Rational Requirements Composer
Drinking Our Own Champagne
JAOO Oct 6th 2009

Rational. software

e business on demand.

© 2009 IBM Corporation

Updated October 5, 2009

DISCLAIMER

- This presentation represents future product direction and strategy and is presented based on best information available. It may change in the future.

© Copyright IBM Corporation 2009. 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.

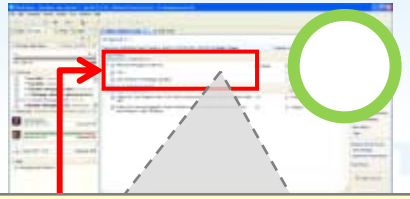
© 2009 IBM Corporation

C/ALM - Align development and test activities with requirements

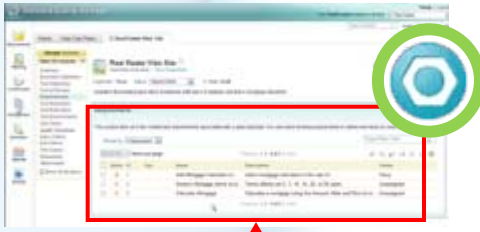
Breakdown silos for better project execution

Analyst, Product Manager
User Experience Professional
Rational Requirements Composer v2.0

Developer
Rational Team Concert 2.0

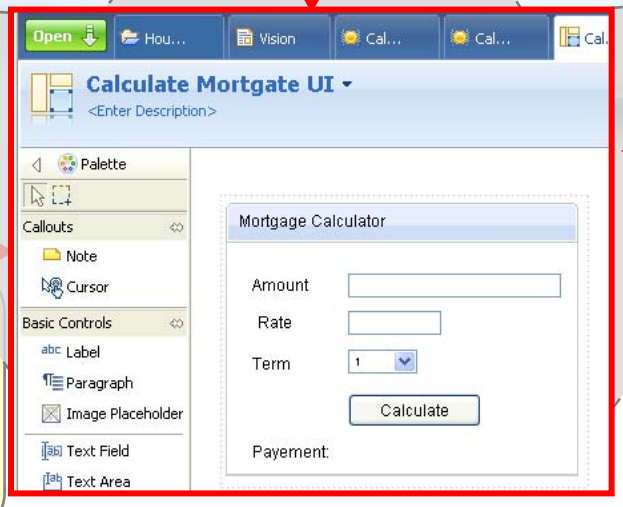


Tester
Rational Quality Manager v2.0



Developers create work items that deliver the requirements

Testers create test plans and report test results against requirements



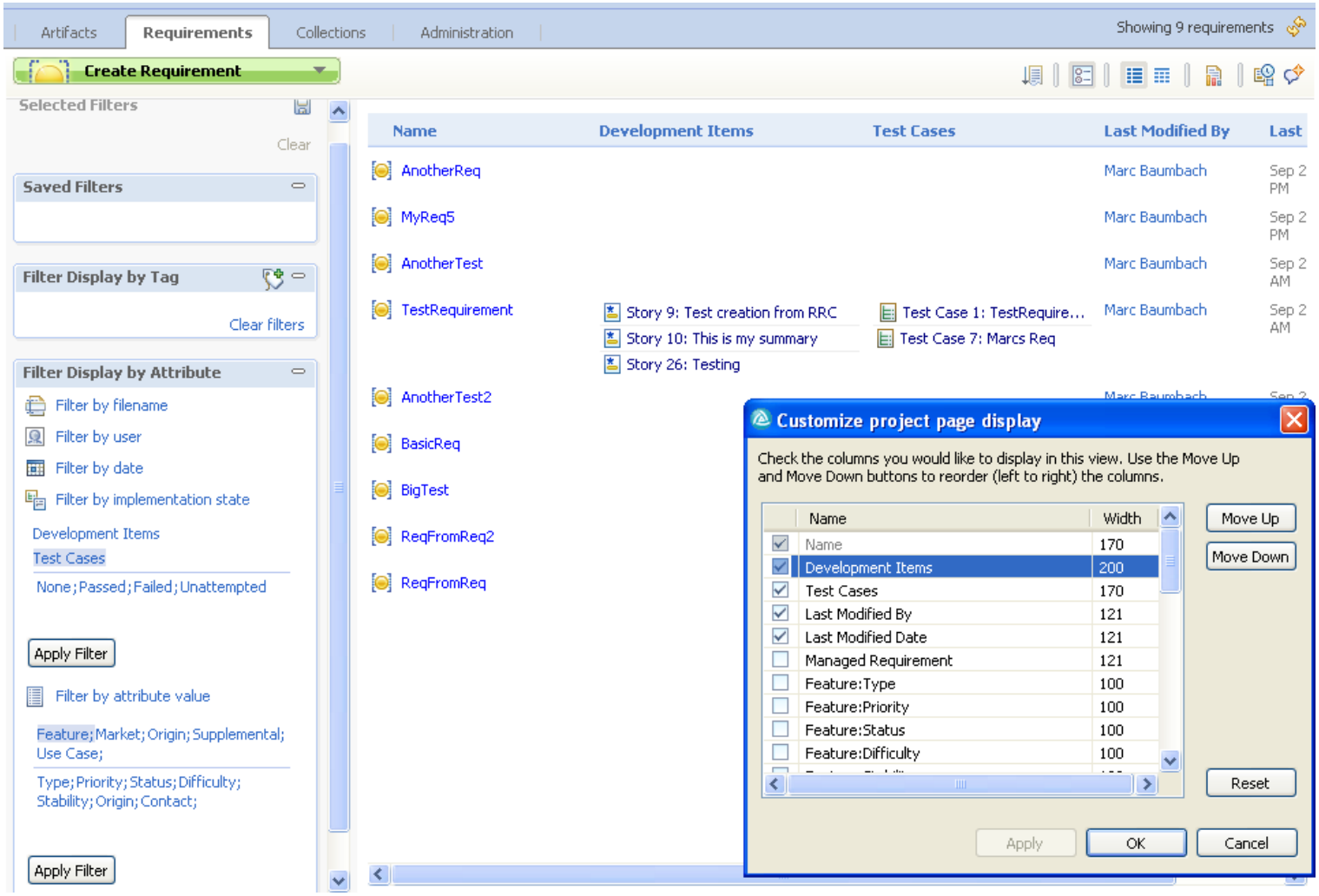
Analysts define, analyze and organize requirements

User Experience Professionals elaborate work items in storyboards and other artifacts

© 2009 IBM Corporation

Traceability – Dashboard Views on rich client

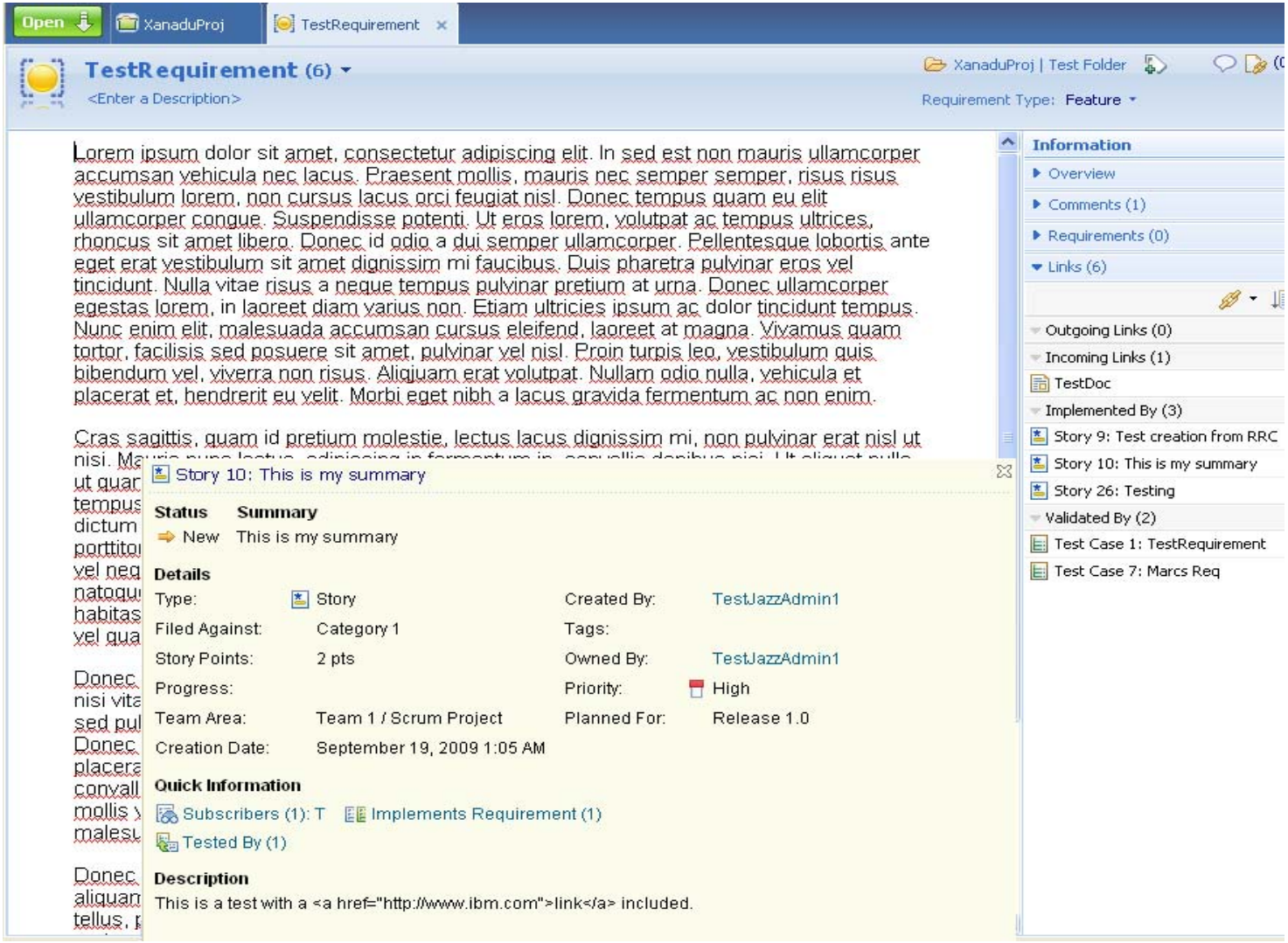
- **Filters in project dashboard, Links to development and test items in dashboard, rich hovers**



© 2009 IBM Corporation

Traceability – Links on editor sidebar on rich client

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



© 2009 IBM Corporation

Purpose

- Give you a vision of how IBM Rational® Requirements Composer (RRC) can improve team performance and project outcomes
- Use the RRC product team's experience as a case study
  - Where the whole team authors content in the development of Version 2
- Show you a "real" usage model that is improving the product
  - Rational "Drinking our own Champagne!" Campaign
- Get you thinking about how you and your teams can benefit from RRC

© 2009 IBM Corporation

Rational Tools Used During RRC Development

- Rational Requirements Composer
  - Requirements definition and refinement
- Rational Team Concert
  - Collaborative development
- Rational Quality Manager
  - Test management system: test planning, construction, and execution
- Rational Performance Tester
  - Performance testing tool used to identify the presence and cause of system performance bottlenecks

For more information please visit the following on Jazz.net (registration required)

Team Blog: http://jazz.net/blog/index.php/2009/07/16/rrc-drinking-our-own-champagne/
RSC 2009 RDM 15 Presentation: http://jazz.net/library/presentation/170

© 2009 IBM Corporation

IBM Teams using Rational Requirements Composer

- Rational Requirements Composer
- Rational Quality Manager (UX team)
- Developer Cloud RAD/RSA project (UX team)
- Rational C/C++ Debugger (whole team)
- MCIF – Web based Self Check
- Non Rational Teams (Just starting)
  - IBM GBS
    - GBS Canada RRC Pilot
  - IBM Systems &Technology Group
    - - STG Rational Solutions
    - - OVS Console
    - - MS System Center Integration
    - - Systems Software Development at IBM China Investment Co. Ltd (Wei Feng Li)
  - SWG UX teams

© 2009 IBM Corporation

IBM Rational Requirements Composer Core Team – 45 Total

- Development team - 25
- User Experience - 3
- Documentation - 2
- System Validation Testing - 4
- Others (Product Management, Field, Support, etc.) - 11

© 2009 IBM Corporation

The RRC team is quite distributed

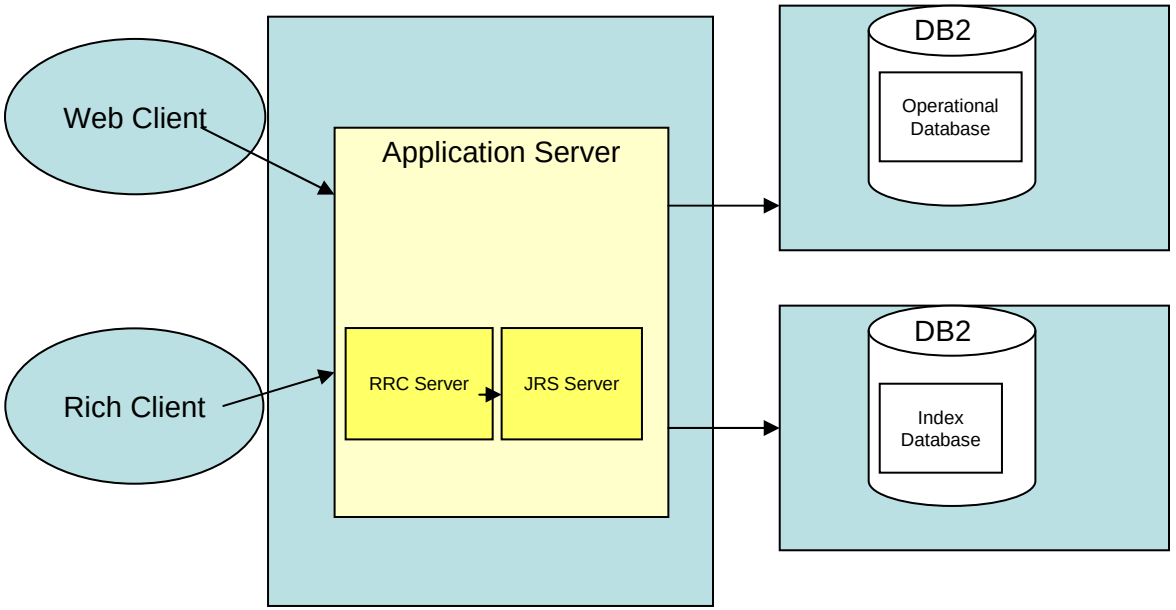
- Number of users: 159 Author licenses assigned, 9 reviewer licenses assigned, 1 system license assigned, total 174 active users.

*note: There are also several users in the UK not shown

© 2009 IBM Corporation

"Drink Our Own Champagne" RRC Server – As of July 23rd, 2009

- Located in Ottawa, Canada
- Single Server machine running
  - Database Data server
  - Database Index server
  - RRC Jazz server
- Database Data server database (backup compressed) 3.5 GB



Recommended:

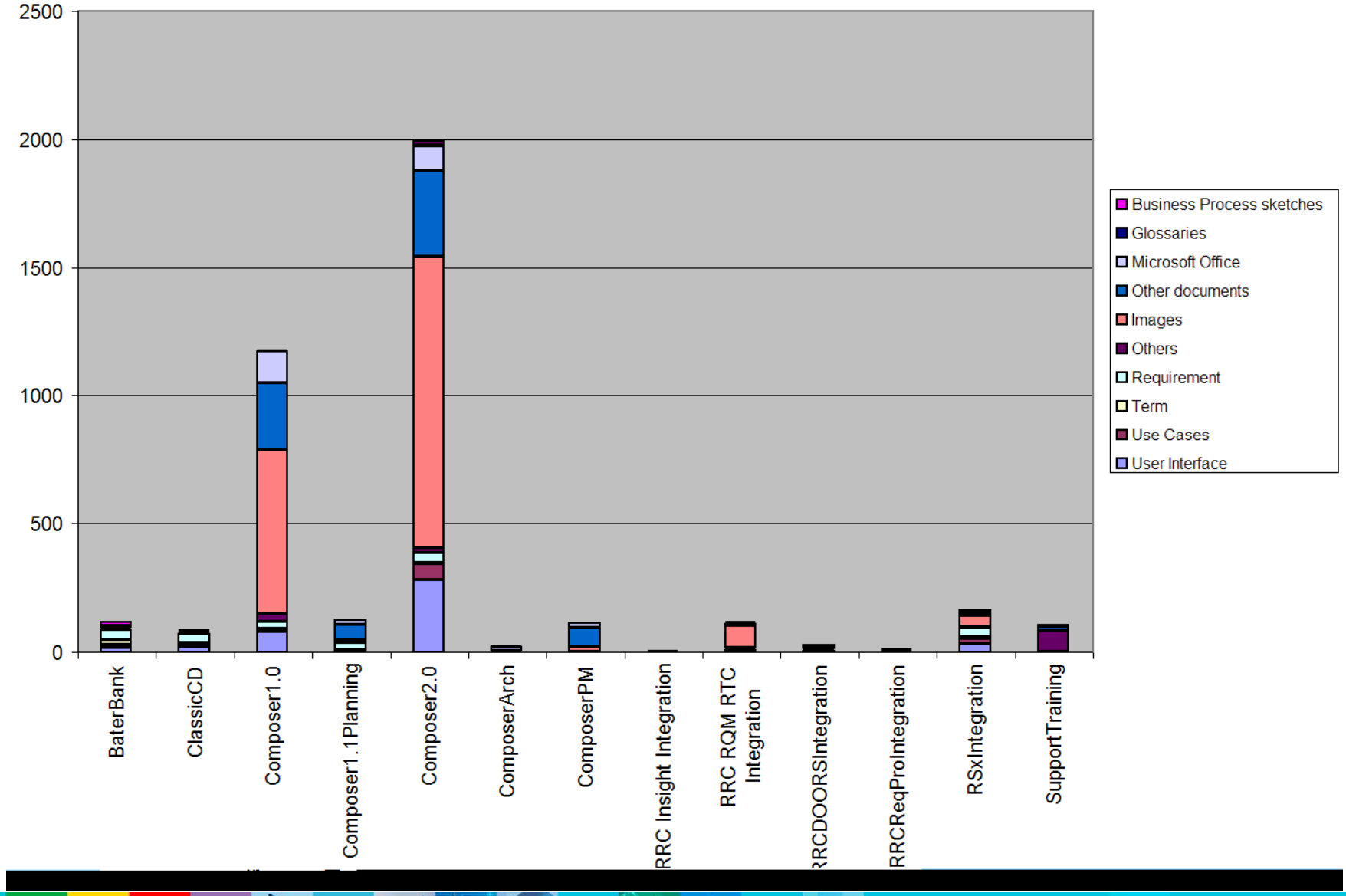
Database for both operational and index databases:
IBM DB2 9.5

Application Server:
IBM WebSphere Application Server 6.1

Minimum Machine Specifications:
2.66 GHz, 4 GB RAM

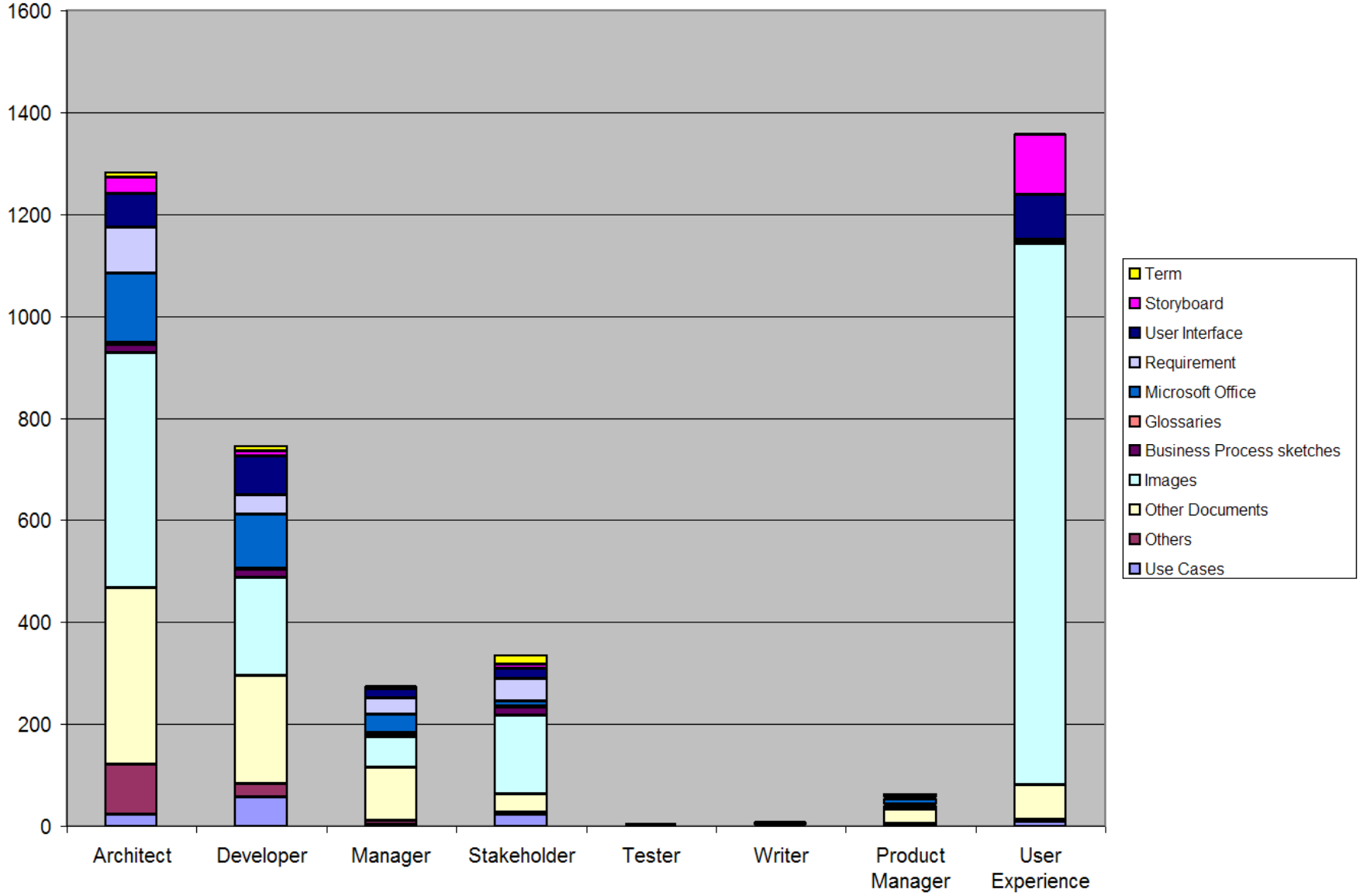
© 2009 IBM Corporation

Artifacts by Projects



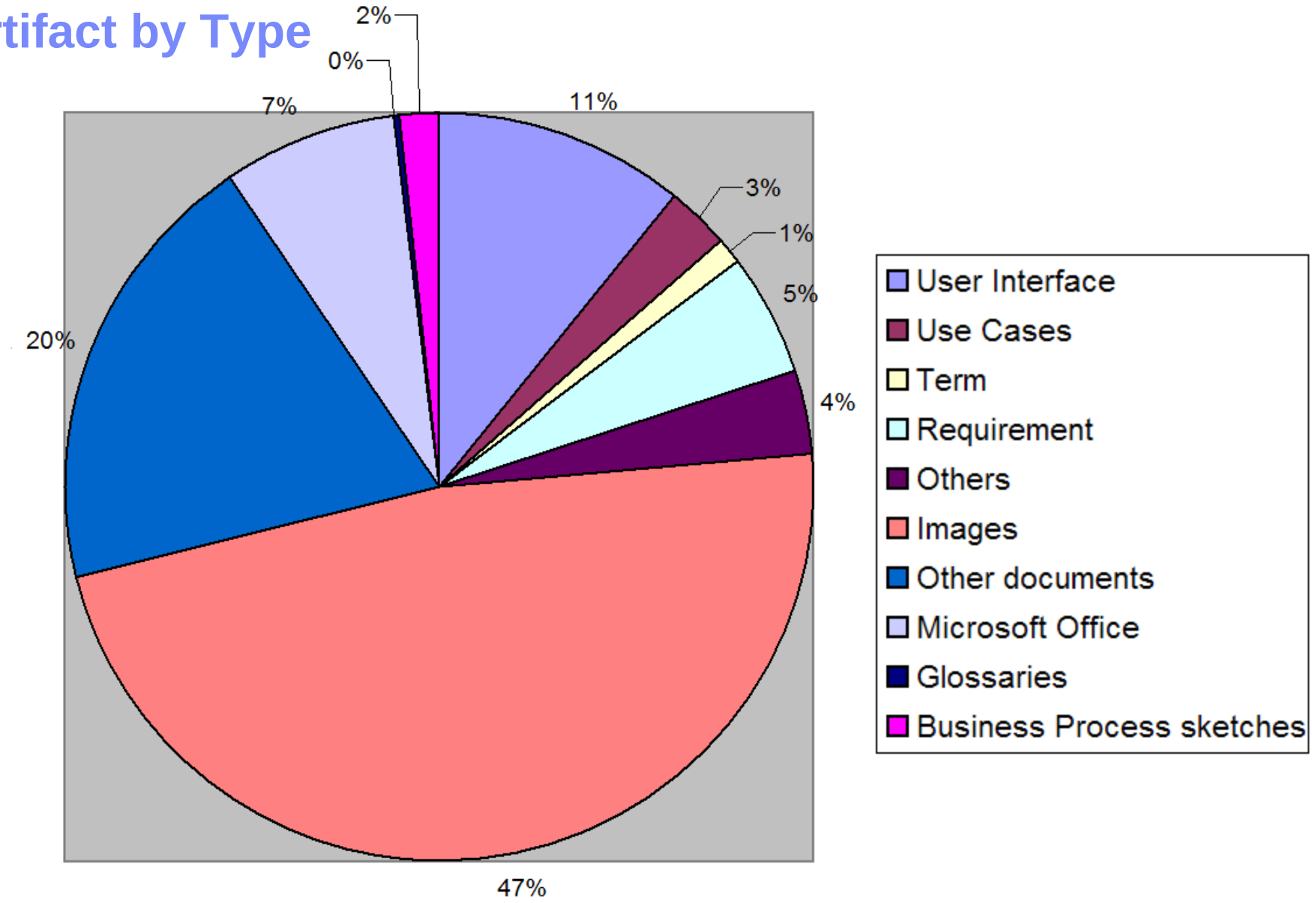
© 2009 IBM Corporation

Artifacts Created by Role – All projects



© 2009 IBM Corporation

Artifact by Type



© 2009 IBM Corporation

The iterative development process

- Iterations
  - Overlap: Planning begins for the next iteration during the SVT testing of the previous iteration
  - Vary in length:
    - Different work effort needed for Inception, Elaboration, Construction and Transition
    - Team Events
      - Rational Software Conference
      - Holidays, Etc
  - Each milestone is consumable and includes
    - Planning,
    - Requirements,
    - Design,
    - Development,
    - Function and
    - System Testing
  - Milestone 6 – is a Beta milestone

	Start Date	End Date
Milestone 1	11/18/08	12/19/2008
Milestone 2	01/05/09	2/20/2009
Milestone 3	02/09/09	4/3/2009
Milestone 4	03/23/09	6/12/2009
Milestone 5	05/25/09	8/7/2009
Milestone 6	07/27/09	9/28/2009
Milestone 7	…	…

https://jazz.net/projects/rational-requirements-composer/release-plan-2.0/

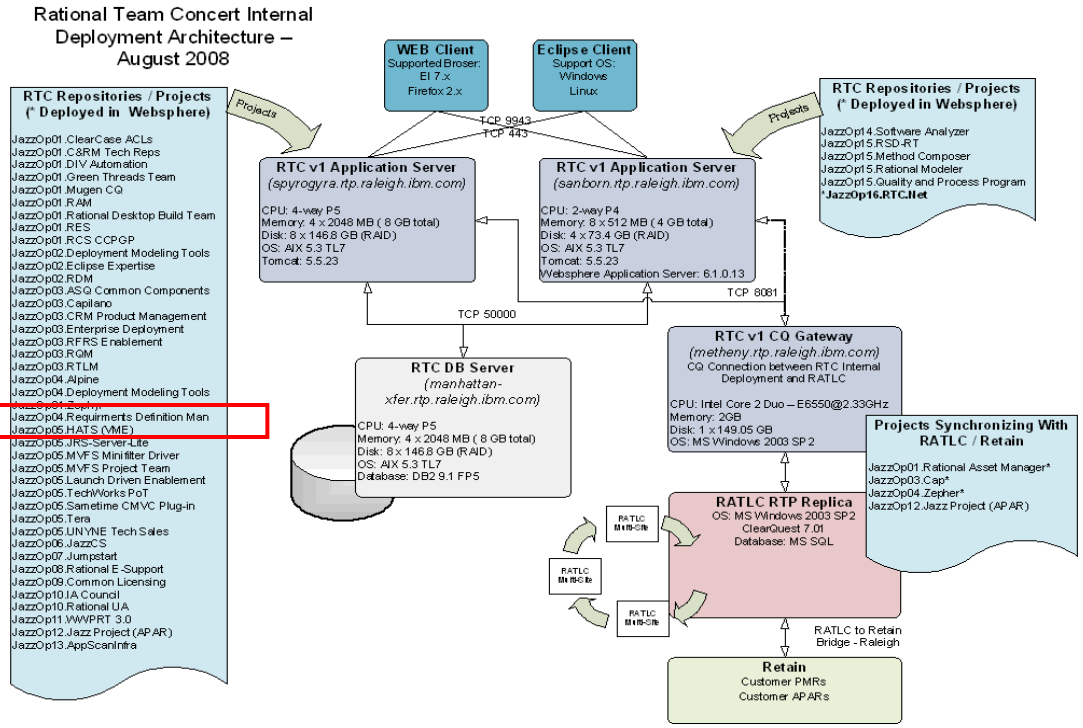
- Rational Team Concert helped with our process
  - Highly productive work items, code library system, build system
  - Process enablement
    - required fields and approvers (changes based on place in cycle)

© 2009 IBM Corporation

IBM's Internal Deployment of Team Concert

- In Rational, today we have 2,000+ developers and testers
  - 80 development projects, two application servers (hosting 20 RTC server instances), one DB2 server
- In IBM we have 40,000 developers - By end of 2009 1/3 of these will be using RTC
  - And this is NOT being mandated, but rather through viral adoption and real productivity gains

Requirements Definition & Management



© 2009 IBM Corporation

The Product Backlog and Future functionality

RTC

Product Backlog

Defects, Change Requests

User Stories, Scenarios

RRC

Stakeholder Requests

Vision Document

Use-Case Model

Supplementary Specification

Design Specifications

User Documentation Specifications

© 2009 IBM Corporation

Open Commercial Software Development of RRC V2

- Many of the open source practices without open source licensing
- Customers and community influencing the direction of products through direct, early, and continuous conversation
  - Communication with development teams
  - Track progress of builds and milestones
  - Provide direct feedback on what is working, what isn't
  - Submit and track defect and enhancement requests (work items)
  - Full visibility to detailed product plans, status and progress
  - Use of product via web interface
  - Transparency allows customers to become part of continuous feedback loop that drives development decisions
  - Opportunity to provide feedback early and often permits customer understanding and influence of release direction and priorities before decisions are finalized
- Customer-driven development

© 2009 IBM Corporation

Jazz.net Project – Rational Requirements Composer

Home | About Jazz | Projects | Downloads | Forums | Library | Community | Robin Bater | Edit Profile | Log Out

Jazz Projects

Type to search

Projects >

Rational Requirements Composer

Release Plan

Related Resources

Rational Requirements Composer 1.0.0.1 Online Help

Library

Rational Requirements Composer

Collaborative requirements definition and refinement

Rational Requirements Composer provides a platform for collaborative requirements definition that enables business analysts, client stakeholders and development teams to define requirements, and collaborate with each other.

What's new in RRC 2.0 Beta 2

This beta release includes improvements in editors, dashboards and viewlets; report and document generation; collections and project snapshots for defining baselines; and workflow for review and approval. See a more comprehensive list of the features and benefits of Beta 2

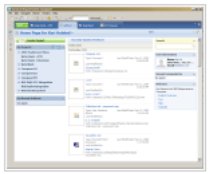
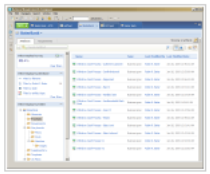
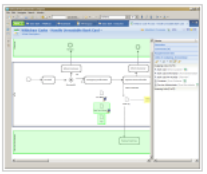
Download RRC 2.0 Beta 2 >

Collaborate to define textual and visual requirements

- Elicit, capture, and refine requirements by using a variety of definition techniques: text, storyboards, use-case diagrams, UI sketches, and business process diagrams
- Use collaboration capabilities to review and comment on requirements among team members
- Organize and create meaningful relationships between requirement artifacts
- Create project glossaries to develop and manage a common vocabulary
- Gather requirements into specification documents
- Store artifacts in a Jazz-based, secure, RESTful server, with artifact access that is controlled by role definitions

Screenshots:

User home page | Create and view artifacts | Collaborate and refine

→ Buy at ibm.com

IBM.

Download
1.0.0.1 Release
May 25, 2009

Download 2.0 Beta 2
September 21, 2009
This is a pre-release Early Access Trial beta of Rational Requirements Composer.

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

See the project dashboard

Explore detailed iteration plans

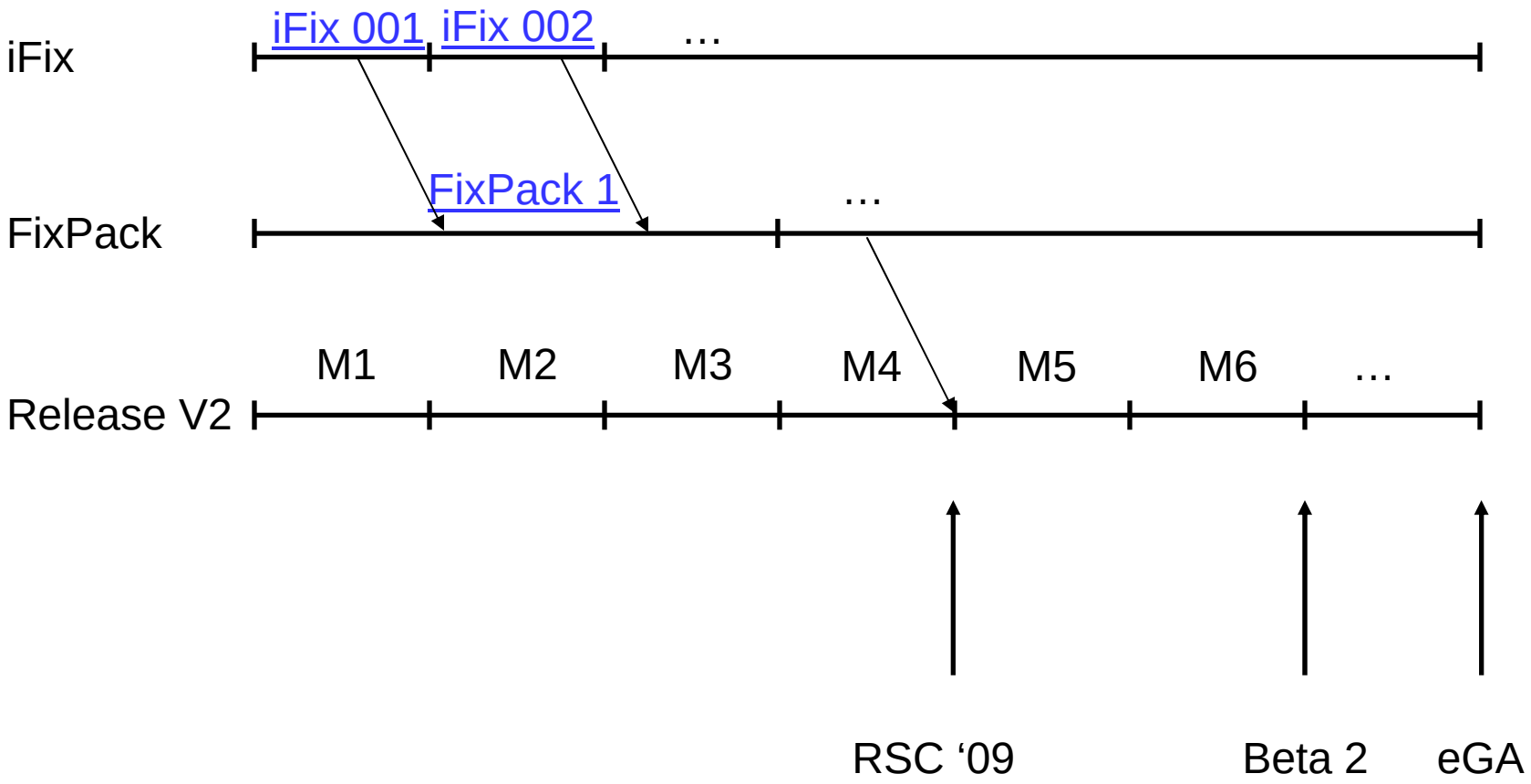
View reports

Review release plans

© 2009 IBM Corporation

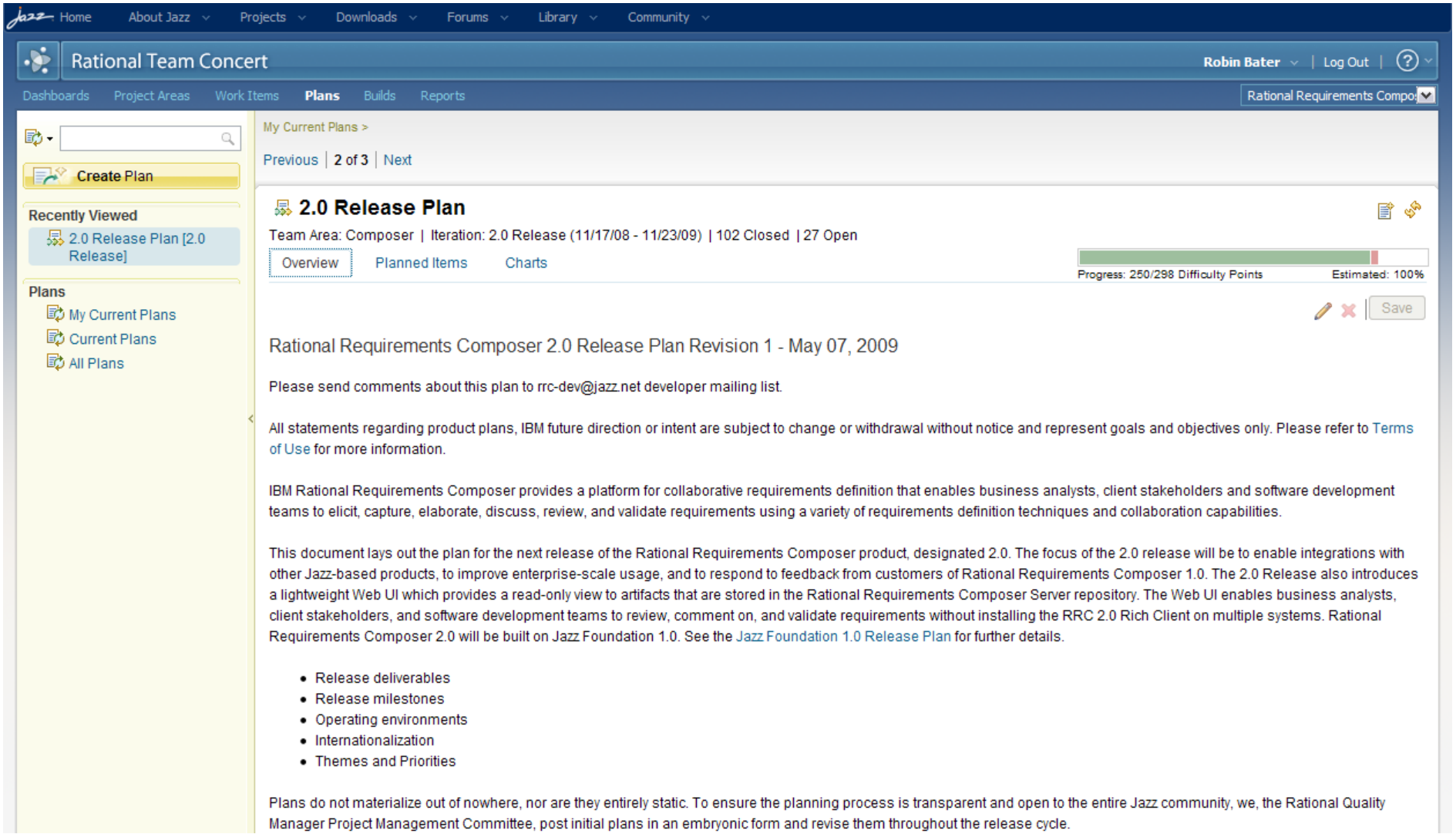
RRC Releases

- Backlog (iFix, FixPack) and Release (V2) themes artifacts are mapped to the various iteration plans



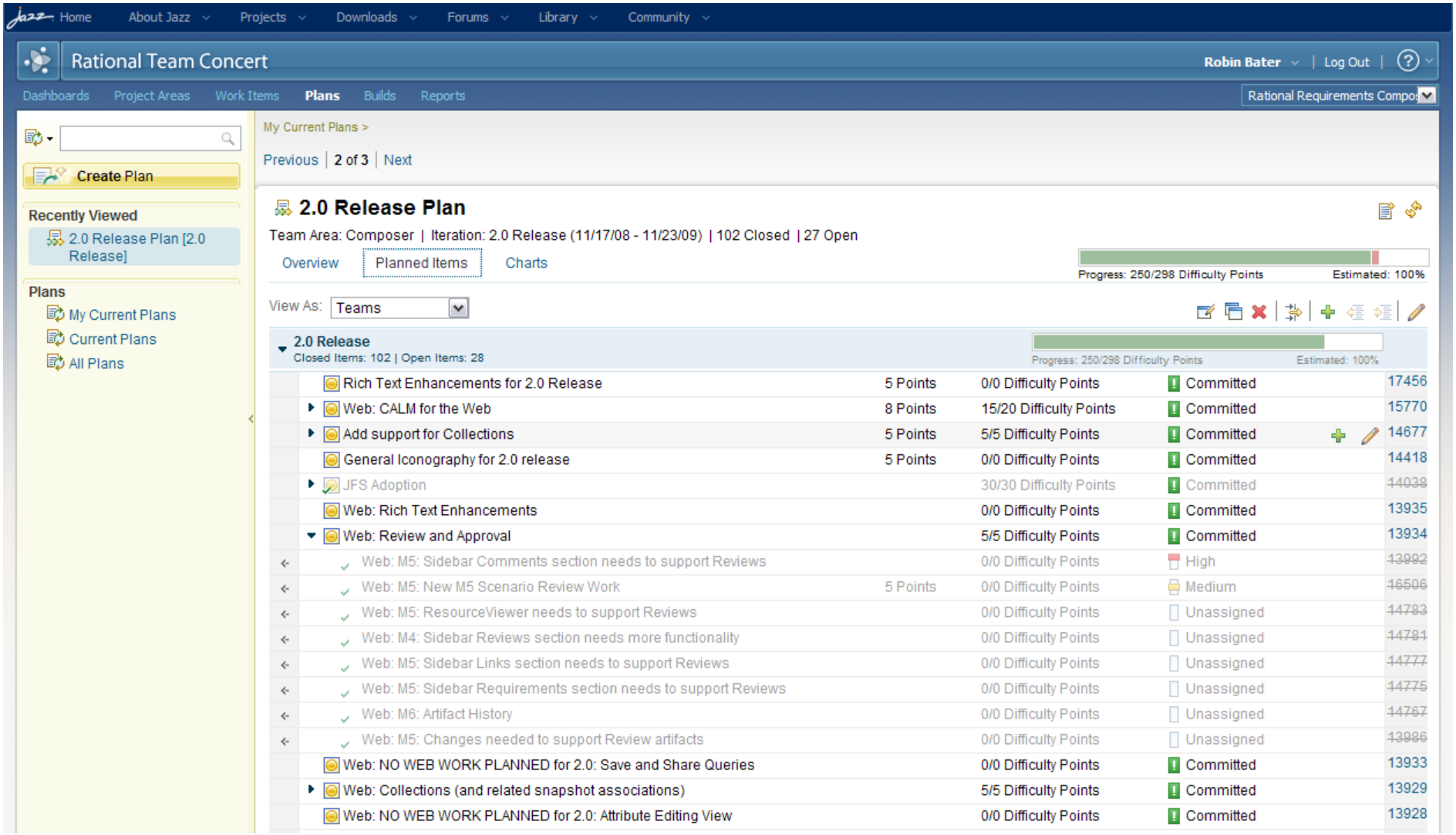
© 2009 IBM Corporation

RRC V2 Release Plan



© 2009 IBM Corporation

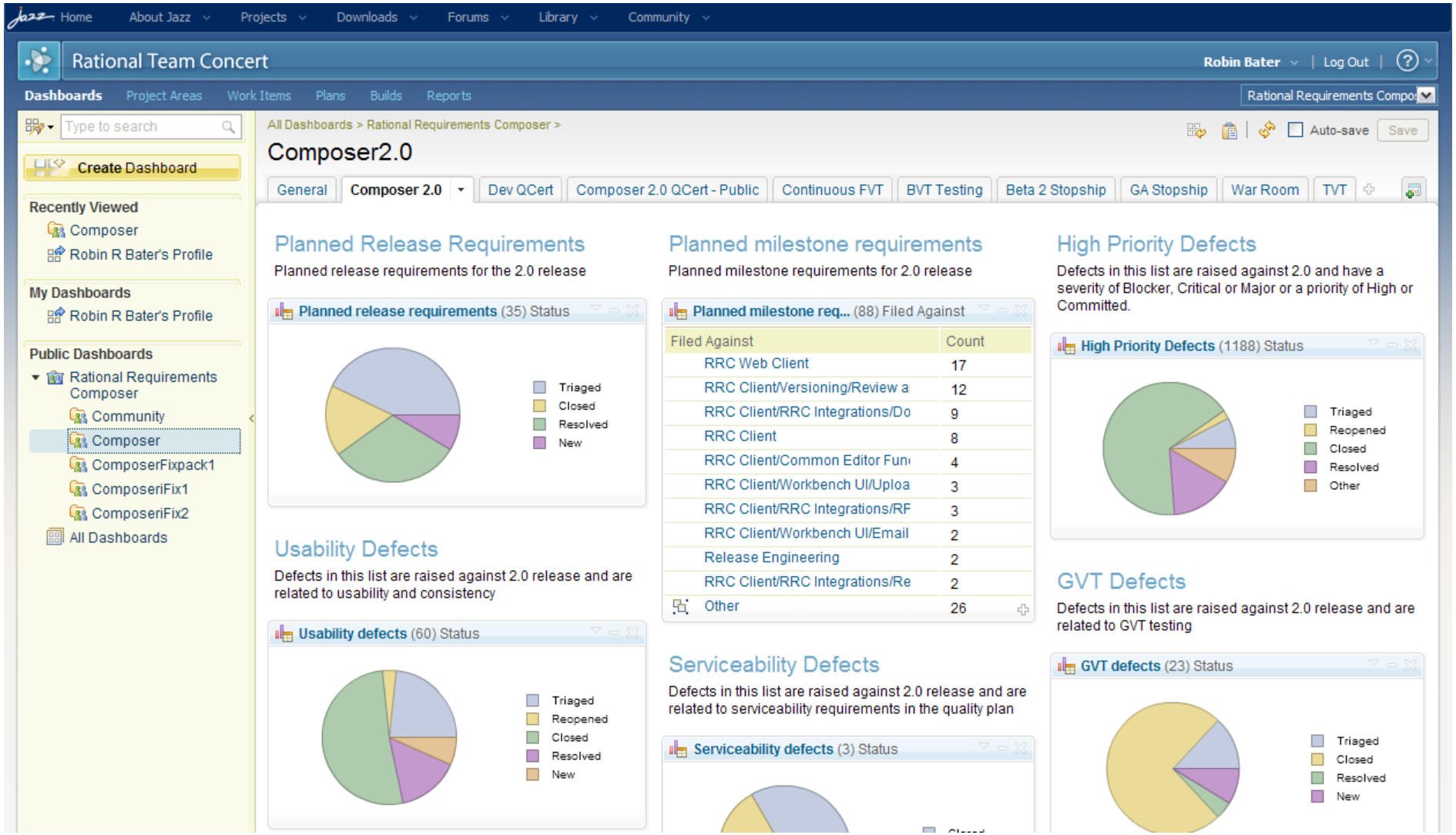
RRC V2 – Release Planned Items



© 2009 IBM Corporation

RRC V2 – Composer 2.0 RTC Dashboard

https://jazz.net/jazz03/web/projects/Rational%20Requirements%20Composer#action=com.ibm.team.dashboard.viewDashboard&team=Composer&tab=_Yk8hAL41Ed2E7tg3d2LZqA



© 2009 IBM Corporation

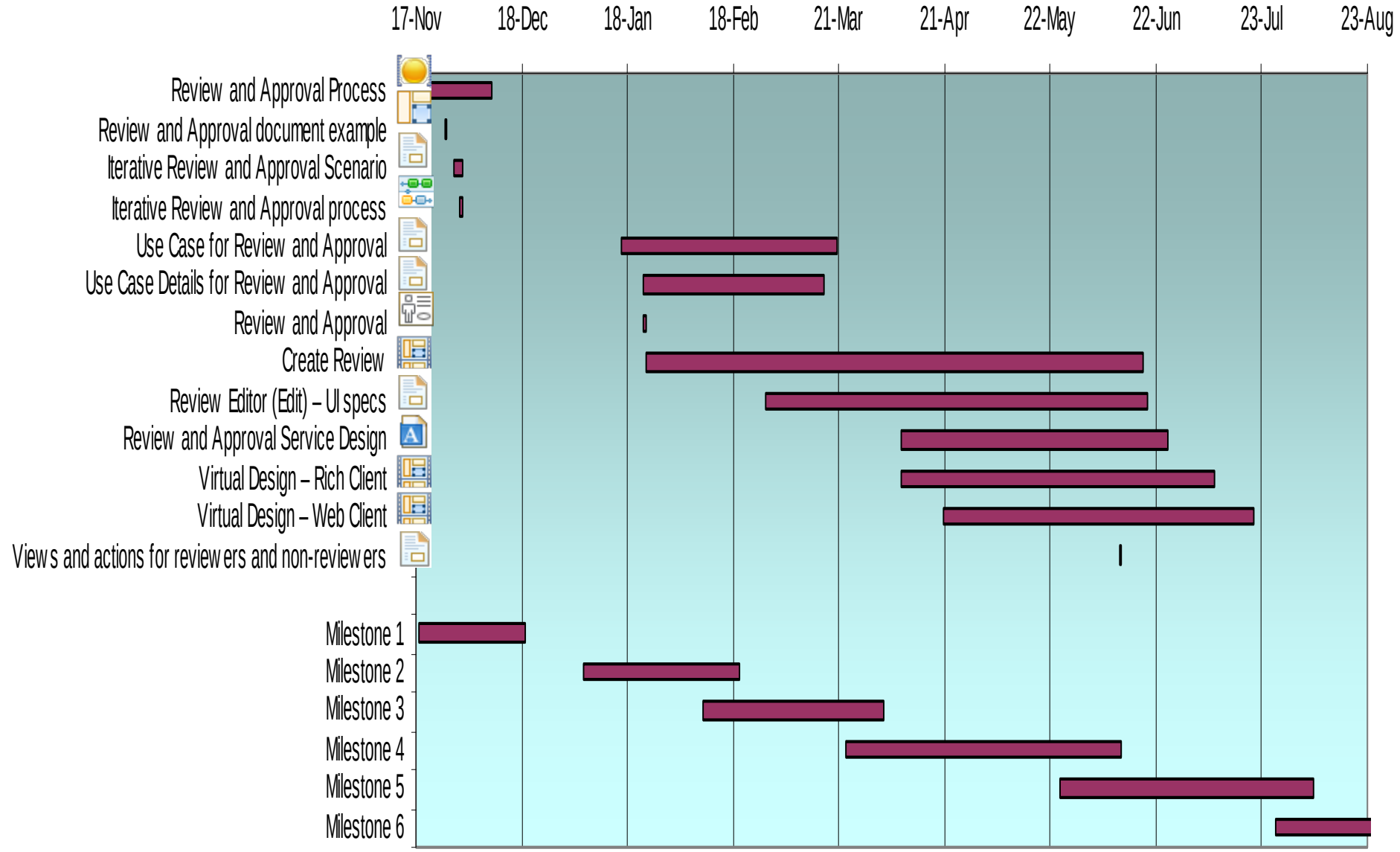
Drinking Our Own Champagne

- The following slides show the evolution of the RRC V2 Review and Approval feature
  - Using IBM Rational Requirements Composer V1.0.0.1
- The review and approval process is firstly described in textual scenario
  - Composer1.1Planning project
- The UX and dev team members then developed increasingly realistic versions of the mock-ups and identified "how it's really going to work".
  - Composer 2.0 project
  - The UX team have benefitted greatly from deep, on-going conversations with developers (and including product management now and again)

© 2009 IBM Corporation

Team-based artifact creation

more effective, with easier collaboration, and communication



© 2009 IBM Corporation

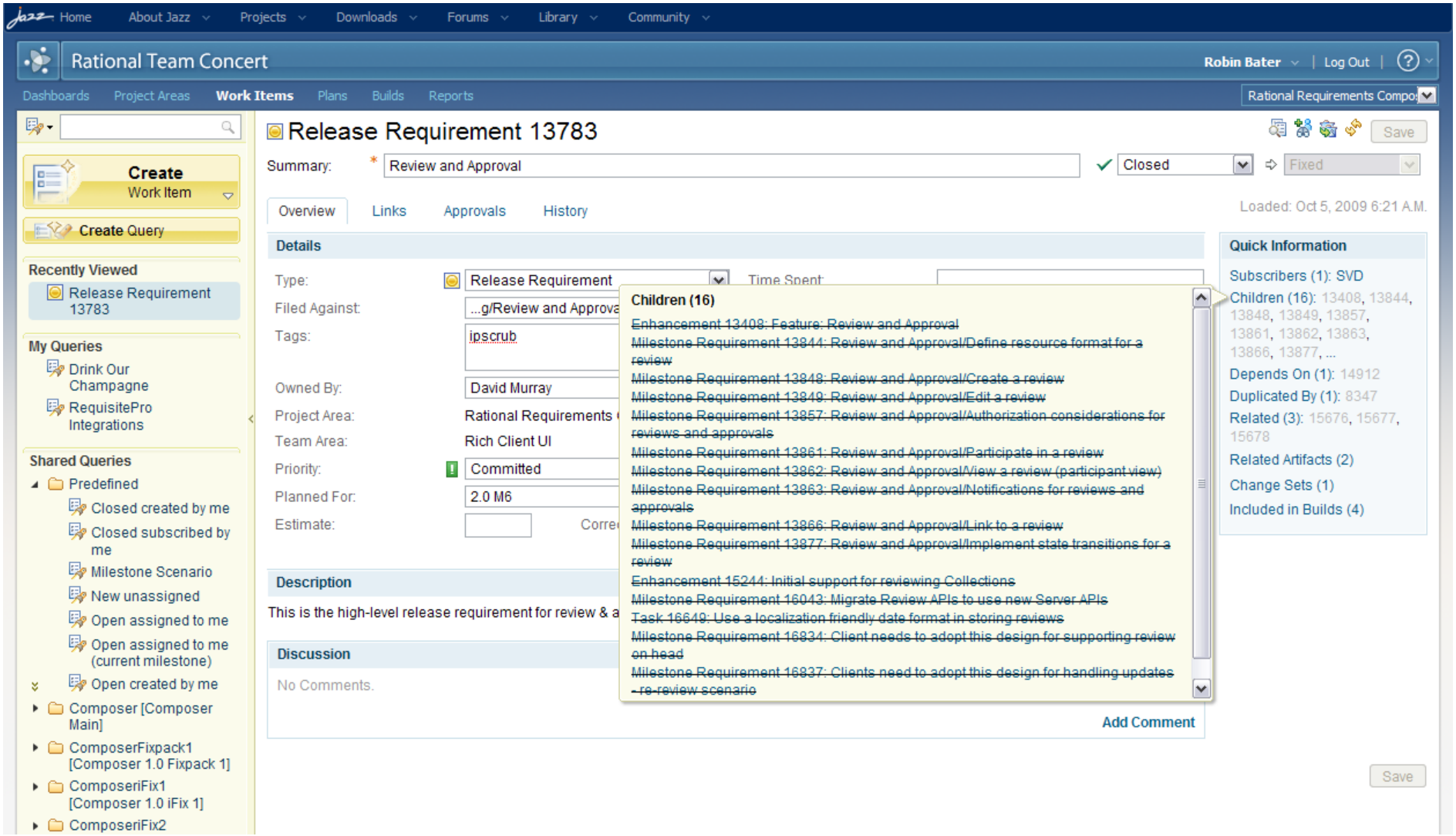
RRC Release V2 “Champagne” artifacts - Details

Artifact		Created On	Last Modified	Revisions	Links
Review and Approval Process	Requirement	Nov 17,2008	9-Dec-08	59	(12) (6) (11)
Iterative Review and Approval Scenario	Document	Nov 28, 2008	25-Nov-08	18	(1) (4) (7)
Iterative Review and Approval process	Business Process Sketch	Nov 29, 2008	30-Nov-08	6	(0) (0) (0)
Review and Approval document example	Sketch	Nov 25, 2008	30-Nov-08	5	(2) (1) (0)
Review and Approval	Use Case Diagra	Jan 22, 2009	20-Mar-09	4	(0) (3) (0)
Use Case for Review and Approval	Document	Jan 16, 2009	16-Mar-09	33	(0) (9) (0)
Use Case Details for Review and Approval	Document	Jan 22, 2009	23-Jan-09	46	(8) (6) (0)
Create Review	Storyboard	Jan 23, 2009	18-Jun-09	79	(55) (10) (0)
Virtual Design – Rich Client	Storyboard	Apr 8, 2009	19-Jun-09	16	(34) (38) (0)
Virtual Design – Web Client	Storyboard	Apr 20, 2009	25-Jun-09	8	(12) (13) (0)
Review Editor (Edit) – UI specs	Document	Feb 27, 2009	9-Jul-09	45	(4) (4) (0)
Views and actions for reviewers and non-reviewers	Document	Jun 11, 2009	20-Jul-09	2	(0) (1) (0)
Review and Approval Service Design	MS Word Doc	Apr 8, 2009	11-Jun-09	19	(6) (0) (0)

© 2009 IBM Corporation

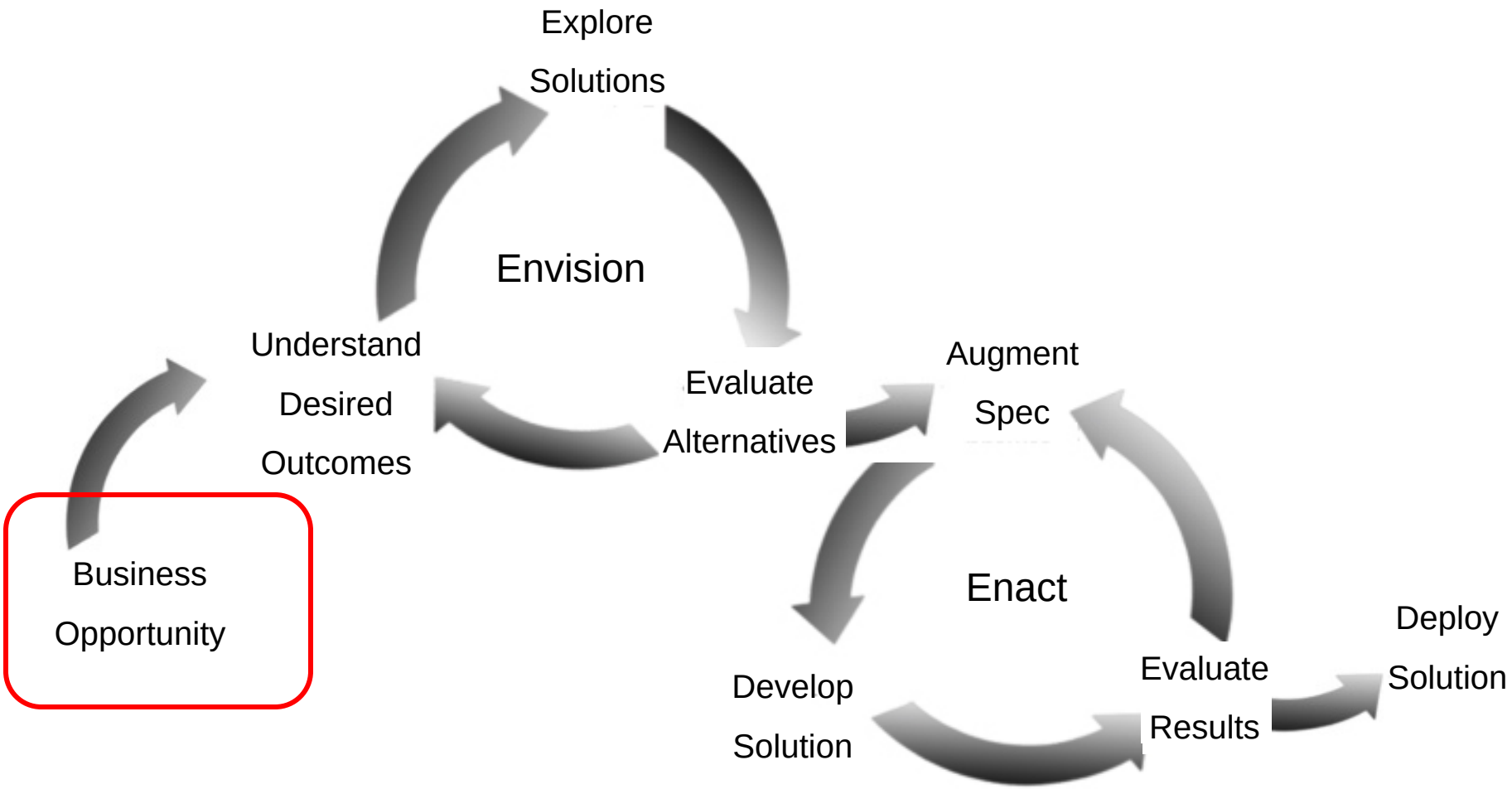
RRC V2 – Review and Approval RTC Release Requirement

https://jazz.net/jazz03/web/projects/Rational%20Requirements%20Composer#action=com.ibm.team.workitem.viewWorkItem&id=13783



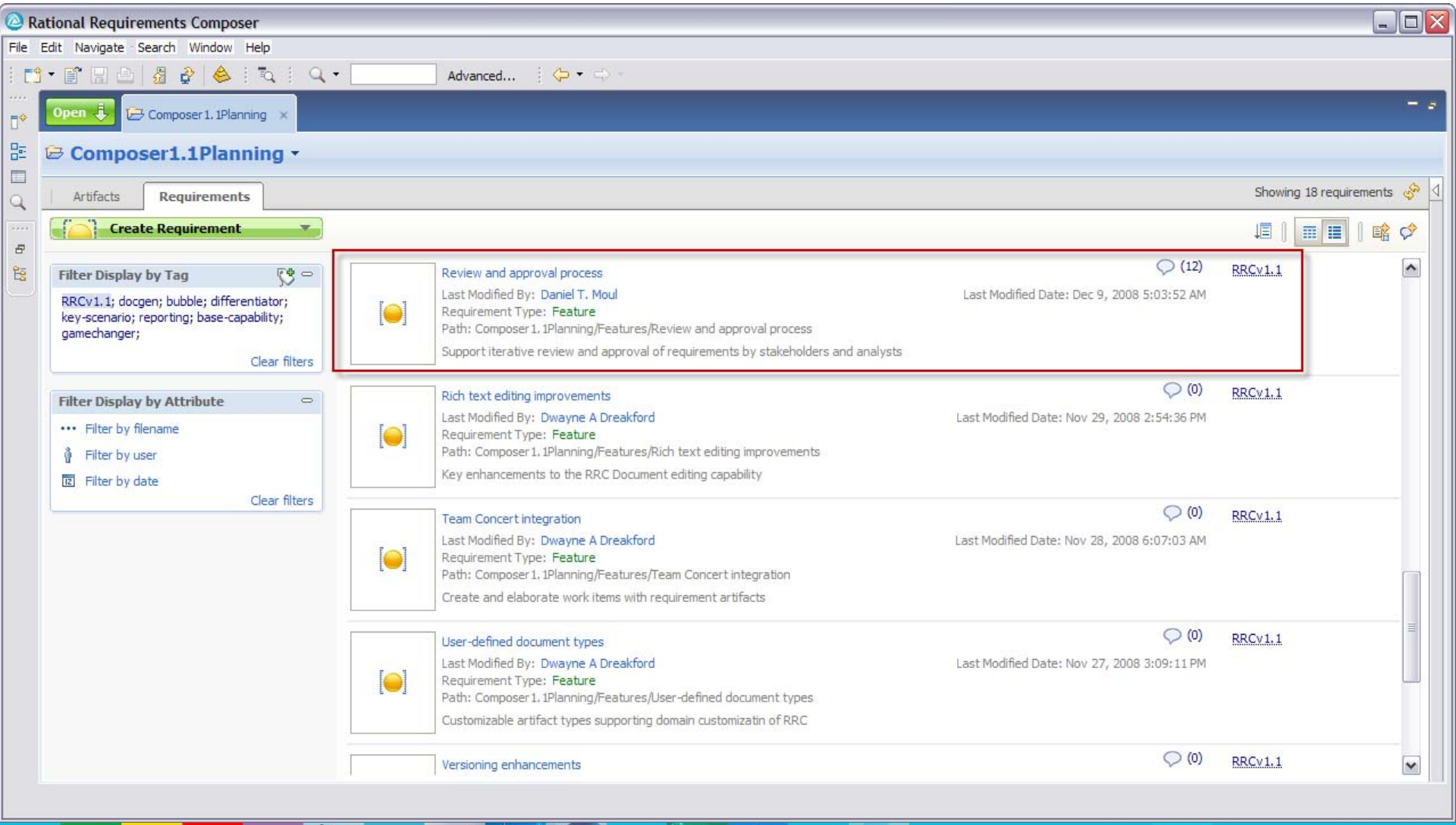
© 2009 IBM Corporation

A roadmap for building the right solution



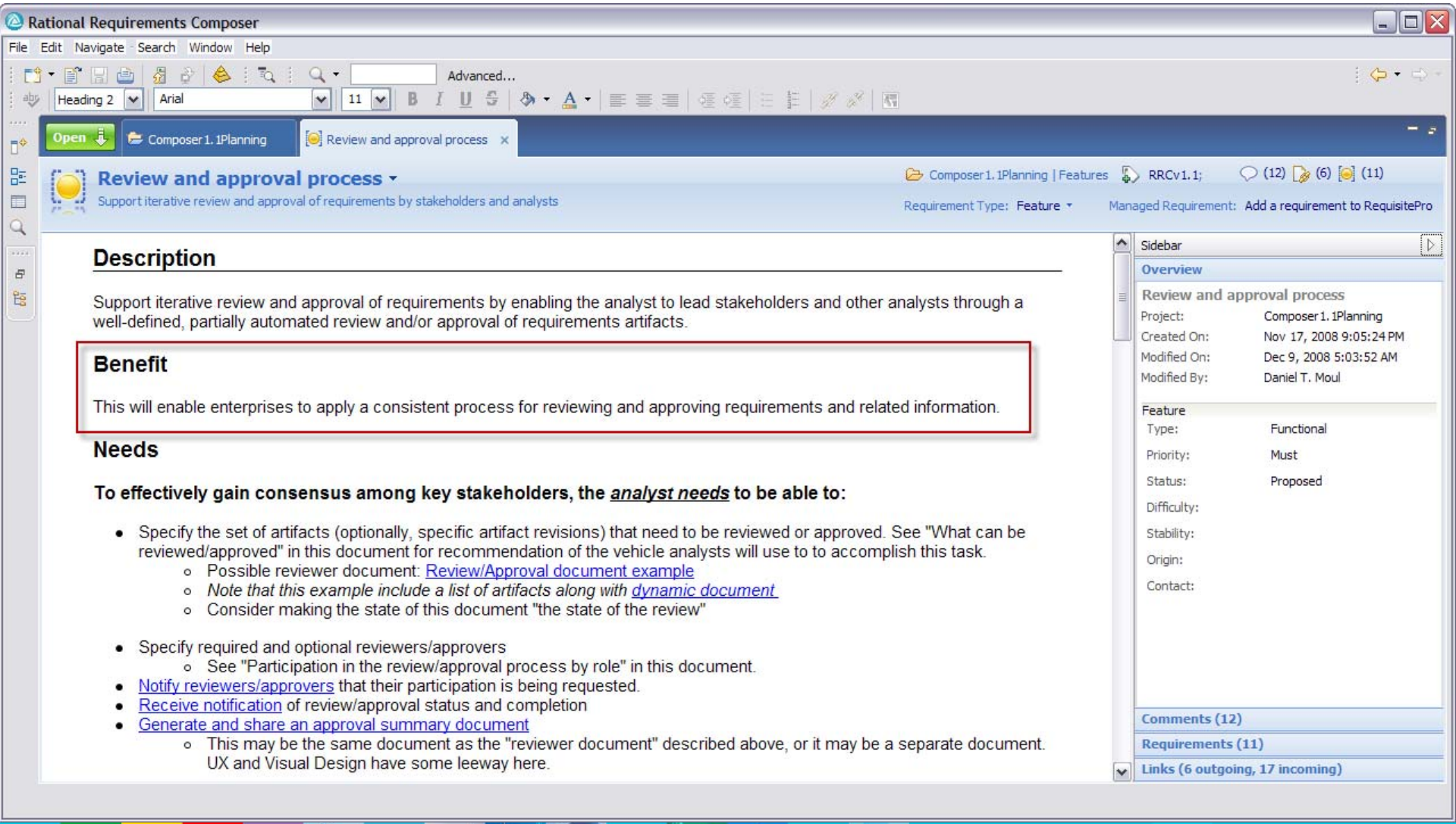
© 2009 IBM Corporation

Dashboard: V1 customer feedback defined list of features



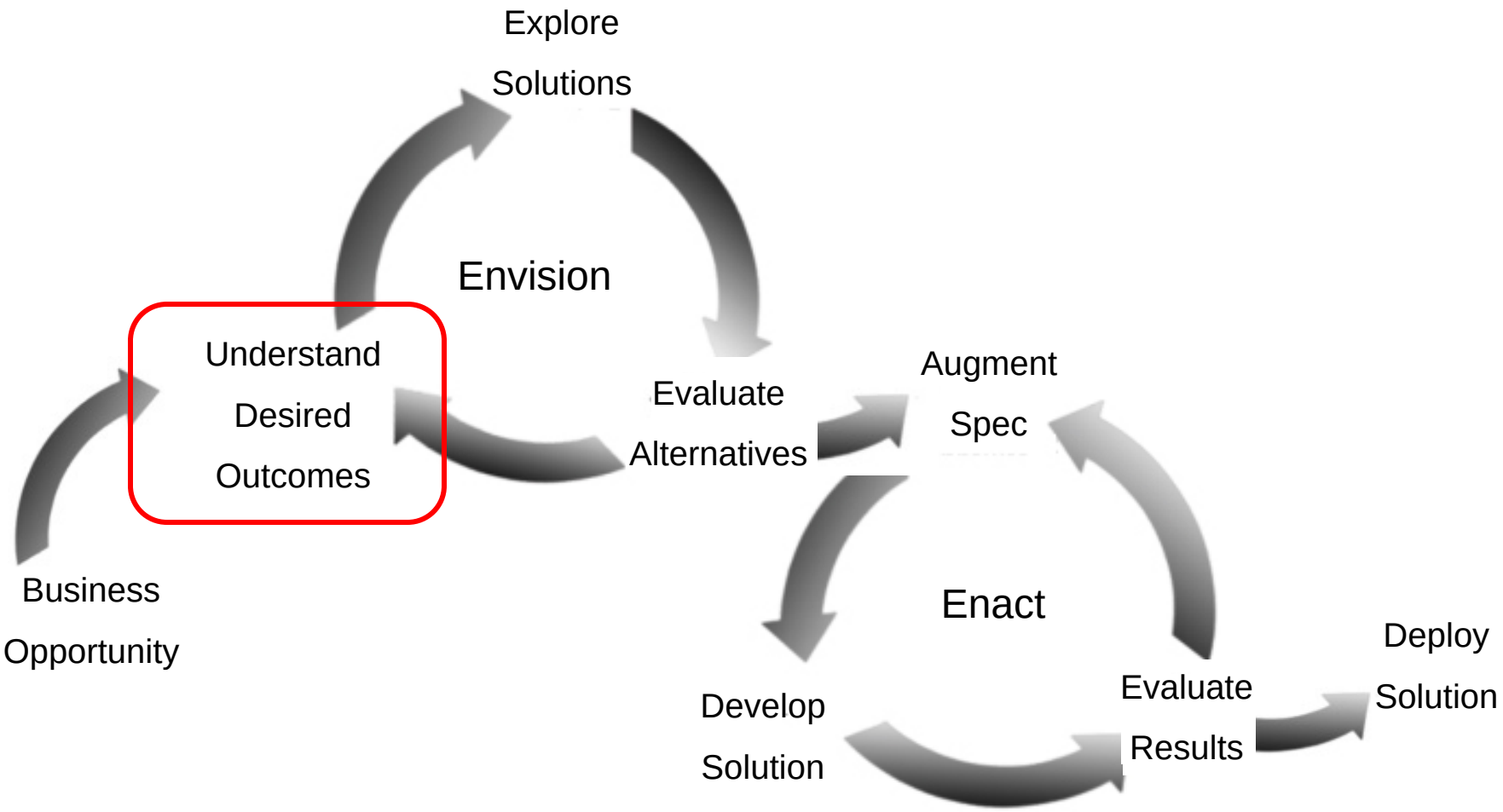
© 2009 IBM Corporation

Feature: Review and Approval process



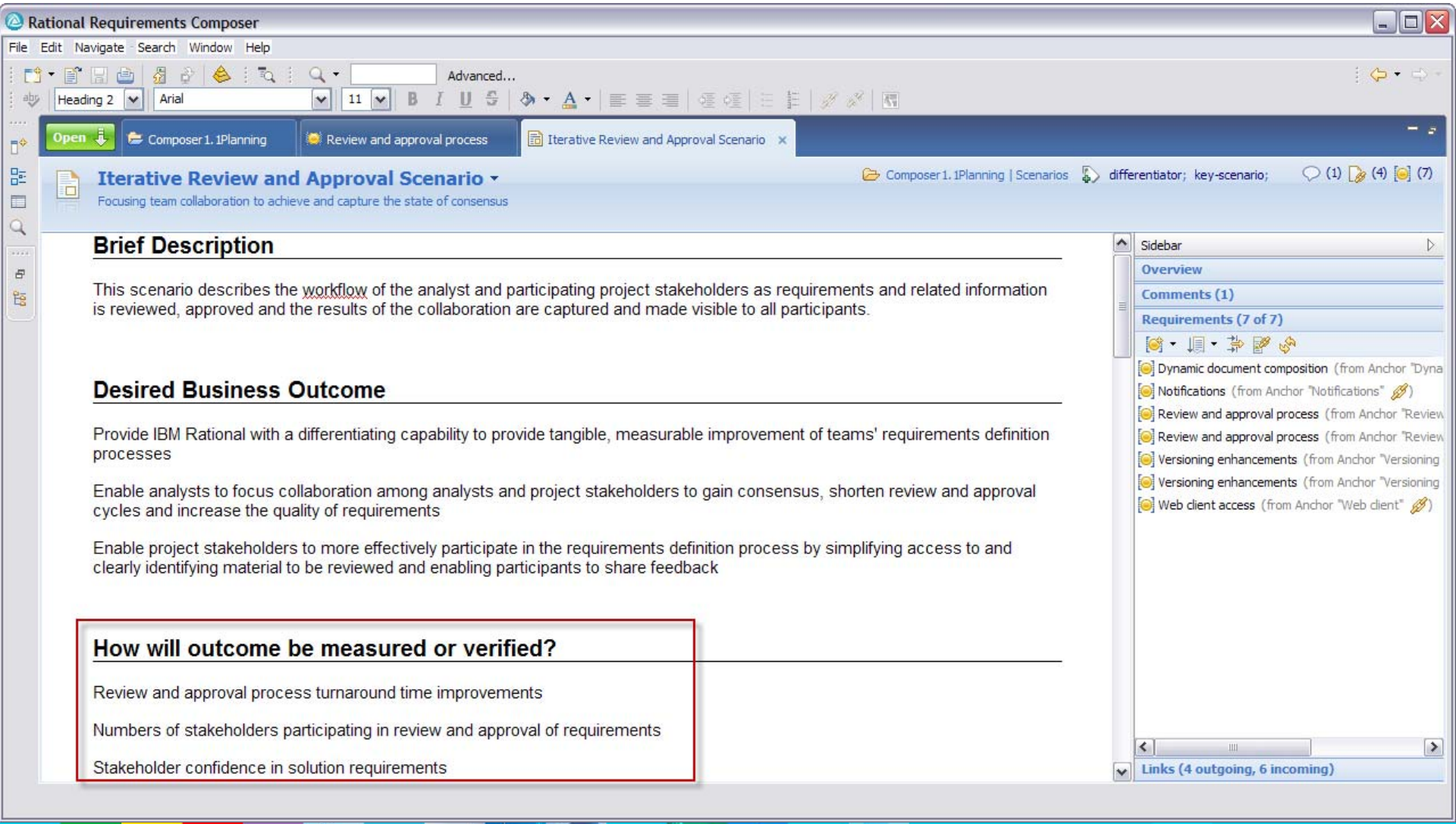
© 2009 IBM Corporation

A roadmap for building the right solution



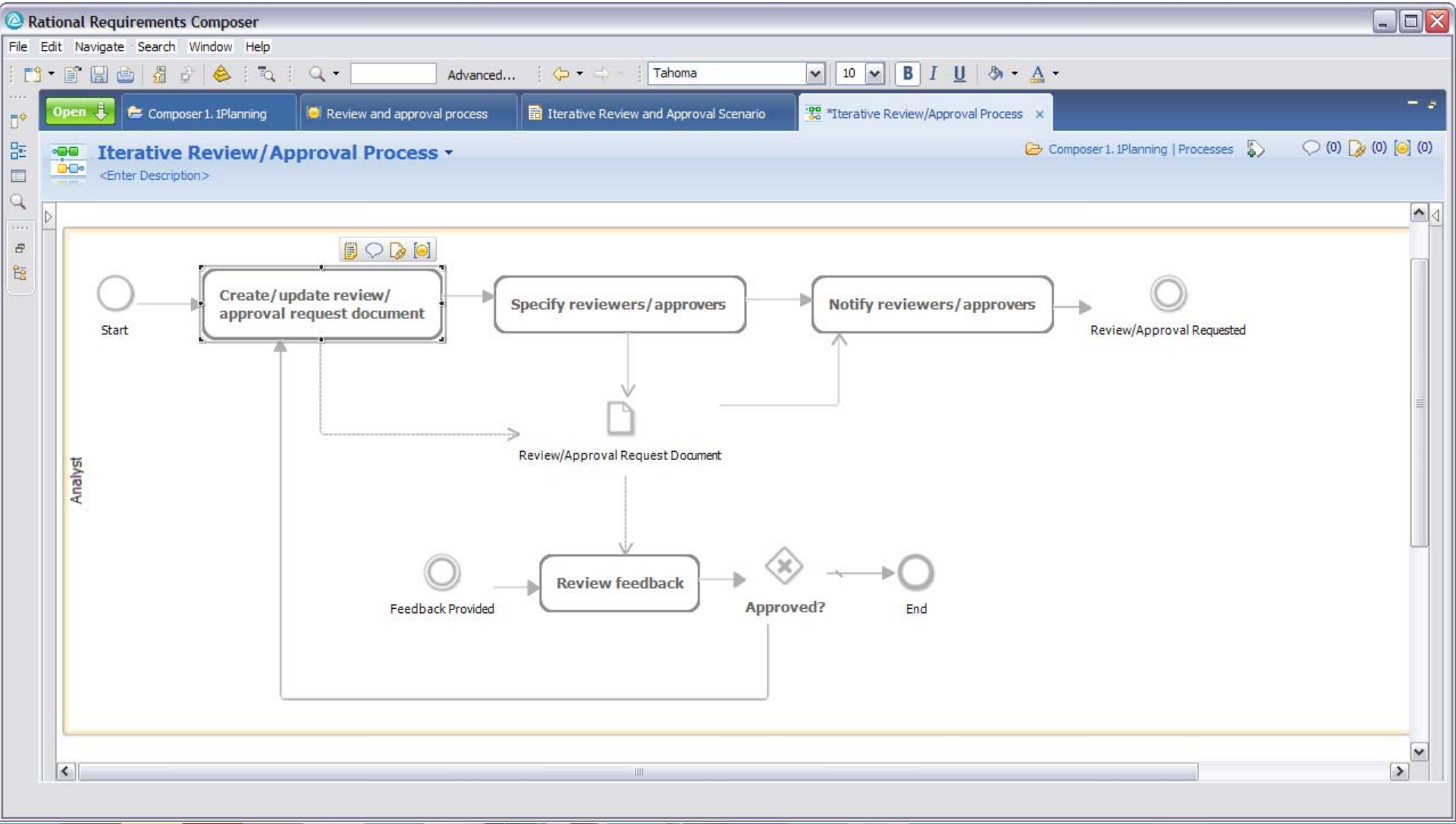
© 2009 IBM Corporation

Document: Iterative Review and Approval scenario



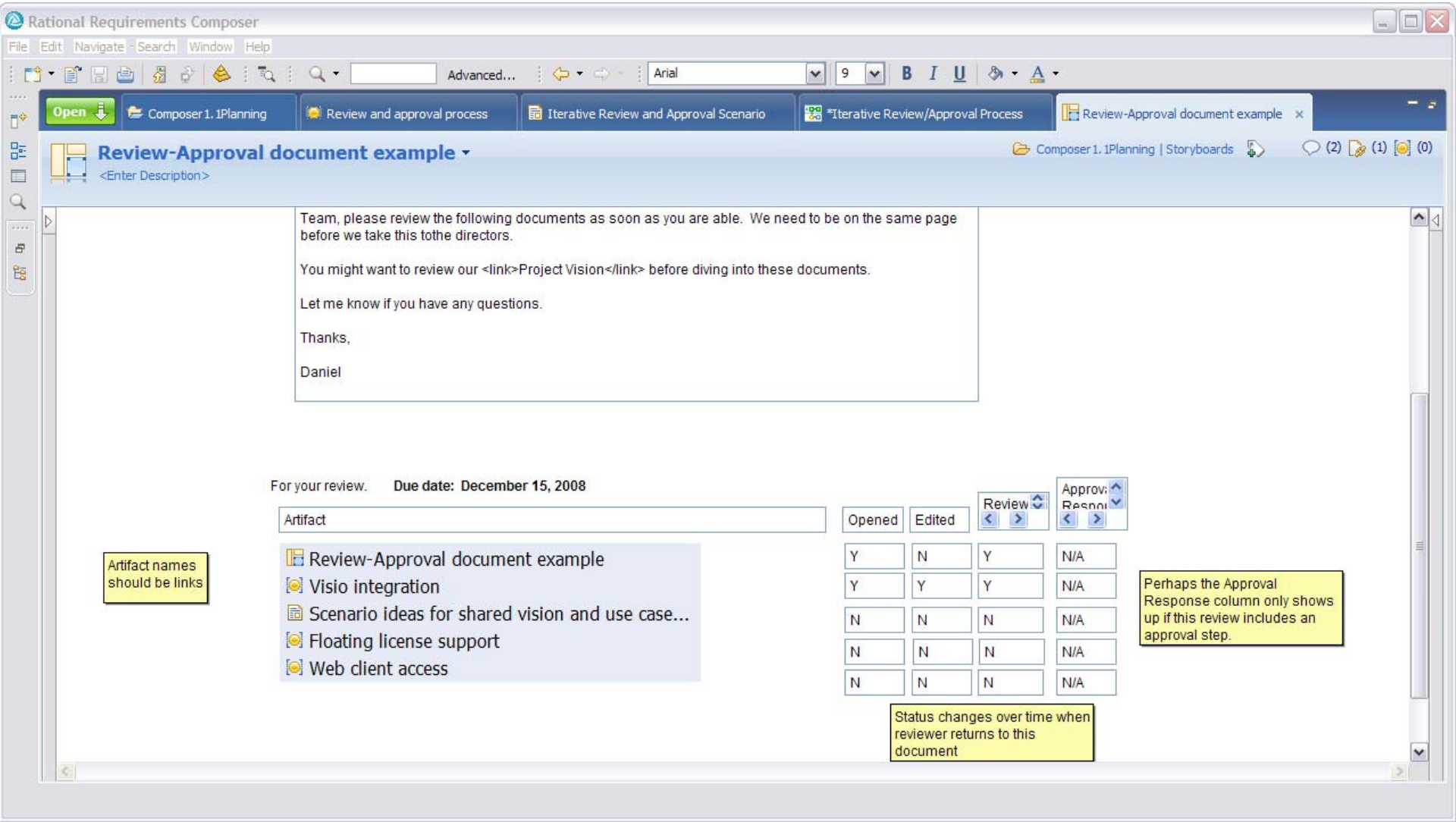
© 2009 IBM Corporation

Business Process Sketch: Iterative Review and Approval process



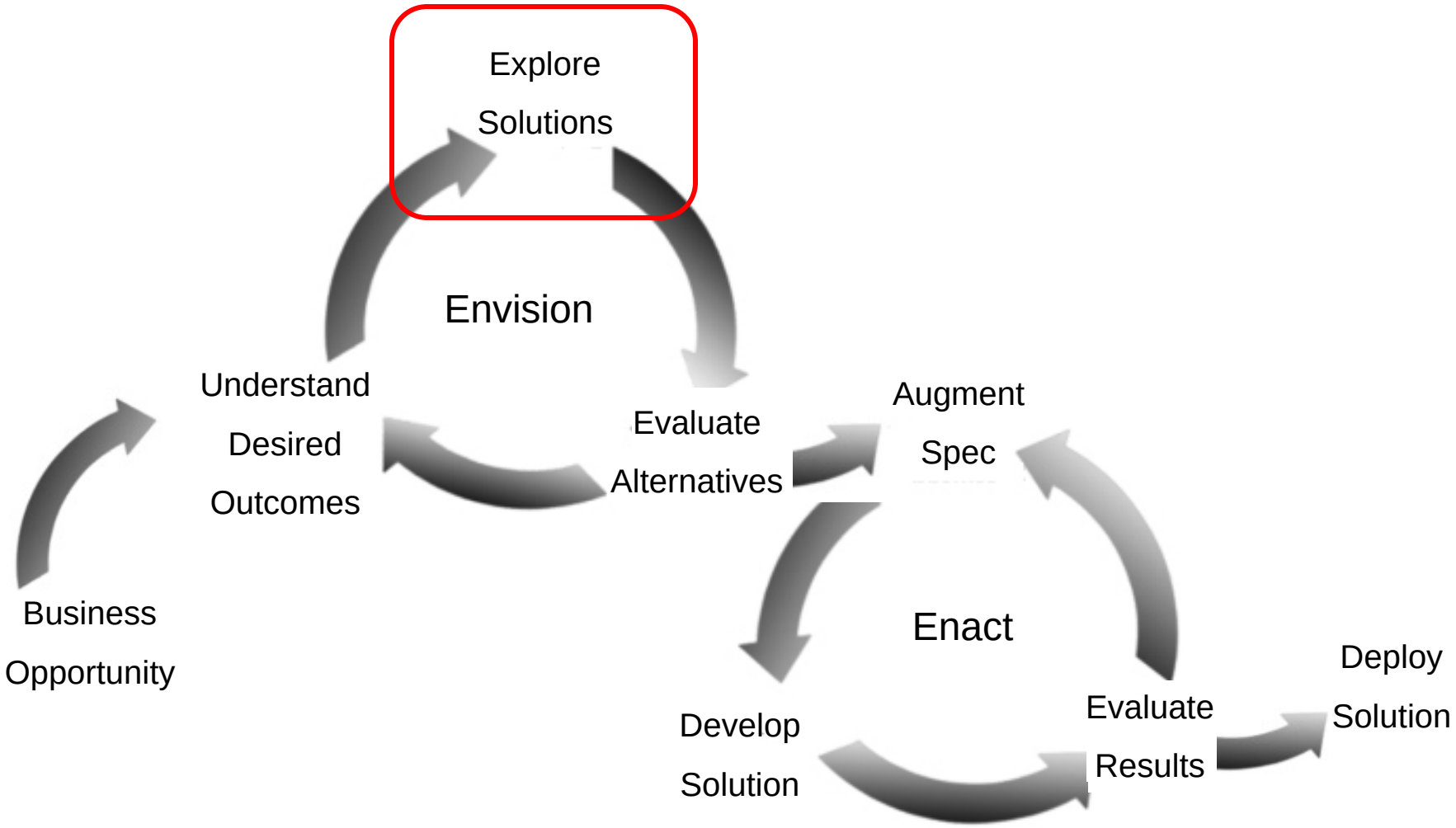
© 2009 IBM Corporation

Sketch: Review and Approval document example



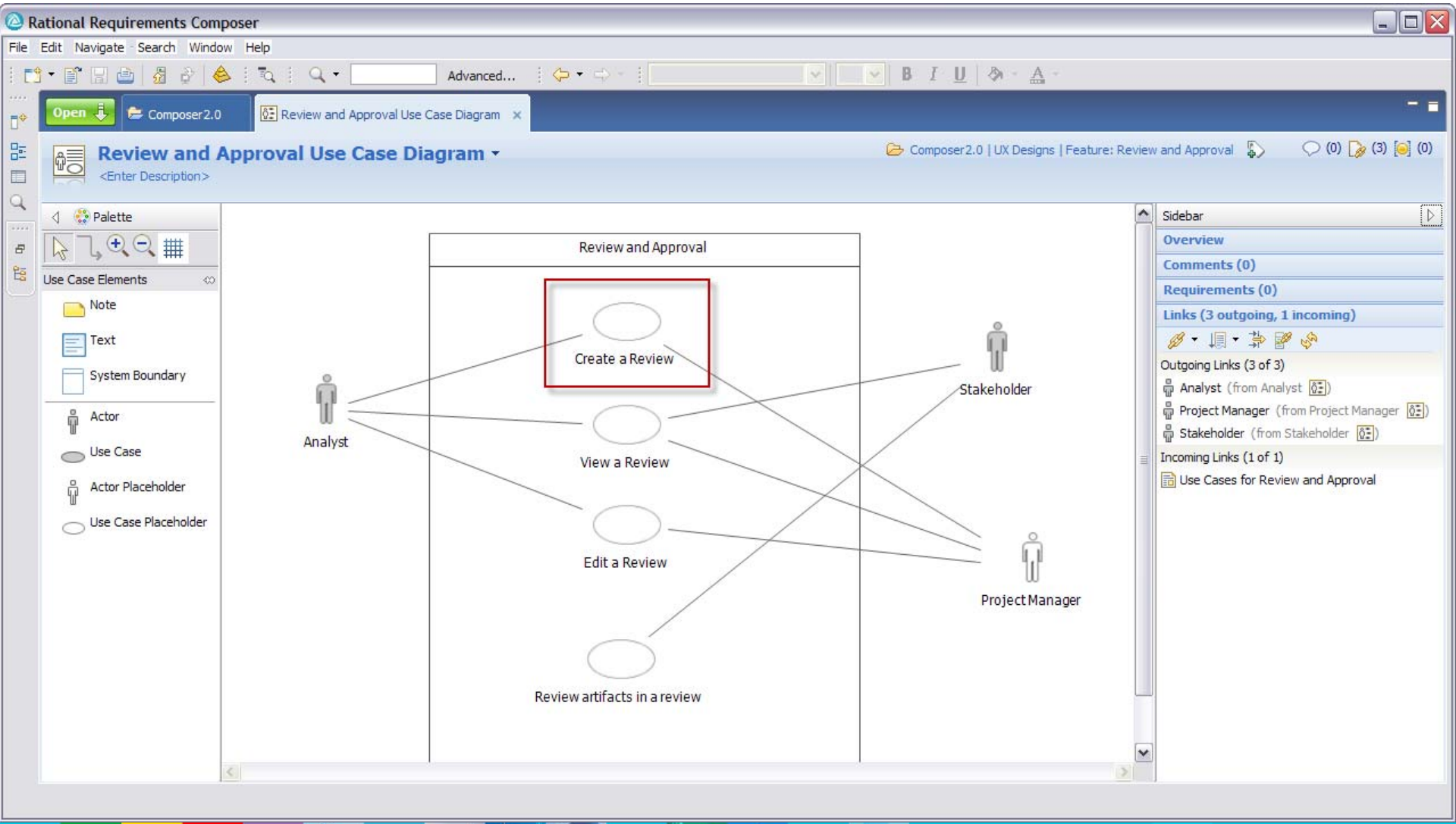
© 2009 IBM Corporation

A roadmap for building the right solution



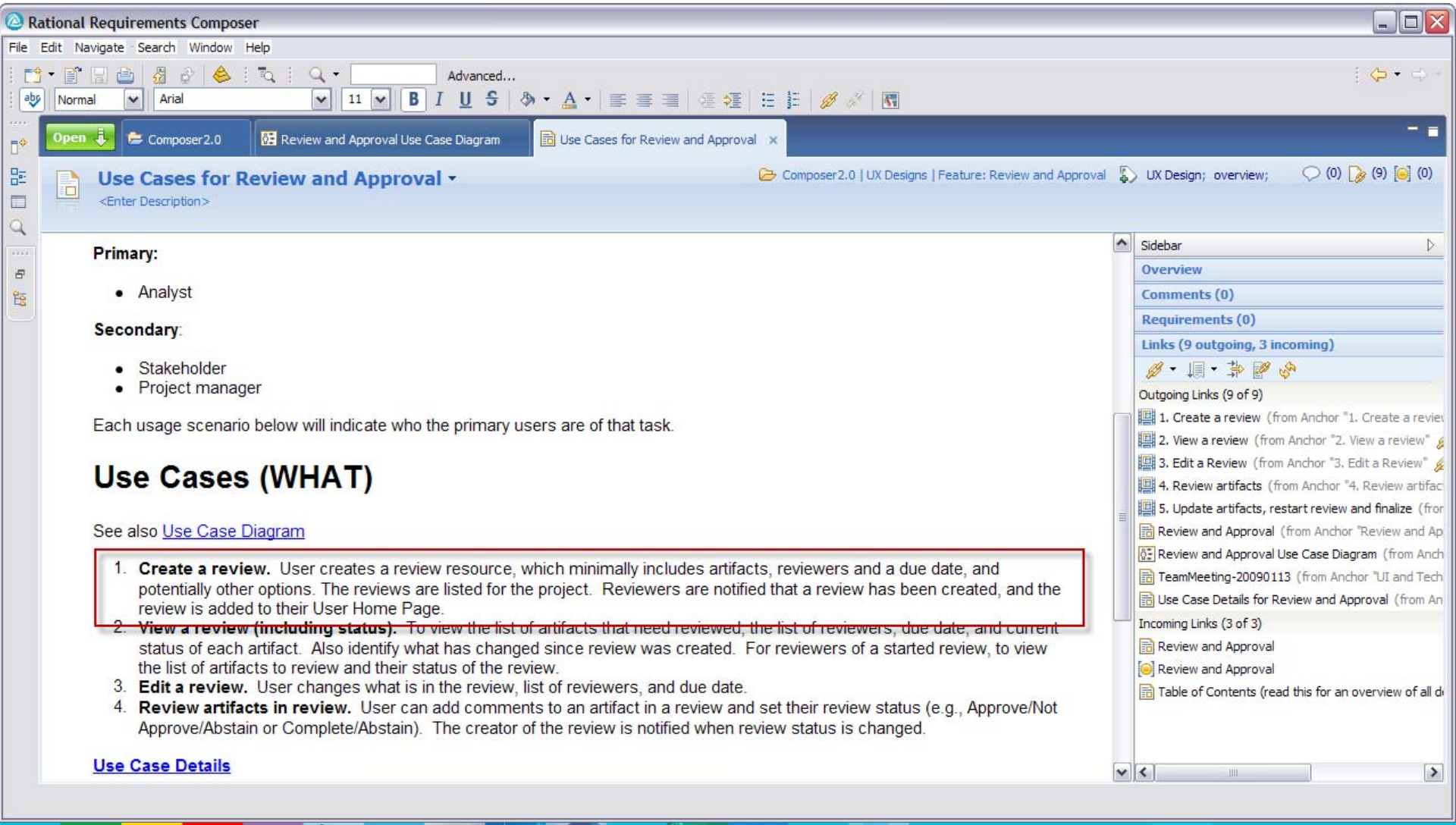
© 2009 IBM Corporation

Use Case Diagram: Review and Approval



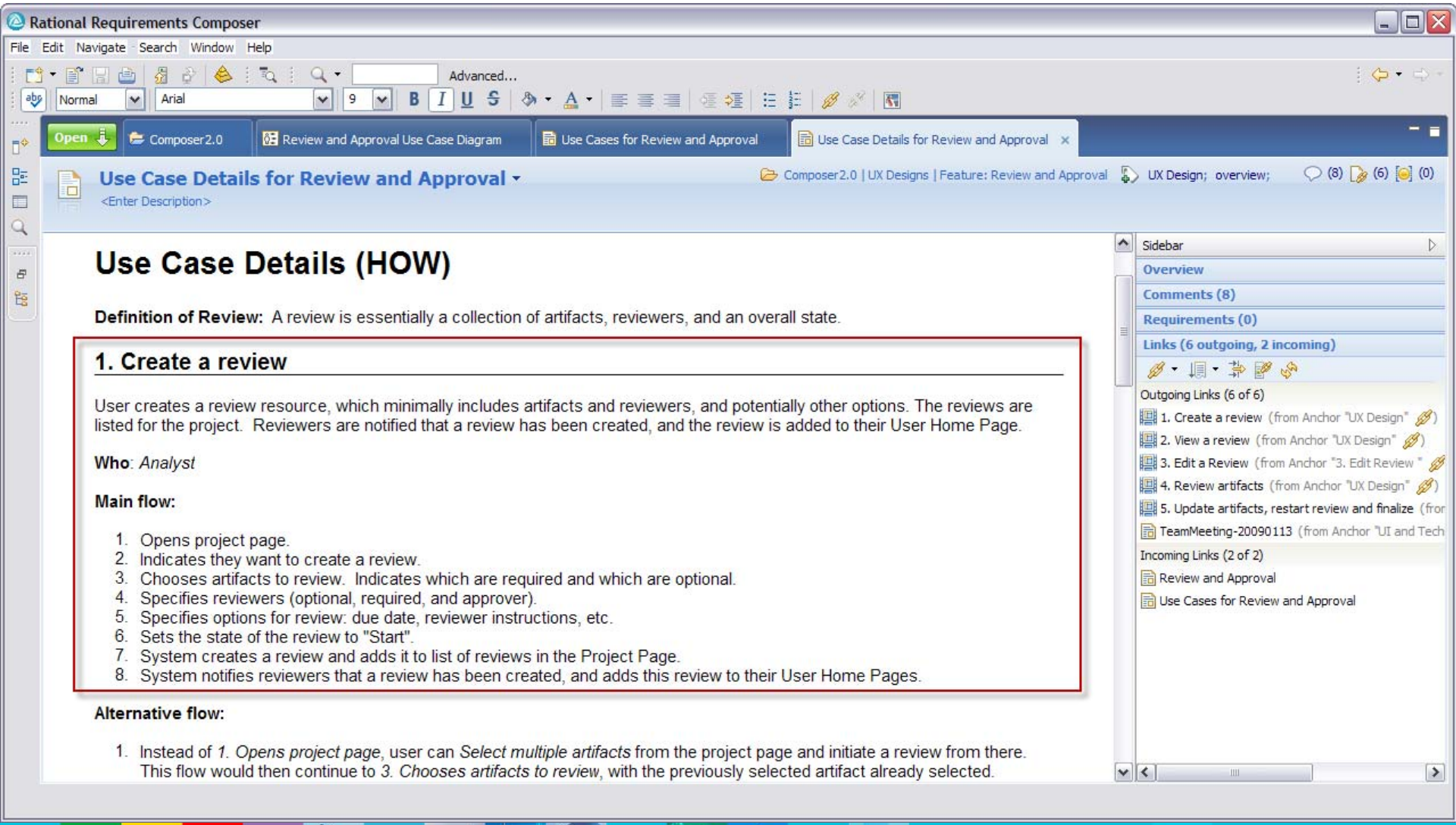
© 2009 IBM Corporation

Document: Use Case for Review and Approval



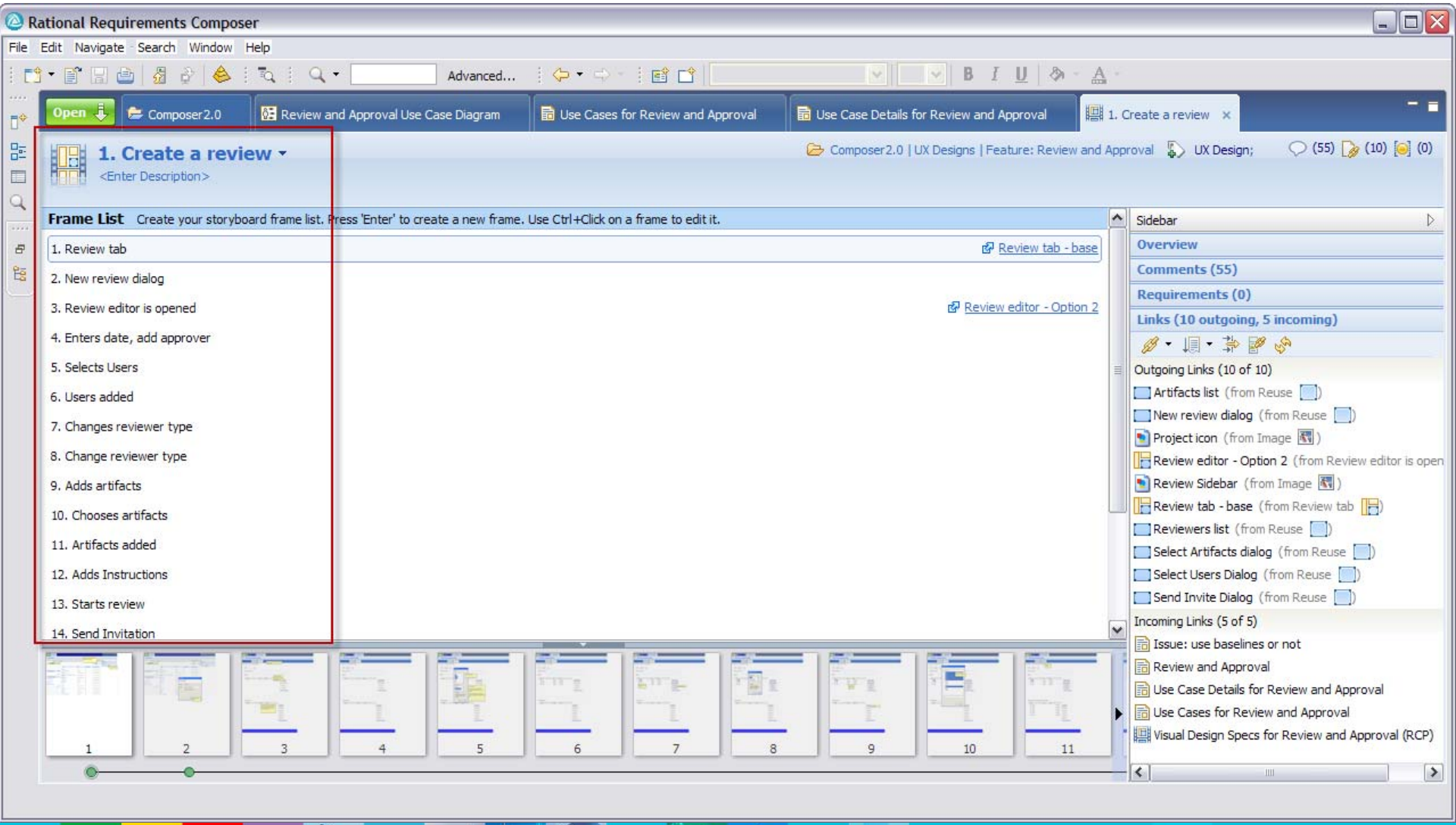
© 2009 IBM Corporation

Document: Use Case Details for Review and Approval



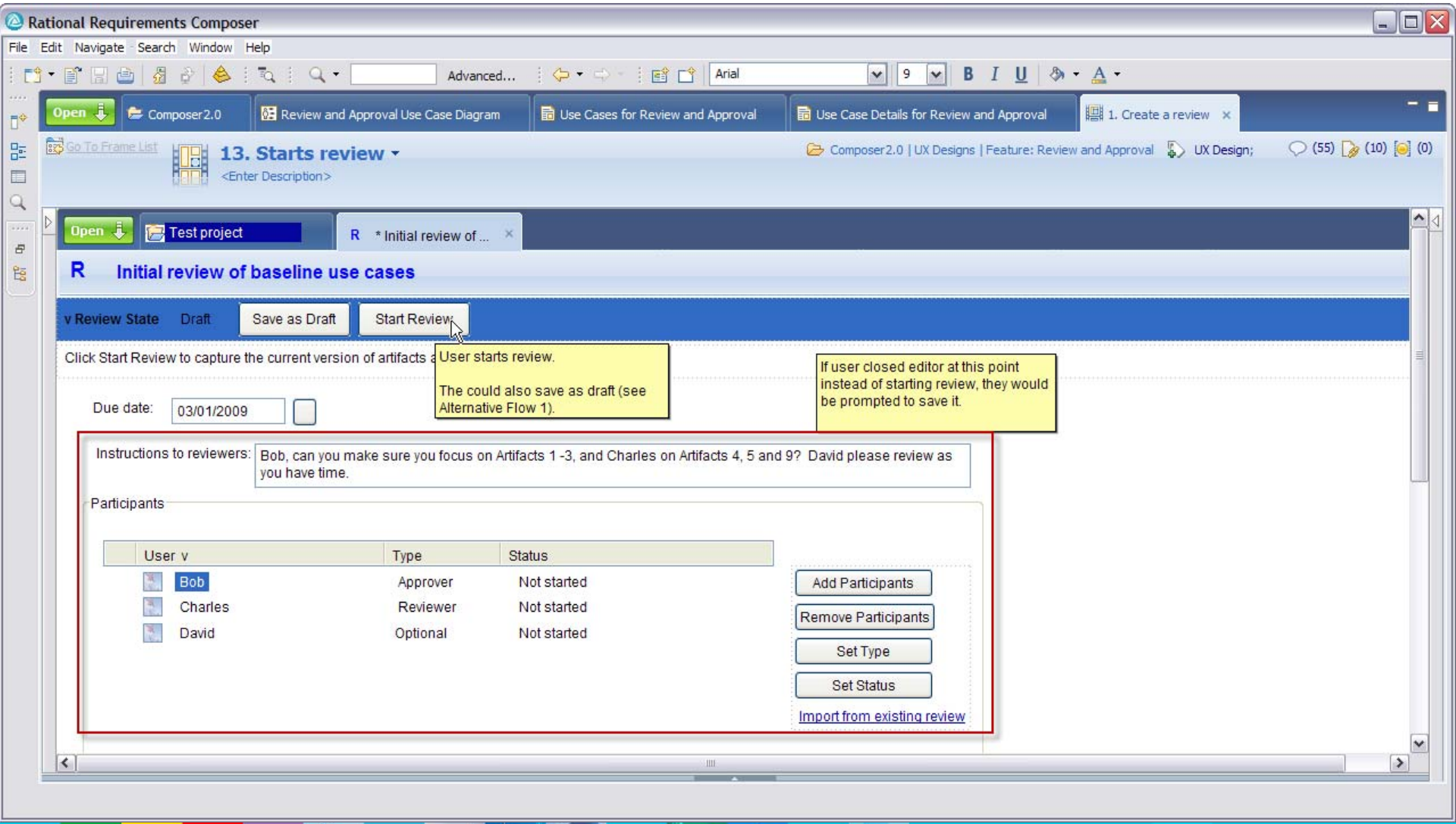
© 2009 IBM Corporation

Storyboard: Create Review – Frame List



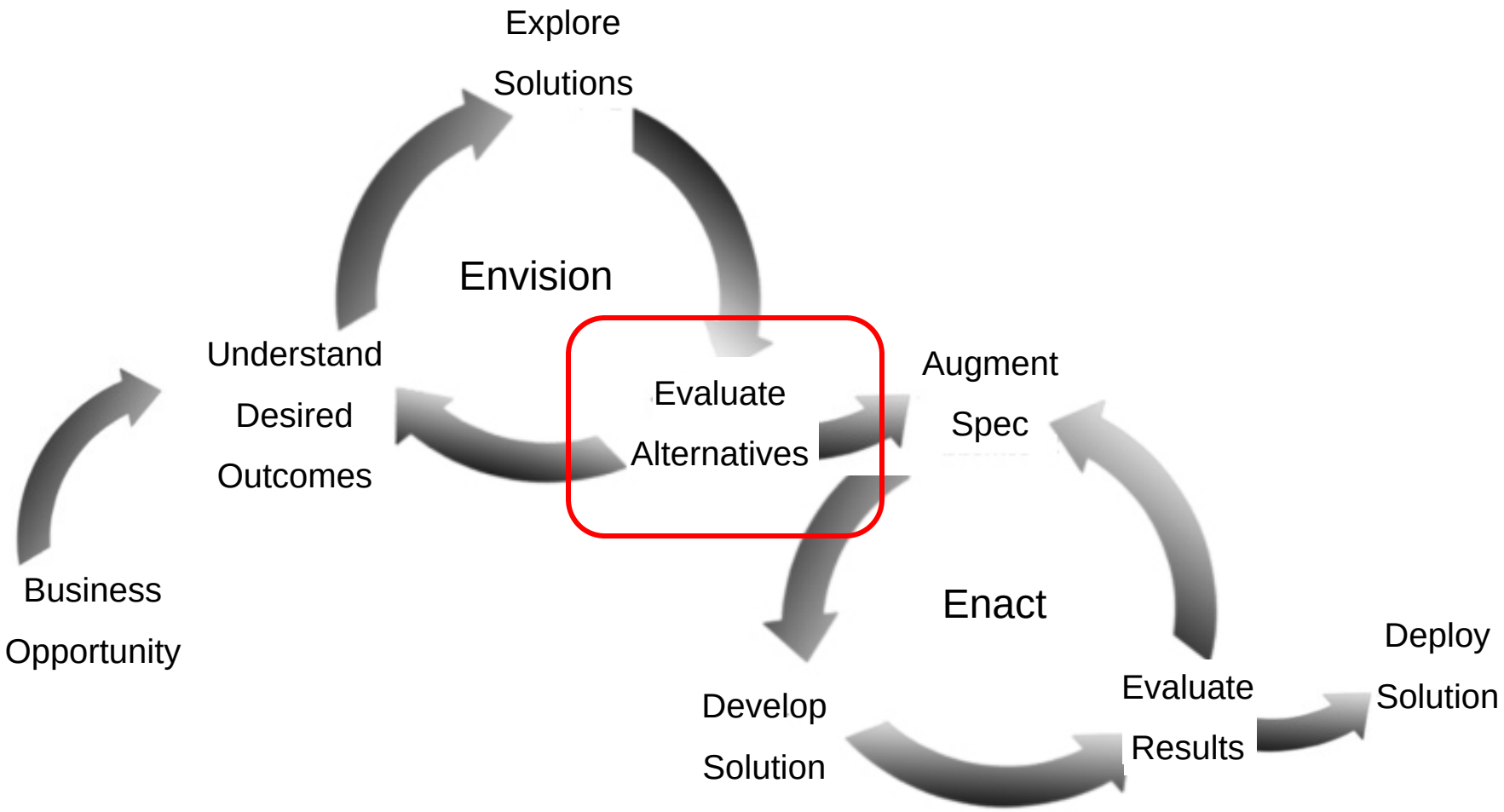
© 2009 IBM Corporation

Storyboard: Create Review



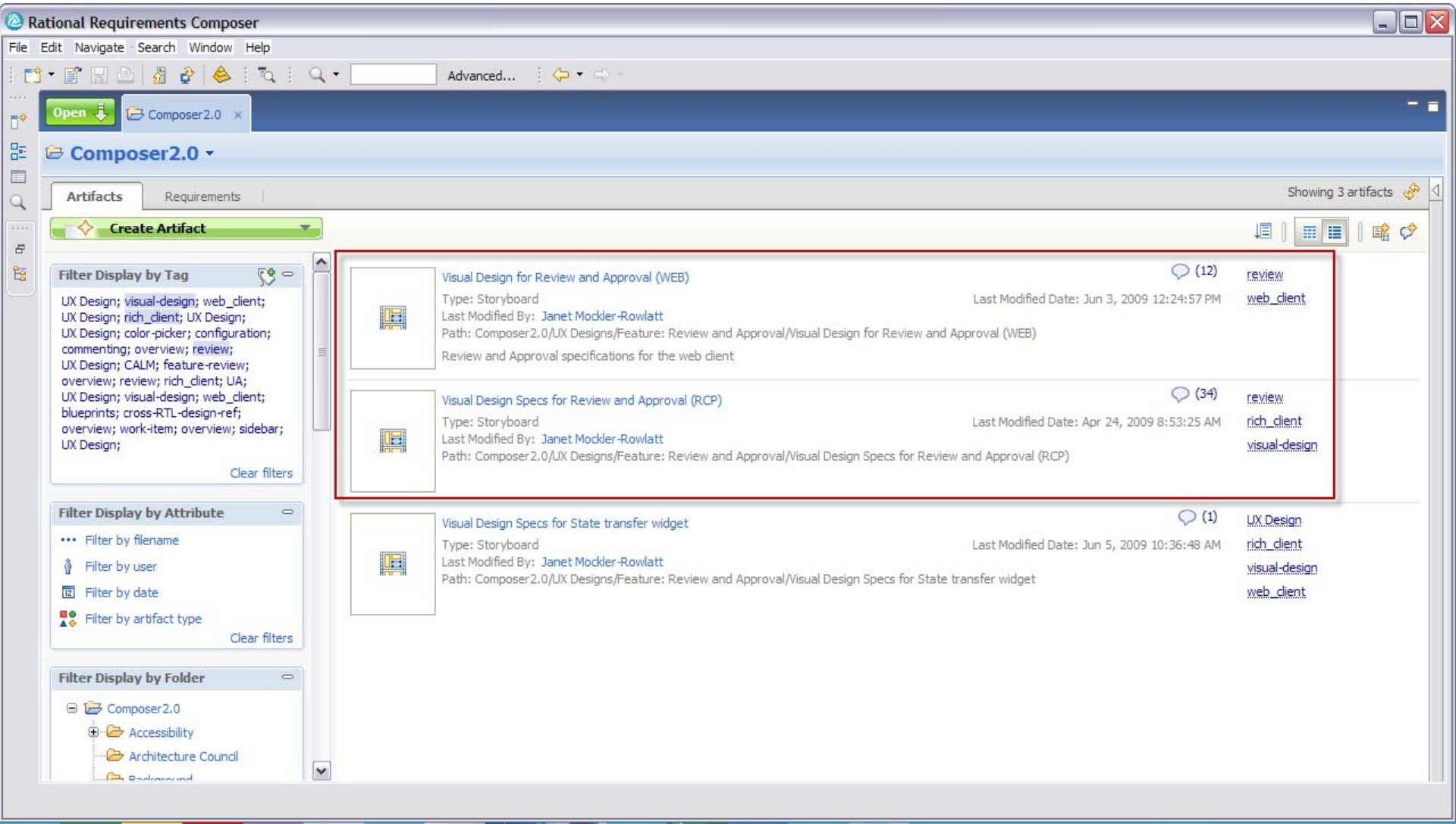
© 2009 IBM Corporation

A roadmap for building the right solution



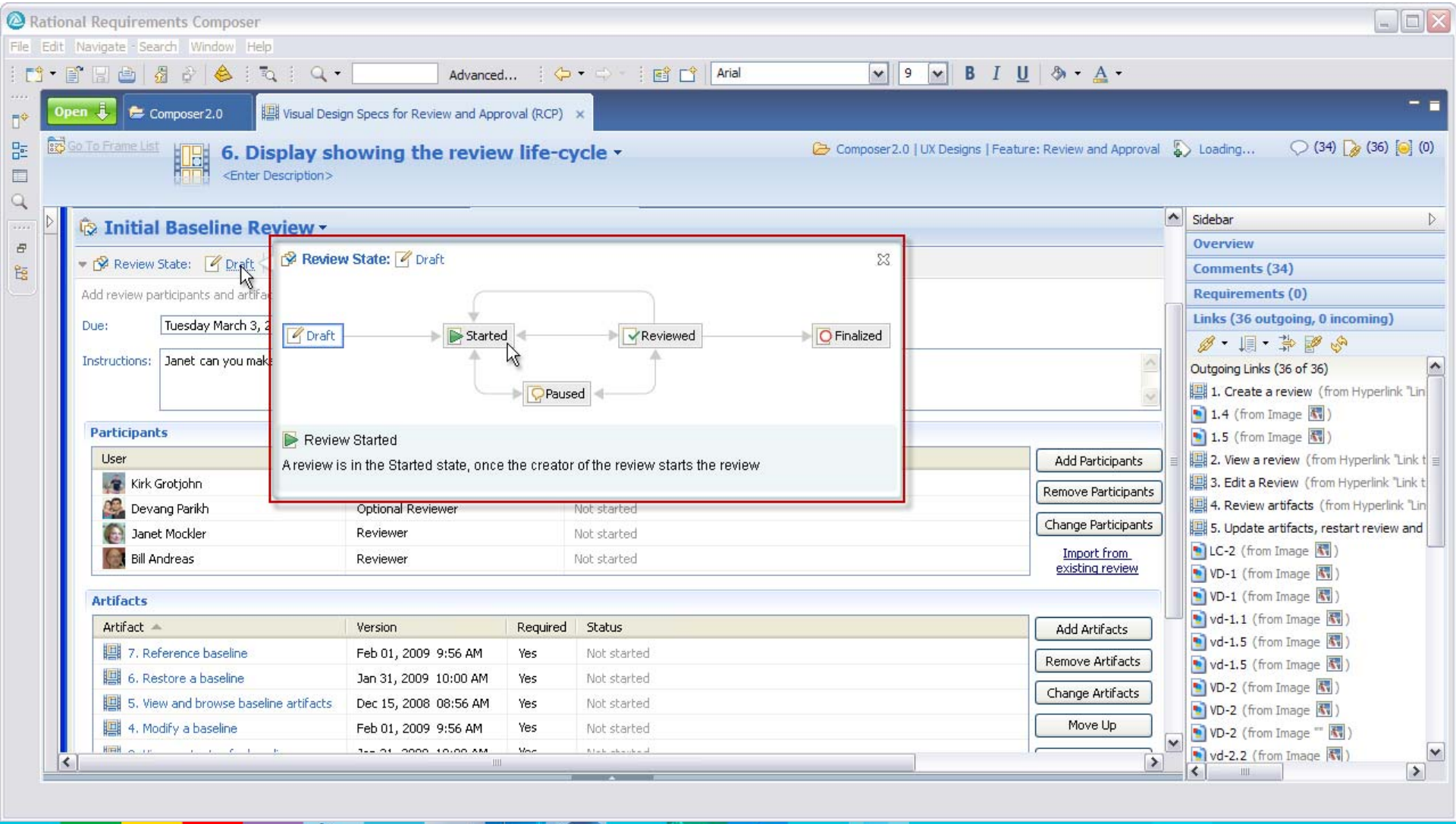
© 2009 IBM Corporation

Dashboard: Virtual Design for Review and Approval



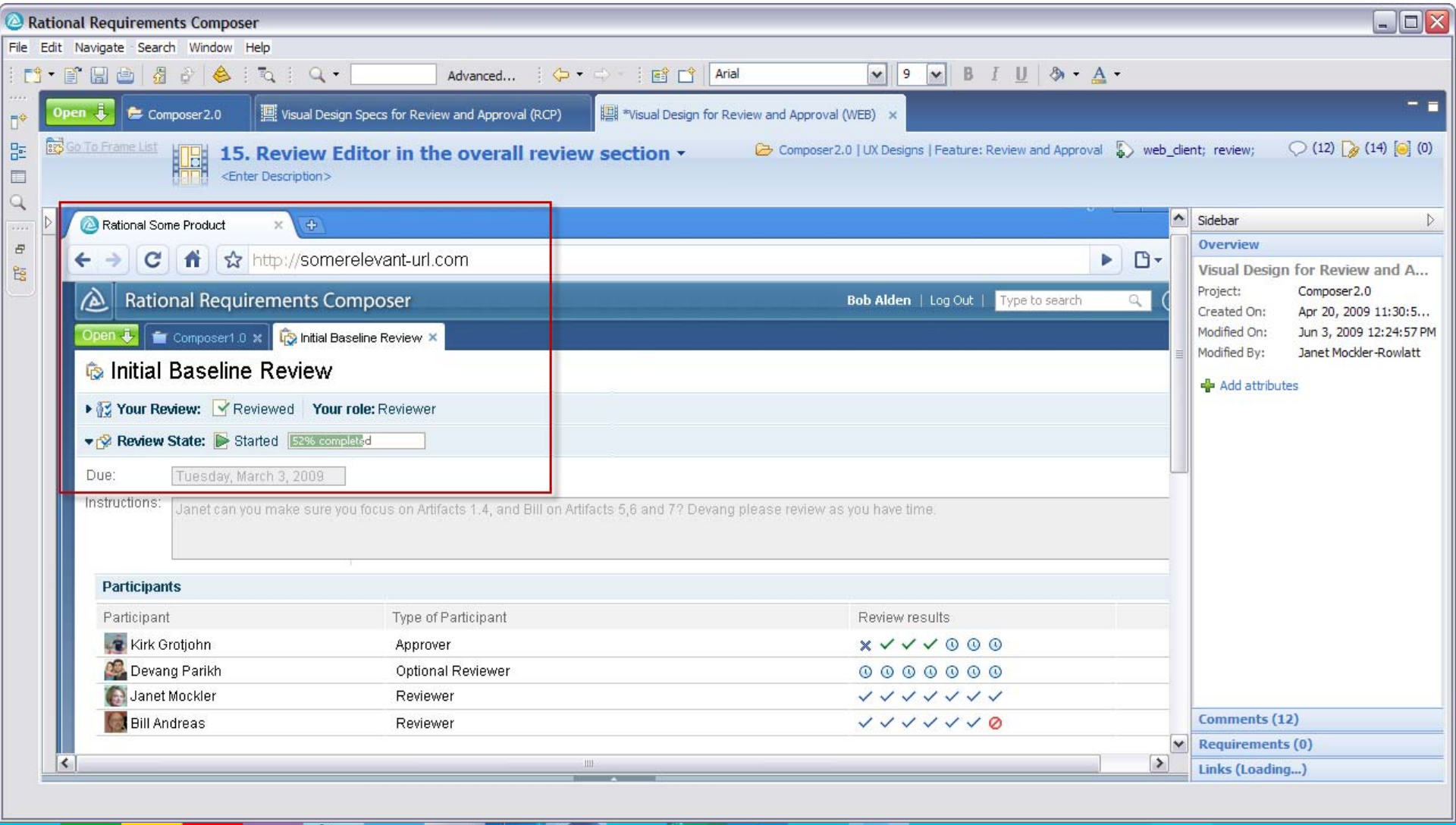
© 2009 IBM Corporation

Storyboard: Virtual Design – Rich Client



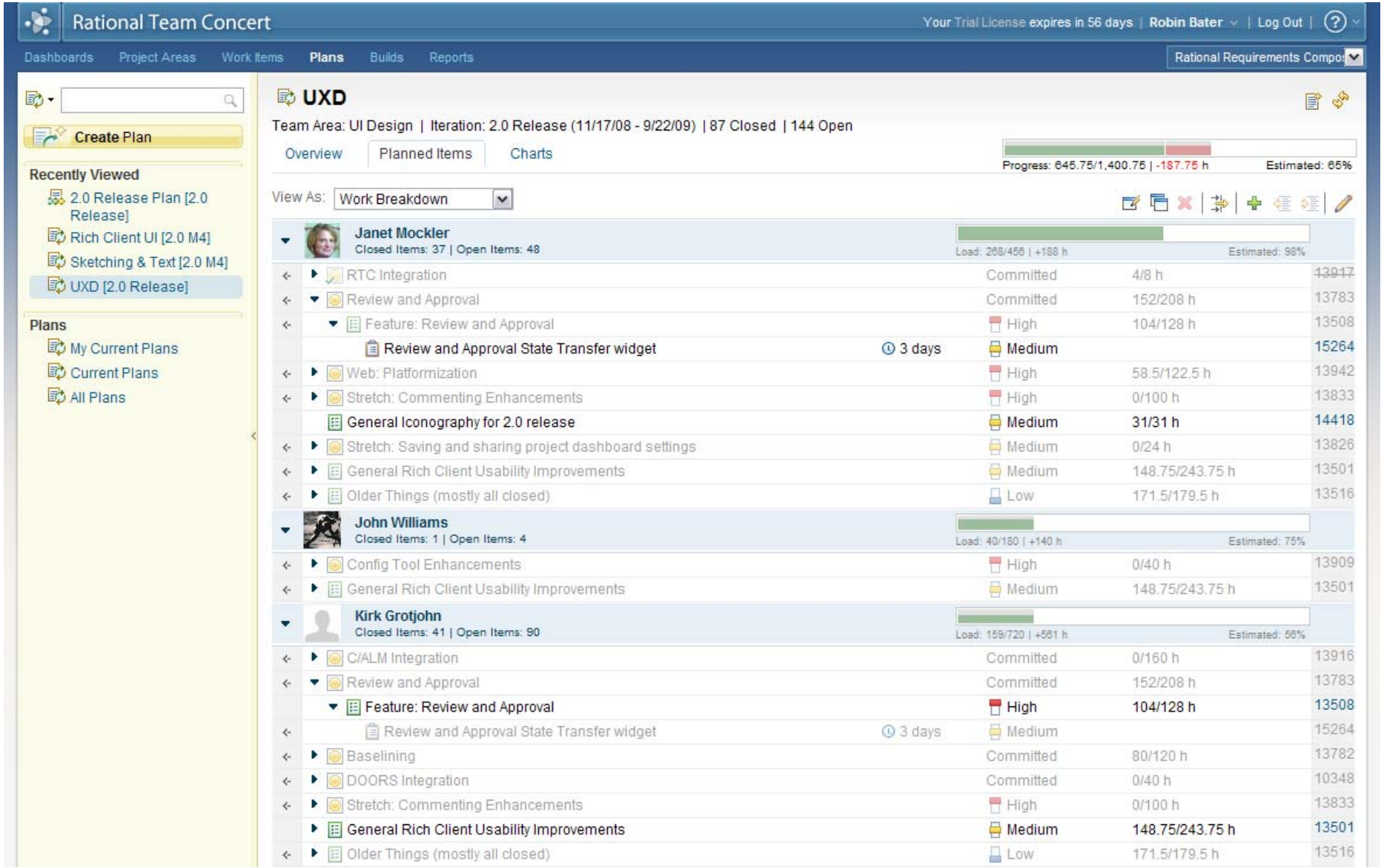
© 2009 IBM Corporation

Storyboard: Virtual Design – Web Client



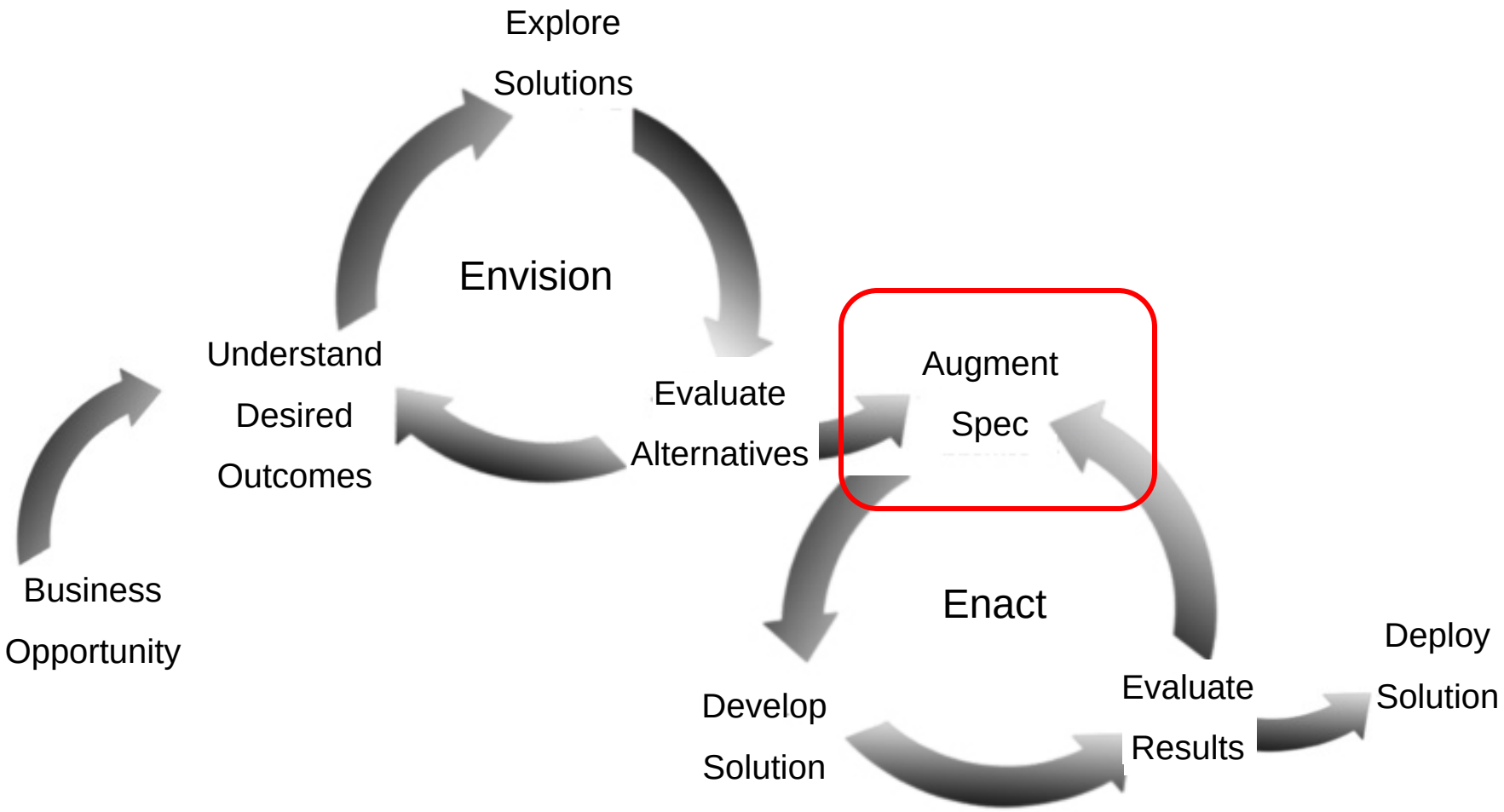
© 2009 IBM Corporation

RRC V2 – UXD Planned Items



© 2009 IBM Corporation

A roadmap for building the right solution



© 2009 IBM Corporation

Document: Review Editor (Edit) – UI specs

Rational Requirements Composer

File Edit Navigate Search Window Help

Open | Composer2.0 | 01. Review Editor (Edit) - UI specs

01. Review Editor (Edit) - UI specs

<Enter Description>

Composer2.0 | UX Designs | Feature: Review and Approval | UX Design; (1) (4) (0)

This document describes UI specifications, focusing on UI layout, functions and interaction. Images are meant to convey rough visual designs. Detailed visual blueprints, where necessary, will be described separately.

1. Review Editor - "Review State" section

This section of the review is for editing the review as well as showing the current state of the review. It is a different view of the same review as shown in the "Your Review" section (Section 2). It will primarily be used by the creator of the review.

All users can see this section, although only creators of the review or users with admin rights can edit it (for all others, the action buttons should be hidden). When the review is in draft state, this will be the top section (after the review is started, "Your Review" will be the top section.

The following document describes what a user can see and do based on their role and the state of the review: Views and actions for reviewers and non-reviewers

Review Editor components map

Sidebar

Overview

Comments (1)

Requirements (0)

Links (4 outgoing, 1 incoming)

Outgoing Links (4 of 4)

Artifacts list (from Embedded Diagram)

Review Editor components map (from Embedded Dia

Reviewers list (from Embedded Diagram)

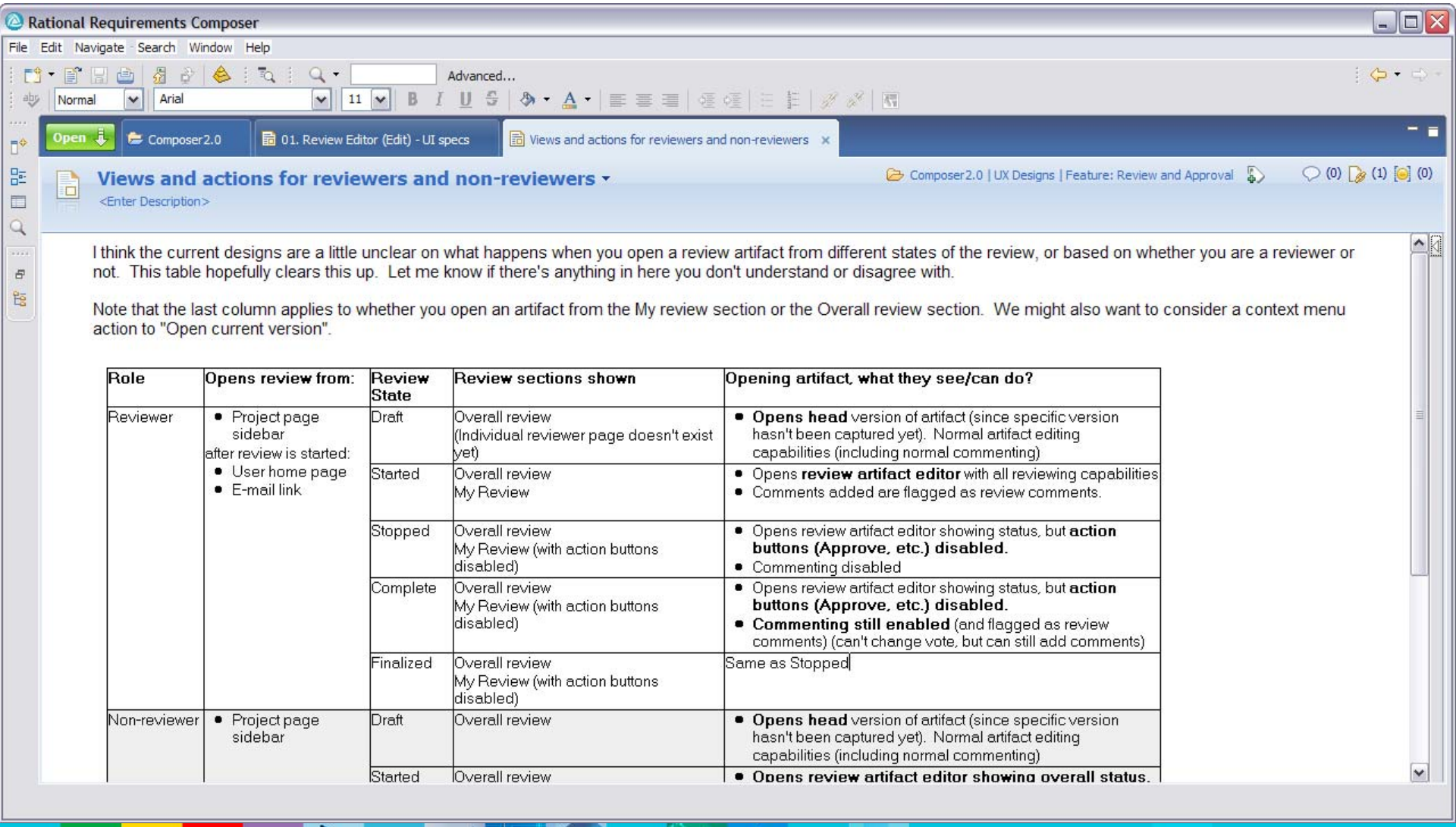
Views and actions for reviewers and non-reviewers

Incoming Links (1 of 1)

0. UI Component Specs - TOC

© 2009 IBM Corporation

Document: Views and actions for reviewers and non-reviewers



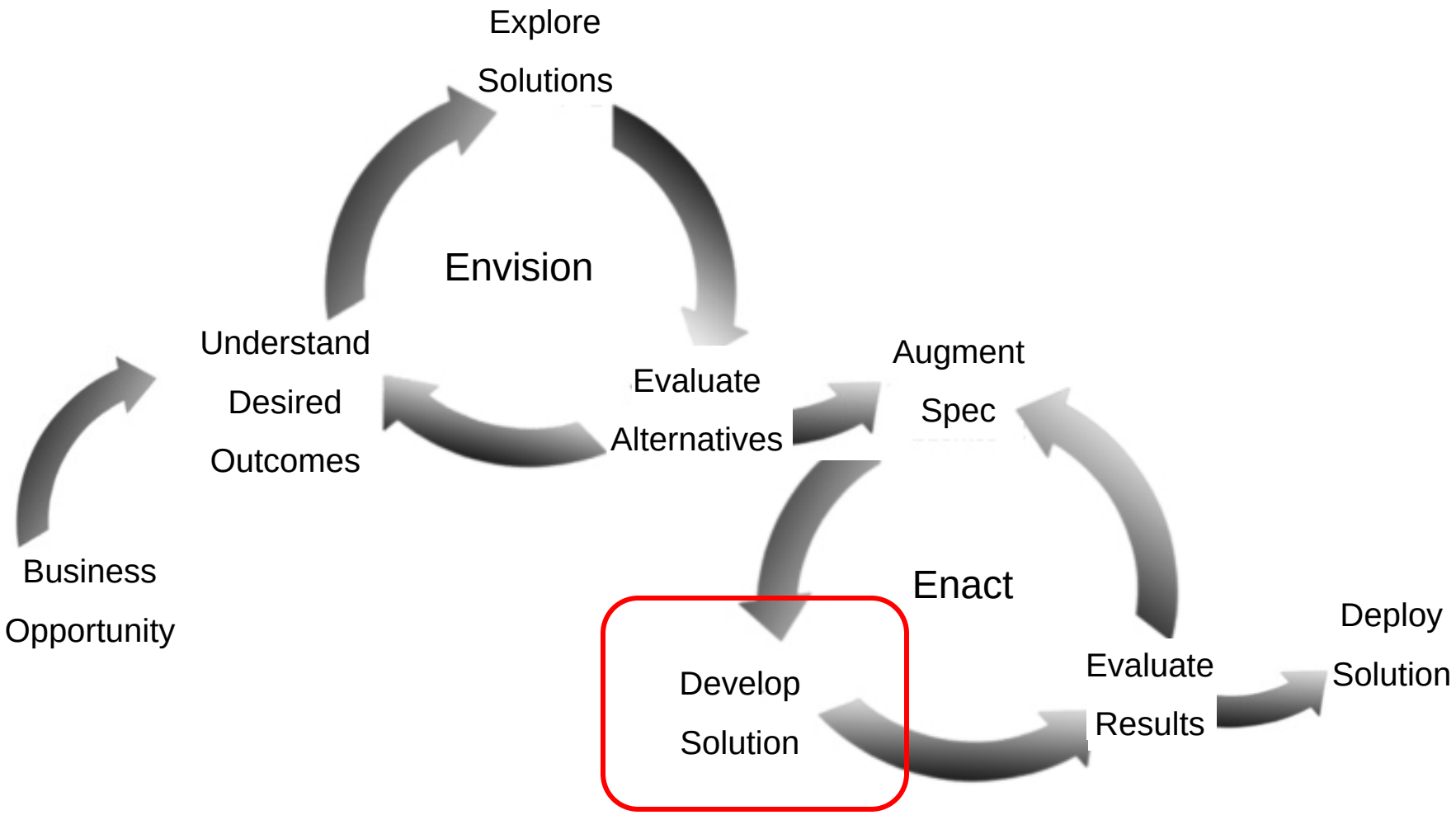
I think the current designs are a little unclear on what happens when you open a review artifact from different states of the review, or based on whether you are a reviewer or not. This table hopefully clears this up. Let me know if there's anything in here you don't understand or disagree with.

Note that the last column applies to whether you open an artifact from the My review section or the Overall review section. We might also want to consider a context menu action to "Open current version".

Role	Opens review from:	Review State	Review sections shown	Opening artifact, what they see/can do?
Reviewer	• Project page sidebar after review is started: • User home page • E-mail link	Draft	Overall review (Individual reviewer page doesn't exist yet)	• **Opens head** version of artifact (since specific version hasn't been captured yet). Normal artifact editing capabilities (including normal commenting)
		Started	Overall review My Review	• Opens **review artifact editor** with all reviewing capabilities • Comments added are flagged as review comments.
		Stopped	Overall review My Review (with action buttons disabled)	• Opens review artifact editor showing status, but **action buttons (Approve, etc.) disabled.** • Commenting disabled
		Complete	Overall review My Review (with action buttons disabled)	• Opens review artifact editor showing status, but **action buttons (Approve, etc.) disabled.** • **Commenting still enabled** (and flagged as review comments) (can't change vote, but can still add comments)
		Finalized	Overall review My Review (with action buttons disabled)	Same as Stopped
Non-reviewer	• Project page sidebar	Draft	Overall review	• **Opens head** version of artifact (since specific version hasn't been captured yet). Normal artifact editing capabilities (including normal commenting)
		Started	Overall review	• **Opens review artifact editor showing overall status.**

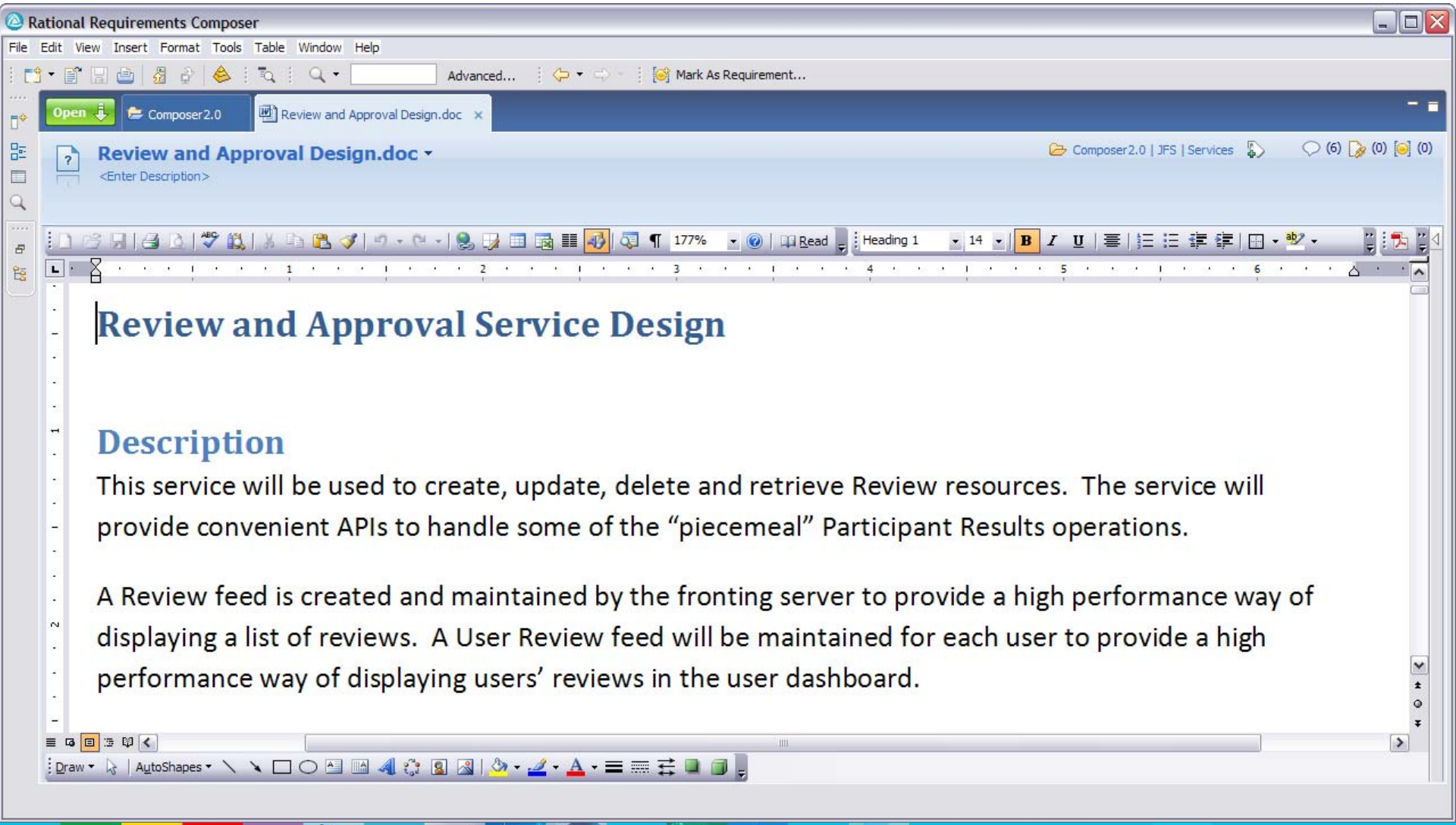
© 2009 IBM Corporation

A roadmap for building the right solution



© 2009 IBM Corporation

Word Document: Review and Approval Service Design



© 2009 IBM Corporation

RRC V2 – http://open-services.net/bin/view/Main/RmHome

Requirements Definition and Management

Introduction

The goal of this effort is to define a common set of resources, formats and RESTful services for the *use in* Requirements Definition and Management tools and *use by* (for example) systems engineering or ALM tools that will enable the effective use of requirements across a development life-cycle.

Requirements Management (RM) resources define the requirements or capabilities of a system or the outcome of some programme of work. We use the term *system* entirely generally and without prejudice to software, hardware, IT, regulatory, business etc.

Requirements are the basis for defining what the system stakeholders (users, customers, suppliers and so on) need from a system and also what the system must do in order to meet those needs, and how the surrounding processes must be orchestrated so that quality, scope and timescale objectives are satisfied. The discipline of requirements management involves many activities including elicitation, definition, documentation, analysis, decomposition, validation and justification; it involves managing changes to requirements and establishing and reasoning about their interrelationships. In involves the creation of assets which are invariably related to other systems and processes. RM is inherently multi-disciplinary and across the life-cycle; it brings together all assets across the life-cycle, ascribing them meaning, captures their dependencies and interrelationships and ultimately, captures the way in which they together produce the desired system.

Requirements management activities

- Elicitation, definition and communication introduce requirements in such a way that relevant stakeholders can understand and agree that the system will meet their needs.
- Gathering requirements and related assets into a coherent set, for example as a set of requirements documents, as part of forming a precise overarching specification of the system. Such a set of requirements is often part of a contractual agreement between parties, for example.
- Analysis aids the understanding of requirements and enables the expression of refactored, decomposed, refined, elaborated, or clarified requirements. Assets such as system models may be created during such analysis.
- The need to validate the system induces the need to create test requirements that demonstrate that all requirements have been met, and thus, for example, that contractual obligations have been discharged.
- The need to deal with ambiguity, complexity or testability of a requirement leads to lower-level, related *derived* artefacts. These derived artefacts may be requirements, or they could be from some other system, such as change requests, diagrams stored in a PLE system and so on.
- The recording and management of the *links* created as part of each of these processes is central to understanding the meaning of requirements, assessing the impact of proposed or actual change and, through the recording of rationale information, validation that lower-level requirements are correctly derived from higher-level requirements.
- The need to plan the scope of a release or milestone in order that the requirements that are to be met are known and meaningful and will meet the stakeholders' needs.

Traceability analysis

Impact analysis exploits the links created during such derivation and analysis in order to understand the ramifications of actual or proposed changes to higher-level requirements. In this way the impact to the system, or aspects of its delivery can be assessed and acted upon.

Derivation analysis exploits links in order to inform cost/benefit analysis processes, or to understand the rationale for some parts of the system, and the ramifications of actual or proposed change to those parts.

Participants

IanGreen (lead)
GeorgeDeCandio
SteveAbrams
AndyBerner

Feedback

Specification comments and issues

Mailing List

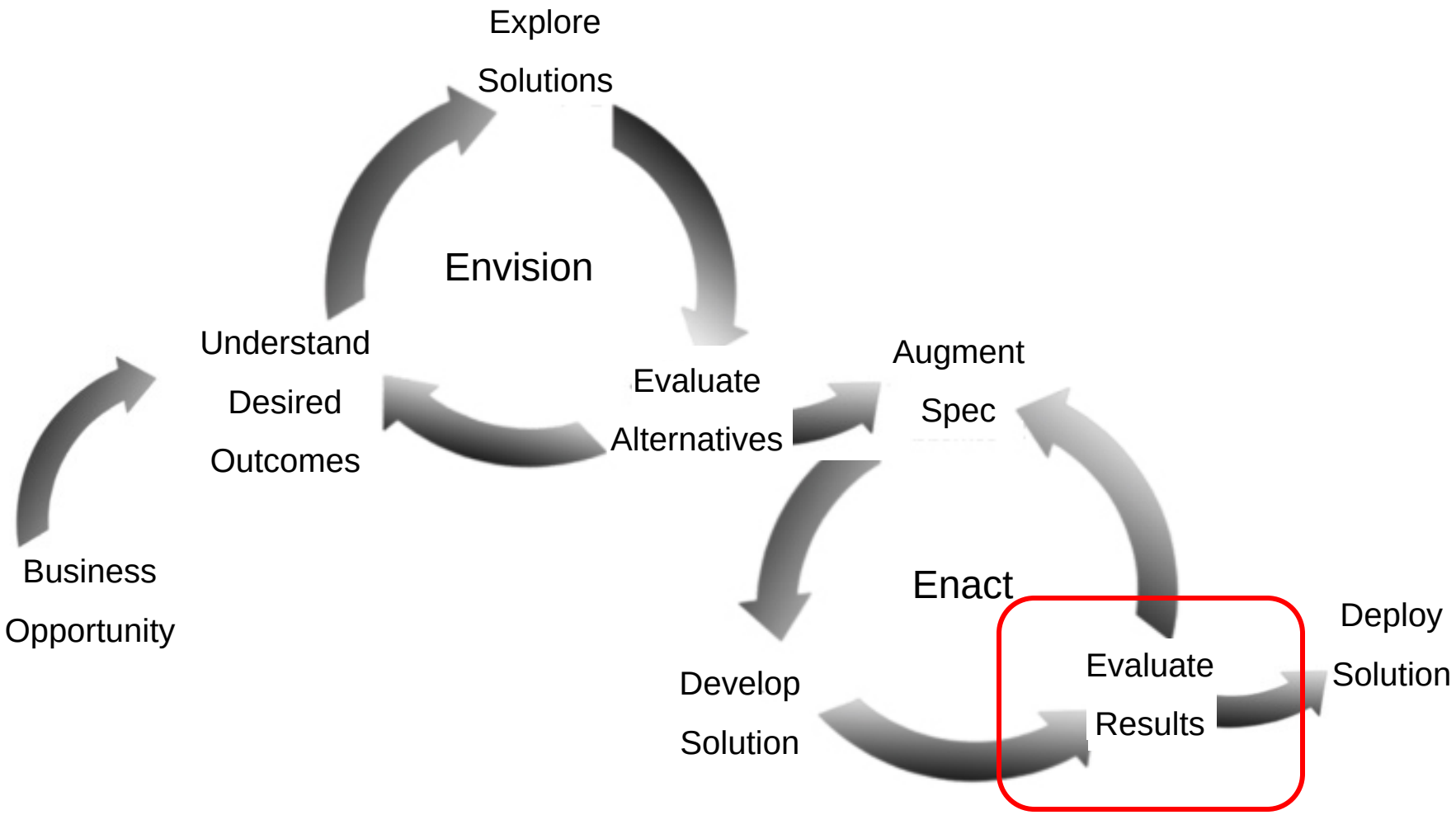
General Open Services

Requirements for discussion

Please add stakeholder requirements here

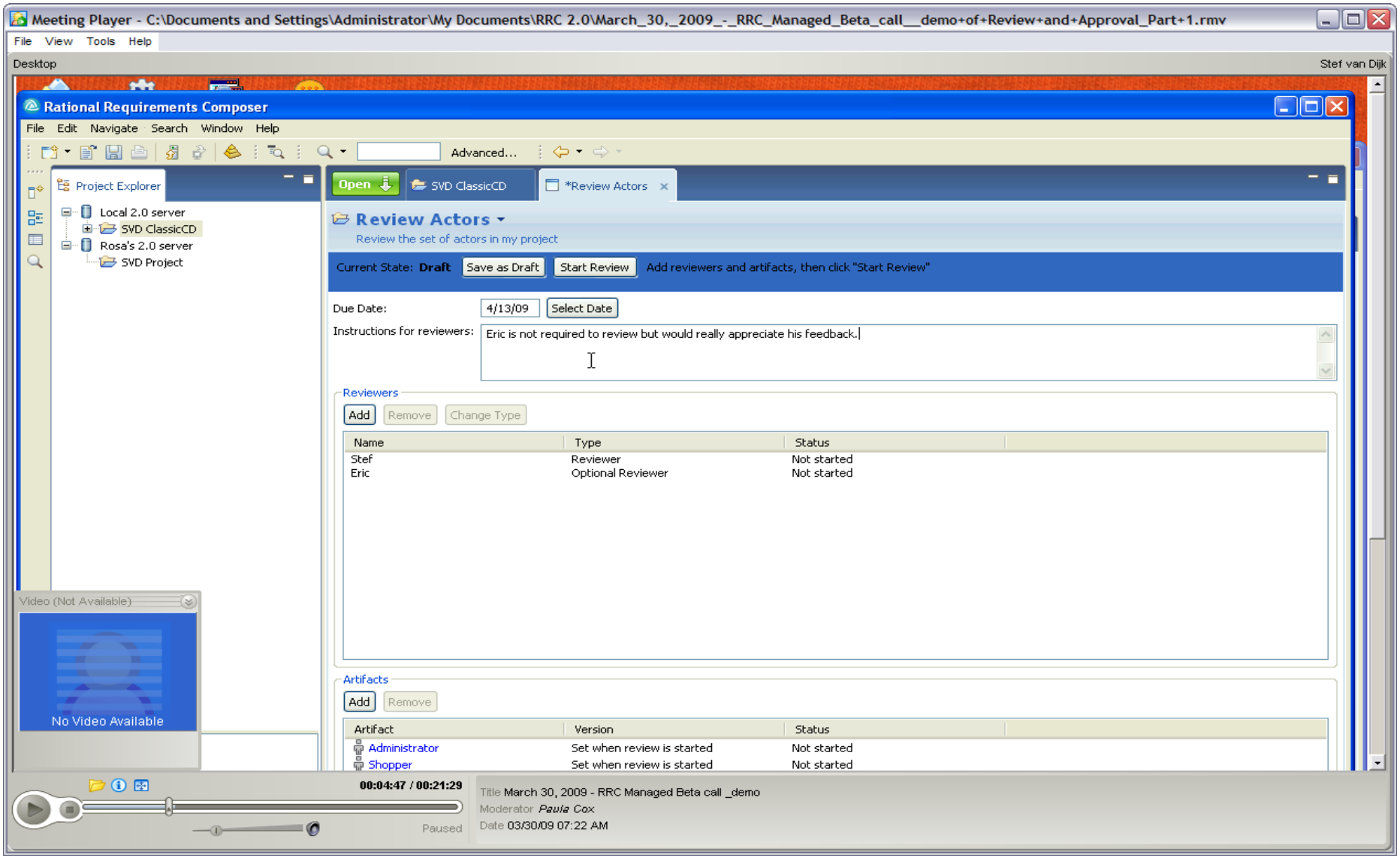
© 2009 IBM Corporation

A roadmap for building the right solution



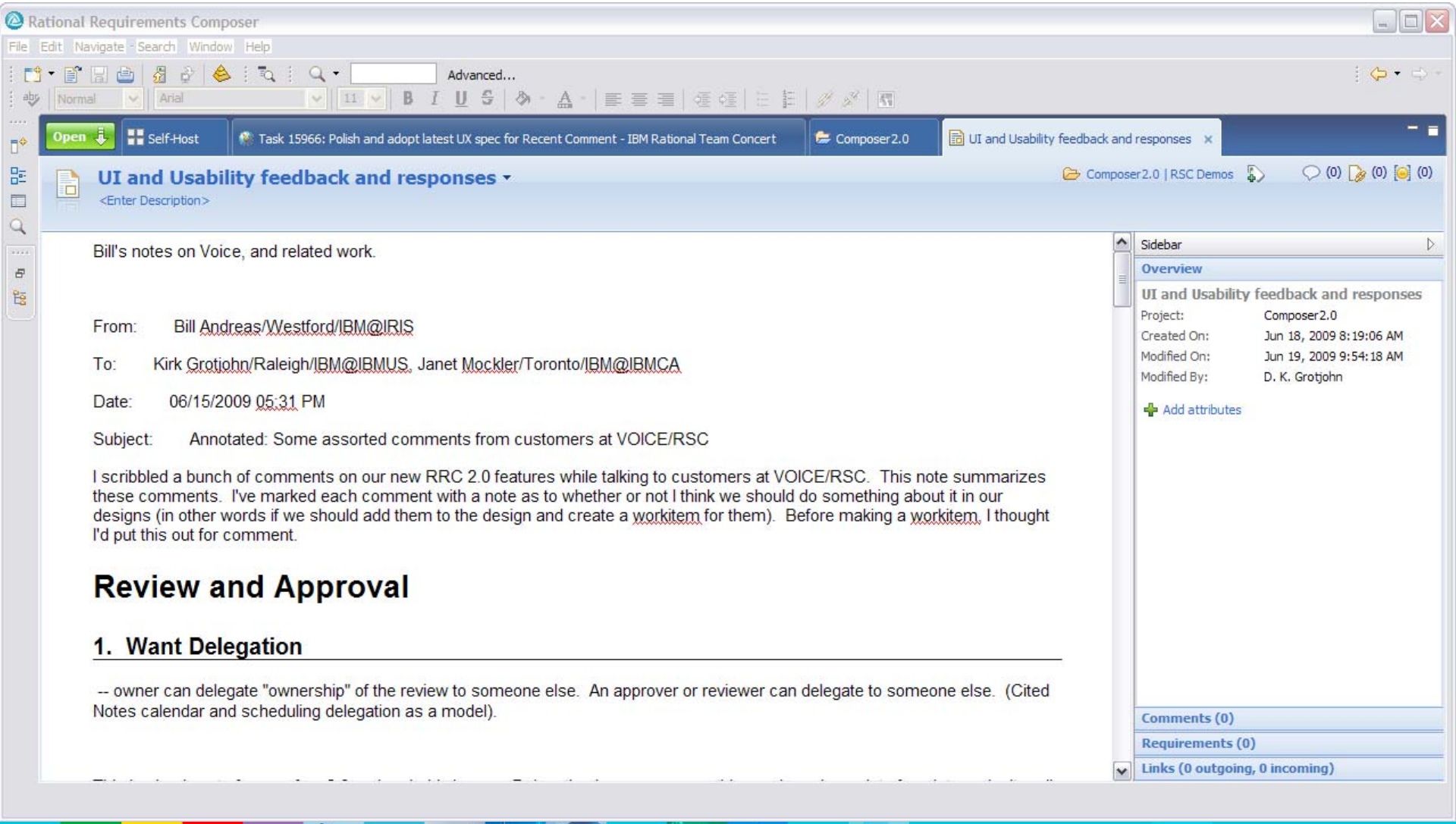
© 2009 IBM Corporation

RRC V2 – early development demo



© 2009 IBM Corporation

RSC 20009 UI and Usability feedback and responses



© 2009 IBM Corporation

RRC – Key benefits experienced by the RRC team

- Elicited more and better feedback before code is written (both requirements and feature design). Enabled the team to converge faster on the “right” requirements and reduced churn/rework that would have resulted otherwise.
  - Increased range and depth of stakeholder participation.
  - Better information visibility enabled the team to identify gaps and clarify misunderstandings more quickly.
- RRC developers communicated better among themselves, especially across component teams. This “lower cost, higher value communication” made the development team more productive

© 2009 IBM Corporation

Stakeholders

- Product manager
  - *“RRC made possible a new level of transparent collaboration between our various stakeholders...the product management team was more involved in early reviews of feature designs. Information was more visible, and RRC's collaborative platform made it easier to provide feedback on a daily basis. This led to better decisions made earlier, fewer surprises, and less rework due to misunderstood requirements*.”– Daniel Moul
- User Experience Design
  - *"Using the UI sketches and storyboards, you can very quickly create some high level designs that illustrate the basic flow, UI components, layout, etc. You can then get immediate feedback from anybody on the product team and easily iterate on UI designs based on the feedback.“* – Kirk Grotjohn, User Experience Engineer
- Architect
  - *"Composer allowed me to enable scenario driven development and address the key pain points in defining integrations with other products. Composer has definition techniques that brought clarity to aspects like Release Requirements, Green Thread/ Scenario Definitions, Technical Architecture, Data Model (Resource Representations), System Use Cases, Usability, and Integrations with other products. The Collaboration capabilities allowed me to quickly socialize with key stakeholders…"*– Vishy Ramaswamy, Rational Requirements Composer Architect

© 2009 IBM Corporation

Stakeholders

- Quality Test Engineers
  - "*It was a "one stop shop" to get everything needed related to design and proposed implementation. This makes collecting input for testing easier"* – Mario Maldari, System Test Lead
- Developers
  - "*As a collaborative leader, Composer is a perfect fit for me. It allowed me to share ideas easily using comments, composer sketches and textual documents"* - Devang Parikh, Development Lead
- Consultants
  - *"..As a member of the field I am not invited or able to attend many of the development discussions, so by spending 10 minutes, I log onto the self-host project dashboard and see what is new…and if there is clarification or contribution I can make I add a comment and direct it to the appropriate team member"* – Robin Bater

© 2009 IBM Corporation

Learn more at:

- IBM Rational software
- Rational launch announcements
- Rational Software Delivery Platform
- Accelerate change & delivery
- Deliver enduring quality
- Enable enterprise modernization
- Ensure Web security & compliance
- Improve project success
- Manage architecture
- Manage evolving requirements
- Small & midsized business
- Targeted solutions
- Rational trial downloads
- developerWorks Rational
- Leading Innovation
- IBM Rational TV
- IBM Business Partners
- IBM Rational Case Studies

© Copyright IBM Corporation 2009. 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.

© 2009 IBM Corporation

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

ClearCase IBM Rational

Rational is a trademark of International Business Machines Corporation and Rational Software Corporation in the United States, Other Countries, or both.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2009. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.

© 2009 IBM Corporation