


# Overview and Demo of Requirements Management in DOORS


Srinivas Mandava  
Rational DOORS



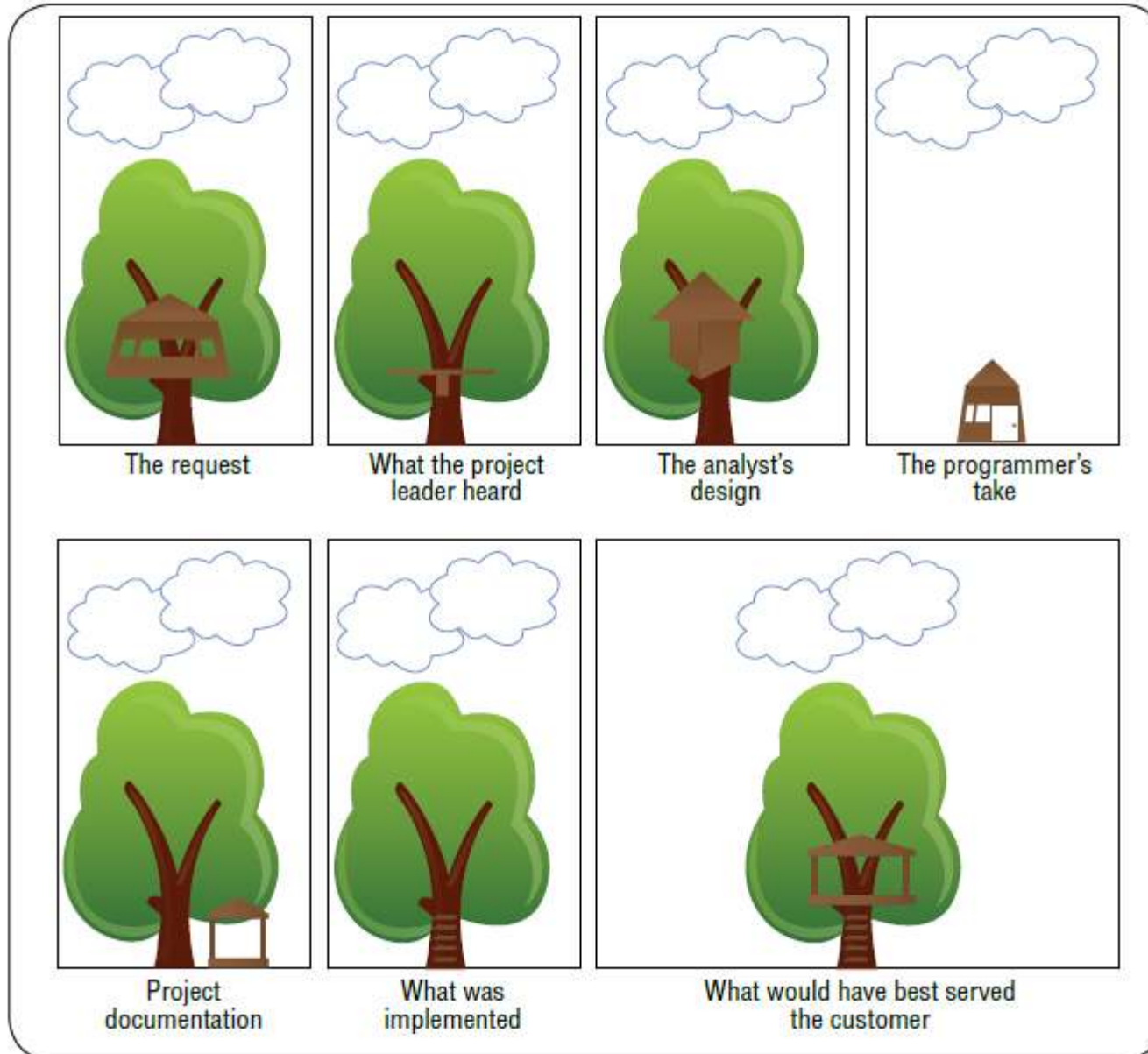
## Agenda

- 
- 1 Why requirements management?
  - 2 Addressing your needs for RM with DOORS
  - 3 Where does RM fit into the wider story
  - 4 Integrations

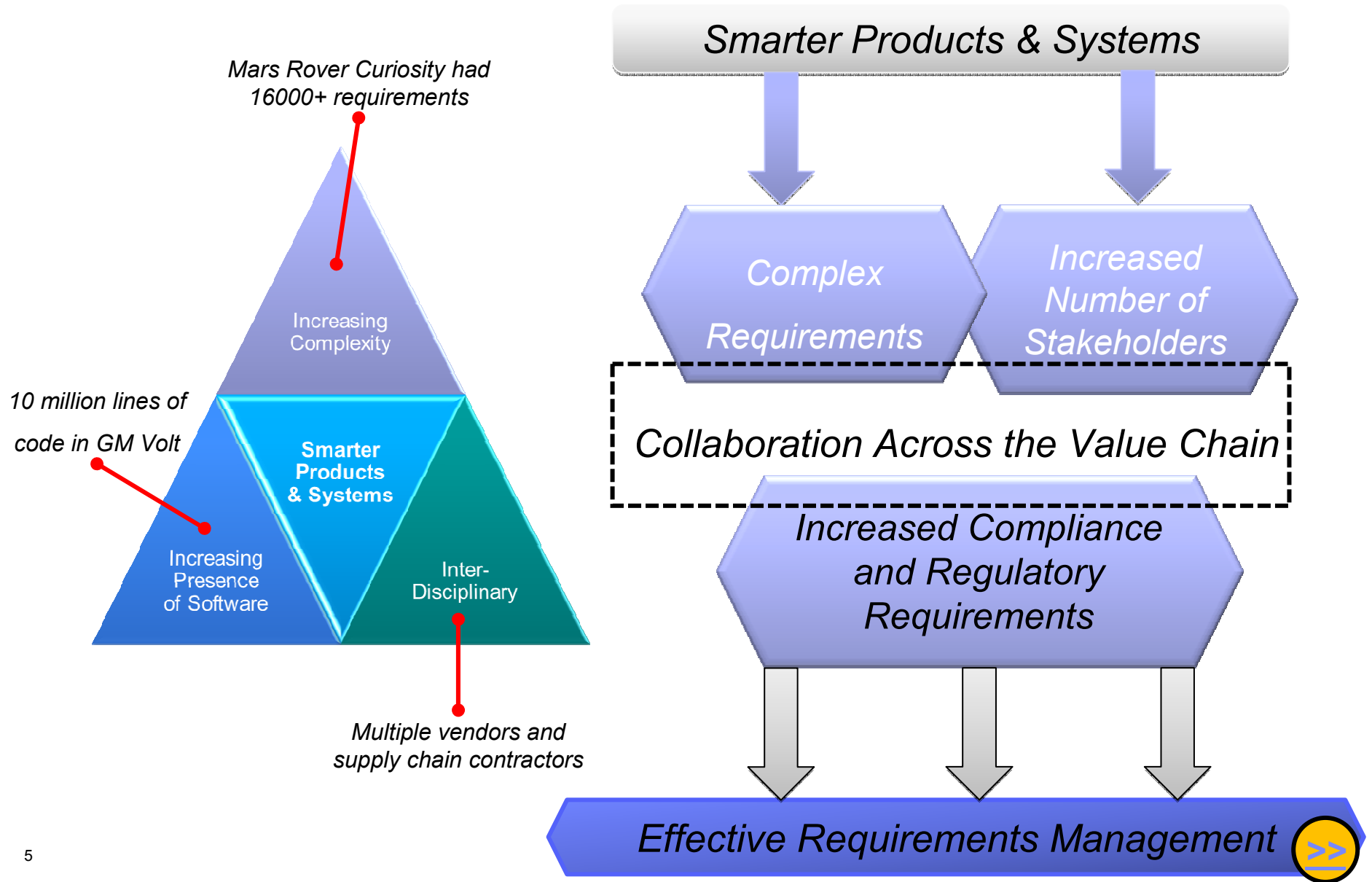
## Agenda

- 
- 1 Why requirements management?
  - 2 Addressing your needs for RM with DOORS
  - 3 Where does RM fit into the wider story
  - 4 Integrations

## Why Requirements Management?



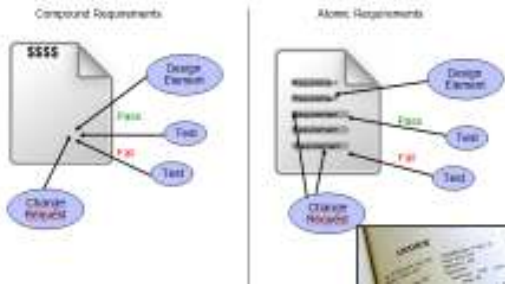
## Why Requirements Management?



## Common Concerns

- Access to timely and accurate Information
- Disconnect between business goals and requirements solution
- Lack of visibility to relevant project status
- Process Standardization
- Compliance and Regulation
- Globally Distributed Development Challenge
- Complex Supply Chain Scenario

# Key requirements for an RM system



**Granularity**



Recording who did what

... and when



Controlling who can see and change information

**History and Security**



Disorganized

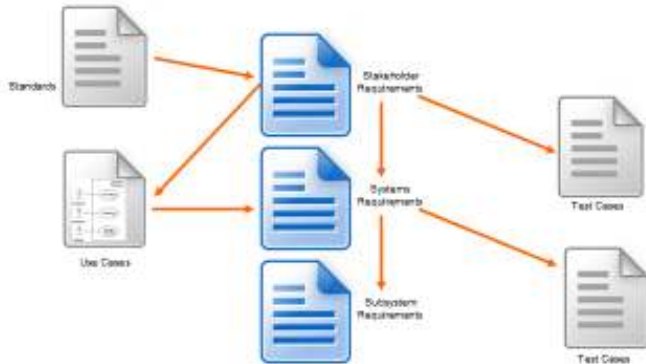


Organization

**Hierarchy**



**Reporting**



**Traceability**

Collaboration



Between customers ...



...Product Managers...

...the business...



...Engineering...

and Test




**Collaboration**



**Attributes**

## Agenda

- 
- 1 Why requirements management?
  - 2 Addressing your needs for RM with DOORS
  - 3 Where does RM fit into the wider story
  - 4 Integrations

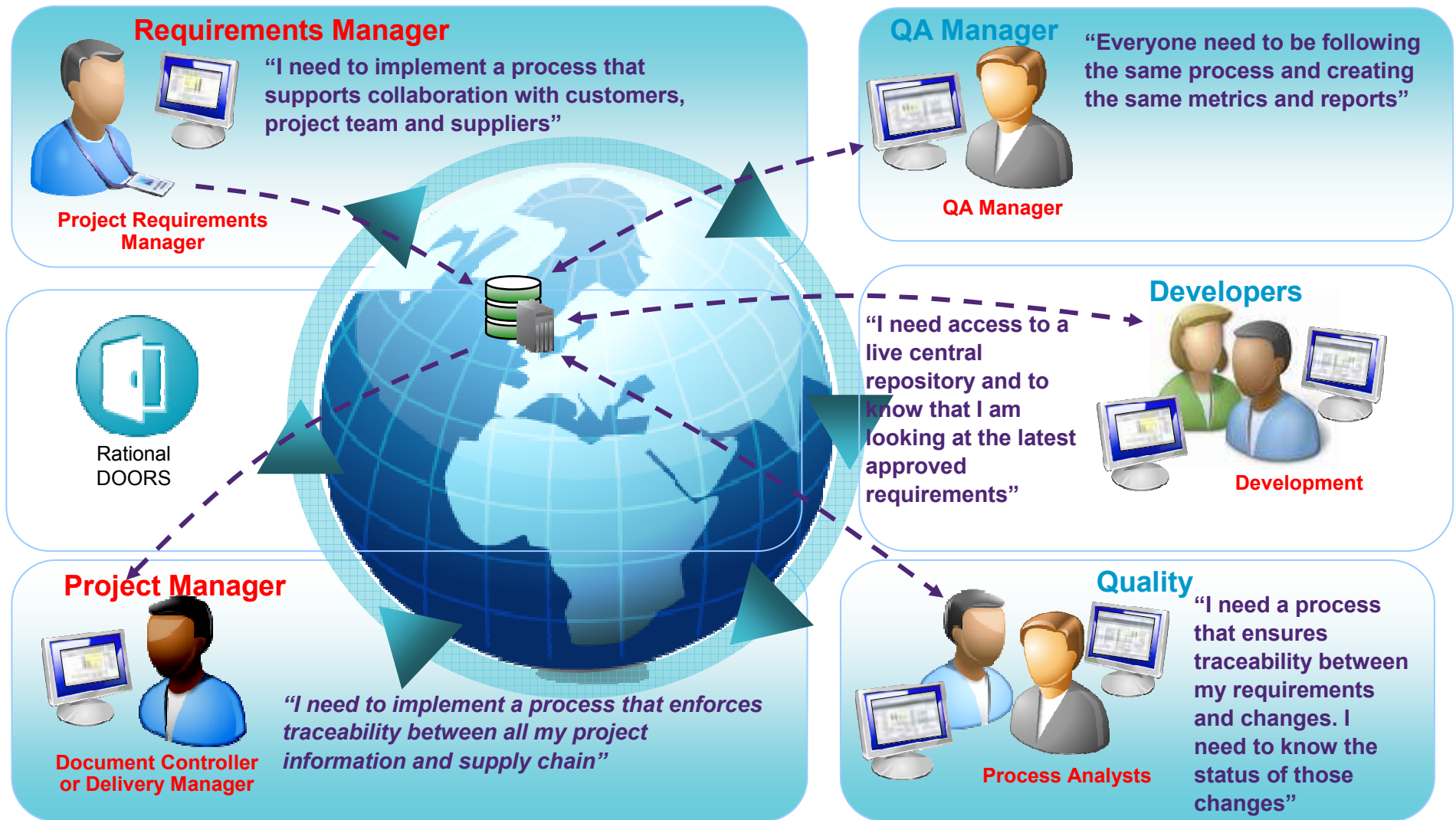


## DOORS

A requirements management tool with:

- Multi-user document access
- Centralized Document Oriented View and Storage
- Extensive access controls
- Change control & tracking
- Linking Support
- Multiple representation of Requirement : View and Filter
- Traceability & impact analysis support
- Baselining
- Multiple data exchange possibility
- Extendibility - DXL

## Addressing your needs for RM with DOORS



## DOORS User Interface and data representation

The screenshot displays the DOORS User Interface for a 'New Family Car Project'. The interface is divided into two main panes: a left-hand navigation pane and a right-hand content pane.

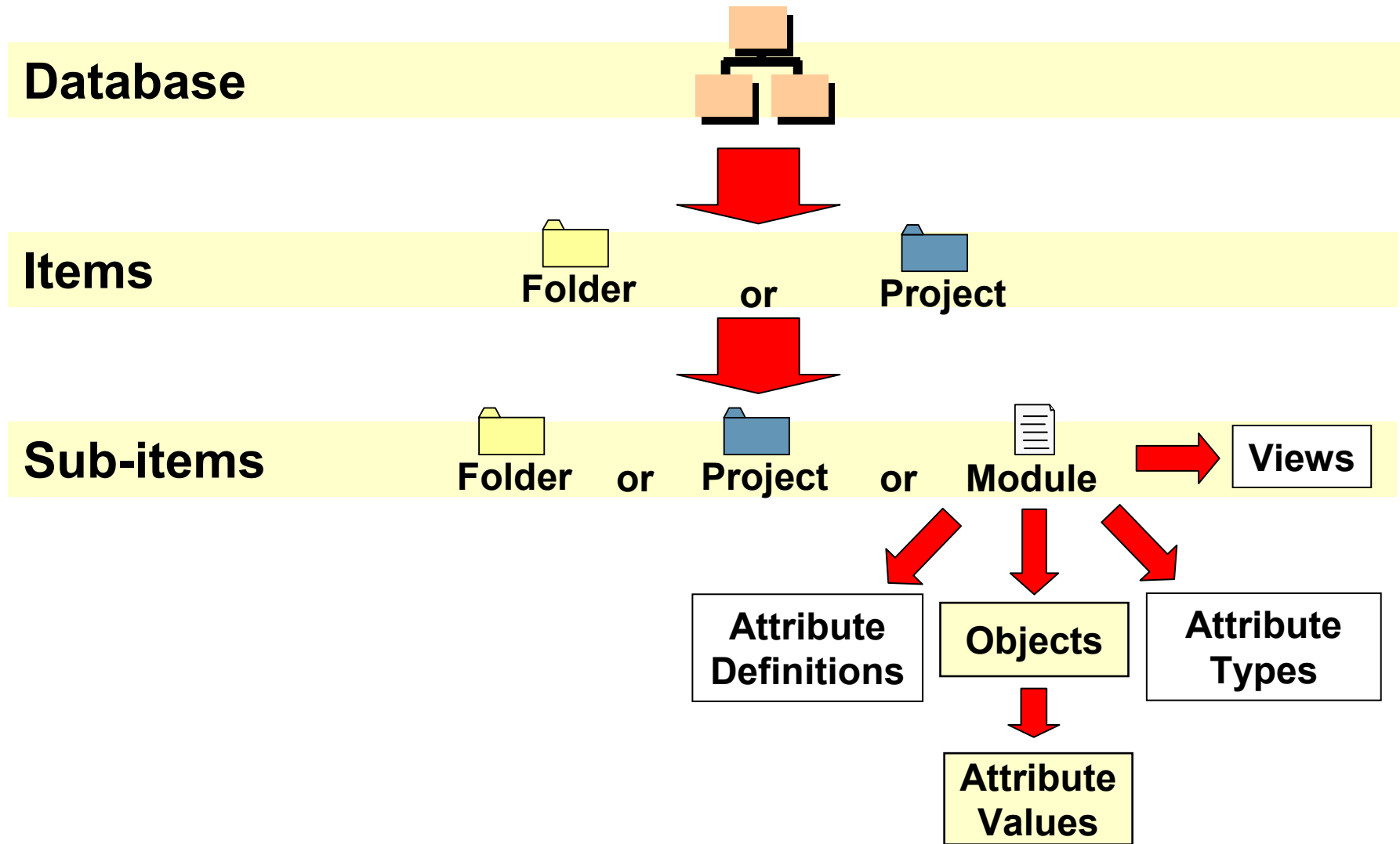
**Navigation Pane (Left):** Shows a hierarchical tree structure starting from the 'Company DOORS Database' root. The tree includes folders for 'Company Standards', 'Product Division', and 'New Family Car Project'. Under 'New Family Car Project', there are sub-folders for 'Admin', 'Architecture', 'Planning', 'Reference', 'Requirements' (highlighted), 'Test Cases', and 'Use Cases'. An arrow labeled 'Navigation' points to this pane.

**Content Pane (Right):** Displays a table of requirements under the 'Current Location' of '/New Family Car Project/Requirements'. The table has columns for 'Name', 'Type', and 'Description'. The requirements listed are:

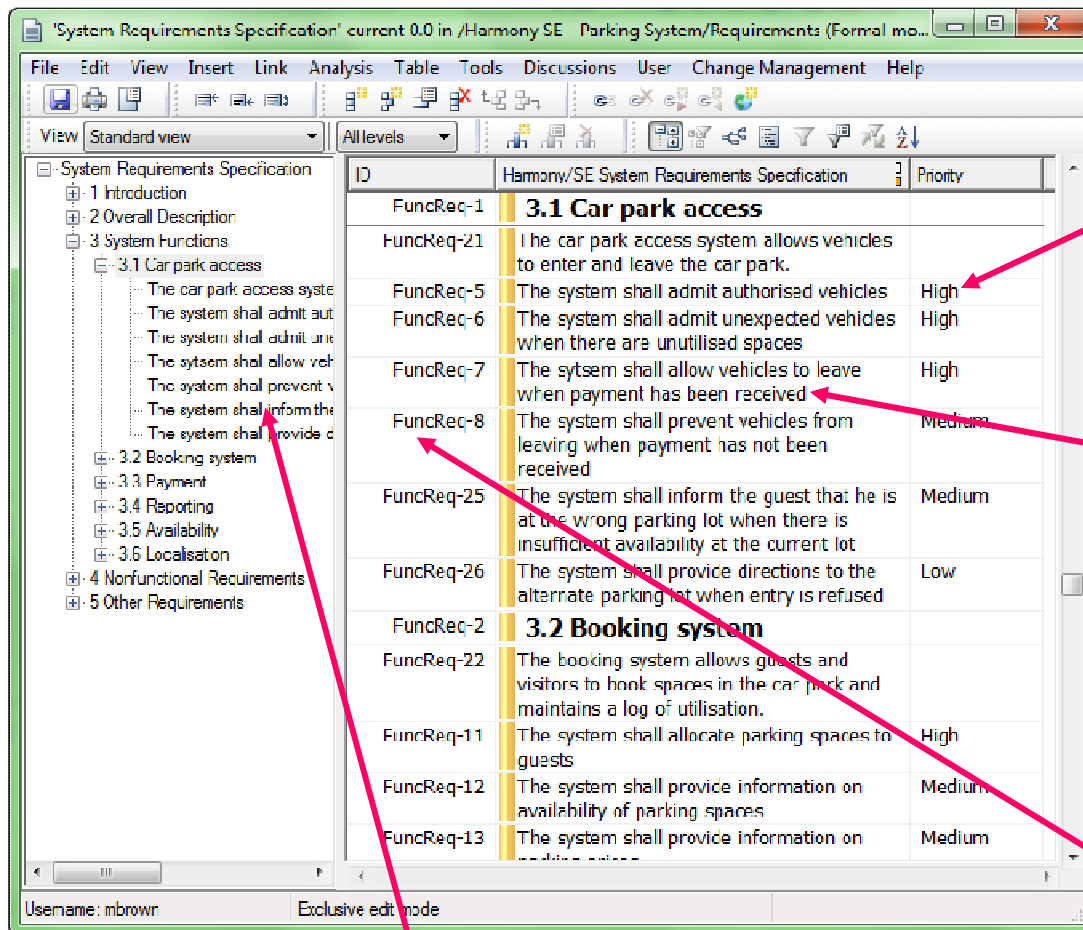
Name	Type	Description
Entertainment System	Formal	In-car entertainment user requirements
Marketing Requirements	Formal	A Vision document of a typical journey..
Stakeholder Requirements	Formal	Car user requirements
System Requirements	Formal	Car system requirements

An arrow labeled 'Content' points to this table. Other annotations include 'Database root' pointing to the top of the tree, 'Project' pointing to 'New Family Car Project', 'Folder' pointing to 'Requirements', and 'Formal module' pointing to the 'Formal' type in the table.

## Hierarchy



## Granularity – Use Atomic Requirements to Accurately Control Dependencies and Acceptance



Set attributes like cost, risk, priority accurately

Create traceability to other requirements, tests or information

Reference requirements with precision

Navigate to specific statements

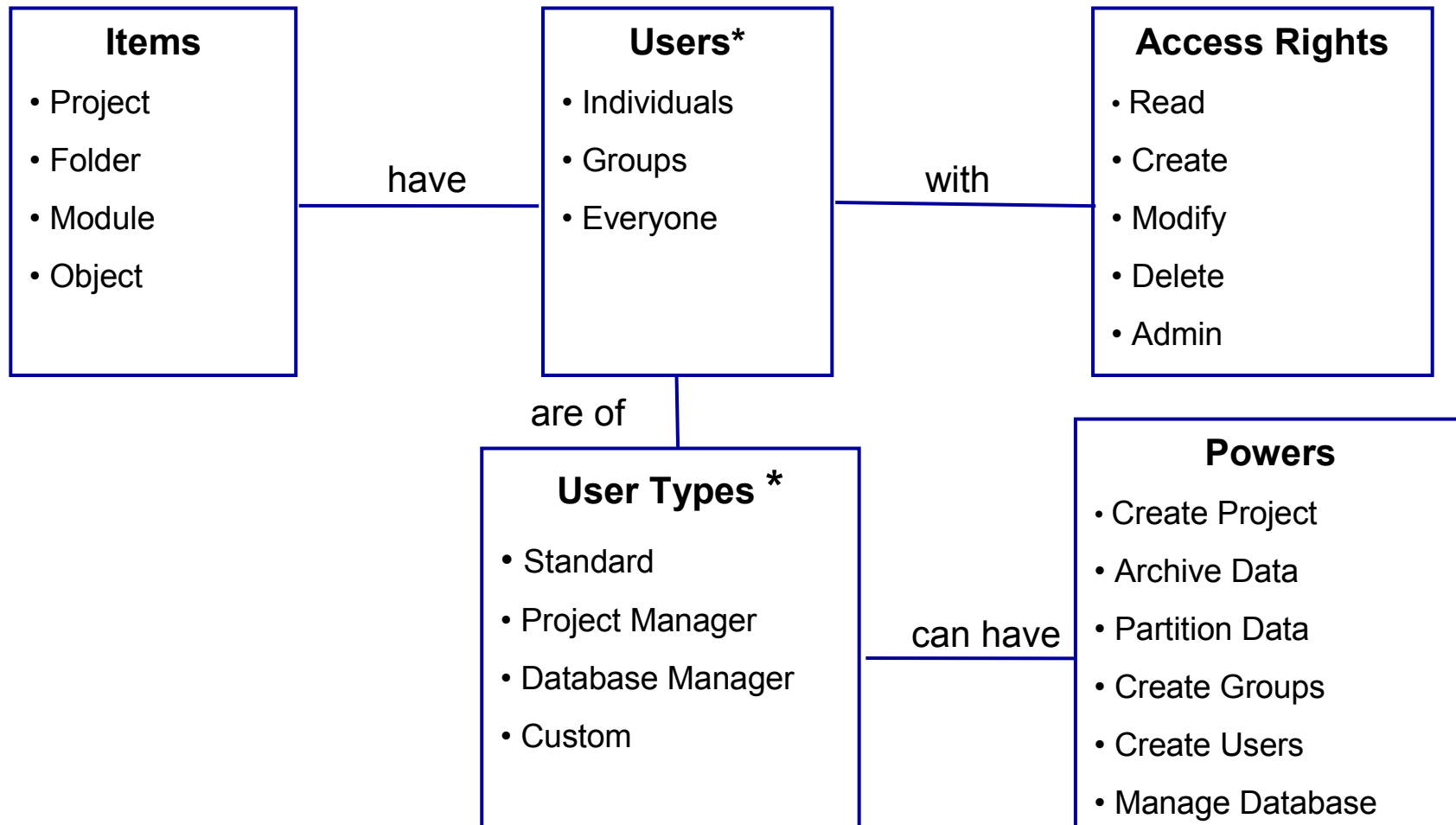
## Attributes – to support and control process

The screenshot shows a window titled "'System Requirements Specification' current 0.0 in /Harmony SE - Parking System/Requirements (Formal module) - DOORS". The window contains a table with the following data:

ID	Harmony/SE System Requirements Specification	Information Type (Rqmt)	Status	Priority	Proposed Verification Method
FuncReq-95	<b>3 System Functions</b>	Heading	Unconfirmed		
FuncReq-1	<b>3.1 Car park access</b>	Heading	Not Applicable		
FuncReq-21	The car park access system allows vehicles to enter and leave the car park.	Context	Not Applicable		
FuncReq-5	The system shall admit authorised vehicles	Requirement	Agreed	High	Test
FuncReq-6	The system shall admit unexpected vehicles when there are unutilised spaces	Requirement	Agreed	High	Simulation
FuncReq-7	The system shall allow vehicles to leave when payment has been received	Requirement	Agreed	High	Simulation
FuncReq-8	The system shall prevent vehicles from leaving when payment has not been received	Requirement	Proposed	Medium	Test
FuncReq-25	The system shall inform the guest that he is at the wrong parking lot when there is insufficient availability at the current lot	Requirement	Supp_Assess	Medium	Demonstration
FuncReq-26	The system shall provide directions to the alternate parking lot when entry is refused	Requirement	Stake_Assess	Low	Demonstration
FuncReq-2	<b>3.2 Booking system</b>	Heading	Not Applicable		
FuncReq-22	The booking system allows guests and visitors to book spaces in the car park and maintains a log of utilisation.	Context	Not Applicable		
FuncReq-11	The system shall allocate parking spaces to guests	Requirement	Agreed	High	Test
FuncReq-12	The system shall provide information on availability of parking spaces	Requirement	Unconfirmed	Medium	Test
FuncReq-13	The system shall provide information on parking prices	Requirement	Proposed	Medium	Test
FuncReq-18	The system shall alert registered visitors to the availability of parking spaces	Requirement	Proposed	Low	Test
FuncReq-19	The system shall maintain a number of spaces as unavailable for visitors	Requirement	Proposed	Medium	Simulation
FuncReq-20	The system shall allow the hotel to set the number of reserved spaces for each day	Requirement	Proposed	Medium	Simulation
FuncReq-24	The system shall dynamically reallocate spaces when a guest arrives at the wrong parking lot	Requirement	Agreed	Low	Simulation
FuncReq-3	<b>3.3 Payment</b>	Heading	Not Applicable		
FuncReq-23	The payment system allows guests and visitors to pay for parking by a variety of methods	Context	Not Applicable		
FuncReq-14	The system shall allow guests to pay as part of their hotel check out	Requirement	Agreed	High	Simulation
FuncReq-15	The system shall allow visitors to pay for parking	Requirement	Proposed	Low	Simulation

At the bottom of the window, it shows "Username: mbrown" and "Exclusive edit mode".

## Roles and Permissions



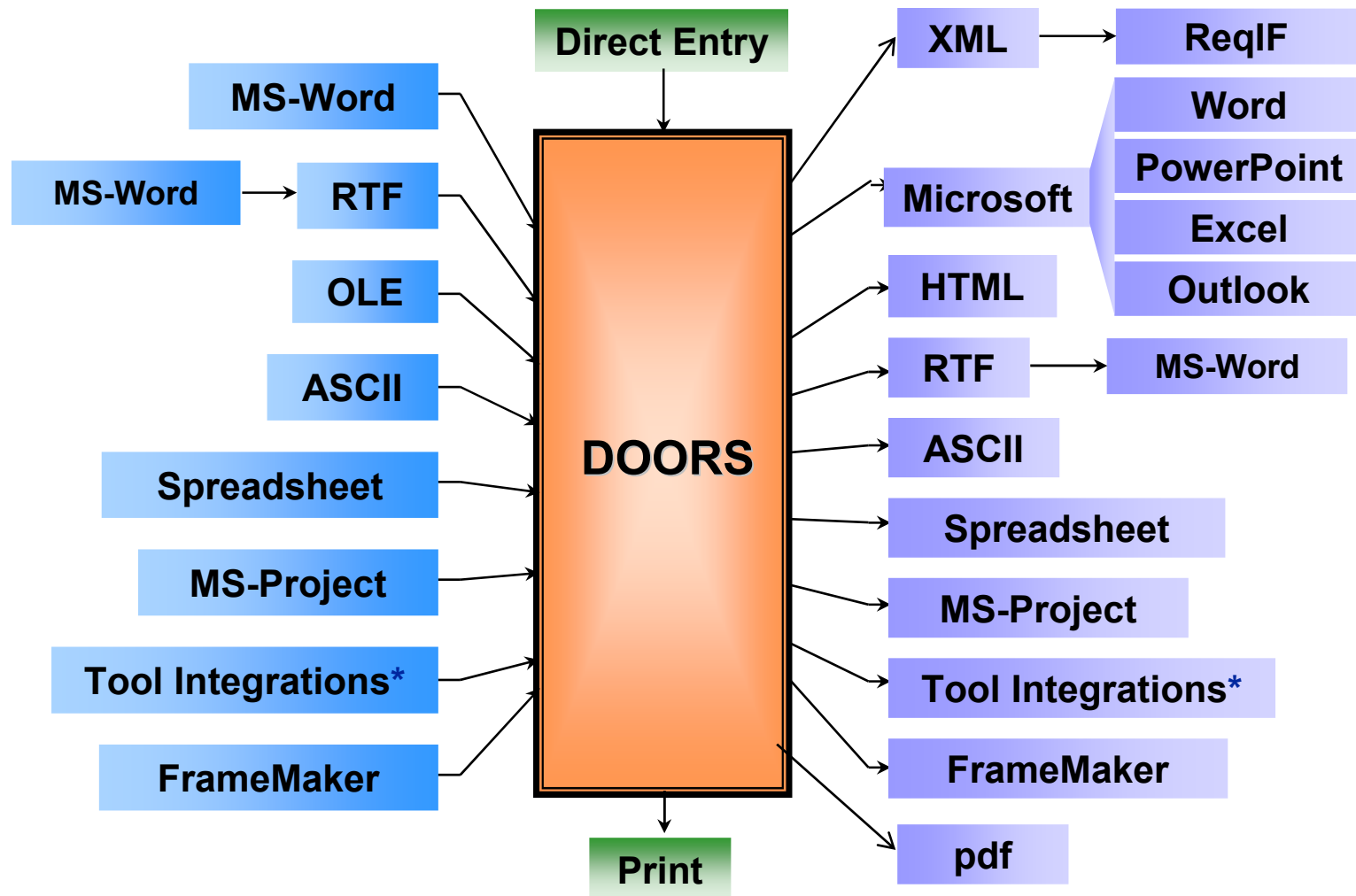


## First Look at DOORS Data:

- Requirement object – Attributes - Modules – Views



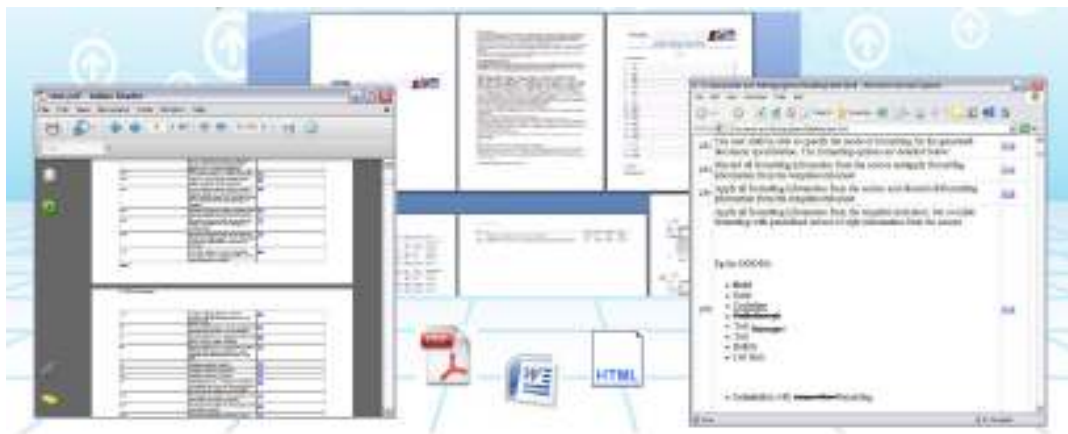
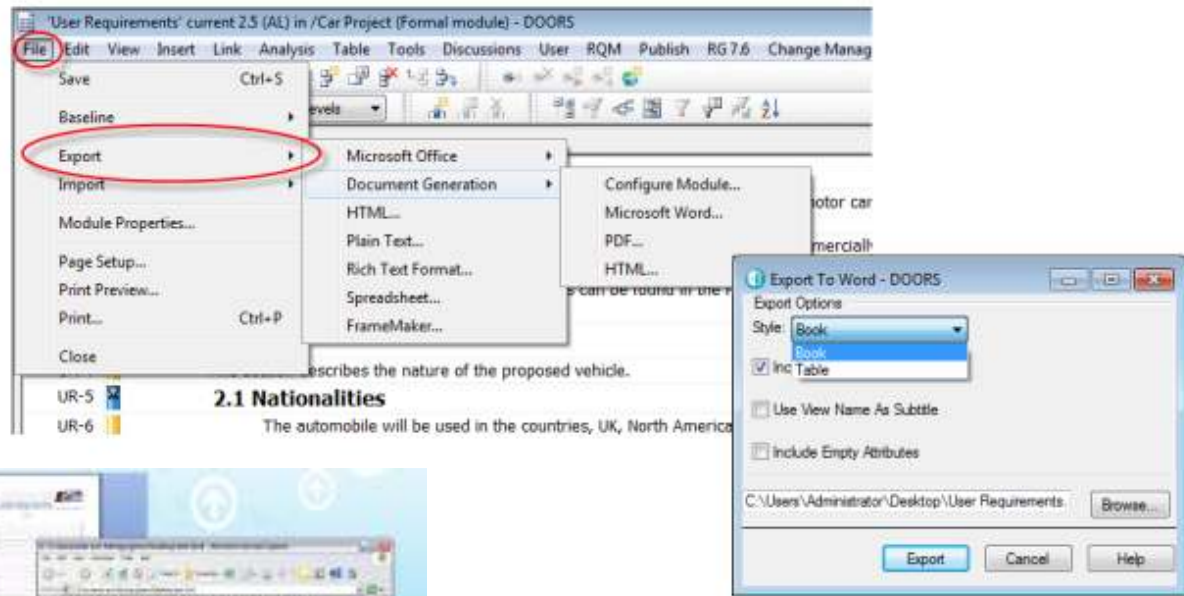
## Import All Your Data & Create Documents



\* See later Integrations slide

## Data Export

- Built in document generation is often crucial to communicate with customers



## Traceability - Encourage Collaboration Across the Lifecycle

User Reqs      Technical Reqs      Design      Test Cases

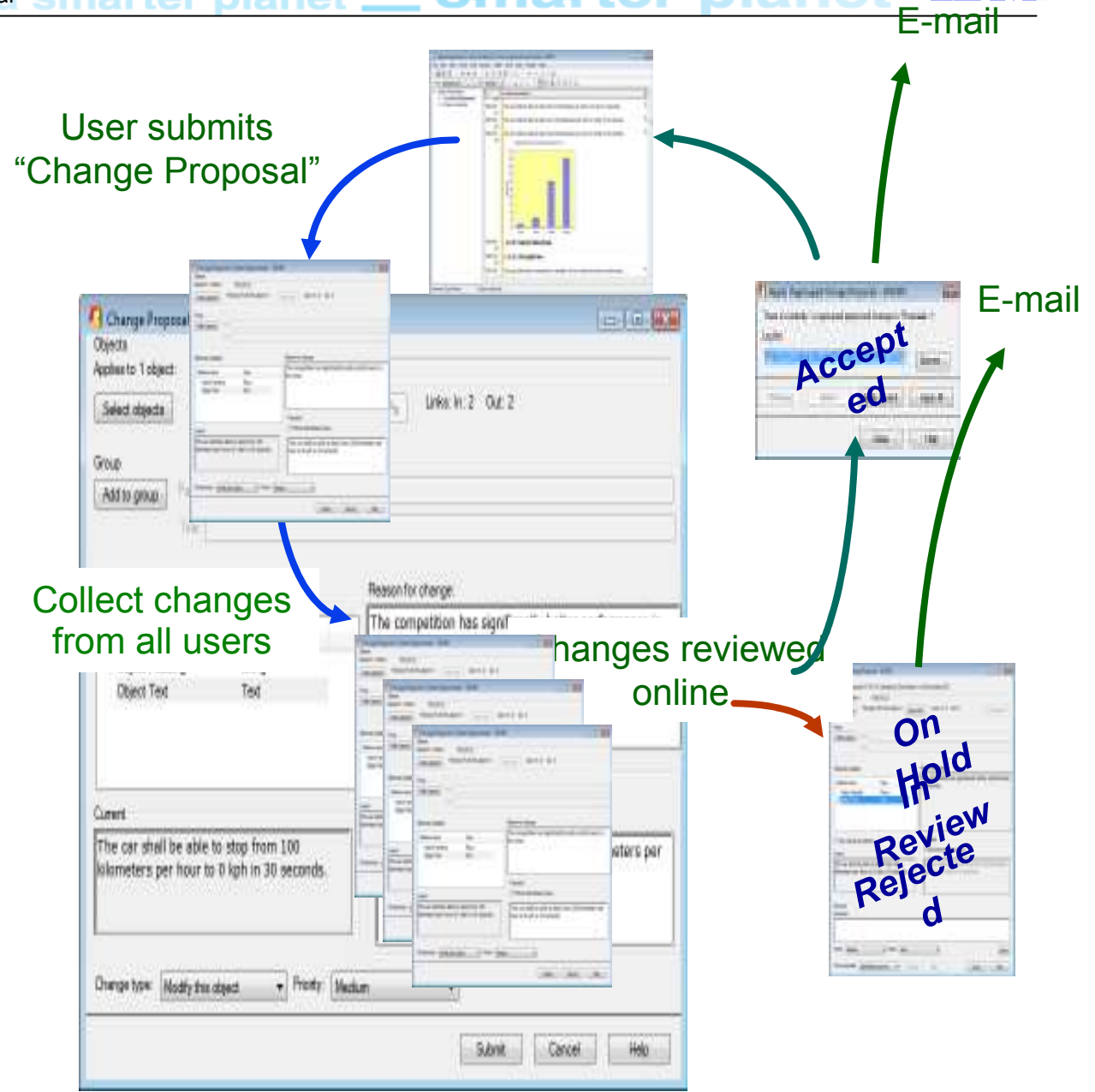
ID	User Requirements	Functional Requirements	Design	Test Plan
TRN-CSR-35	<b>3.1.2.3 Stopping</b>			
TRN-CSR-36	Users shall be able to stop safely.	FR-23 The car shall be able to stop from 10 kilometers per hour to 0 kph in 2 seconds.  FR-24 The car shall be able to stop from 30 kilometers per hour to 0 kph in 6 seconds.	TRN-AD-48 Disc brakes  TRN-AD-48 Disc brakes  TRN-AD-48 Disc brakes	TRN-TP-34 High Speed Braking Test  TRN-TP-35 Low Speed Braking Test  TRN-TP-34 High Speed Braking Test  TRN-TP-35 Low Speed Braking Test  TRN-TP-34 High Speed Braking Test

Username: Dave Mason      Exclusive edit mode

*End-to-end visual validation in a single view*

## Collaboration

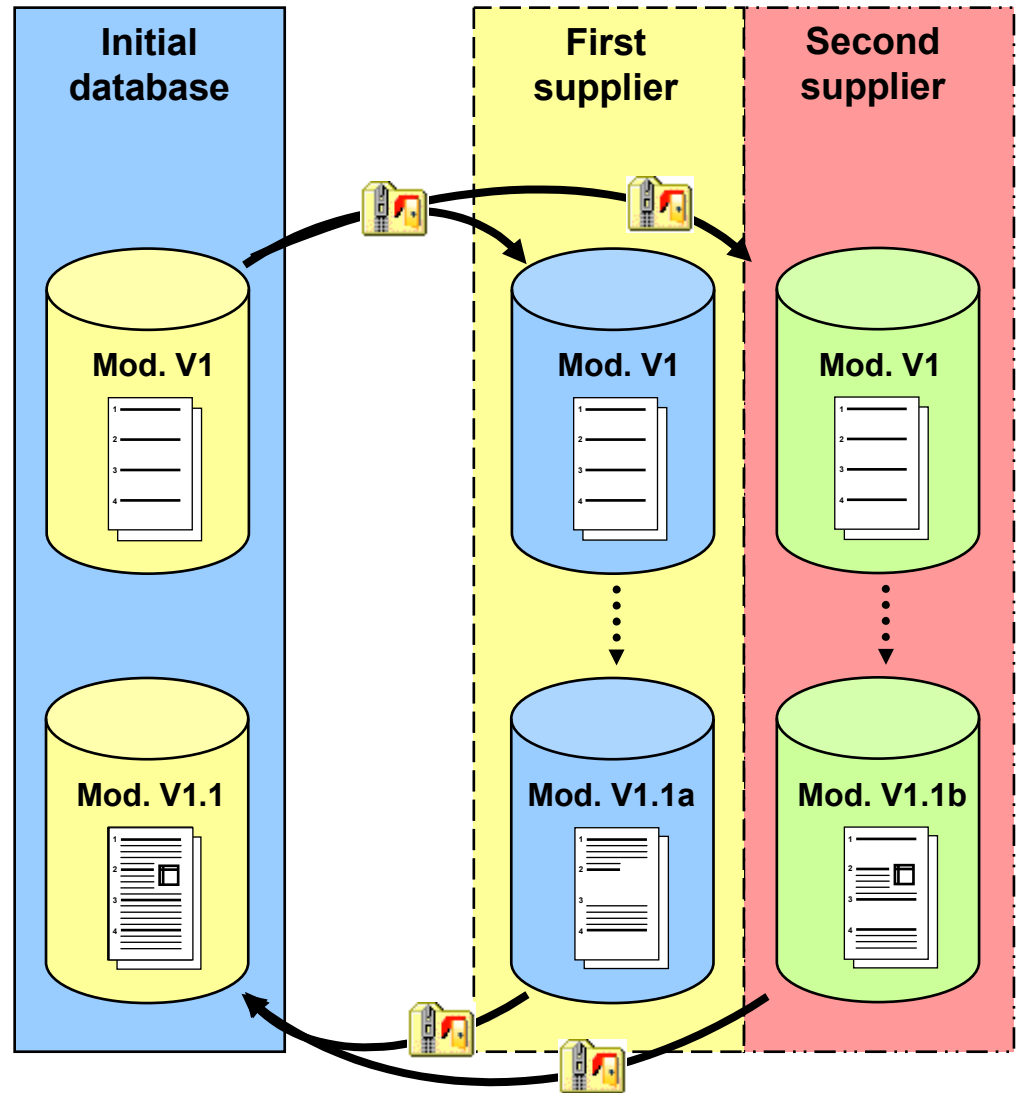
- Collaboration through information discussions or by formal review
- Email notification available for updates



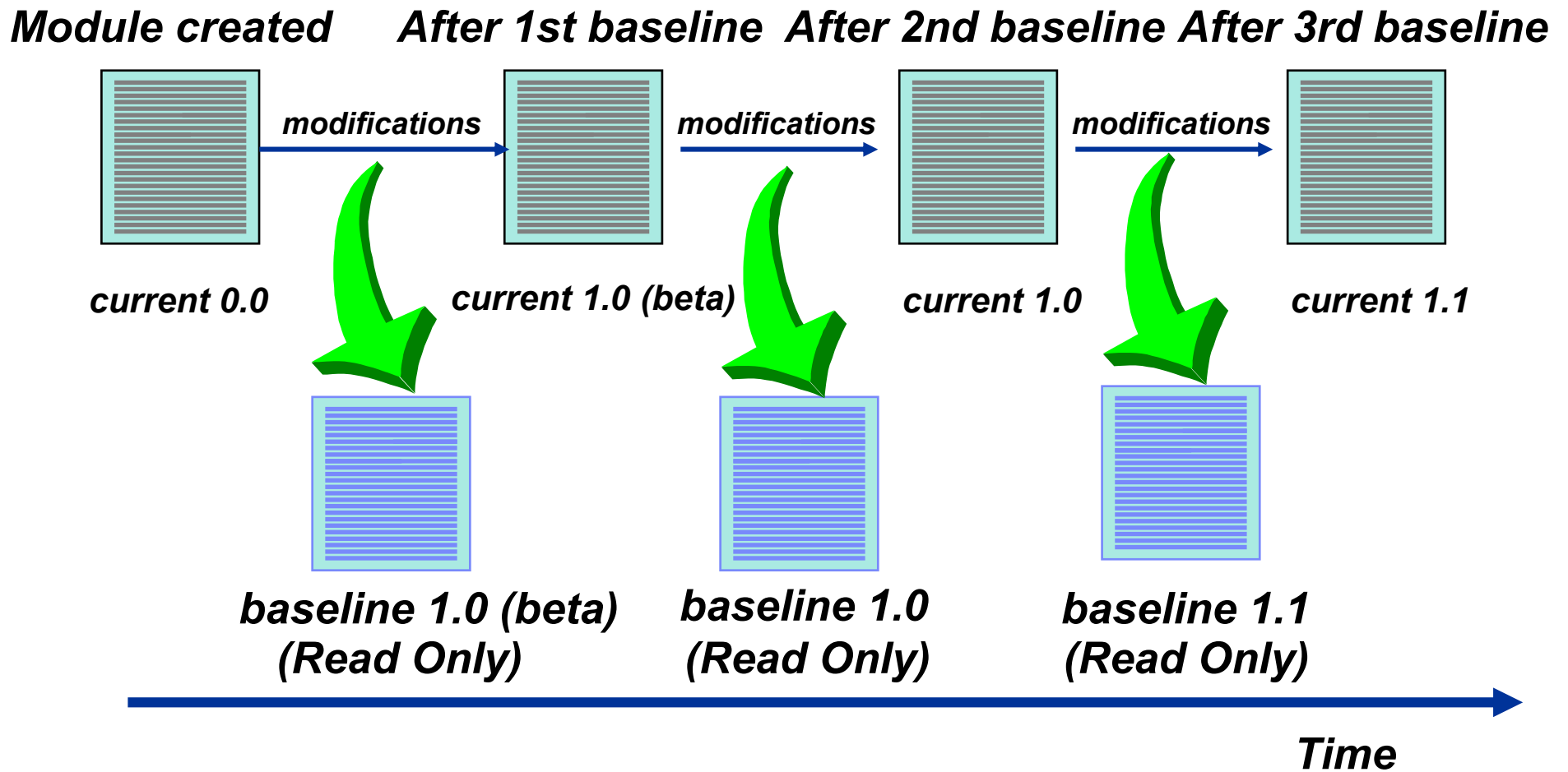
## Supply chain collaboration

### Introducing native support for ReqIF within DOORS

- Fundamental use case to support supplier relationships
- Step 1:
  - Distribute data to supplier database(s)
- Step 2:
  - Concurrent working on data
- Step 3:
  - Managed merge of data
  - With resolution for overlapping data



## A Formal module timeline





## Requirement Review, ReqIF and Simple Versioning

- CPS and Discussion
- Baselining

## Manage Complex Requirements and Get it Right the First Time

*IBM Rational DOORS, a highly secure and scalable solution combined with a robust requirements management process, can help you:*



- Avoid lost requirements and unnecessary development (central repository)


- Improve productivity by automating labour-intensive processes (managing traceability links and changes)

- Reduce requirements errors (traceability, impact analysis)

- Provide comprehensive audit trails (traceability, history, baselines, electronic signature)

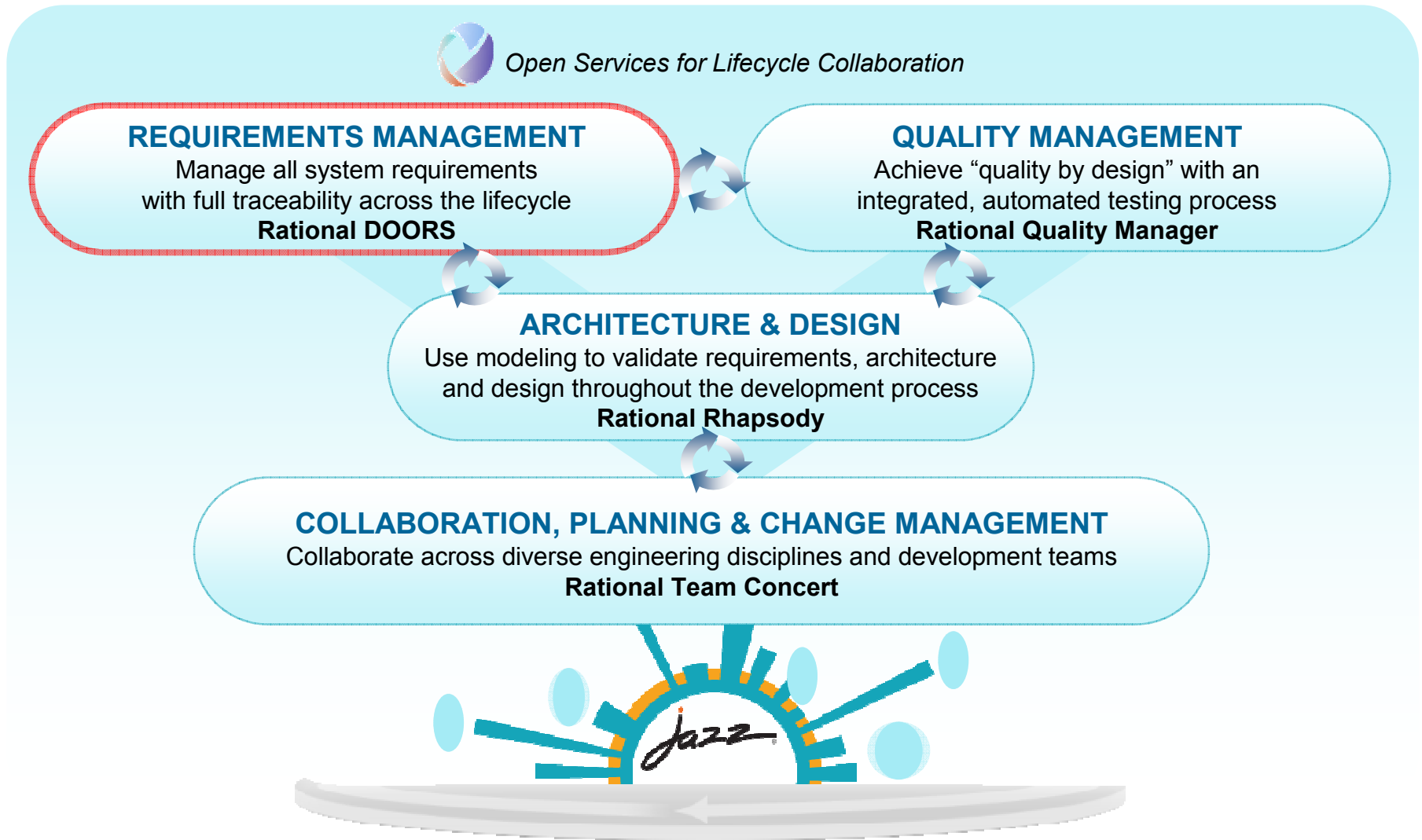


## Agenda

- 
- 1 Why requirements management?
  - 2 Addressing your needs for RM with DOORS
  - 3 Where does RM fit into the wider story
  - 4 Integrations

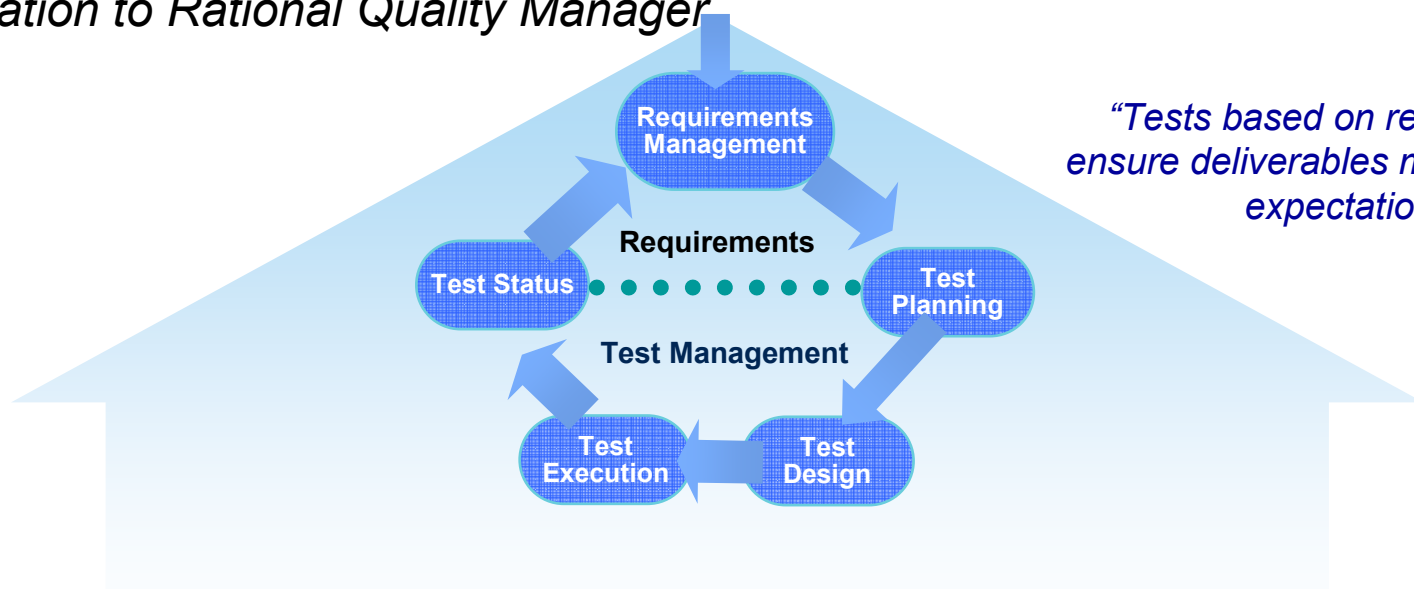
## Rational Solutions for Systems and Software Engineering (SSE)

*Built on a core product set*



# Requirement-driven Testing

## Integration to Rational Quality Manager



*“Tests based on requirements ensure deliverables meet customer expectations”*

### 1. Plan Tests Early

- Plan tests for each requirement as the requirement is written.

### 2. Conduct Tests Early

- Perform tests as early as possible in the development process.

### 3. Relate Tests to Requirements

- Trace tests back to the requirements they are designed to check.

### 4. Relate Defects to Requirements

- Trace defects back to the requirements that they show are not satisfied.

### 5. Measure Progress against Requirements

- Set targets and measure the progress of testing in terms of those requirements that are shown to be satisfied or are not satisfied.

## Requirements-driven development

*Integrations to RTC, ClearQuest and Change*

Requirement / work item traceability

### ▪ 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

The screenshot shows a hover window for a requirement with the following details:

- Object Identifier:** URD-21233
- Module:** User Requirements Document
- Created On:** 08 February 2010
- Created By:** Martin Henderson
- Last Modified On:** 14 March 2010
- Last Modified By:** Martin Henderson

**Attributes**

- Object Heading:** Lorem ipsum et duo paulo nusquam eleifend, his animal inermis intellegbat ex. Sit ex recteque corrumpit, mel ea vocibus posidonium. No pri lobortis tincidunt. Eu duo etiam facilisis, aequo patrioque ea vim.
- Object Text:** <read-denied>
- Object Short Text:** <empty>

**Links**

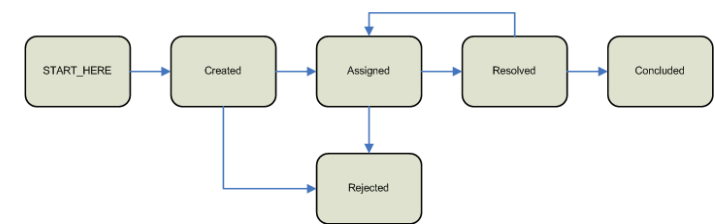
- ▶ DOORS Out-Links (2)
- ◀ DOORS In-Links (7)

## Requirements change management *With RTC, ClearQuest and Change*

### CM system controls changes to requirements Workflow and approvals via CM system

- Requirement/Change Request traceability
  - **Stakeholder requirements submission:** Generation of draft requirements based on enhancements submitted through their Change Management system
  - **Requirements driven development:** Create implementation requests 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

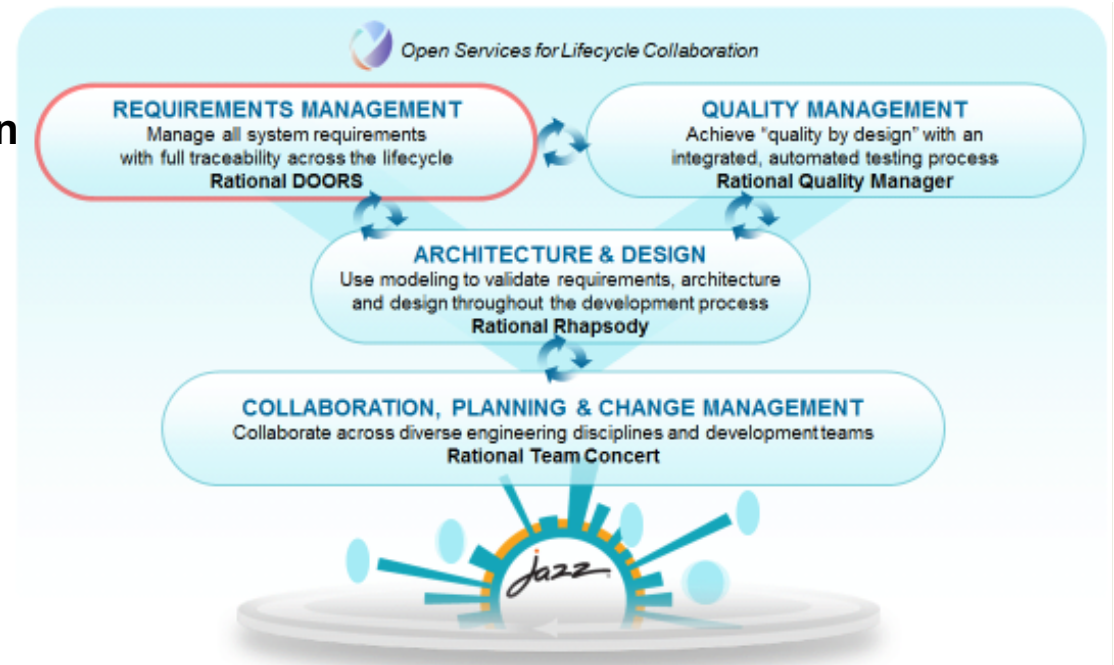
### Process defined and controlled in CM system




## Requirements traceability across product development lifecycle

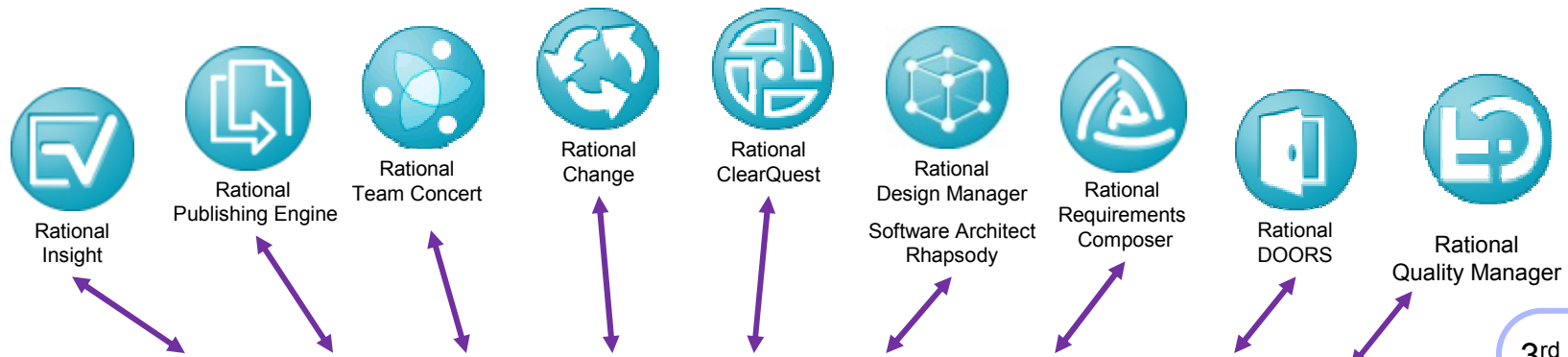
*Better quality, lower risk & higher predictability*

- Get everyone on the same page
  - Collaborate across the **supply chain**
- **Create** what the customer needs
  - Requirements-driven development
- **Deliver** what the customer needs
  - Requirements-driven testing
- Improve **system design optimization**
- Improve **project execution**



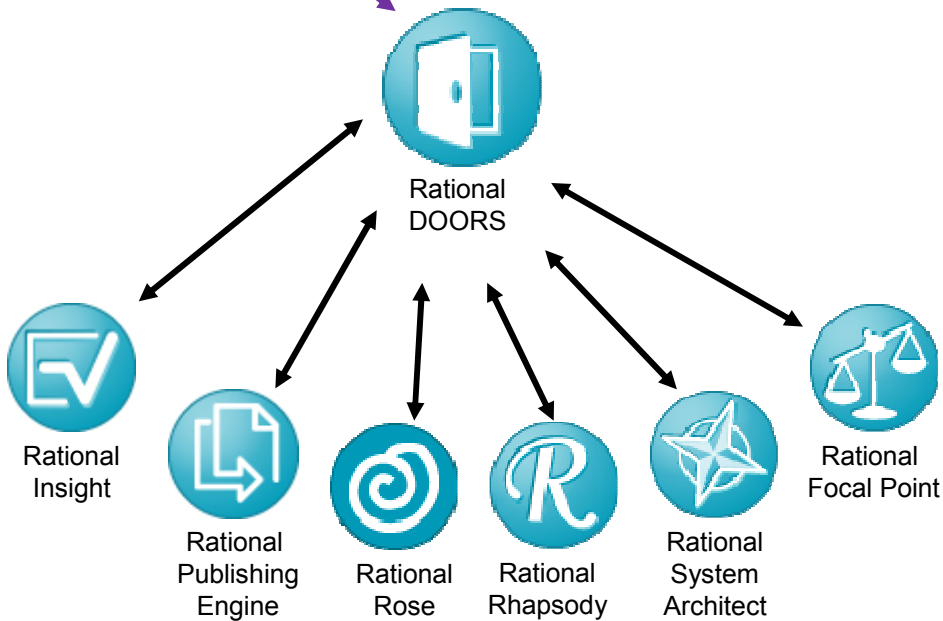
## Agenda

- 
- 1 Why requirements management?
  - 2 Addressing your needs for RM with DOORS
  - 3 Where does RM fit into the wider story
  - 4 **Integrations**



**Open Services for Lifecycle Collaboration**

3<sup>rd</sup> Party tools







# Questions



[www.ibm.com/software/rational](http://www.ibm.com/software/rational)

© Copyright IBM Corporation 2011. 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.