IBM Software Group

Turning Product Development Into Competitive Advantage:

IBM Rational Solutions for Complex Systems and Software Engineering

Eugen PASLARU

Rational Technical Specialist CEEMAS

Go to IBM

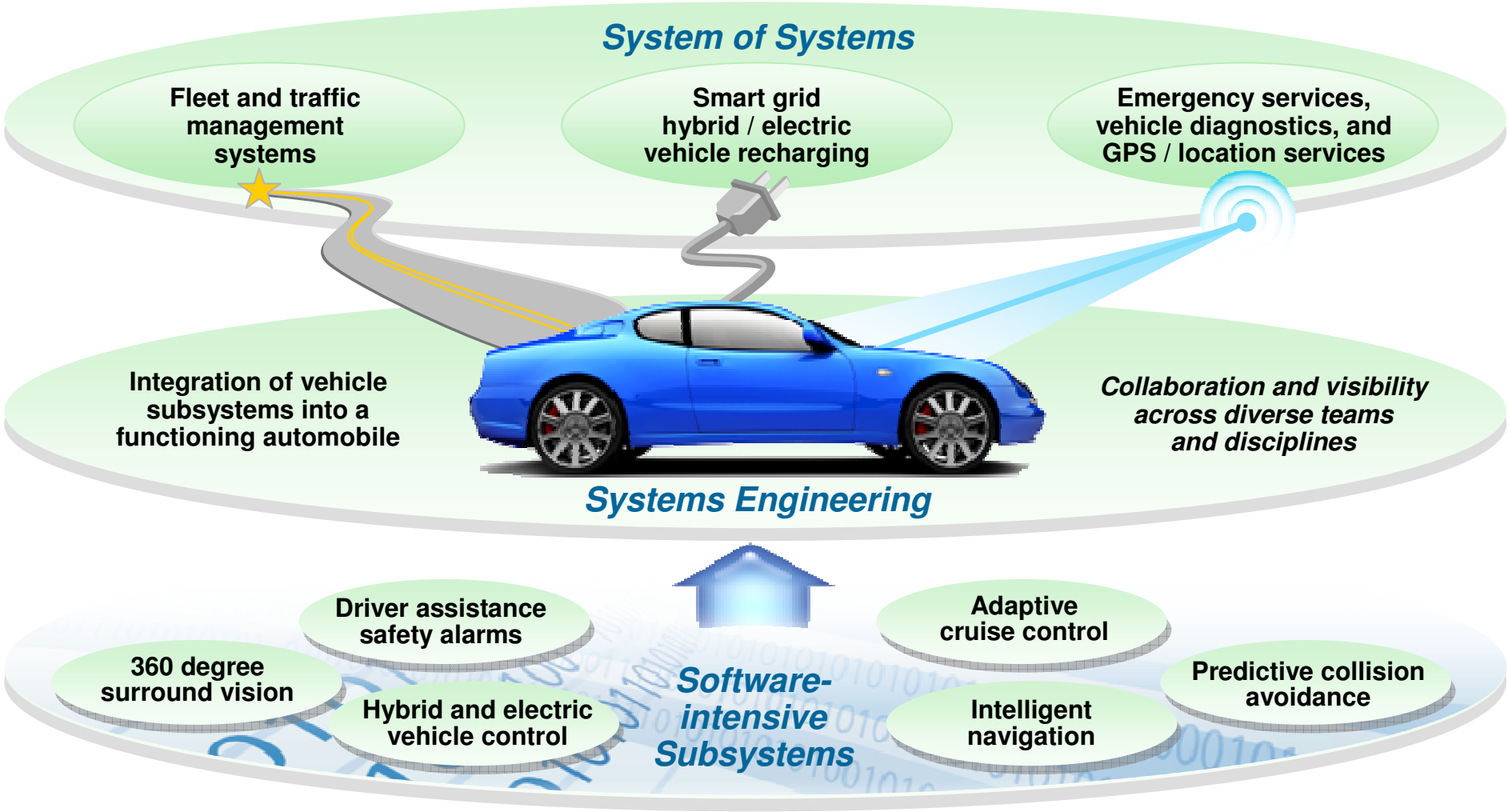
Products are Getting Smarter Every Time We Look

- 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 2012 it's **yesterday's news!**

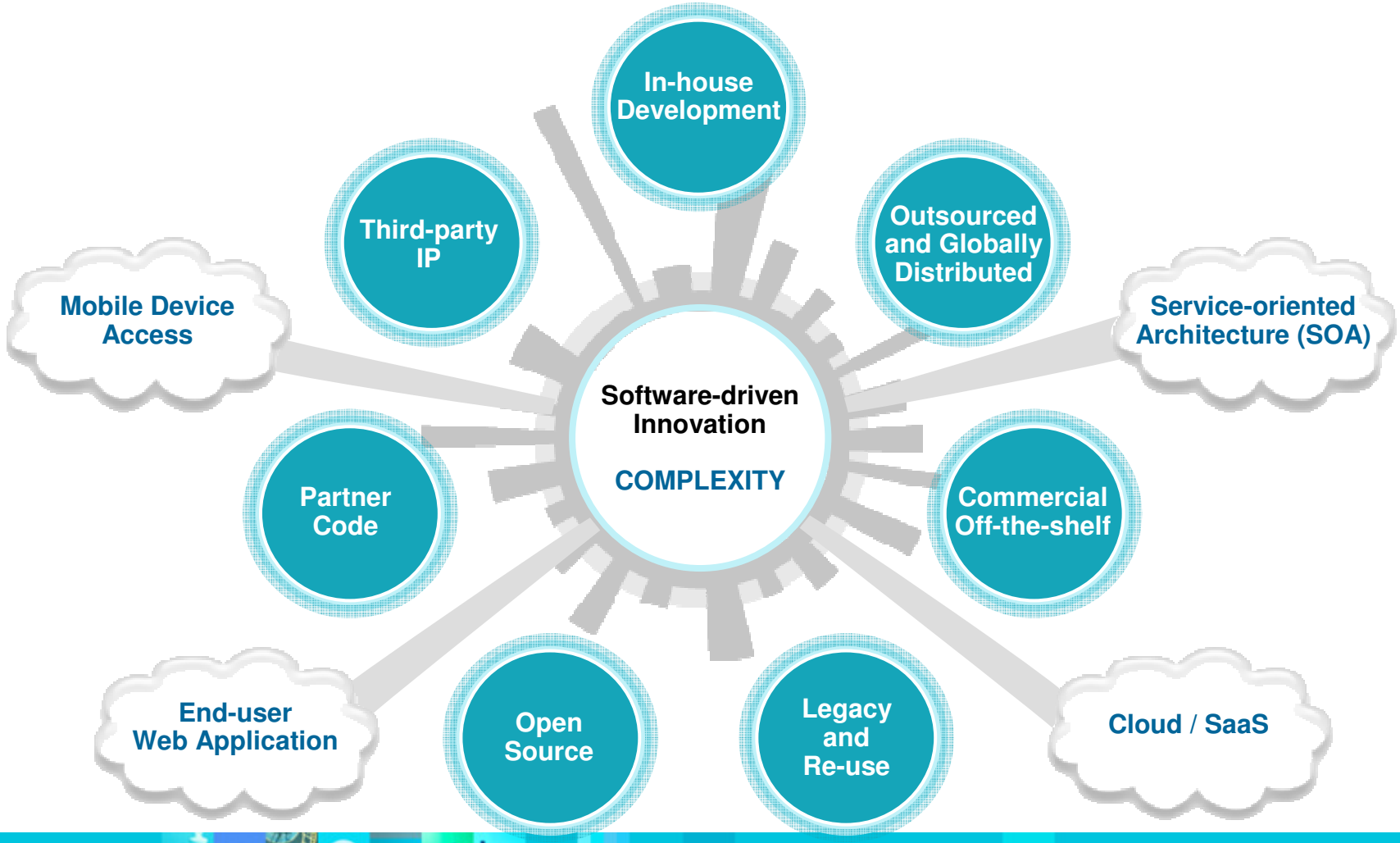
What's possible by 2020?

Smart Products Require Innovative Systems and new Development Methodologies

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



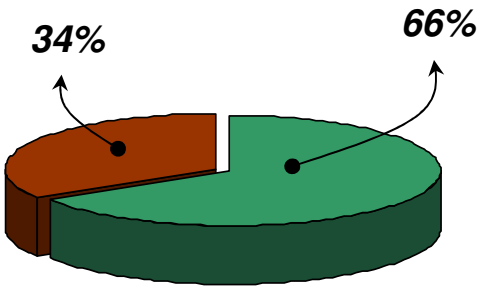
The defining challenge: Managing "systems of systems"



4

Software is the Heart of Today's Systems Innovation

Aberdeen Group
A Harte-Hanks Company



⅔ of all products developed include embedded software components

Source: Aberdeen Group, November 2008

"The medical field is highly dependent on software, which significantly enhances delivery of patient care"

MERGE Healthcare

"Like many of the components that make up today's vehicles, the hydraulic hybrid systems are intelligent software-intensive systems."

EATON

"Software has evolved from a hidden component driving functionality to the keystone of product differentiation and end-user experience."

-- VDC Research

Complexity Creates Development Challenges

Leading to cost overruns, schedule slips and quality issues

Poor requirements engineering = failed projects

Paper-based and manual processes hinder efficiency

Complex architecture is difficult to textually explain

Functionality is poorly distributed across components

Hardware/software integration is often late

Many organizations lack formalized practices

Silos of people, process, and projects

Geographic Barriers

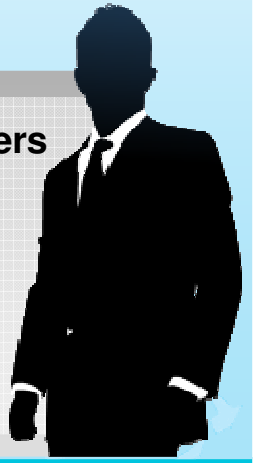
- Poor communication
- Language, culture, time
- Process gaps resulting in rework

Organizational Barriers

- Weak collaboration
- Poor project governance and LOB oversight
- Security of IP

Infrastructure Barriers

- Incompatible tools
- Unreliable access
- Lengthy on-boarding
- Inflexible integration



Modern Approaches for Describing Systems Are Evolving

To Better Manage Complexity and Reduce Time-to-market

Past

Specifications

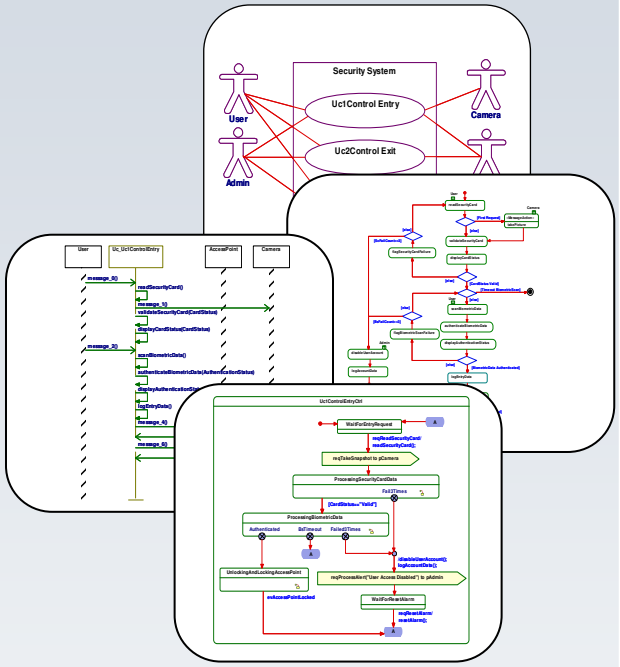
Interface requirements

System design

Analysis & trade-off

Test plans

Future

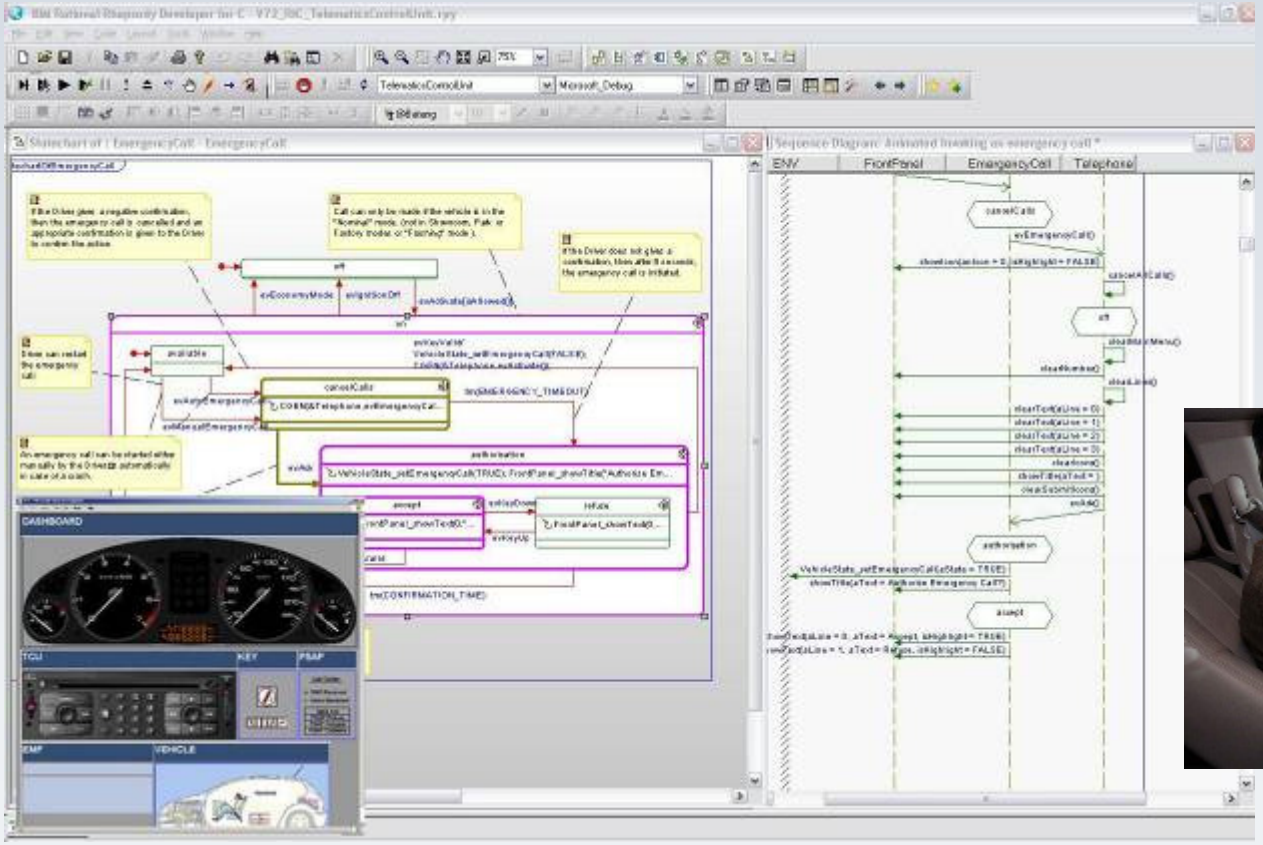


Moving from manual methods to an automated, visual approach

Rational Solutions for Systems and Software Engineering

Collaboratively refine requirements into a robust system

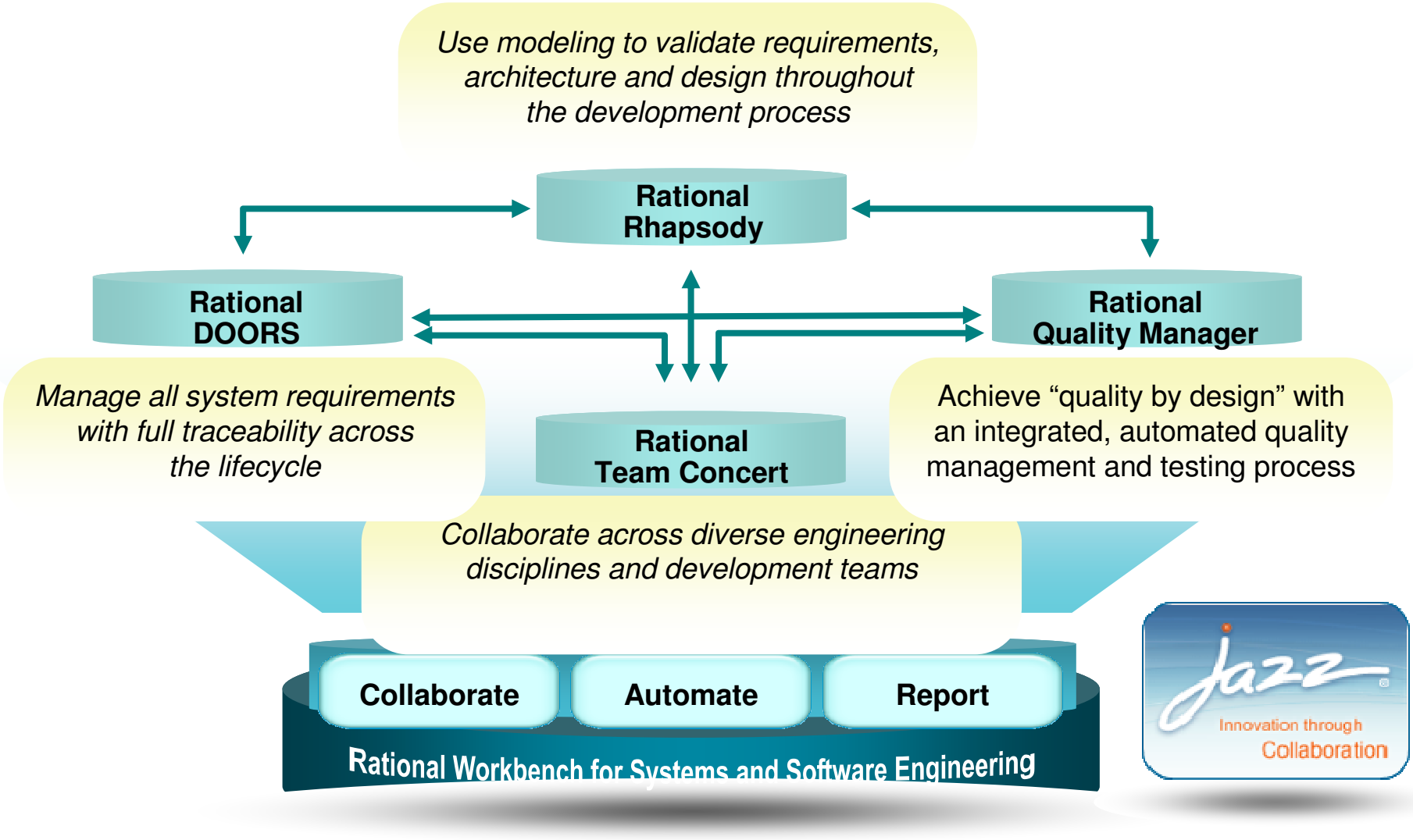
A standards-based practice for the development of complex systems across the mechanical, electronic and software disciplines



Rational Core Components for Systems and Software Engineering

Rational Solutions for Systems and Software Engineering

Built on a core product set



ALM is about connecting the disciplines

Project/Planning

- Business Drivers
- Iterations
- Sign-off
- Contract
- Risk Assess
- User Involvement

Requirements

- Use Cases
- Nonfunctional
- Sign-off
- Contract
- Risk Assess
- Threat Model
- Test Requirements

Development

- Test Driven Development
- Build Management
- Static Analysis
- Source Management
- Pair Programming/ Code Review

Testing

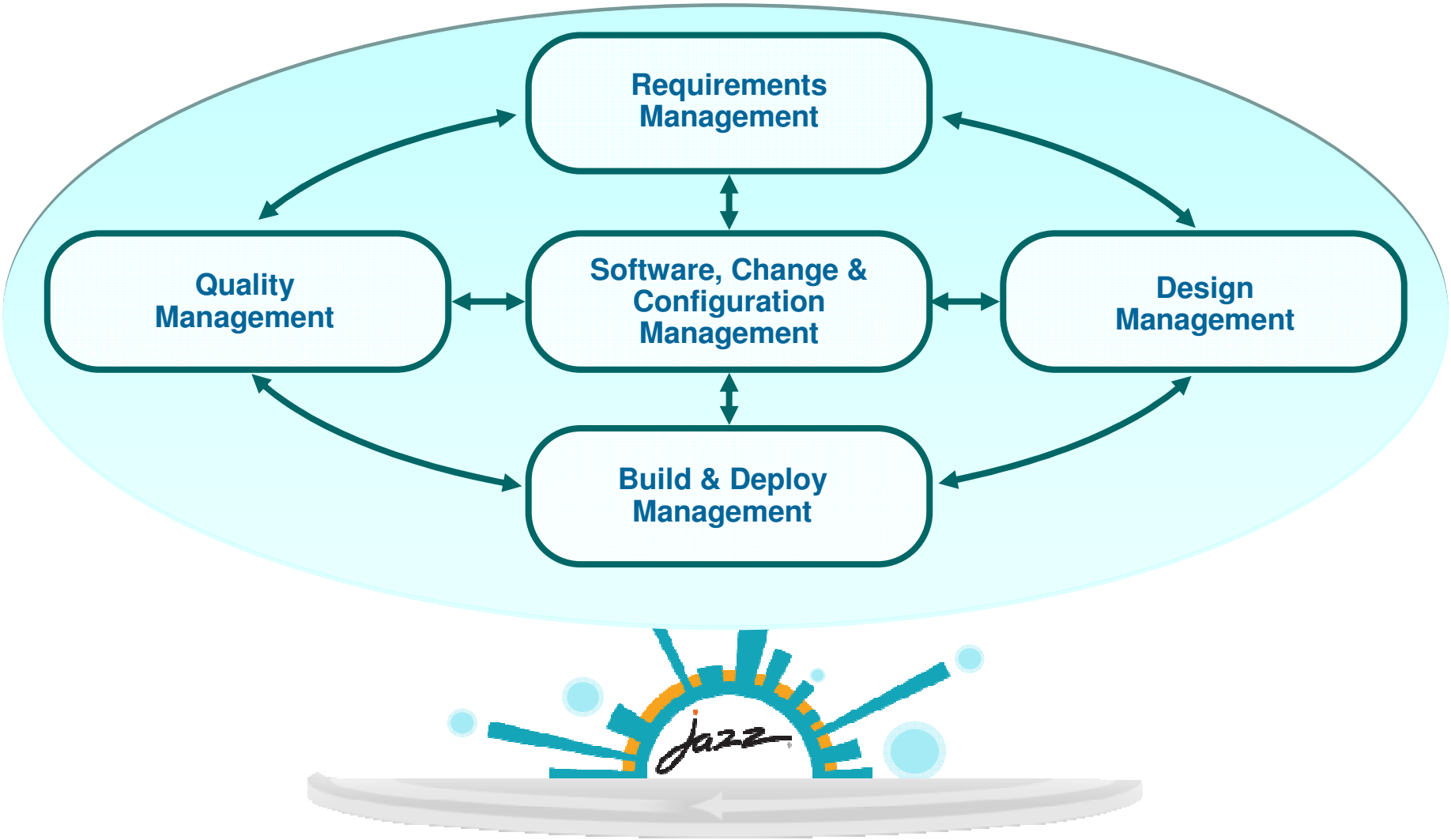
- Scenario-Driven Automation
- Exploratory Test
- User Involvement
- Contract Validation

Continuous Learning and Feedback

Source: Gartner Application Architecture, Development & Integration Summit Presentation, The Future and Present of AD, Thomas E. Murphy, December 2008

Rational Application Lifecycle Management (ALM)

Modular, open and extensible



12

Domain Focused Development

Apply industry and domain standards

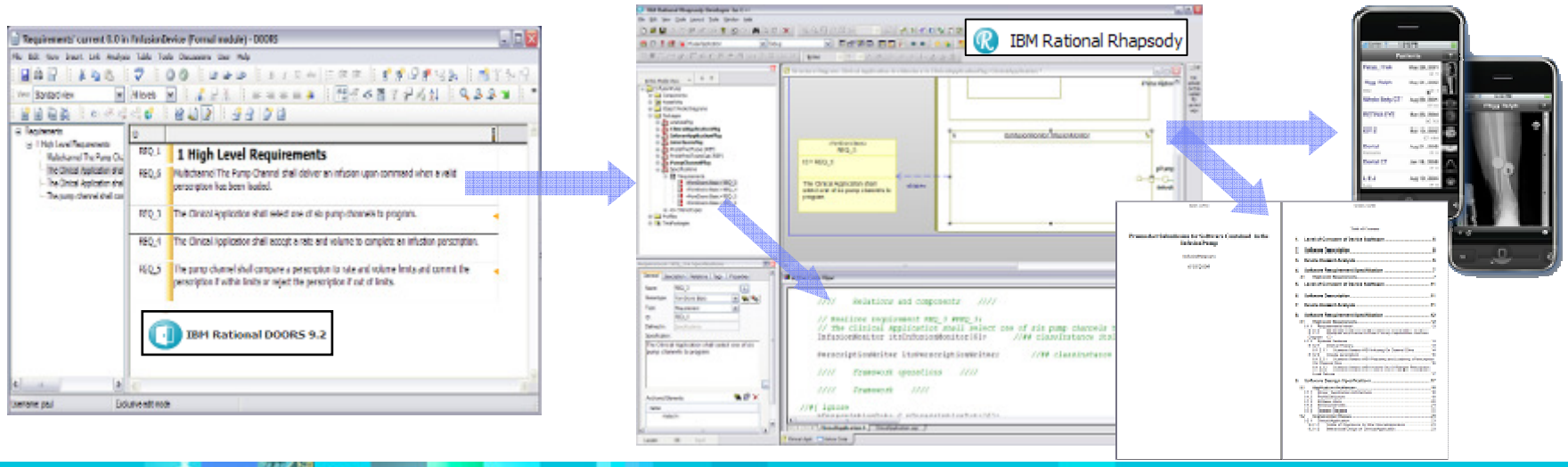
- Interconnected diagrams form multi-dimensional models
  - Can describe even the most complex systems

- Unified Modeling Language – UML 2.x
  - Industry-standard notation for specifying, visualizing, and documenting systems and software designs
- Systems Modeling Language - SysML
  - Extends/specializes UML to address needs of the Systems Engineer
  - Open standard published by the OMG and INCOSE

- Industry notations and frameworks: DoDAF, MODAF, UPDM, AUTOSAR…

Manage Requirements across Lifecycle and Disciplines

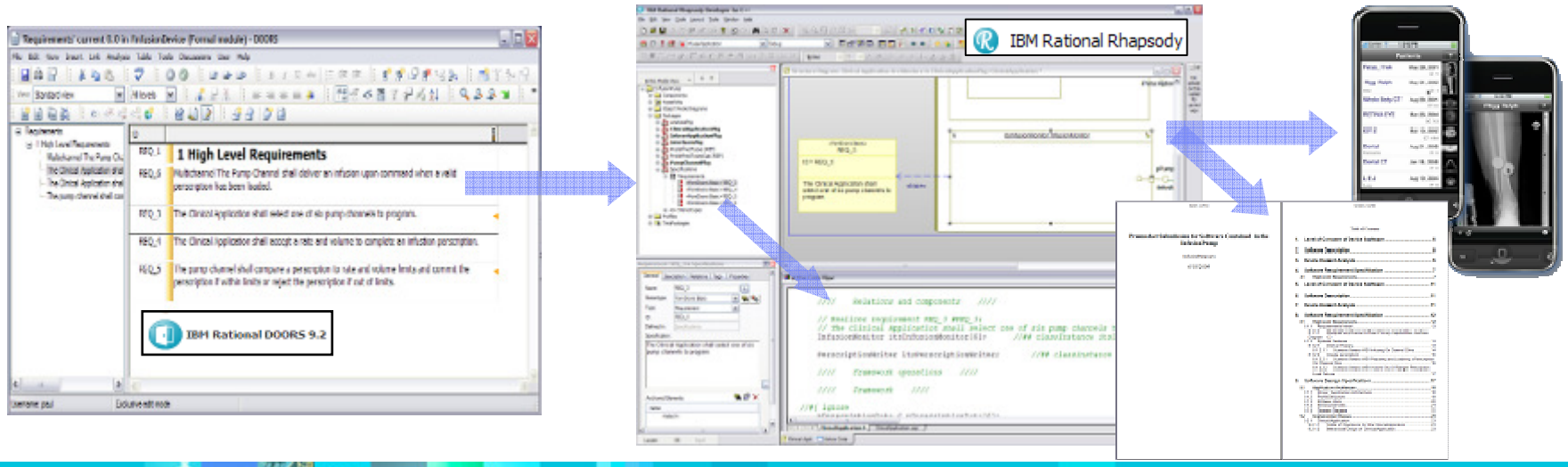
- *Build the right product* because the requirements are visible at all times
  - Prove that all requirements (user, safety, regulatory, etc.) were fully satisfied
- Understand the requirements
  - Analyze stakeholder needs
  - Evaluate coverage and impact analysis
- Validate the requirements
  - Analyze for correctness and to determine next steps



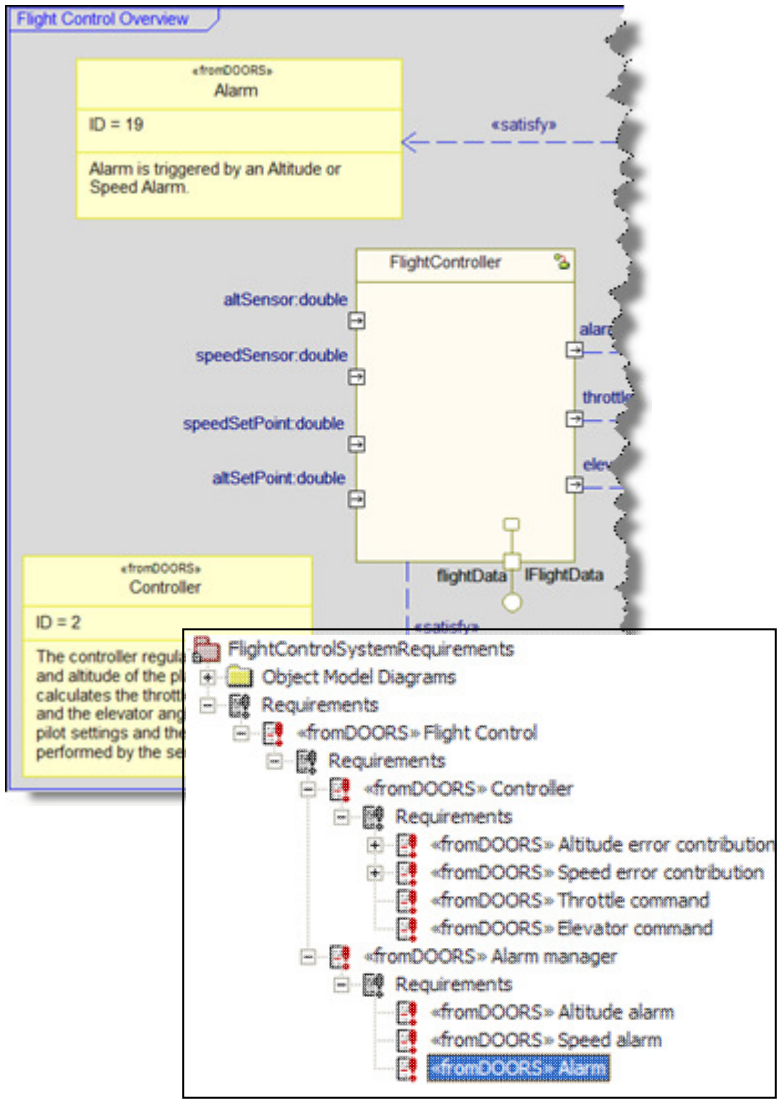
Manage Requirements in Context

Ensure success by meeting real needs

- Specify the system design through visualized requirements
  - Iteratively analyze and assess stakeholder needs
  - Link system requirements with the design for coverage and impact analysis
- Establish traceability throughout development
  - Analyze impact for every changed requirement
- Include functional and non-functional requirements
  - Match performance requirements with physical specifications



Translate Requirements into a System Design



- *Build the product right* with structural and behavioral analysis and design
- Visualize the system
  - Reduce confusion over requirements
  - Specify system functionality
  - Simulate to confirm functionality
- Analyze impact of changes
  - Whether in requirements or design
- Trace requirements in either direction
  - Provide full accountability and understanding
- Specify and develop software
  - Monitor and control the system

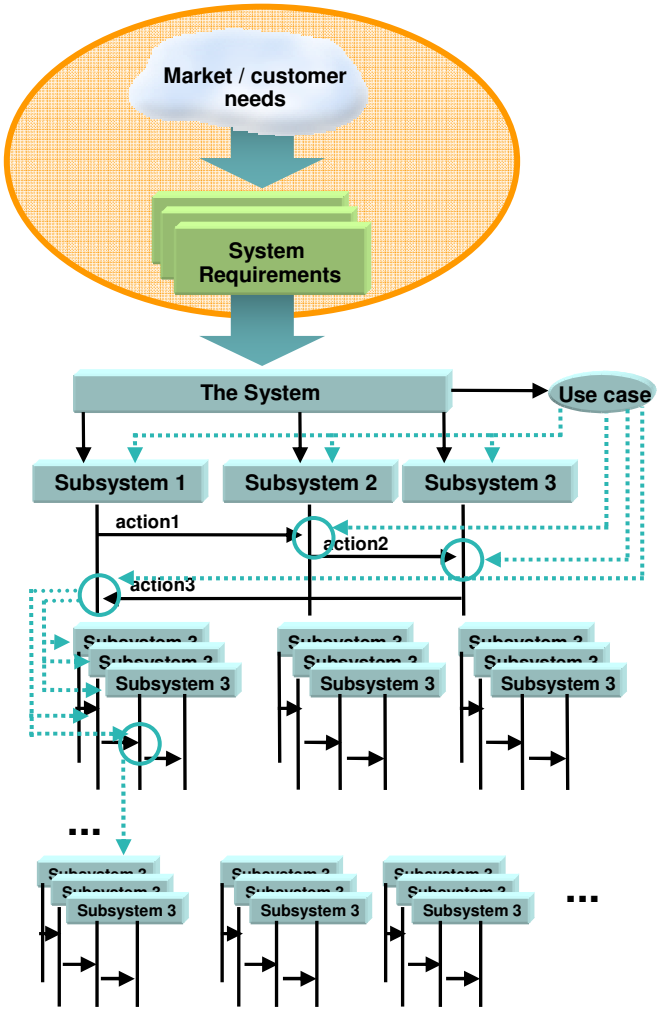
Create and Manage Your Architecture

Transform requirements into a working system

Achieve Quality by Design

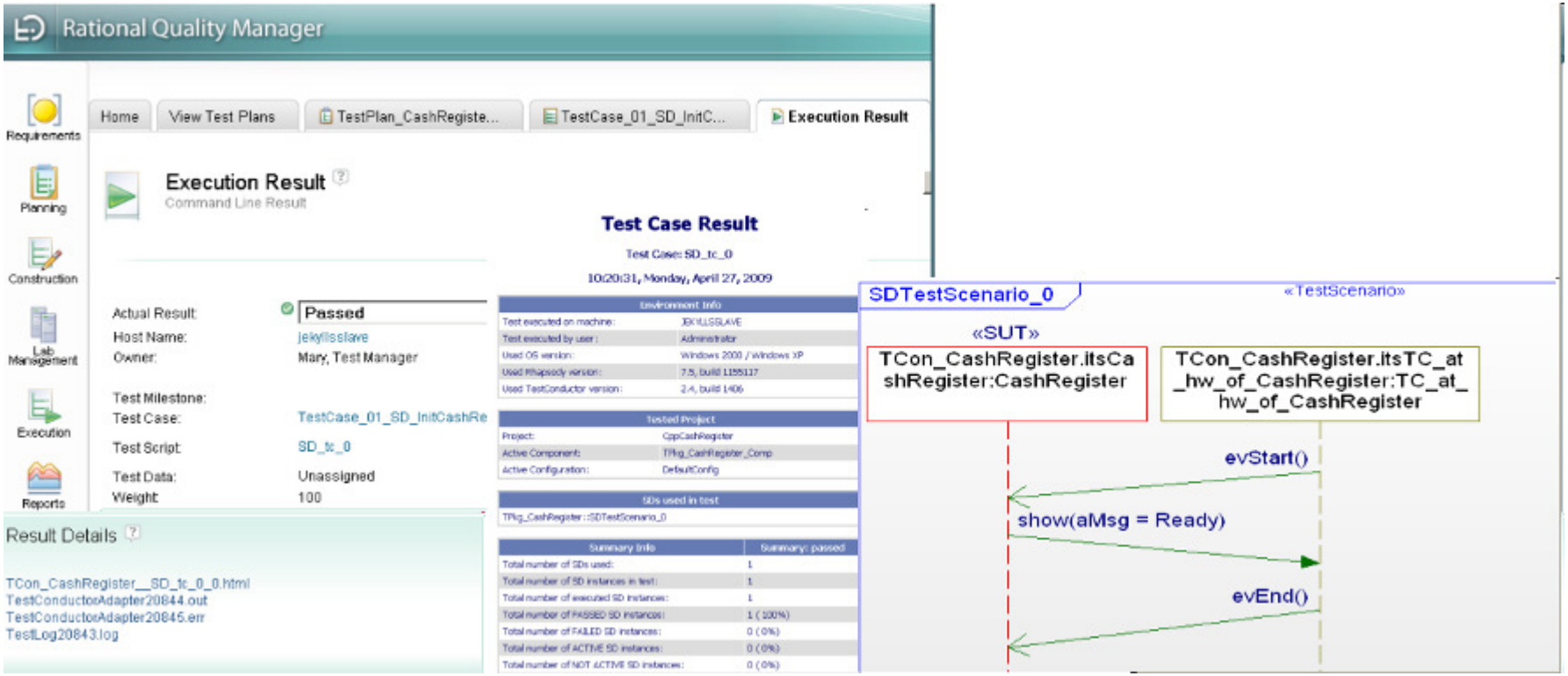
- Derive the system in the context of its environment
  - Reduce confusion over requirements
  - Establish system functionality and its constraints
- Eliminate errors as they are introduced
  - Before they are too expensive to find and repair

- Simulate often
  - Animate and execute the design model
  - Validate functionality and verify correctness
- Automatically create and execute tests
  - Derive from the design model or target platform
  - Create test harnesses for unit testing
- Manage test cases
  - Prioritize the features and functions to be tested



Build in Quality from Concept to Launch

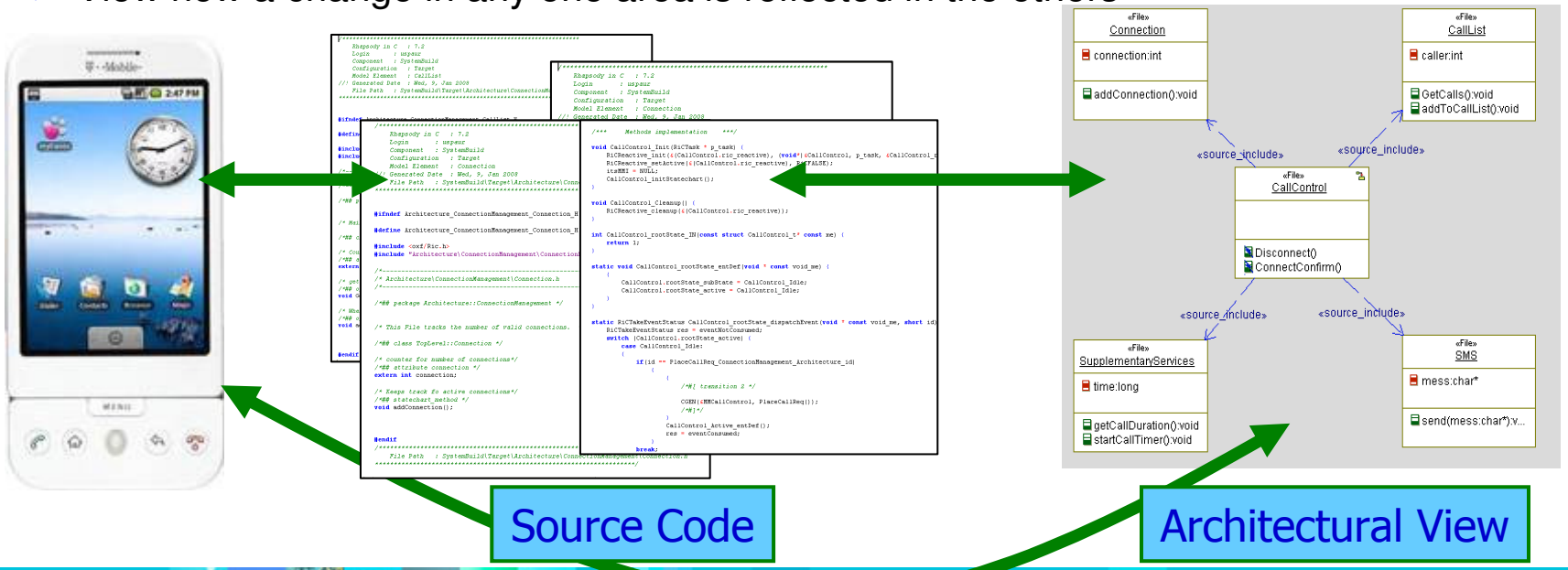
- Simulate often to validate functionality and verify correctness
- Automatically create and execute tests from the design model or target platform
- Manage test cases, while prioritizing the features and functions to be tested



Implement the System

Create software that matches the requirements

- Build efficient embedded software that powers the system
  - Specify and test deployable source code from the system requirements
  - Generate complete C, C++, Java, and Ada applications – including behavior
- Synchronize between architecture and code
  - Simultaneously work with the design model, software and target
  - View how a change in any one area is reflected in the others



Recapture Intellectual Property

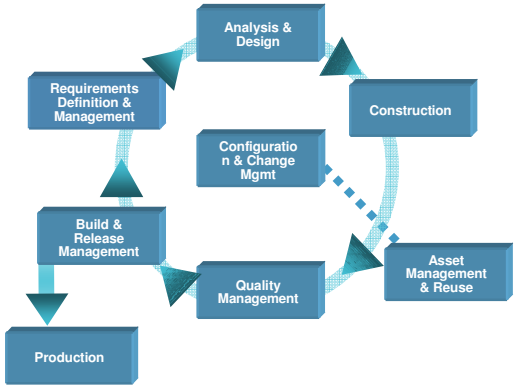
- Preserve intellectual property
  - Visualize and reverse-engineering existing software
  - Create a library of design assets
  - Analyze to best meet requirements
- Work with product lines
  - Expand product offerings
  - Exploit commonality across products
  - Focus efforts on unique product variants

Utility

Speed

Fuel economy

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…

Learn more at:

- IBM Rational software
- IBM Rational Software Delivery Platform
- 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