Integrating DOORS into the Systems and Software Engineering Lifecycles

Prakash S **IT Specialist**

IBM Software
Innovate2011
The Premier Event for Software and Systems Innovation
Software. Everyware.
June 9–12 Delhi, India

Agenda

- Rational solutions for systems and software engineering
- The vision in practice – what do we have so far?
  - Requirements driven development
    - Integrating DOORS with change management
    - Demonstration of DOORS-RTC integration
  - Requirements driven testing
    - Integrating DOORS with test management
    - Demonstration of DOORS-RQM integration
  - Model Based Systems Engineering
    - Integrating DOORS with modeling systems
    - Demonstration of DOORS-Rhapsody integration

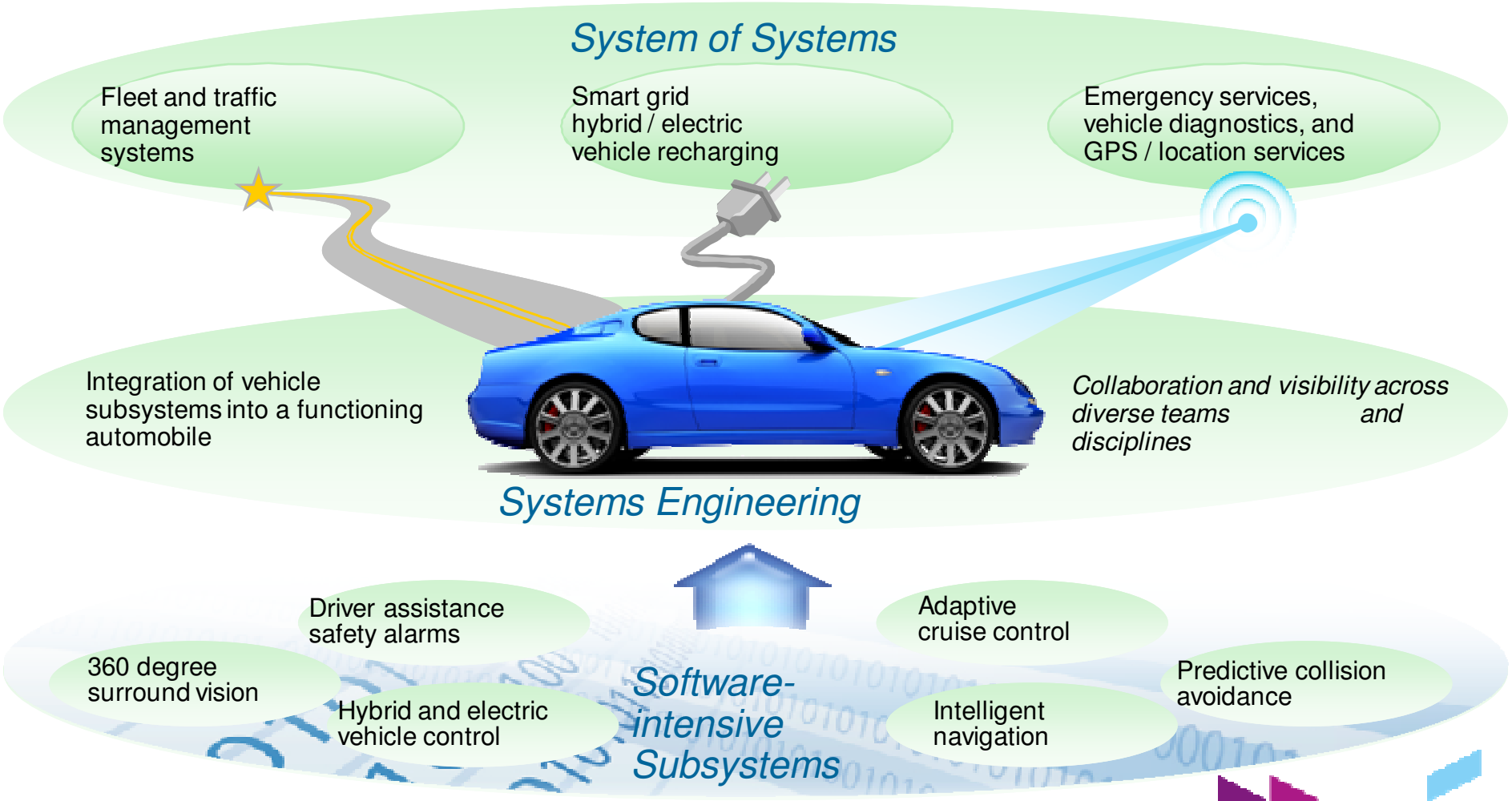
Products are Getting Smarter Every Time We Look

- One billion camera phones were sold in 2007, double that of 2006
- One customizable device: phone, e-mail, music, Web, camera, GPS, apps, video recorder, e-reader, …
- User productivity and enjoyment have skyrocketed
- In 2000 this would have been science fiction
- In 2010 it's yesterday's news!

What's possible by 2020?

Smart Products Require Innovative Systems

Incremental value is created by global interconnection across products, systems, applications and networks



Modern Approaches for Describing Systems Are Evolving

To Better Manage Complexity and Reduce Time-to-market

Past

Specifications

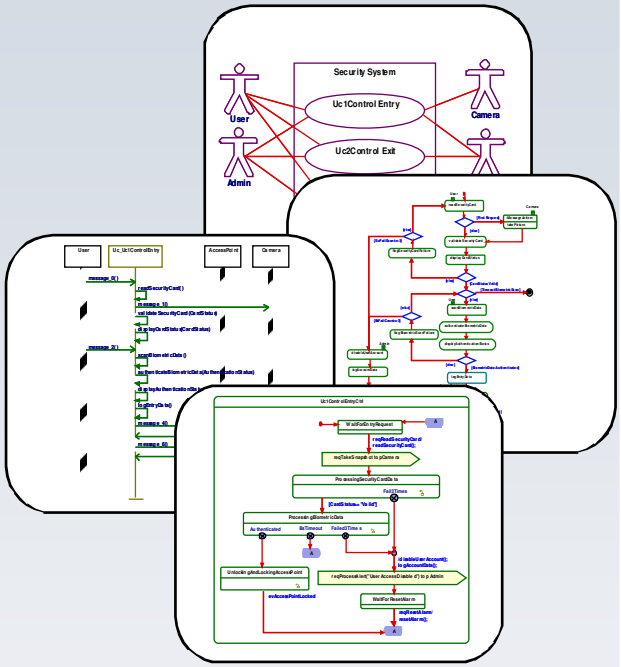
Interface requirements

System design

Analysis & trade-off

Test plans

Future



Moving from manual methods to an automated, visual approach

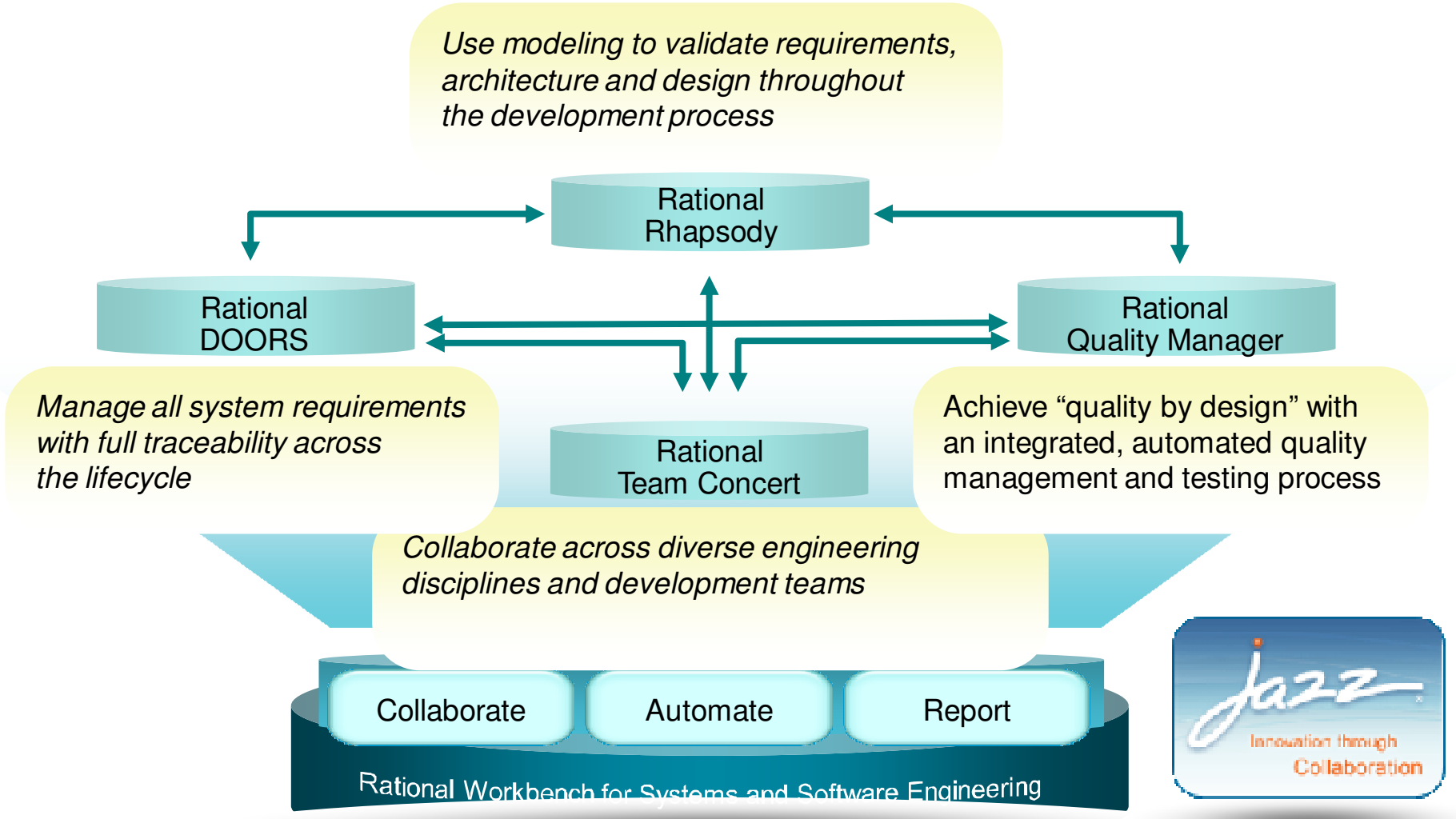
A shift in Product Delivery

The IBM Rational solution for systems and software engineering

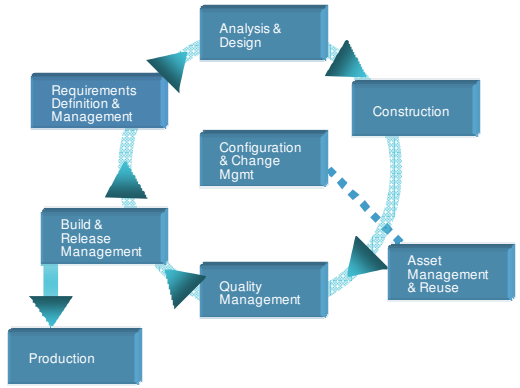
- The growing complexity of modern systems leads to increasing challenges in managing joint development life cycles, testing sequences and rollouts, and in monitoring and managing these systems once they are delivered.
- To meet this there is a need to challenge the way Rational develop and deliver their products
  - Need to move away from only delivering single products
  - The Rational solution for Systems and Software engineering offers world class practices and tools that unite mechanical, electronic and software disciplines.
  - Collaborative capabilities and automation are injected into project management, requirements engineering, architecture design, and quality management. Tools in the solution can share requirements, model artefacts, and other development components and provide comprehensive life-cycle management that can be tailored to your specific industry needs.
  - The solution enacts the systems delivery workflows and task management capabilities to effectively run today's systems delivery projects.

Rational Solutions for Systems and Software Engineering

Built on a core product set



Extend the Solution to Meet Your Needs



The Rational solution can be tailored to meet virtually any systems development workflow :

- *Automated reporting and documentation* with **Rational Publishing Engine**
- *Enterprise systems delivery* with **Rational System Architect**
- *Embedded software testing* with **Rational Test RealTime**
- *Team-based configuration management* with **Rational ClearCase** or **Synergy**
- *Domain specialization with industry-specific profiles such as AUTOSAR, Android, functional safety, and defense architecture frameworks*
- *Embedded platform development* with **Wind River Workbench/VxWorks**
  - Support also exists for **Green Hills Integrity, QNX Momentics/Neutrino** and many other embedded platform operating system environments
- and many others…

Open Services for Lifecycle Collaboration (OSLC)

An initiative aimed at simplifying data linking and tool integration across the lifecycle

Barriers to sharing resources and assets among tools

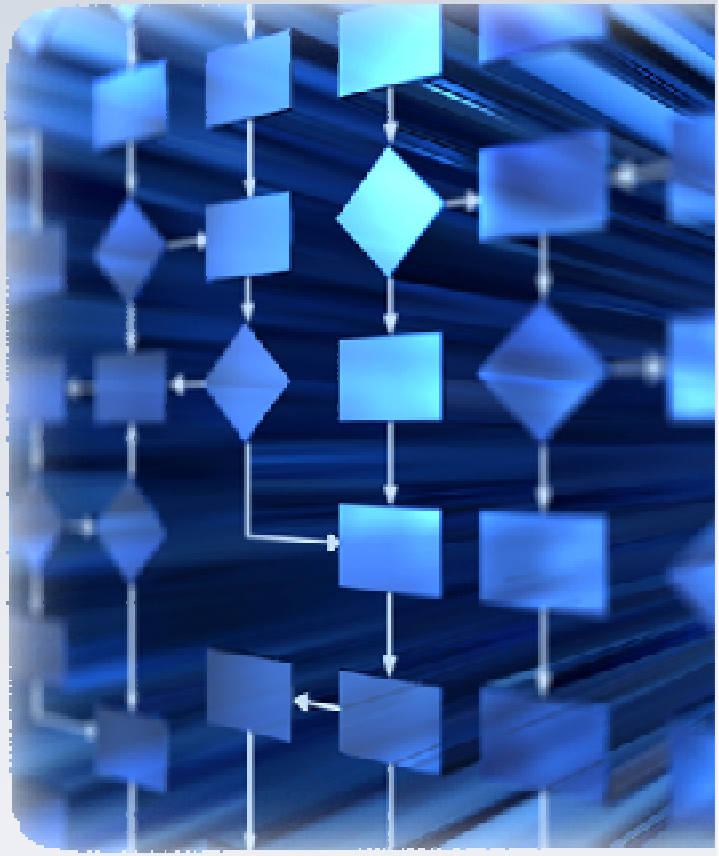
- Multiple vendors, open source projects, and in-house tools
- Private vocabularies, formats and stores
- Entanglement of tools with their data

Open Services for Lifecycle Collaboration

- Community Driven –
  - Specified at open-services.net
- Specifications for ALM, PLM and DevOps Interoperability
- Inspired by Internet architecture
  - Loosely coupled integration with "just enough" standardization
  - Common resource formats and services
- A different approach to industry-wide proliferation

Summary

- Complexity can rapidly increase as you develop products and systems
- Maintaining the various systems relationships manually is very difficult – maybe impossible
- OSLC provides an architectural solution to connect IBM Rational, Partner and competitor solutions together
- IBM's *solution for Systems and Software Engineering* automates the building of structures and dependency relationships to:
  - Manage increasing complexity
  - Enable collaboration across the entire development organization



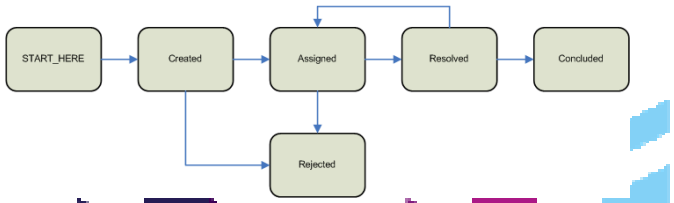
DOORS 9.3 with Generic CM integration via OSLC

Integrations to RTC, ClearQuest and Change

1. Requirements Change Management
   - **CM system controls changes to requirements**
     Workflow and approvals via CM system
2. Requirement Driven Development
   - **Stakeholder requirements submission**
     Generate requirements from enhancements submitted to Change Management system
   - **Requirements driven development**
     Create implementation tasks from requirements and monitor development progress alongside the requirements
   - **Requirement defect tracking**
     Associate a defect with a requirement to investigate a possible change to the requirement

OSLC-style linking & rich hovers as seen in Rational Team Concert

Process defined and controlled in CM system

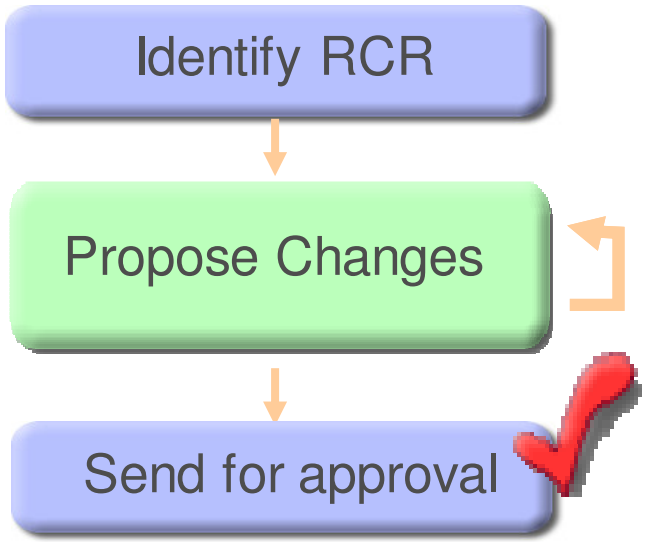


11

Controlling Requirement Changes

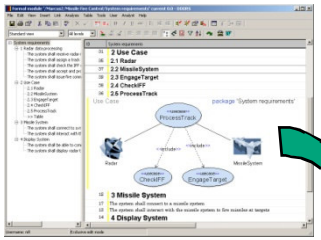
- Stakeholder Requests change(s)
- The Requirements Engineer works inside DOORS
- Selects an assigned Requirement Change Request
- Works inside DOORS
  - "Business as usual"
- Delivers the work for review
- The integration automatically packages the changes together

- Changes are not in the module until they are approved & applied



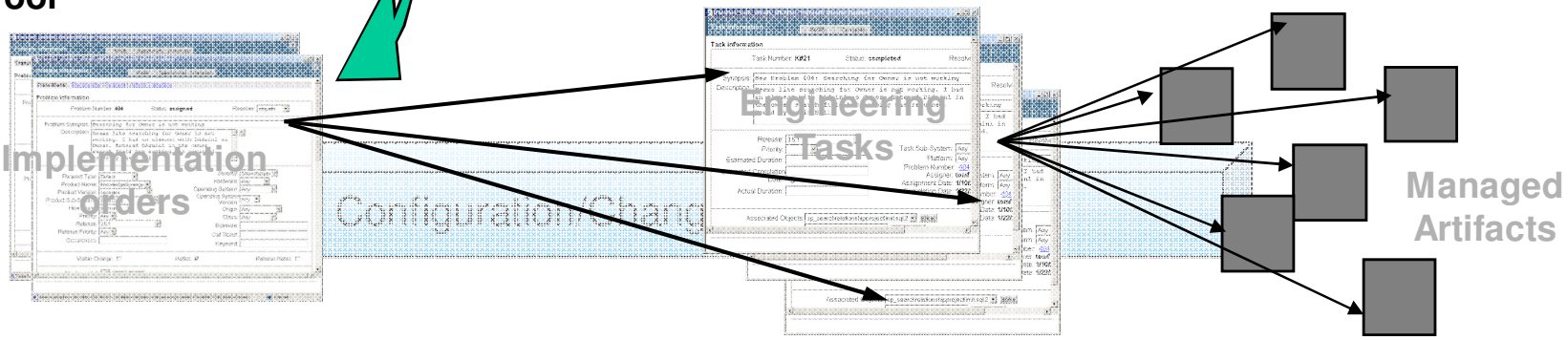
Requirement Driven Development

1) DOORS



- Requirements link to implementation orders
- Defects to be associated with requirements to investigate changes

2) CM Tool



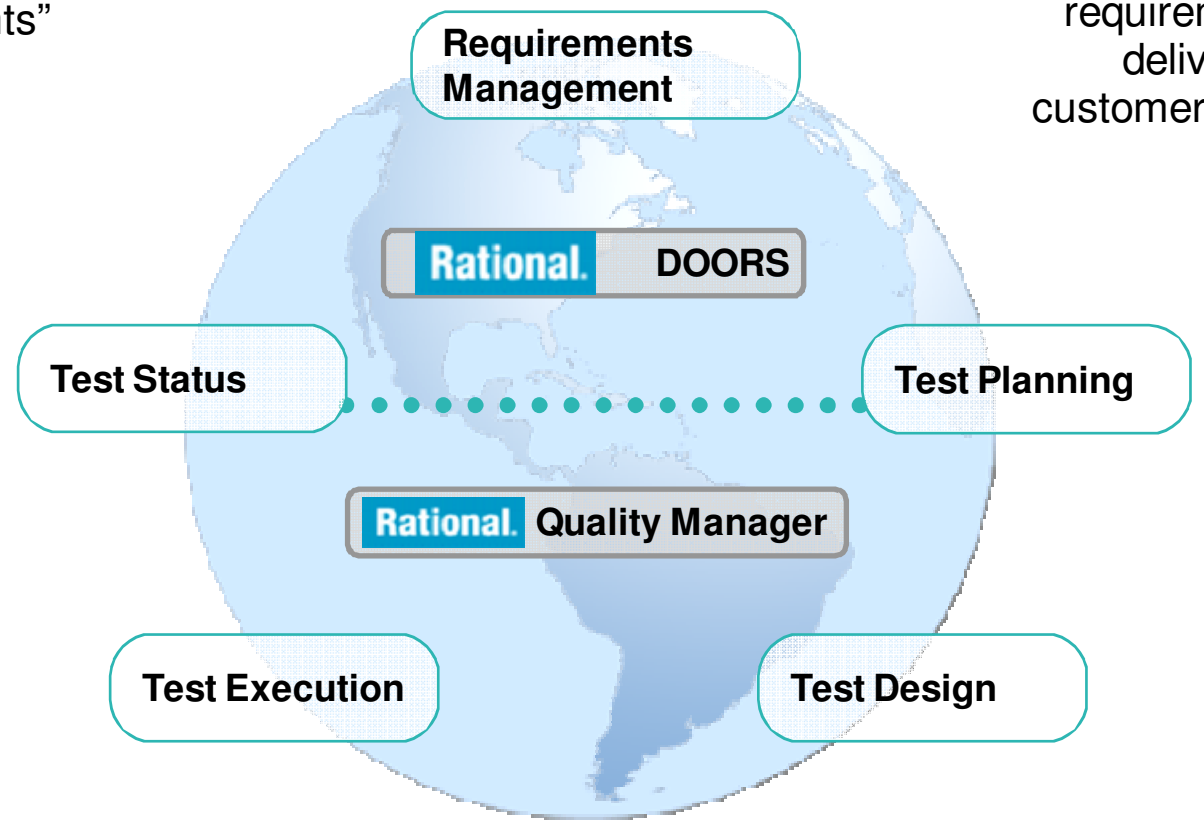
Traceability of actual activities, not just data – better impact analyses

Integrating DOORS and RTC

Integrated Requirements Management

"Quality is conformance to requirements"
Crosby

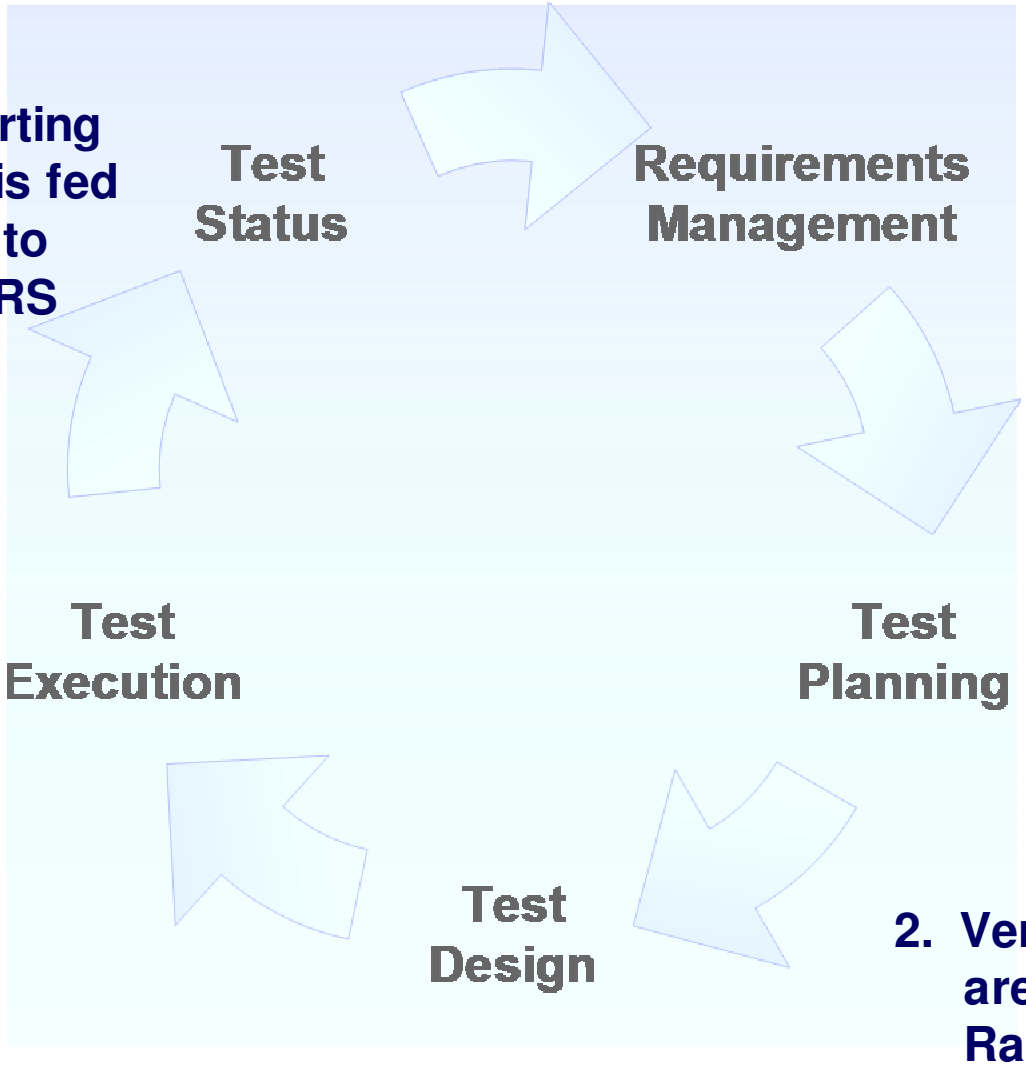
Tests based on requirements ensure deliverables meet customer expectations



Process Automation and Increased Focus

The test team is working against the right set of requirements

How the Integration Works – High-level



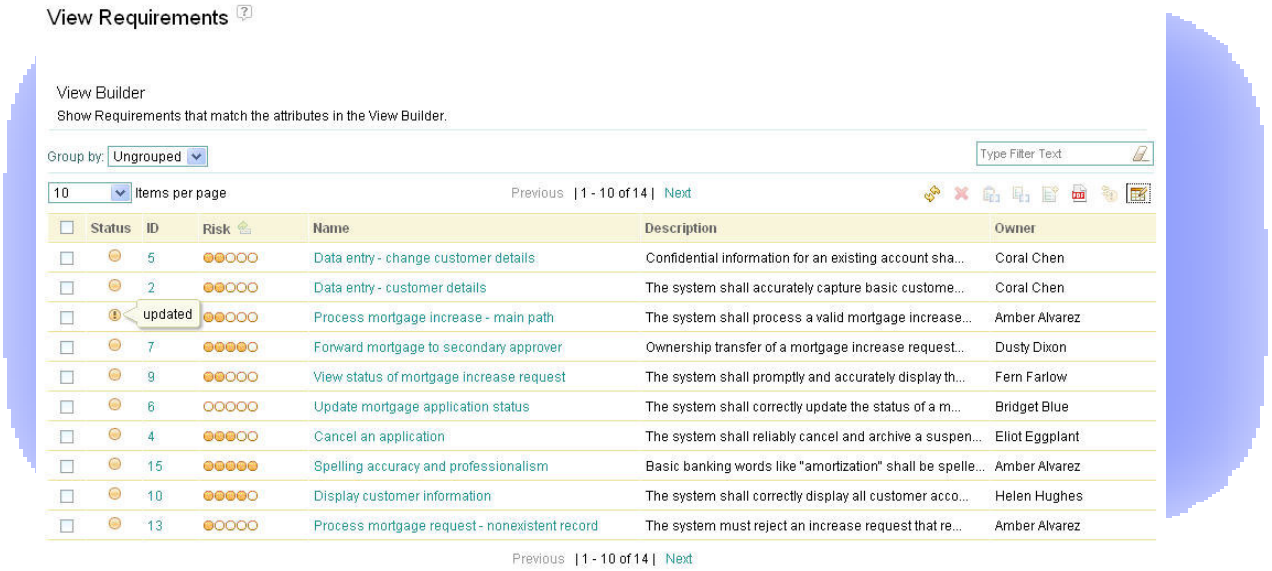
3. Reporting data is fed back to DOORS

1. Requirements in DOORS are pushed to a named test plan in Rational Quality Manager

2. Verification activities are detailed in Rational Quality Manager

Requirements driven testing

Knowing what to test



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

Know you are testing the right things

17

RQM Dashboards
Requirements Reports

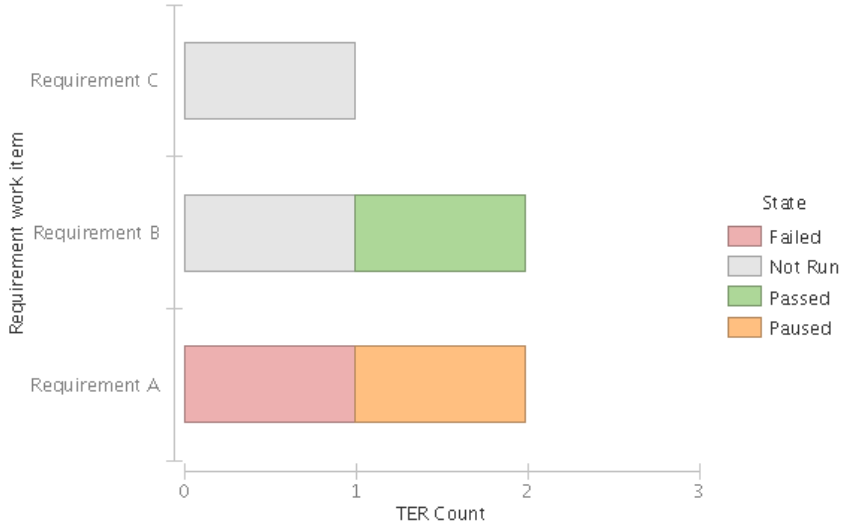
- *Are my Requirements covered and how ?*

Plan Requirements Coverage Detail

Requirements covered by test cases

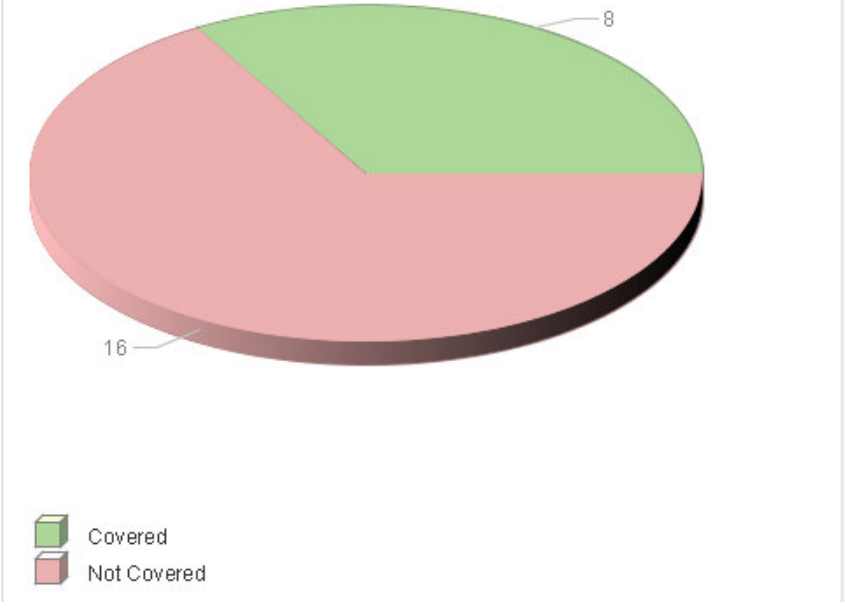
Test plan	Requirement ID	Requirement name	Test case
Test Plan - A			
	808	Requirement A	TCA1
	809	Requirement B	TCA2
	809	Requirement B	TCA3
	902	Requirement C	TCA3

Plan Requirements Execution using TER Count - Test Plan A



November 26, 2010 5:55:29 AM

Plan Requirements Coverage by Test Case



Traceability in DOORS

Requirements Reports

'Automated Meter Reader System Requirements' current 1.0 in /Water Meter (Formal module) - DOORS

File Edit View Insert Link Analysis Table Tools Discussions User RQM Publish Rhapsody 7.5.3 RG 7.5.3
Change Management Help

View: RQM Test Results Views | All levels

ID	System requirements for the AMR system	Test Cases	Verdict
39	The handheld device shall provide for the means for the meter reader to manually enter a meter reading.	(10) Test Usage Data indicator:	
37	The handheld device shall interfaces with the city's backoffice software.	(11) Test Route Configuration: Passed	Passed
40	The handheld software device shall allow for programming of a defined route, advancing to the next meter on the route as the meter reader moves through the route.	(11) Test Route Configuration: Passed (12) Test Run Route: Failed (15) Test Navigate Next Address:	Failed
41	The handheld device shall have the ability to search for Accounts by Last Name, Service Address, Meter Number, and Unread Meters.	(14) Test Search Address:	
42	The handheld device shall have a screen capable of displaying the number of accounts that have	(12) Test Run Route: Failed	Failed

Username: susan | Read-only mode

Integrating DOORS and RQM

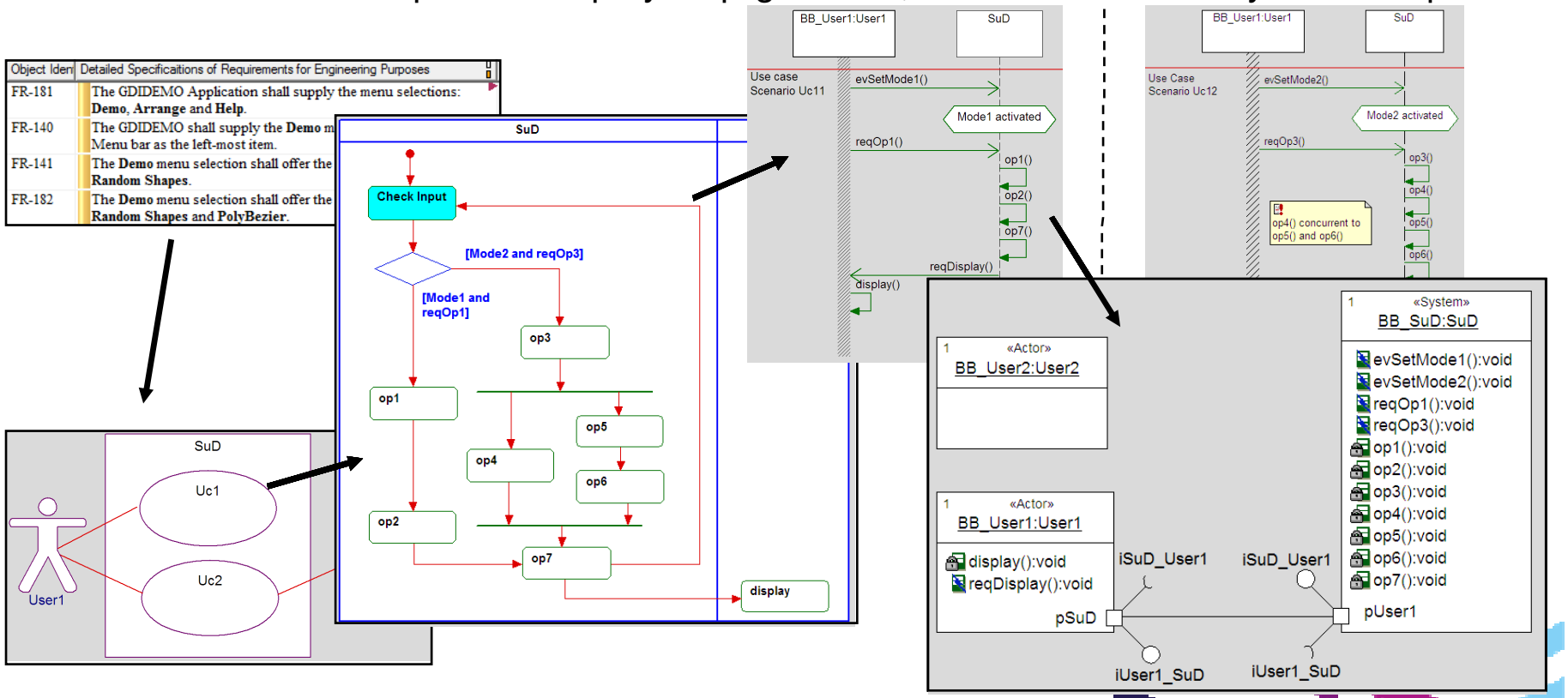
Model Based Systems Engineering (MBSE)

- A Systems Engineering analysis and design practice
- A visual approach to understanding requirements and realizing them into a robust system design
  - Used to refine and improve upon the system's requirements
- Helps manage complexity through the use of abstraction and separation of concerns

21

Modeling in Requirements Engineering

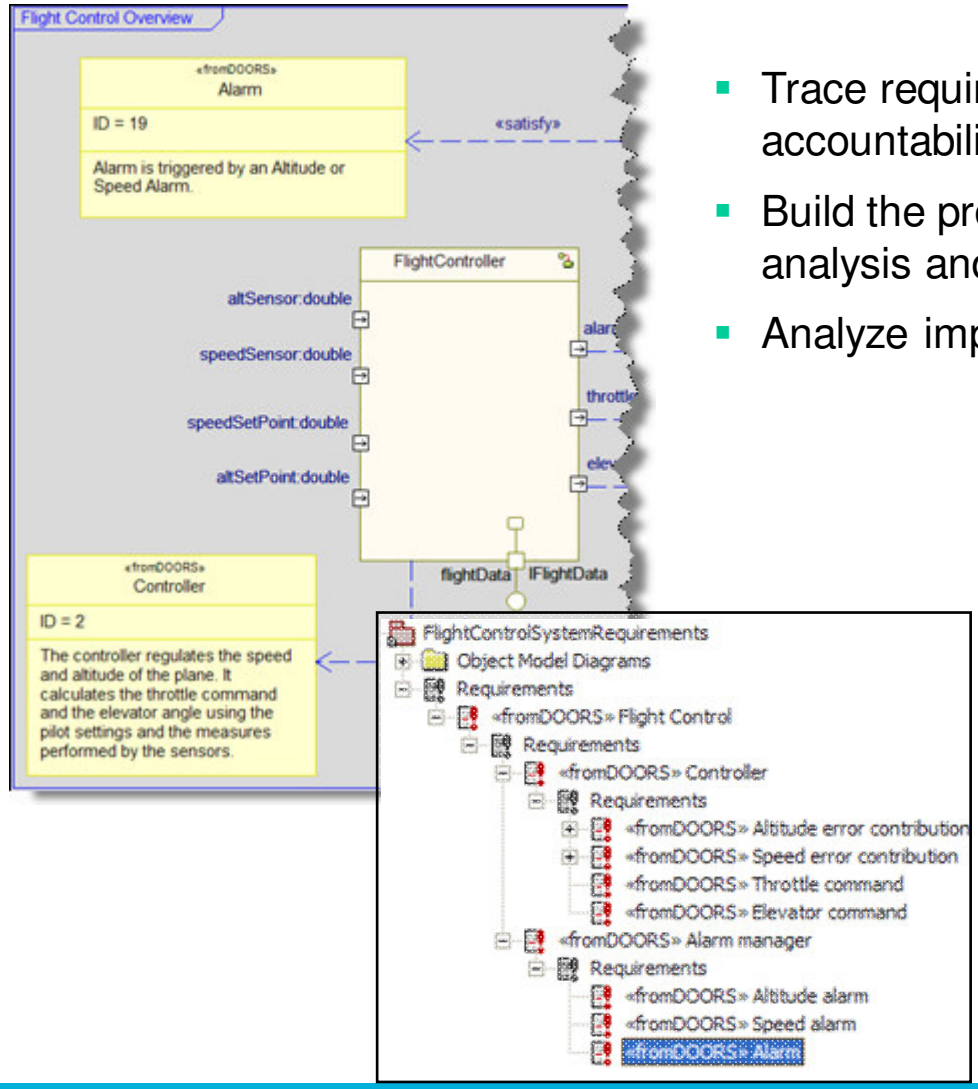
- MBSE complements traditional requirements analysis techniques
  - during Requirements Analysis, we organize requirements into functional groups (use cases)
  - during Functional Analysis, we identify system functions and explore the system's dynamic behavior using sequence diagrams and model execution
- Rational SE Practices provide step-by-step guidance, and automates many of the steps



22

Model driven analysis and design for Systems Engineers

using Rational Rhapsody and Rational DOORS



- Trace requirements in either direction for full accountability and understanding
- Build the product right with structural and behavioral analysis and design
- Analyze impact of changes in requirements or design

Integrating DOORS and Rhapsody

www.ibm.com/software/rational

www.ibm.com/software/rational