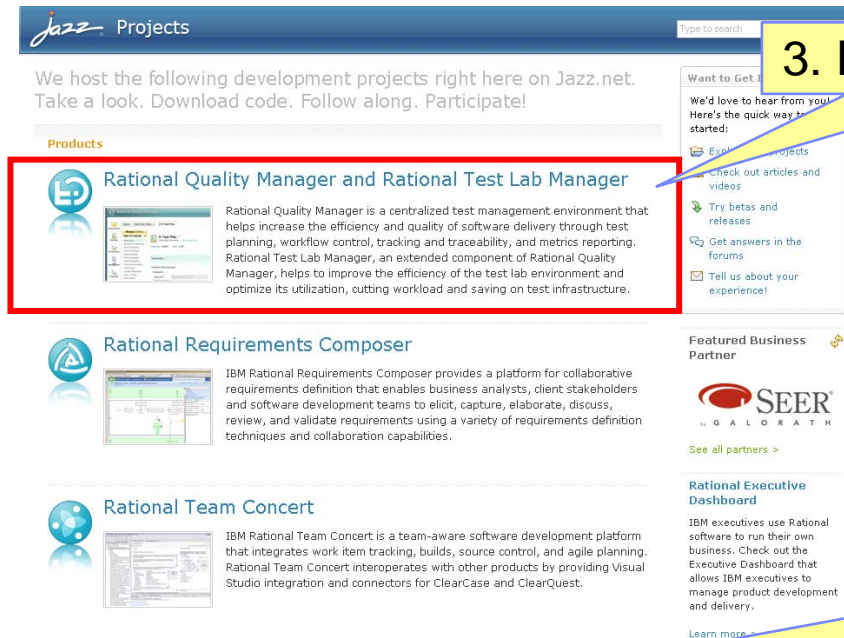
User ID:

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

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

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

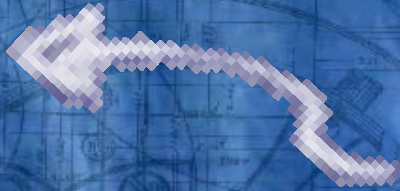


fig. 2.



fig. 3.



fig. 4.



fig. 5.



Distance	Time	Speed
1000	10	100
2000	20	100
3000	30	100
4000	40	100
5000	50	100
6000	60	100
7000	70	100
8000	80	100
9000	90	100
10000	100	100
11000	110	100
12000	120	100
13000	130	100
14000	140	100
15000	150	100
16000	160	100
17000	170	100
18000	180	100
19000	190	100
20000	200	100

(G. P. Delany)