



IBM Software Group

# IBM Rational Requirements Composer CLM 2012

*Better, faster business outcomes  
through better requirements practices*

*Raghu P S*

*Brand Architect*

**Rational** software



# Agenda

- Part 1: Why good requirements practices matter
- Part 2: Achieving better business outcomes with Rational Requirements Composer

## Challenges to the effective delivery of software

*A history of cost overruns, schedule slips and quality issues*



Only **22%** of executives felt that their IT and business strategy were tightly integrated



**30%** of project costs associated with rework



**49%** of budgets suffer overruns and **62%** fail to meet their schedules

**Focus on  
business impact**

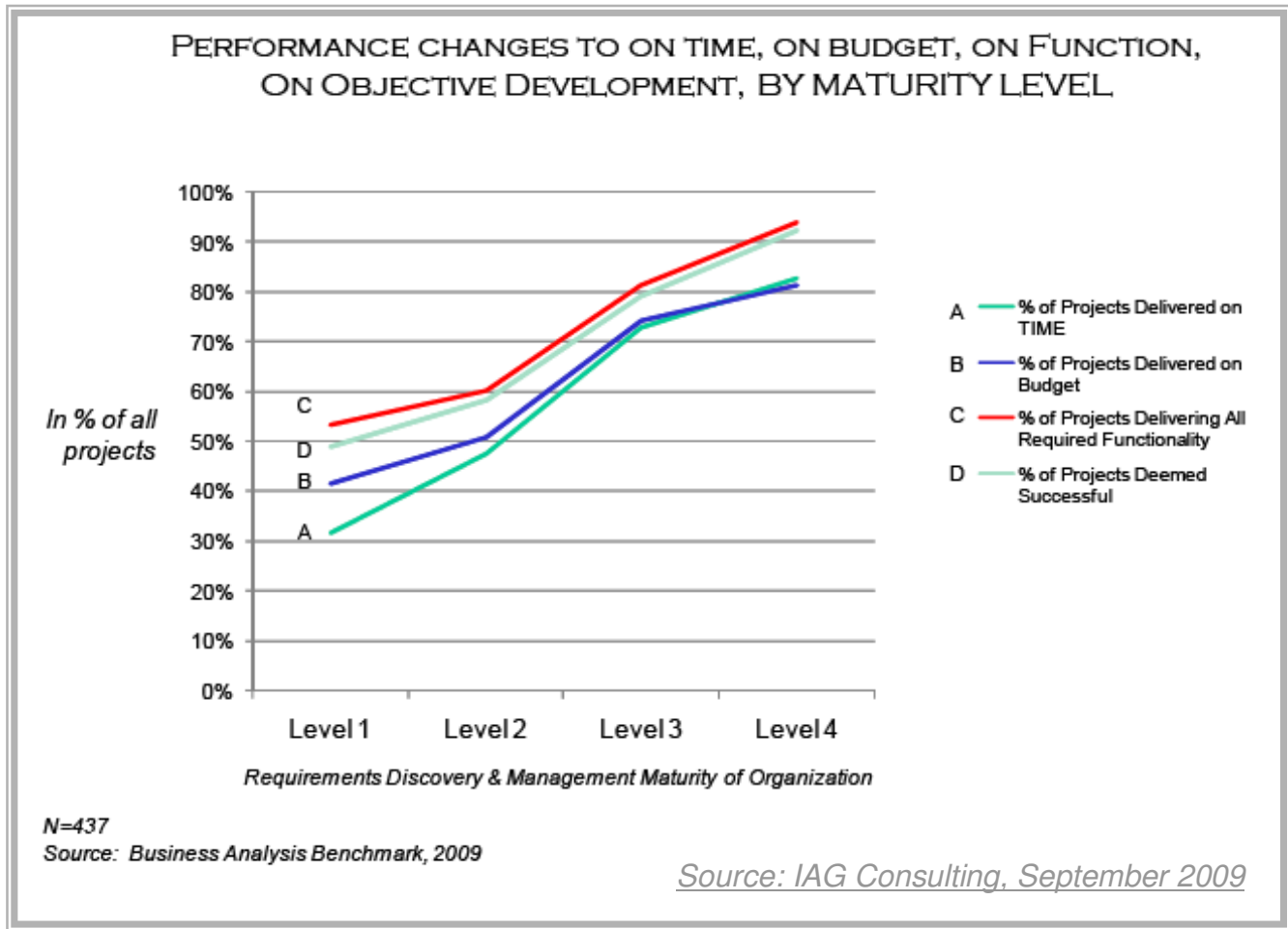
*CIOs are trying to find the balance between the need to reduce costs, improve services and deliver innovation*

– IBM CIO Survey, 2009



# Good requirements are key to project success

## Reclaim up to a third of project budget and schedule

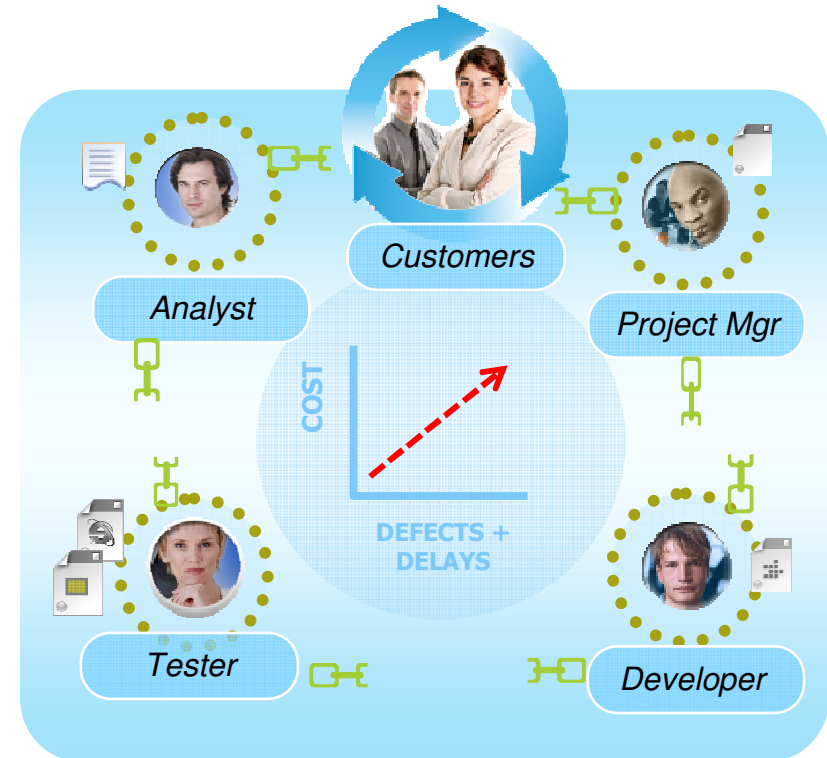


- Average on time performance **increased 161%**
- Time overruns were **reduced 87%**
- Average on budget performance **improved over 95%**
- Budget overruns **reduced about 75%**
- Percentage of projects that deliver the functionality needed by the business **rose by about 75%**

## Symptoms of requirements issues

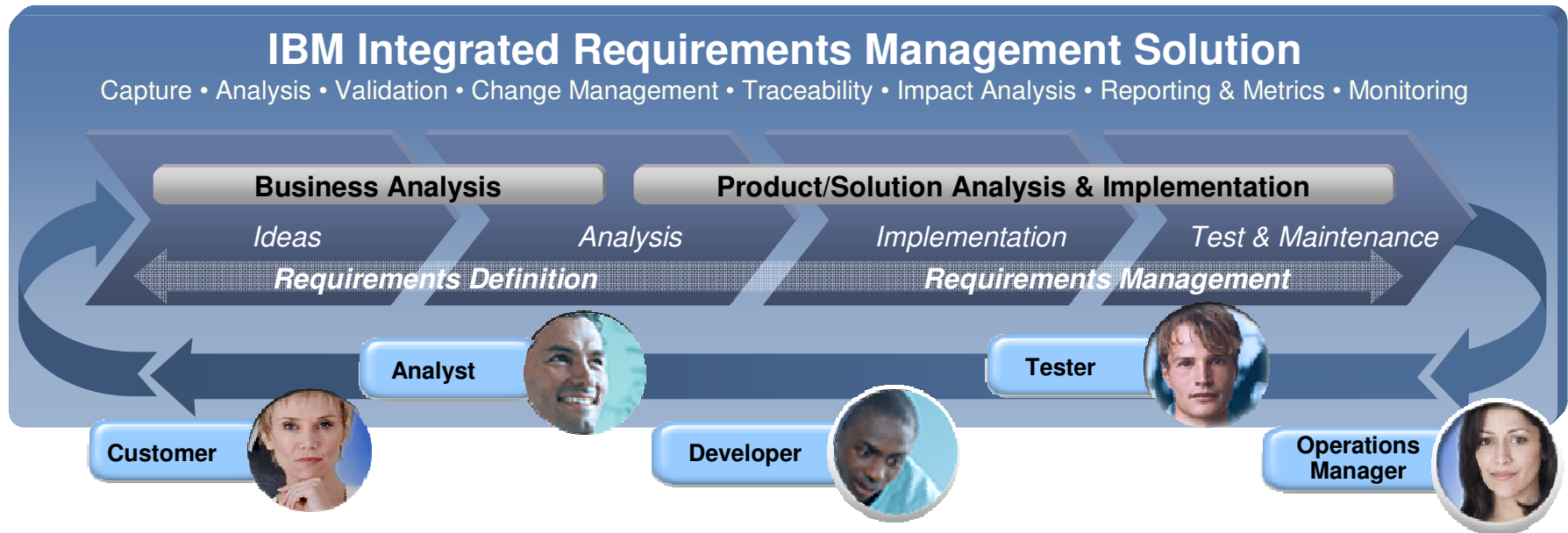
### *Suggesting a need to invest in requirements practices*

- Too many project surprises, overruns and failures
- Customers are not satisfied with the process or the results
- Analysts, developers and testers find it difficult to work in tandem
- Team is bombarded by change but struggling to keep up
- You aspire to greater agility but are unsure how to achieve it



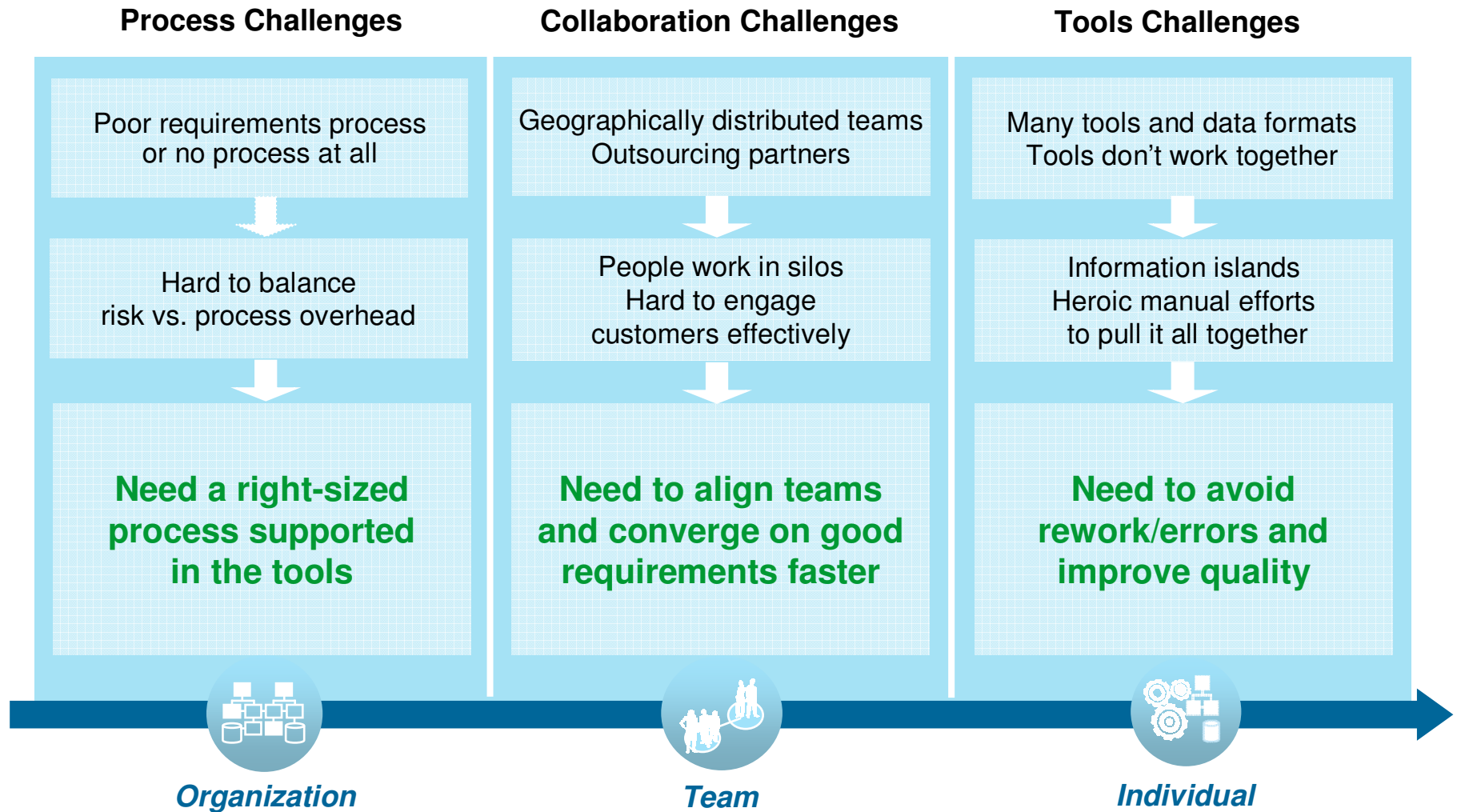
# A “requirements view” across the lifecycle

## For Programs, Projects, Products, Systems and Systems-of-Systems



# Good requirements *practices* are key to project success

## Address the big challenges

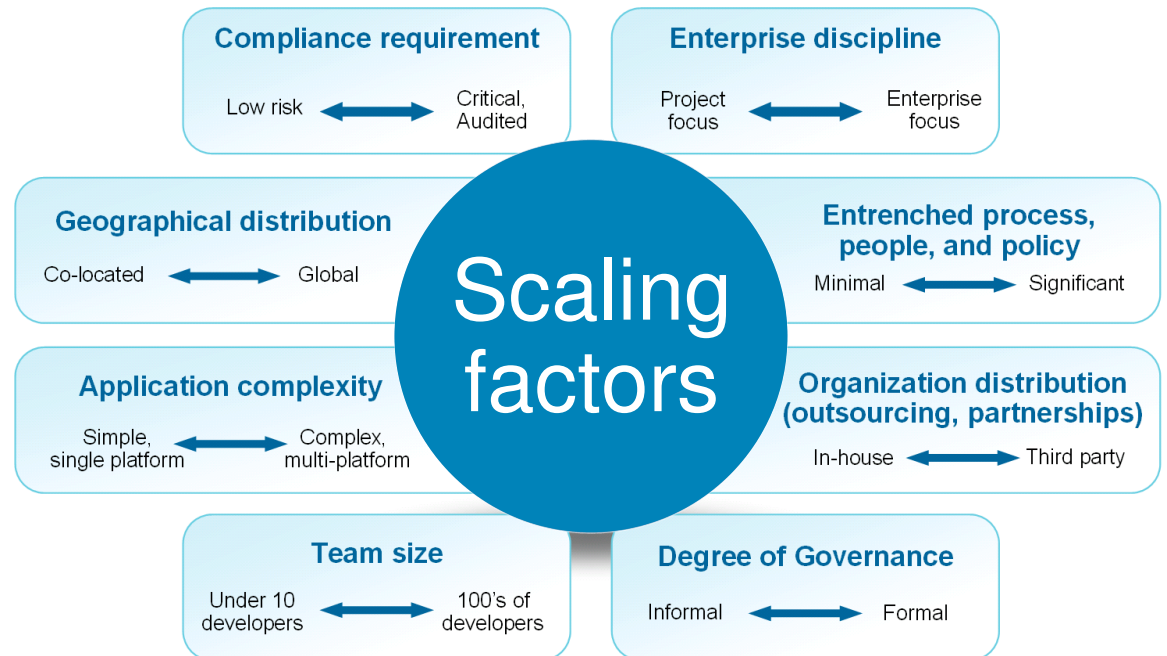


# Become more agile at scale

*Scaling factors require more discipline, automation and communication*

What doesn't change when scaling

- Stakeholders actively participating in describing, prioritizing and evolving requirements
- High level requirements envisioning during project inception
- Just-in-time requirements elicitation during each iteration
- Organizing and prioritizing requirements in ways that are transparent to the team

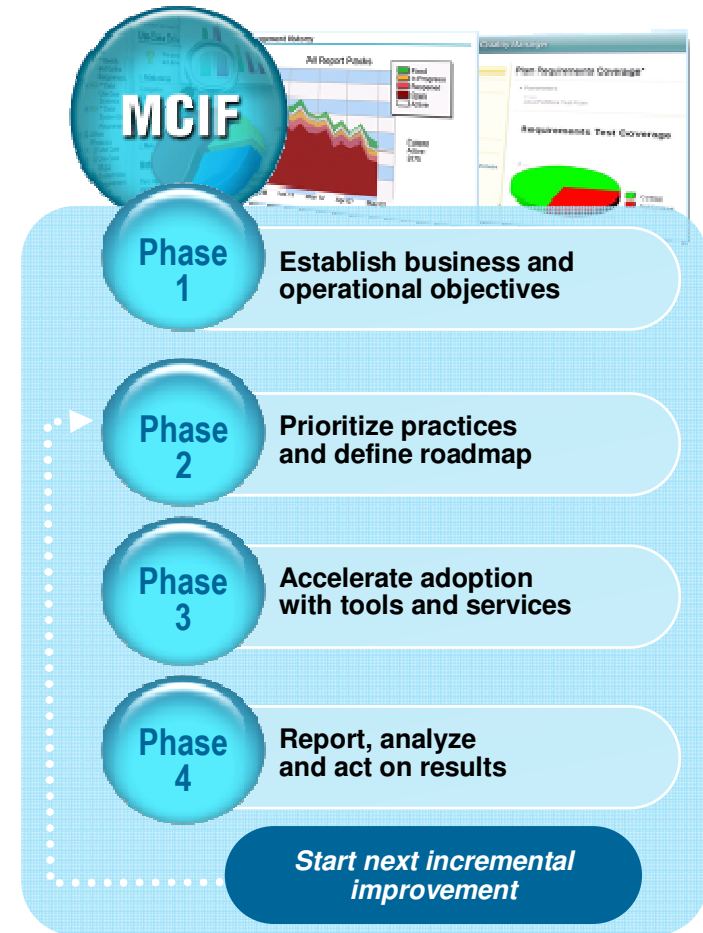




# Applying the right requirements practices

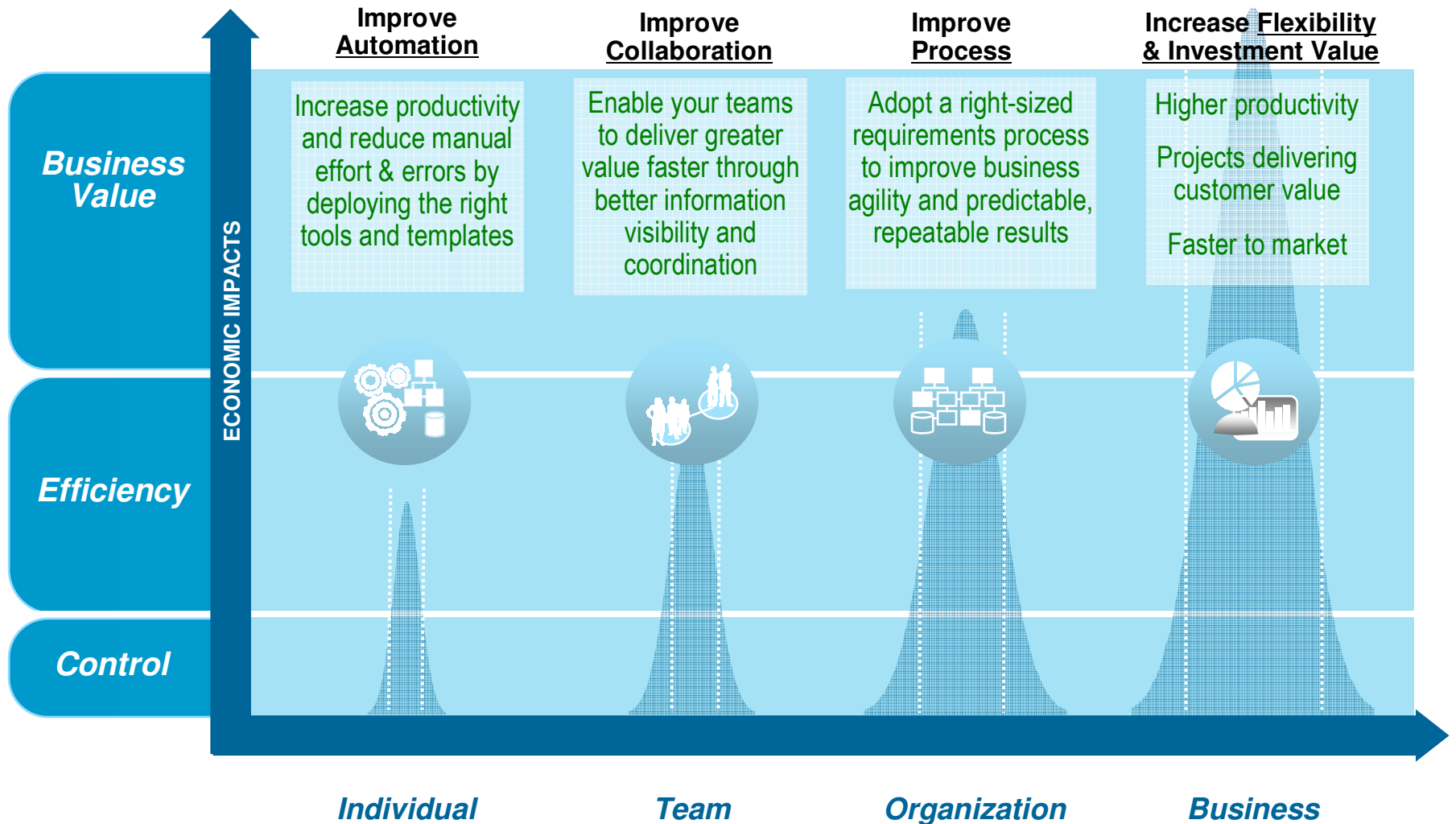
*Manage, measure and improve your practices with a proven, repeatable approach*

- Understand where you are at today, and establish a roadmap forward
- Define the simplest, leanest requirements process that will meet your needs; choose the appropriate tools to support it, and implement the requirements practices of highest value to you
- Accelerate and monitor the adoption of the process and tools
- Continuously assess the business and operational benefits and replicate best practices across your teams



## Measured Capability Improvement Framework

# Improve your ROI with better requirements and requirements practices



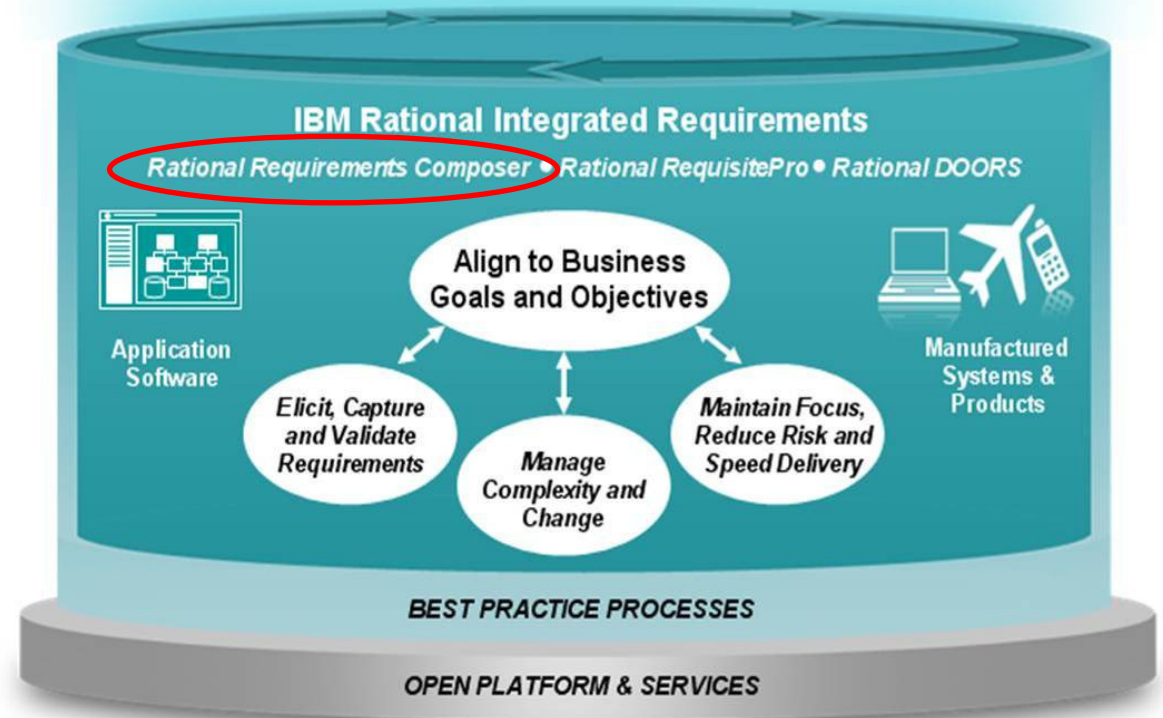
# Agenda

- Part 1: Why good requirements practices matter
- Part 2: Achieving better business outcomes with Rational Requirements Composer

# Rational Requirements Composer

Better business outcomes through *light-weight* requirements practices

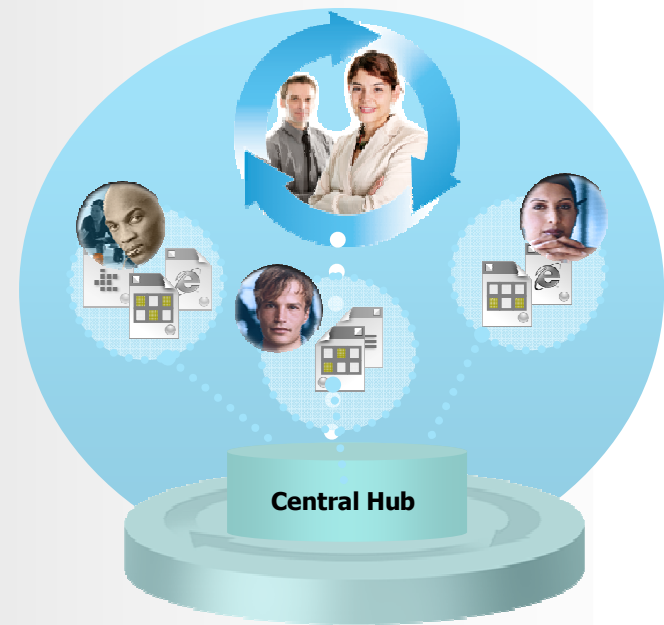
- **Engage stakeholders** early and often to converge on the “right” requirements faster
- Improve agility and align project activities with customer value in **light-weight practices**
- **Reduce rework and get to market faster**



**Better requirements. Less rework. Better results**

## Good requirements are key to project success

- **Understand what your customers really need**
  - Engage a wide range of stakeholders to capture real business needs and project constraints
  - Uncover missing, incorrect, and ambiguous requirements that drive project overruns and cancellations
  - Work across boundaries: organizational, geographical, time zones



*“Rational Requirements Composer has been engineered with the business analyst in mind. The web access to these artifacts brings the requirements domain into the current technology space. I can simply send a URL out for review.”*  
 - Randy Haven, IBM Global Business Services



*“In general, [RRC] is very simple and easy to use with a nice and customizable UI. The Web UI is a great addition ... so that we can review, comment and verify artifacts with clients and other team members.”* - Anthony Gock, Deloitte



# Use scenarios to uncover customer needs

**1. Brief Description**  
A registered customer has applied to open a new Account. Based on existing Customer information, the eligibility system will use predefined heuristics to determine whether or not the customer is eligible. In circumstances, intervention by the Branch Manager may be necessary.

**2. Basic Flow of Events**

Describe flows and capture requirements in **Rich-text Documents**

Define and reuse common terms in **Glossaries**

Outline flows with **Process Diagrams**

Group related scenarios, describe actors, system boundaries and user goals with **Use Cases**

Visualize scenarios with **Storyboards and Screen Flows**

Use whiteboard snapshots and other **Informal Documentation**



# Many notations, extensive team involvement, rapid iterative refinement

## Example: Using RRC V1.0 to create V2.0

**1**

**1. Product managers record and organize stakeholder requests**

Create a review for a project and specify the users that are to approve or review a given set of artifacts. Reviewers/approvers comment and sign off on the artifacts. The status of the review is updated accordingly.

Design: [Review and Approval](#)

Use Cases: [Use Cases for Review and Approval](#)

**2**

**2. Product managers collaborate with development on milestone scenarios and use cases**

**4**

User	Type	Status
John	Approver	Not started
Charles	Reviewer	Not started
David	Optional	Not started

**4. UI Storyboards help team to rapidly iterate on scenario options and flesh out design**

**3**

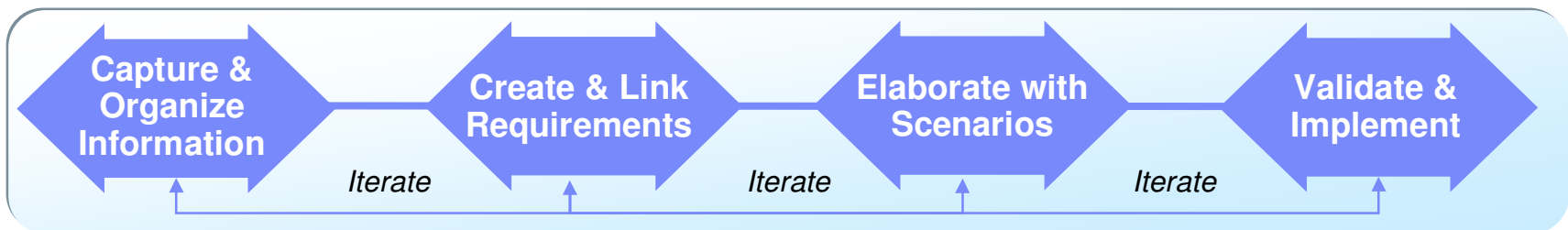
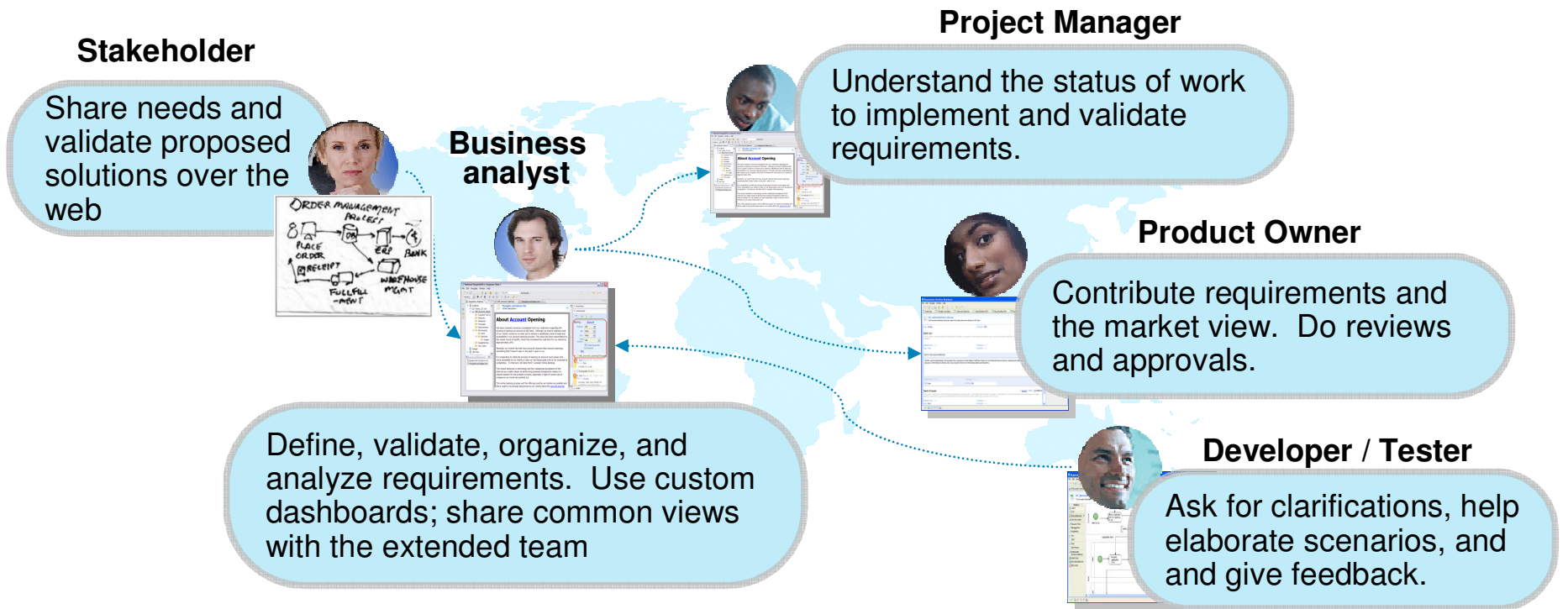
```

    graph TD
      Draft --> StartsReview[Starts review]
      StartsReview --> InProgress
      InProgress --> Review
      Review --> Draft
      Review --> StartsReview
      Review --> InProgress
  
```

**3. Architects and subject matter experts use process diagrams and text to outline scenarios**

# Distributed teams collaborate around the world

## Capture, analyze, review and approve requirements





# Good requirements *practices* are key to project success

## More agility through light-weight requirement practices that scale

- **Respond quickly to changing priorities**
  - Learn and adapt as you go
  - Every project is a "voyage of discovery"
  
- **Work smarter**
  - Bridge your information islands
  - Become more agile at scale
  - Automate what you can to reduce errors and increase productivity
  - Reuse existing knowledge and good practices
  
- **Work together more effectively**
  - Align development and test activities with customer value
  - Improve information visibility and management of change

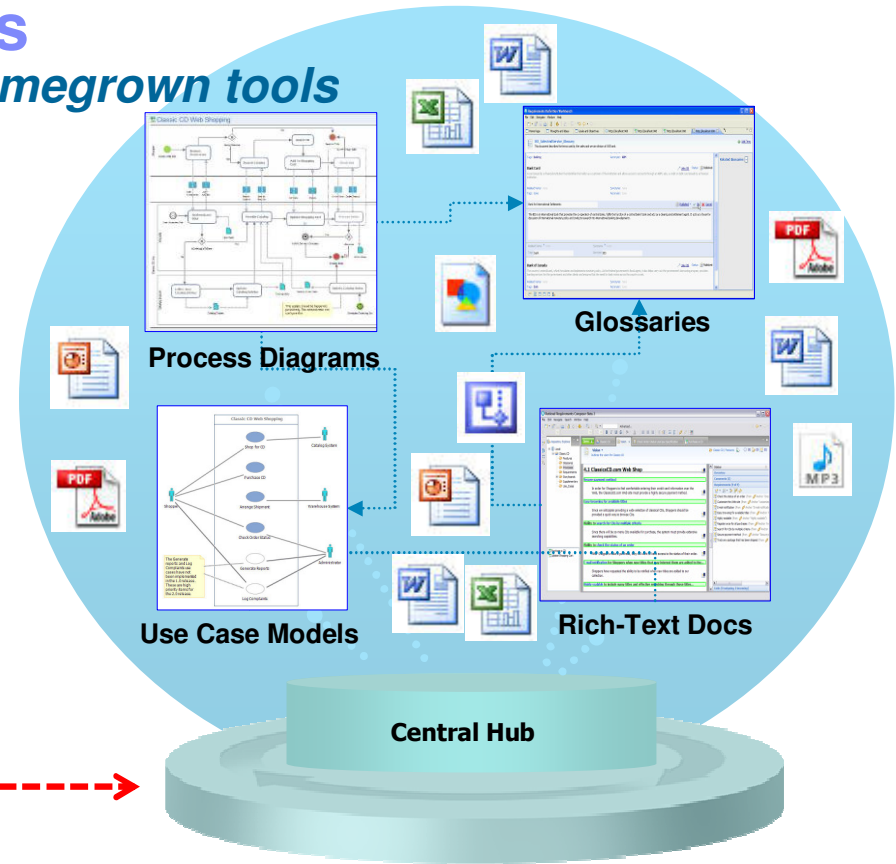
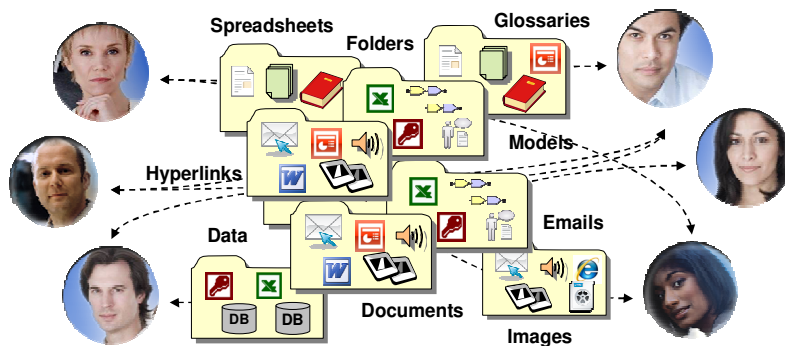


*"We are using RRC 1.0 to capture requirements for a major web development project and are definitely sold on the product. Version 2.0 will take it to a whole new level."*  
 - Robert Huet, Ascendant Technologies



# Bridge your information islands

## Embrace but move beyond Office and homegrown tools



- There are **many kinds** of requirements artifacts
- Many tools, data formats and repositories create **information islands**
- A large **extended team** participates in the requirements “conversation”

*“The vast array of options linking artifacts sensibly with one another [in RRC] give it a definite benefit over using older, document-based approach to defining requirements ... This product is clearly a step up from our current methodology.”*

- Randy Haven, IBM Global Business Services



# Use of general tools for requirements poses challenges

Value in RDM

## Difficult to address:

- ✗ Artifacts are not interrelated (are on islands)
- ✗ No team collaboration & workflow
- ✗ Ad-hoc procedures for relating and versioning artifacts
- ✗ Difficult to reuse requirements “downstream”
- ✗ Working with artifacts in well-defined groups
- ✗ Coordinated updates across related artifacts
- ✗ Generating requirements reports and documents

### Office documents

Requirements can be captured in ...

- Documents
- Drawings
- Spreadsheets
- Pictures

### Collaborative tools

Requirements can be communicated via:

- Email
- Instant messaging
- Newsgroups
- Wikis and blogs
- Groupware
- Shared file system

# Domain-specific tools help the team go to the “next level”

Value in RDM

- ✓ Secure repository provides central location
- ✓ Team collab. & workflow
- ✓ Progressively structure unstructured information
- ✓ Artifacts are interrelated and versioned
- ✓ Requirements are reused “downstream”

## Office documents

Requirements can be captured in ...

- Documents
- Drawings
- Spreadsheets
- Pictures

- ✓ Collections group related artifacts
- ✓ Baselines for scoping and comparison
- ✓ Views for coordinating updates across artifacts
- ✓ Purposeful documents and dashboards

## Collaborative tools

Requirements can be communicated via:

- Email
- Instant messaging
- Newsgroups
- Wikis and blogs
- Groupware
- Shared file system

## Requirements Domain-specific

### Tools enabling high-productivity practices

- Native support for various RD techniques
- Embrace office documents
- Collaboration in the requirements practices

### Requirements across the lifecycle

- Relate requirements to project milestones and work items
- Testing verifies solution meets requirements
- Coverage and impact analysis, change mgmt



# Foster collaboration and team transparency

## Centralized repository, common dashboards, team-wide conversations

The screenshot shows the Composer2.0 web interface. The main content area displays a list of artifacts with columns for Name, Last Modified By, and Last Modified Date. The interface includes a sidebar on the right with sections for Project Details, Recent Comments, Recent Artifacts, and Recent Requirements. On the left, there are filter panels for tags, attributes, and folders.

**Organized by project**

**Social tagging (public and private)**

**Filter by almost any criteria**

**Organize artifacts into hierarchies**

**In-context team conversations**

**Each comment can be a discussion thread**

**Artifacts matching search criteria (with customizable view of relevant attributes)**

**Transparency into what the team is doing**

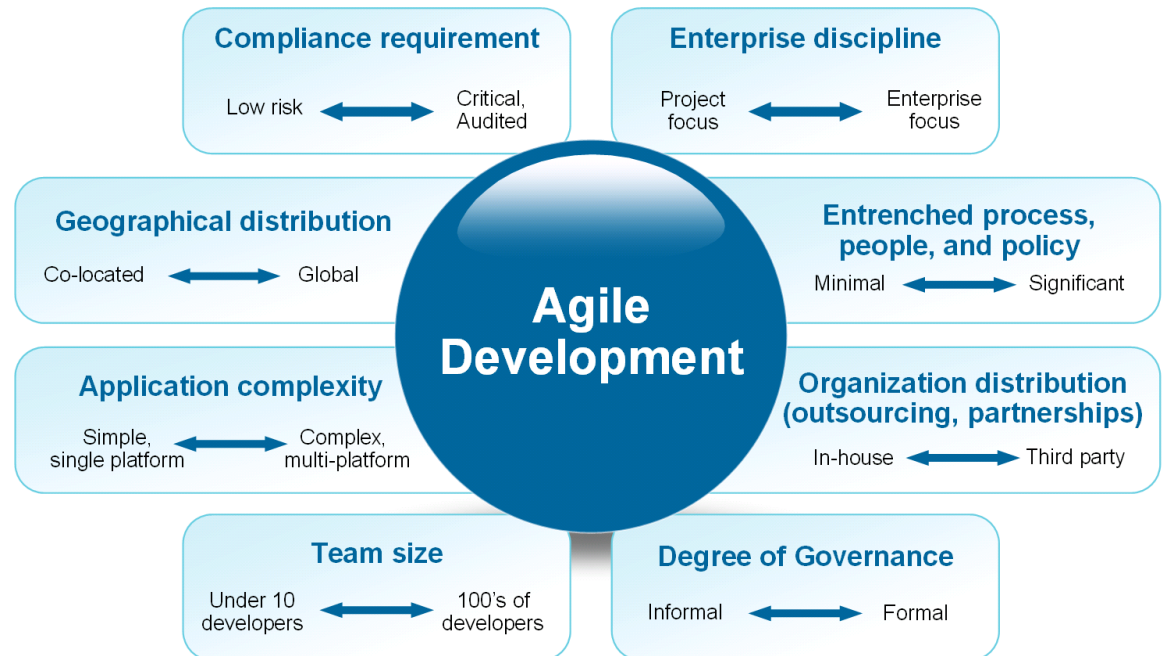
Name	Last Modified By	Last Modified Date
Business Process Diagram		
delete me	D. K. Grotjohn	Jun 30, 2009 2:43:22 PM
Document		
02. Collection editor - UI Specs	D. K. Grotjohn	May 12, 2009 11:37:53 AM
B2B Process 2.0 Format	Nicholas Bennett	Mar 23, 2009 11:51:33 PM
Baselining Test Cases	Eric S. Yokeley	Feb 11, 2009 4:56:00 PM
dummy document	MUHTAR AKBULUT	Oct 22, 2009 3:43:14 PM
Terminology to discuss	D. K. Grotjohn	Aug 4, 2009 2:00:50 PM
Web UI Enablement for RRC Server	MUHTAR AKBULUT	Jan 27, 2009 10:24:53 AM
JPEG Image (JPG)		
Figure 1 -- Project View with Attributes.jpg	Bill Andreas	Jan 27, 2009 12:05:55 PM
Microsoft Word Document		
Review.doc	Thomas F. Mutdosch	Apr 8, 2009 9:54:31 AM
PNG Image		
Reporting-Dialog-1.png	Bill Andreas	May 25, 2009 3:04:25 PM
slide 34.png		2:31:12
vwlt-11		34:48 PM
Storyboard		
2. Compare baselines	Bill Andreas	May 28, 2009 3:28:13 PM
3. Review a collection	Bill Andreas	May 29, 2009 8:47:58

# Become more agile at scale

*Scaling factors require more discipline, automation, and communication*

What doesn't change when scaling

- Stakeholders actively participating in describing, prioritizing and evolving requirements
- High level requirements envisioning during inception
- Just-in-time requirements elicitation during each iteration
- Organizing and prioritizing requirements in ways that are transparent to the team

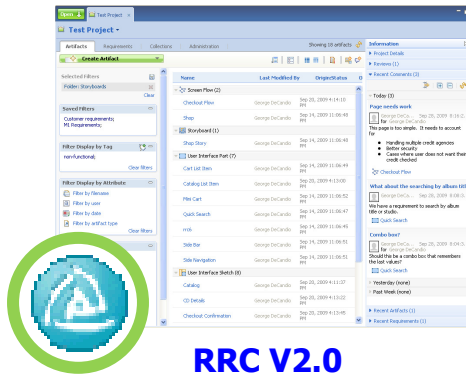


# Automate what you can

## Improve productivity and reduce errors



### Rational Requirements Composer



**RRC V2.0**

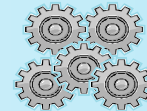
*“RRC dramatically reduced the time to do internal assembly, discussion, and finalization of deliverable documents internally.”*  
 - Brian Beveridge, Greenridge Business Systems



- Deliver work products in standard forms



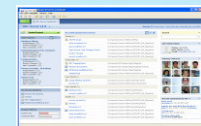
- Generate and publish reports based on IBM® Rational® Publishing Engine



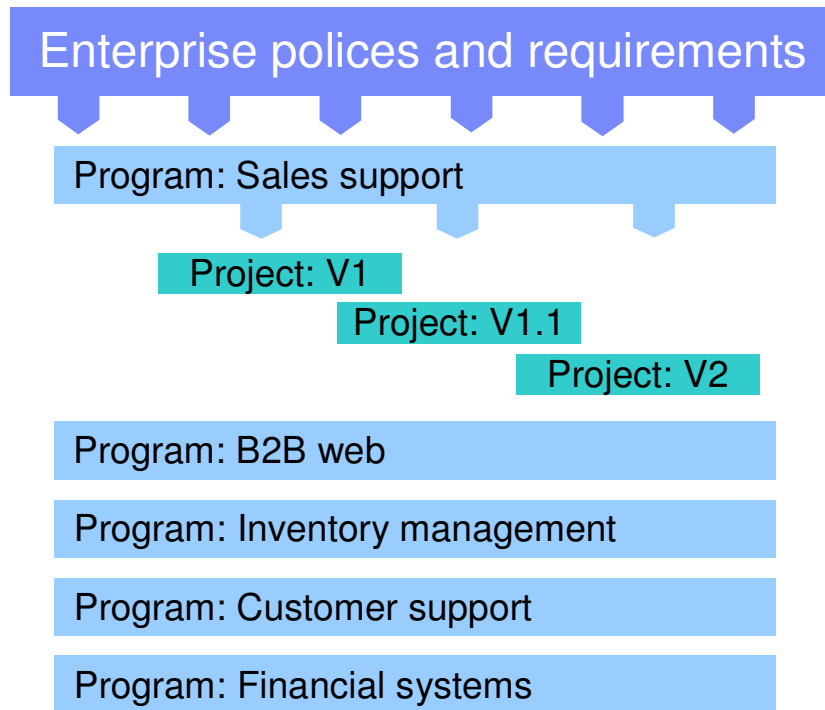
- Organize and retrieve information quickly using tags, filters, collections, searches and queries



- Analyze requirements from various perspectives using dashboards and traceability reports

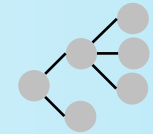


# Reuse existing knowledge and good practices



**Rational Requirements Composer**

- Align projects with enterprise, program and product-level requirements



- Reuse requirements artifacts



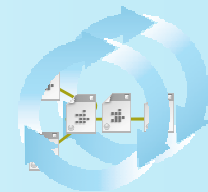
- Use templates to promote repeatable best practices



- Remove ambiguity and unify teams around common glossaries



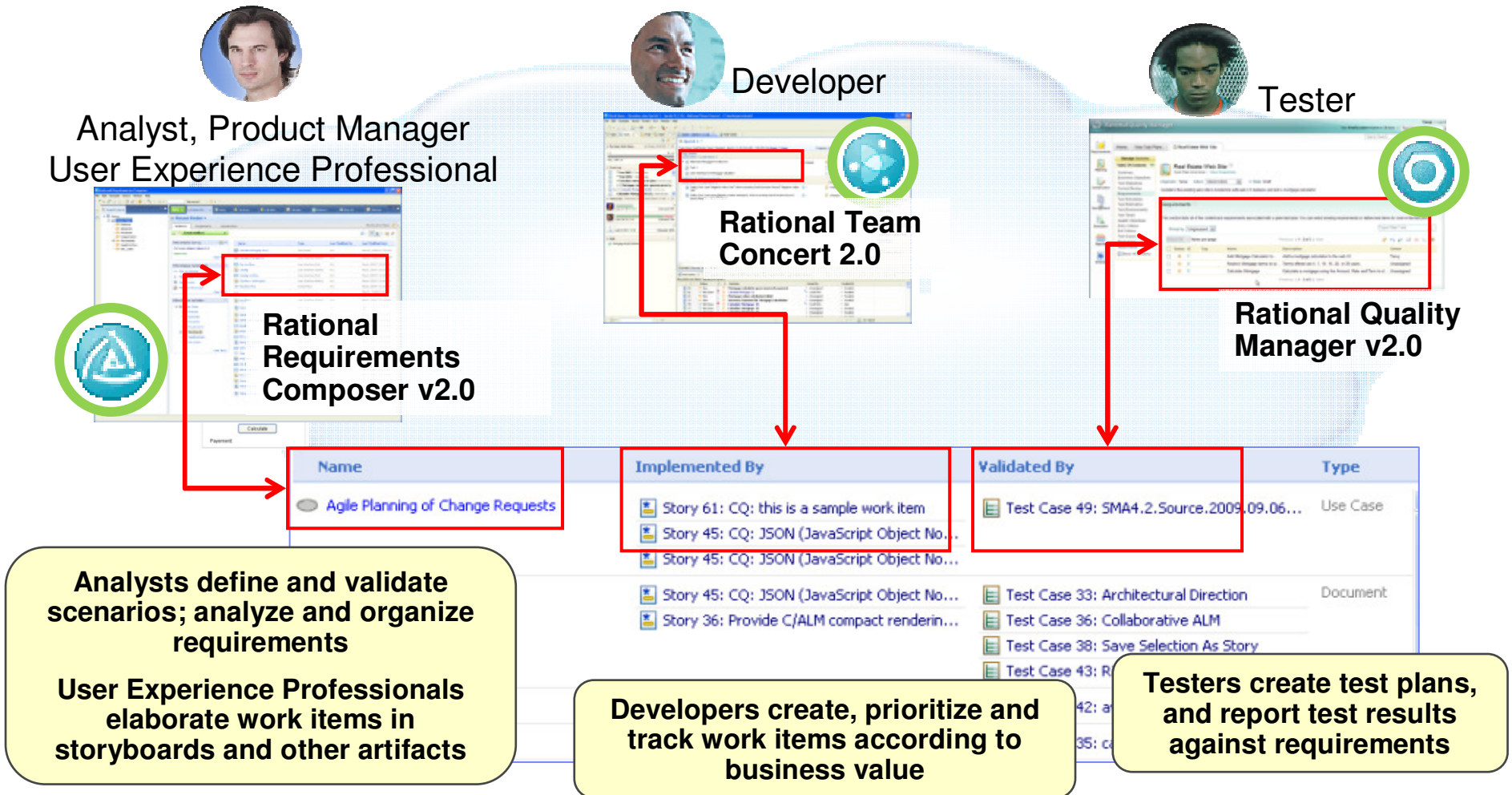
- Provide lightweight process guidance in practitioners' tools where and when they need it



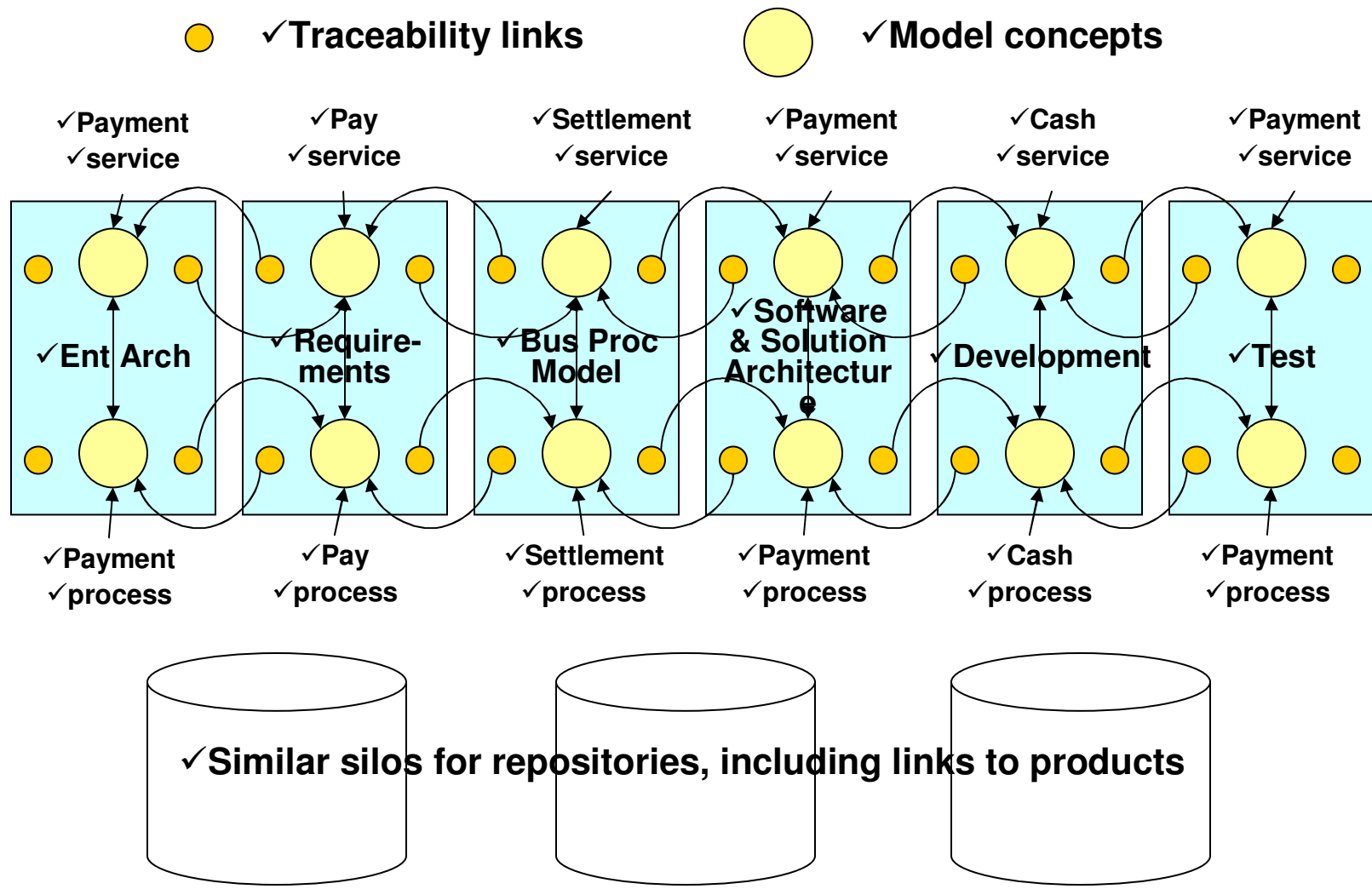


# C/ALM - Align development and test activities with customer value

## Break down role-based information silos for better project execution

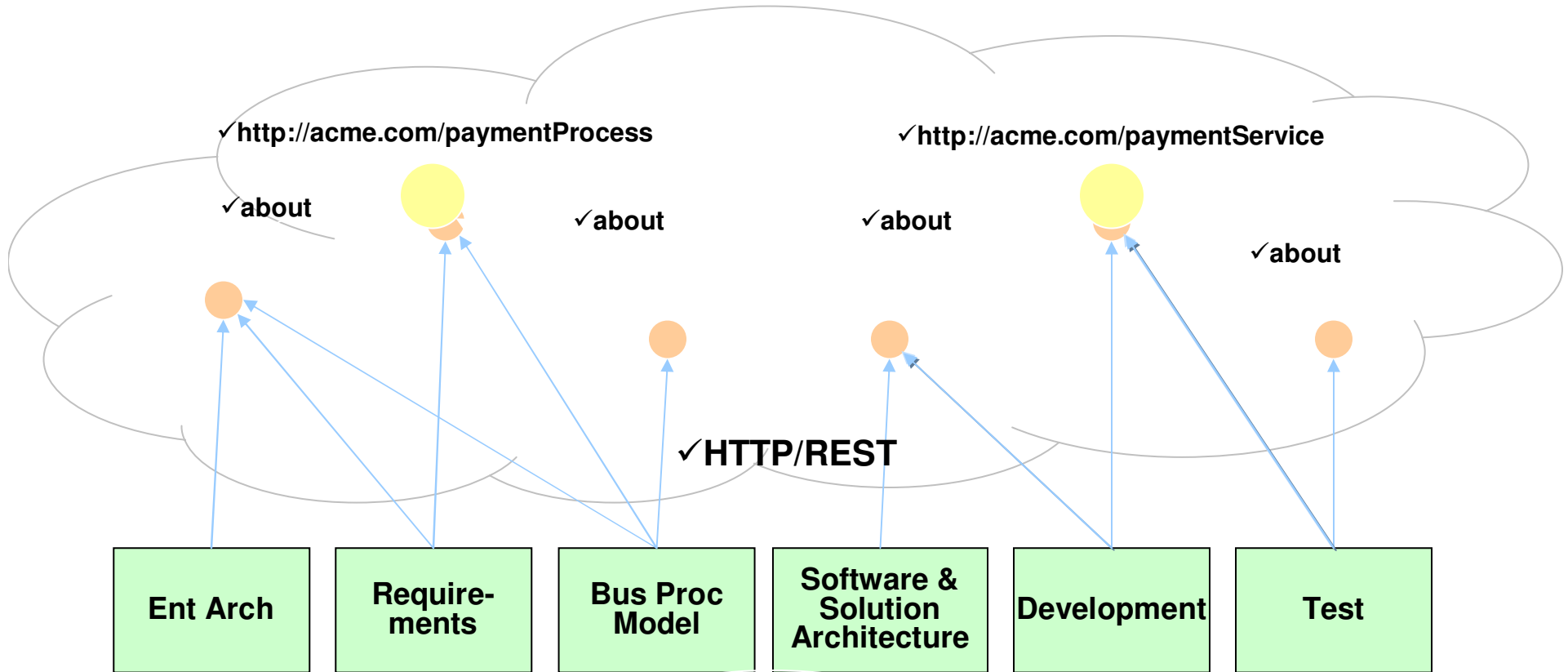


# Tool integration - the old way



# Tools – the new way: “www linked data”

## Taking integration to a new level



✓ Similar fix for repositories

# C/ALM Cross Product Viewlets

The screenshot shows the Rational Requirement Composer interface for Bob Alden. The dashboard includes several viewlets:

- My Projects:** Lists projects like JUnit Project and Alpha Project with sub-items like Requirements, Development, and Test.
- My Documents:** Lists recent documents such as 'JUnit 4.4 Release Plan' and 'Failed Color Definition'.
- My Tasks (4 open):** Lists tasks like 'Review all with team' and 'Draft usecases for next iteration'.
- Requirements With Defects (5):** A viewlet (highlighted with a red box) showing failed requirements like 'Fix background (69)' and 'Theme not consistent (54)'. It includes a 'Launch dialog for picking colors' link.
- Requirements With Failing Tests (2):** Shows tests that failed, such as 'Test launch trigger' and 'Failed Color Definition (1 failed)'.
- Recently Modified (27 by Priority):** A bar chart showing the number of items modified recently, with a peak of 18 items.
- Recent Comments for Me:** Shows comments from Patricia Bader and Marco Cassice regarding the 'Release plan 4.4 ready'.
- Current Agile Planning Plans (2):** Shows the current iteration 'JUnit 4.4 M1' and progress bars for '4.4 M2 Iteration Plan' and '4.4 Release Plan'.
- Primary Team Contacts (8):** Lists team members like Patricia Bader (Project Manager), Marco Cassice (Development Lead), and James Joyce (Writer).

✓ C/ALM viewlets provide answers to questions about lifecycle status  
 ✓ Information from RTC and RQM is visible in RRC (and vice versa)

# Traceability – Lifecycle Viewlets on Web Dashboard

General Tab

RRC Server [xanadu.raleigh.ibm.com]

RQM [xanadu.raleigh.ibm.com]

Rational Requirements Server [xanadu.raleigh.ibm.com]

RTC [xanadu.raleigh.ibm.com]

- Project/Team
- Requirements
  - Artifacts
  - Comments
  - My Projects
  - Requirements
  - Requirements With Development Items
  - Requirements With Test Cases
  - Reviews

### Viewlet Chooser

Select a viewlet from the chooser to add it to your dashboard.

✓ Choose viewlets from RRC, RTC or RQM

✓ Requirements with development items

✓ Requirements with test cases

Open Stories (5)

- 9: Test creation from RRC
- 10: This is my summary
- 11: as dfasa?sdfsa dsaasdfa
- 13: fas dfasa?sdfsa
- 26: Testing

Requirements Validated by Test Cases

Project: XanaduProj Test Collectic

Show: All

- TestRequirement (Loading...)

Requirements With Development Items

Project: XanaduProj Test Collectic

Show: All

- TestRequirement
- Story 9: Test creation from RRC

**Welcome to Rational Requirements Composer**

Rational Requirements Composer provides a platform for collaborative requirements definition that enables business analysts, client stakeholders and software development teams to define

## Traceability – Predefined Filters for C/ALM viewlets

General Tab +

**Requirements With Development Items**

Project: Base Pr RQM Cc Show: Resolver

- All
- Open
- Resolved

✓ Requirements with development items:  
All, Open, Resolved

General Tab +

**Requirements Validated by Test Cases**

Project: Base Pr RQM Cc Show: Missing

Burak Requirement  
4: Test Case with unattempted

- All
- Missing Test
- Successful
- Failed

✓ Requirements with test cases:  
All, Successful, Failed, Missing Test



# Traceability – Rich Hover

**✓ Hover over work items, test plans and test cases**

**✓ Rich hovers display information from RTC and RQM**

**Status Summary**  
 ➔ New Share code with Jazz Source Control

**Details**  
 Type: Task  
 Filed Against: Backlog  
 Team Area: Team 1 / Scrum Project  
 Creation Date: September 18, 2009 11:33 PM  
 Created By: TestJazzAdmin1  
 Tags:

**Quick Information**  
 Subscribers (1): T Implements Requirement (1)

**Description**  
 Share code with Jazz Source Control  
 To share code in your Eclipse workspace with the Jazz stream created during process initialization:  
 1. Select a project you wish to share, right-click and select **Team->Share Project...**

Open Item

**Welcome to Rational Requirements Composer**  
 Rational Requirements Composer provides a platform for collaborative requirements definition that enables business client stakeholders and software development team requirements using a variety of techniques, and collaborate using a suite of platform capabilities.



## Product Overview

The Rational Requirements Composer provides a lightweight solution for collaborating, reviewing, and commenting on requirements artifacts throughout the development lifecycle with the rich client.



## Product Tour

This tour shows the major components of Rational Requirements Composer, including using the Web client to

# Traceability – Dashboard Views on rich client

The screenshot shows the Rational Requirements Composer interface. The main window displays a list of requirements and their associated development items and test cases. A dialog box titled 'Customize project page display' is open, allowing users to select and reorder columns for the dashboard view.

Name	Development Items	Test Cases	Last Modified By	Last
AnotherReq			Marc Baumbach	Sep 2 PM
MyReq5			Marc Baumbach	Sep 2 PM
AnotherTest			Marc Baumbach	Sep 2 AM
TestRequirement	<ul style="list-style-type: none"> <li>Story 9: Test creation from RRC</li> <li>Story 10: This is my summary</li> <li>Story 26: Testing</li> </ul>	<ul style="list-style-type: none"> <li>Test Case 1: TestRequire...</li> <li>Test Case 7: Marcs Req</li> </ul>	Marc Baumbach	Sep 2 AM
AnotherTest2			Marc Baumbach	Sep 2
BasicReq				
BigTest				
ReqFromReq2				
ReqFromReq				

**Customize project page display**

Check the columns you would like to display in this view. Use the Move Up and Move Down buttons to reorder (left to right) the columns.

Name	Width
<input checked="" type="checkbox"/> Name	170
<input checked="" type="checkbox"/> Development Items	200
<input checked="" type="checkbox"/> Test Cases	170
<input checked="" type="checkbox"/> Last Modified By	121
<input checked="" type="checkbox"/> Last Modified Date	121
<input type="checkbox"/> Managed Requirement	121
<input type="checkbox"/> Feature:Type	100
<input type="checkbox"/> Feature:Priority	100
<input type="checkbox"/> Feature:Status	100
<input type="checkbox"/> Feature:Difficulty	100

**Filters in project dashboard**

**Links to development and test items in dashboard**



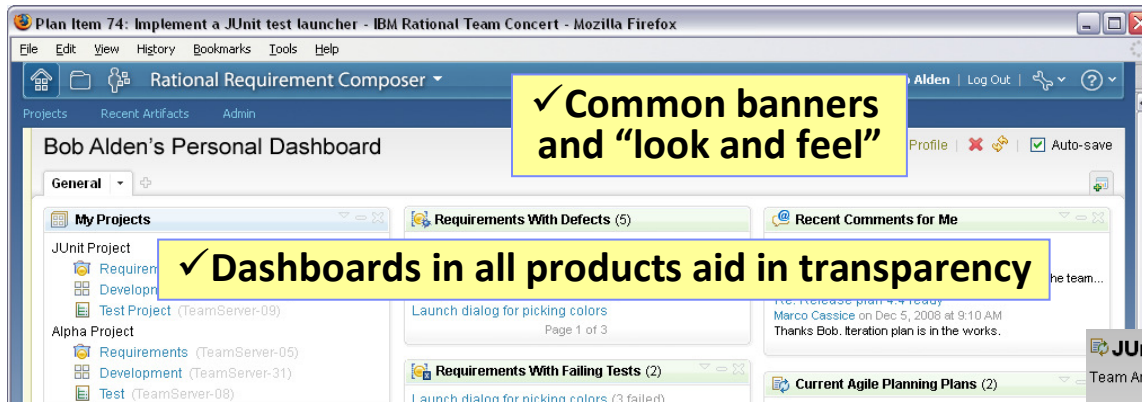
## Traceability – Links on editor sidebar on rich client

- Links to development and test items in sidebar and rich hovers

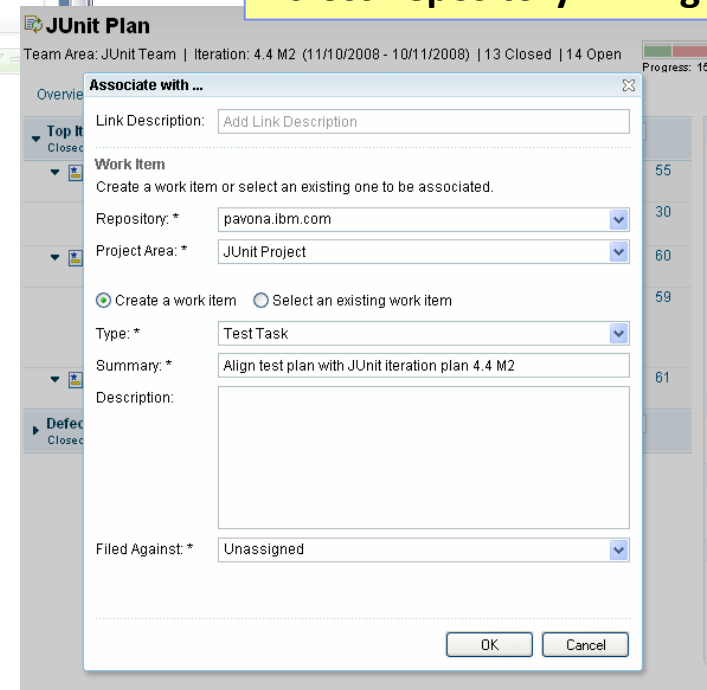
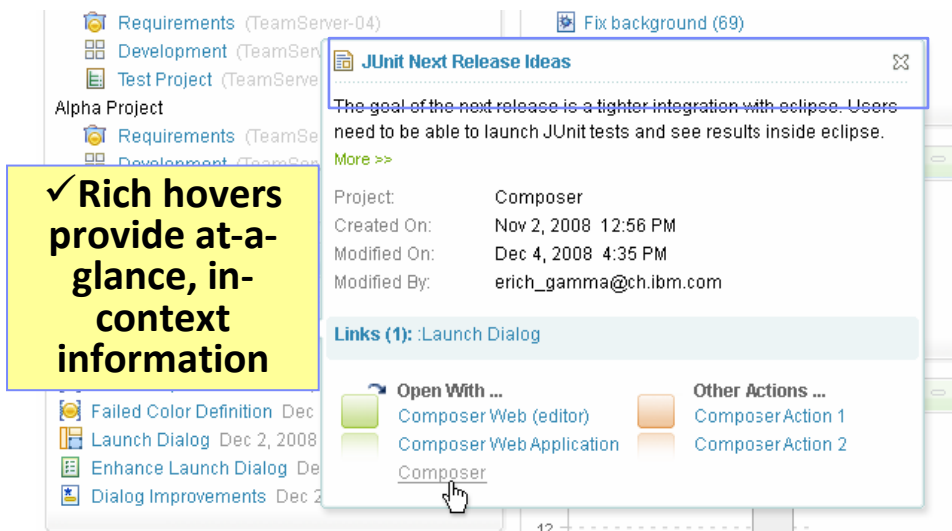
The screenshot displays the IBM Rational software interface for editing a TestRequirement. The main editor area contains placeholder text (Lorem ipsum) and a link to 'Story 10: This is my summary'. The sidebar on the right, titled 'Information', shows a tree view of links. A red box highlights the 'Links (6)' section, which includes 'Outgoing Links (0)', 'Incoming Links (1)', and 'Validated By (2)'. The 'Incoming Links (1)' section is expanded, showing a list of links: 'TestDoc', 'Story 9: Test creation from RRC', 'Story 10: This is my summary', 'Story 26: Testing', and 'Tested By (2)'. The 'Tested By (2)' section is also expanded, showing 'Test Case 1: TestRequirement' and 'Test Case 7: Marcs Req'. A yellow callout box with a blue border points to the sidebar and contains the following text:

- ✓ Links to development and test items
- ✓ In sidebar
- ✓ In "rich hovers"

# Common UI elements aid in ease-of-use



✓ Link Dialogs enable cross-repository linking



# C/ALM Mash-up Dashboard – View from RTC

**✓ Developers have insight into requirements in Rational Requirements Composer**

**Recently Changed Requirement Artifacts**

Artifact Name	Date	Time
Vision	Dec 4, 2008	4:52 PM
Eligibility Subsystem	Dec 4, 2008	4:35 PM
Balance of Account	Dec 4, 2008	3:59 PM
Supplemental	Dec 4, 2008	11:05 AM
Initial Account Opening Sketch	Dec 3, 2008	10:38 AM

# Improve management of change by improving visibility

*Change is inevitable, so invest in managing it well*

## Requirements

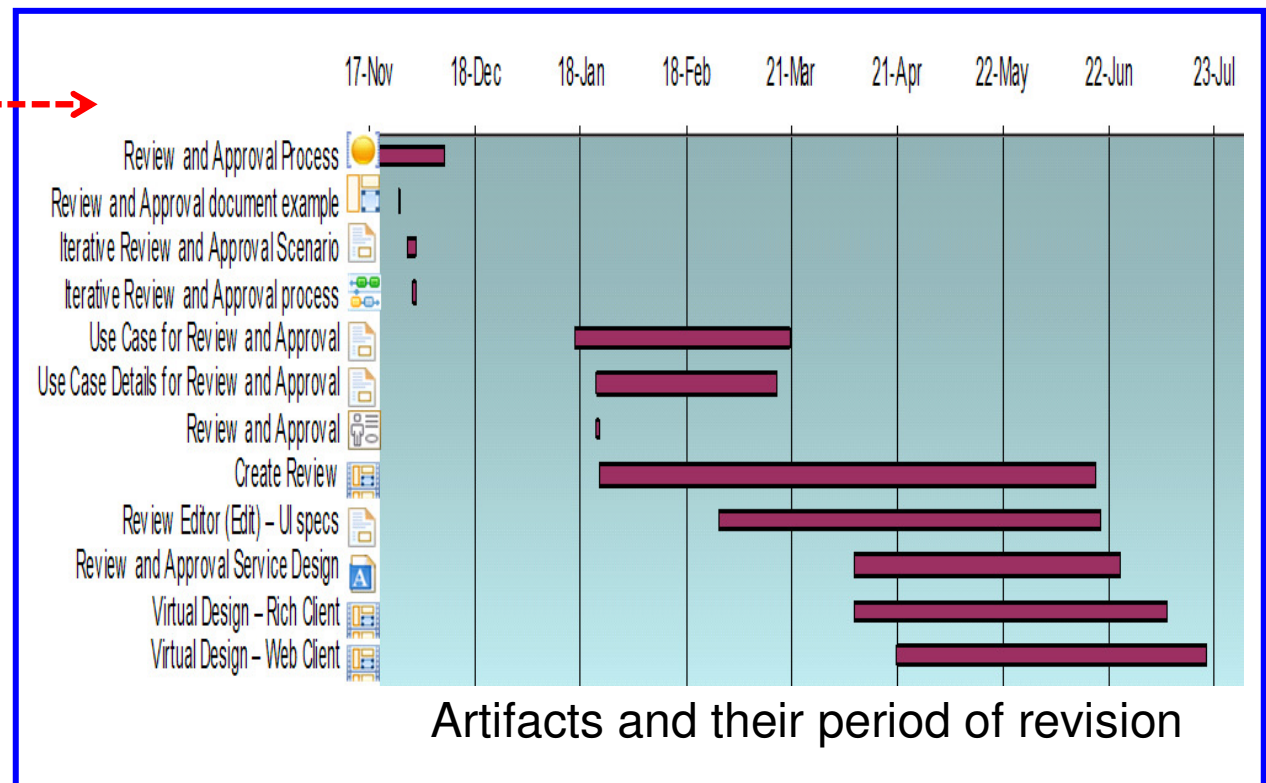
- Define clear scope; everyone works on the same version of the requirements – and knows when they change

## Project activities and project artifacts

- Keep them aligned with changing requirements

## Team members

- Retain project knowledge. Enable new staff to become productive faster and work smarter





## Better business outcomes

*Through good requirements and good requirements practices*

- **Reduce rework and get to market sooner**
  - Converge on good requirements faster
  - Less rework later
  - More automation
  - Fewer errors
  - Better project ROI



*"This tool enables us to extend and improve our client's requirement practices, which significantly increases the probability of a successful software project."*



*- Kenny Lew, CEO, Noblestar International*

# Requirements Composer Return on Investment (ROI)

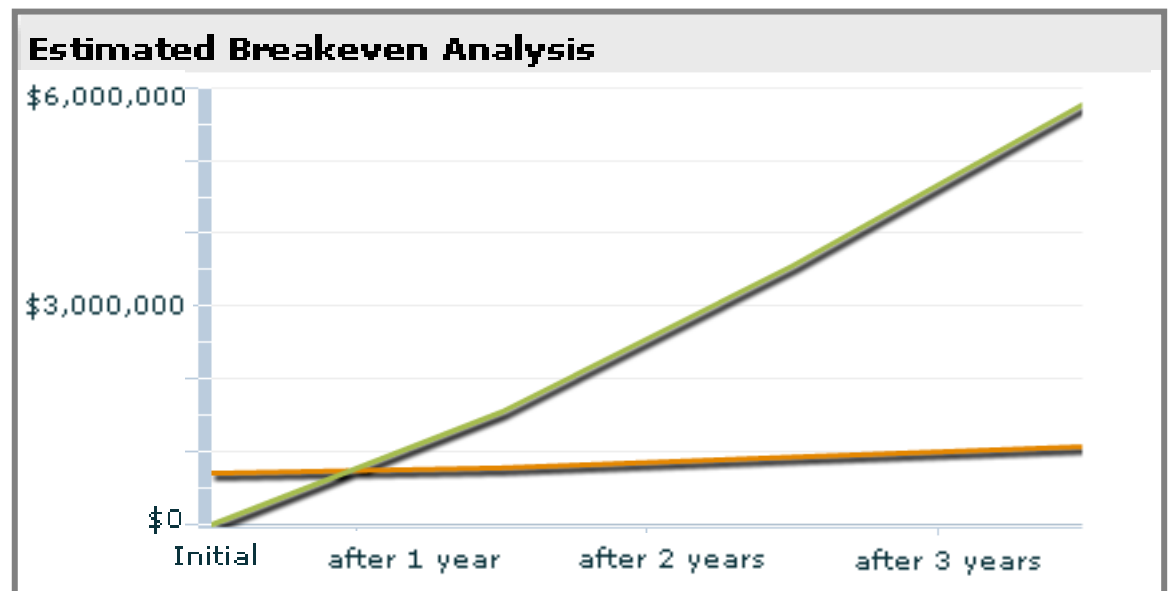
*Sample calculations for an organization completing five projects in one year*

Requirements Composer key cost-saving benefits include:

- ★ **Reduction in rework:** Get requirements right sooner; avoid wasted effort
- ★ **Faster time to market:** Start enjoying the outcomes of your project sooner
- ★ **Avoiding missing and lost requirements:** Reduce project overages and the risk of failure

[tinyurl.com/RDQROI](http://tinyurl.com/RDQROI)

Estimated Return On Investment				
	Initial	After 1 year	After 2 years	After 3 years
Cumulative Benefits		\$1,551,228	\$3,545,664	\$5,761,704
Cumulative Costs	\$707,300	\$782,211	\$925,141	\$1,068,072
Cumulative Cash Flow /	(\$707,300)	\$769,017	\$2,620,523	\$4,693,632
Benefit to cost ratio		1.98	3.83	5.39
Return on Investment		98%	283%	439%



## IAG Consulting achieves significant savings *with IBM Rational software*

**Situation:** IAG consultants needed a better way to engage stakeholders in defining requirements as well as implement an end-to-end requirements lifecycle.

**Solution:** IBM's Requirements Composer, IBM RequisitePro and (for certain engagements) Rational Software Architect.

**Benefits:** IAG saw **significant overall savings** for customers across the entire requirements lifecycle when taking into account requirements efficiencies, automated generation of tests, enhanced modeling and analysis capabilities, and **greater ability to share and collaborate** on larger scale projects.



*“Rational Software has the flexibility to address a wide range of IAG engagements, the pre-configuration functionality to support our demanding standards of consistency, and the right vision for end-to-end requirements life cycle management.”*

*A. Ross Little  
EVP IAG Consulting*

## Executive summary

### *Better, faster business outcomes with Rational Requirements Composer*

- **Understand what your customers really need**
  - Engage stakeholders to capture real business needs
  - Work across boundaries to uncover missing, incorrect and ambiguous requirements
  
- **Be more agile with light-weight requirements practices**
  - Align development work with business objectives, policies and requirements
  - Improve management of change by improving visibility
  
- **Reduce rework and improve time to market**
  - Converge on good requirements faster to reduce rework later
  - Get everyone on the same page and adapt quickly to change



*Delivering greater, faster business value with  
“lean” requirements practices*



# See for yourself at jazz.net



Home About Jazz Projects Downloads Forums Library Community

**jazz** Projects

Projects >

Rational Requirements Composer

Release Plan

Related Resources

Rational Requirements Composer 2.0 Online Help

Library

**Rational Requirements Composer**  
**Collaborative requirements for the whole team**

Rational Requirements Composer provides a platform for lean requirements practices involving a large set of stakeholders and team members in defining, analyzing and using requirements across the project lifecycle.

**What's new in 2.0**

Version 2.0 includes a new Web client, review and approval workflow, customizable dashboards, collections and snapshots, embedded reporting, and collaborative ALM integrations.

**Get the scoop:**

- What's new in 2.0
- Videos

**License information**

- Jazz licensing in ten minutes
- Additional RRC license details

[Download Rational Requirements Composer 2.0 >](#)

[Buy a com.co](#)

**Download 2.0 Release** **NEW!**  
 November 24, 2009

[All Downloads >](#)

**Join the conversation**

- Ask a question in the Forum
- Subscribe to the mailing list

**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 [Read guidelines >](#)
- See the project dashboard
- Explore detailed iteration plans

- Download Requirements Composer 2.0
- Learn more from the blog and library
- Have discussion with developers and community
- Report defects or request enhancements
- Stay informed!

## Resources

- Webcast: [Delivering more value: What's new in RRC 2.0](#)
- [Business Analyst eKit](#)
- [ROI Calculator](#) for requirements-driven quality
- [Integrated Requirements Management Web page](#)
- RRC on [jazz.net](#)
- Follow us on Twitter [@RationalRDM](#) and on [Facebook](#)
- Rational [RDM Blog](#)



**Learn more at:**

- [IBM Rational software](#)
- [Integrated Requirements Management](#)
- [Process and portfolio management](#)
- [Change and release management](#)
- [Quality management](#)
- [Architecture management](#)
- [Rational trial downloads](#)
- [Leading Innovation Web site](#)
- [developerWorks Rational](#)
- [IBM Rational TV](#)
- [IBM Business Partners](#)
- [IBM Rational Case Studies](#)

**© Copyright IBM Corporation 2008. All rights reserved.** The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, Rational, the Rational logo, Telelogic, the Telelogic logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.