# Cross Product Reporting for Improved Collaborative Lifecycle Management

Fariz Saracevic (fariz@us.ibm.com)
*Offering Strategy and Delivery Leader*

IBM Software
Innovate2011
The Premier Event for Software and Systems Innovation
Software. Everyware.
**August 9-11,** Bangalore | **August 11,** Delhi

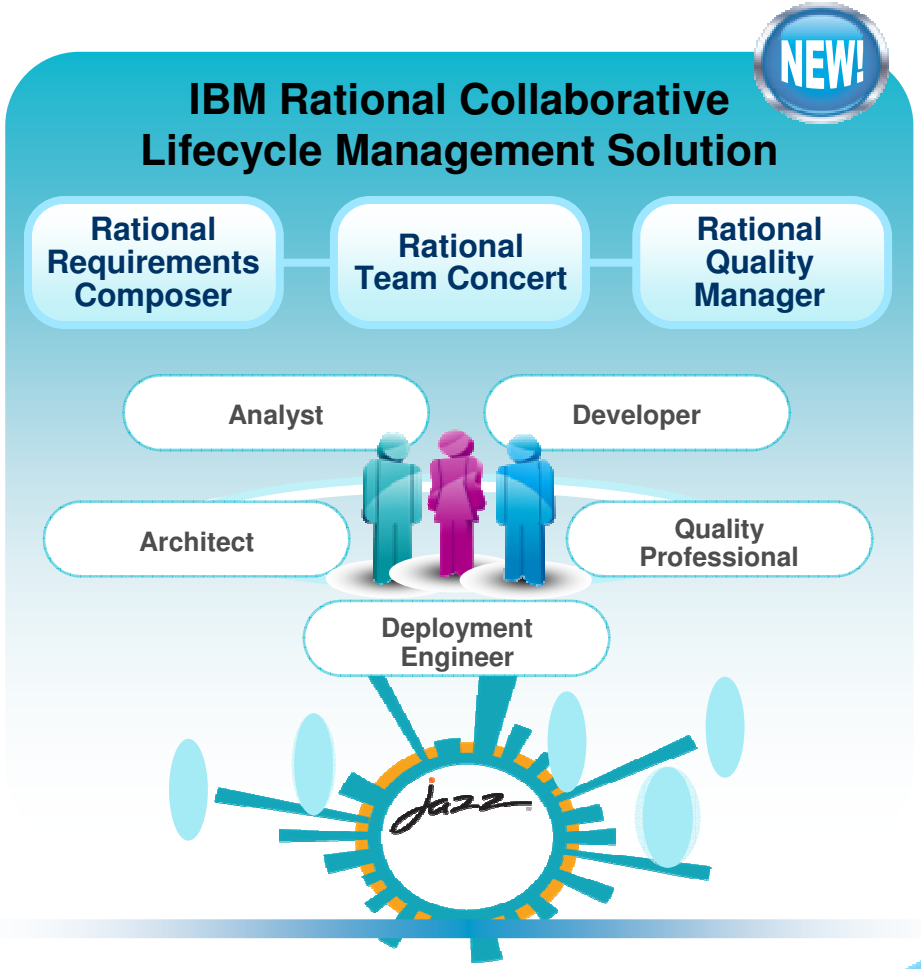
# Topics

- **Overview - Collaborative Lifecycle Management**
- **Reporting Needs**
- **CLM Reporting Architecture**
- **Cognos Overview**
- **The CLM Data Model**
- **Cognos report authoring tooling**
  - Query Studio
  - Report Studio

# IBM Rational Collaborative Lifecycle Management Solution

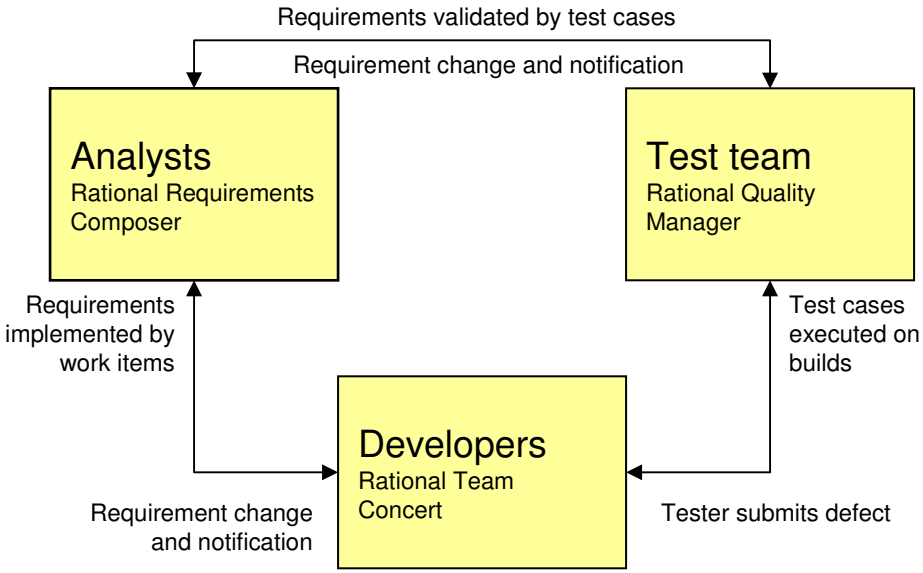
*Improve productivity with an integrated ALM solution*

- **Optimize your team through support of the 5 ALM Imperatives**
  - Real-time planning
  - Lifecycle traceability
  - In-context collaboration
  - Development Intelligence
  - Continuous Improvement
- **Get up and running quickly**
- **Extend as your needs evolve**
- **Support heterogeneous development across multiple platforms and technologies**



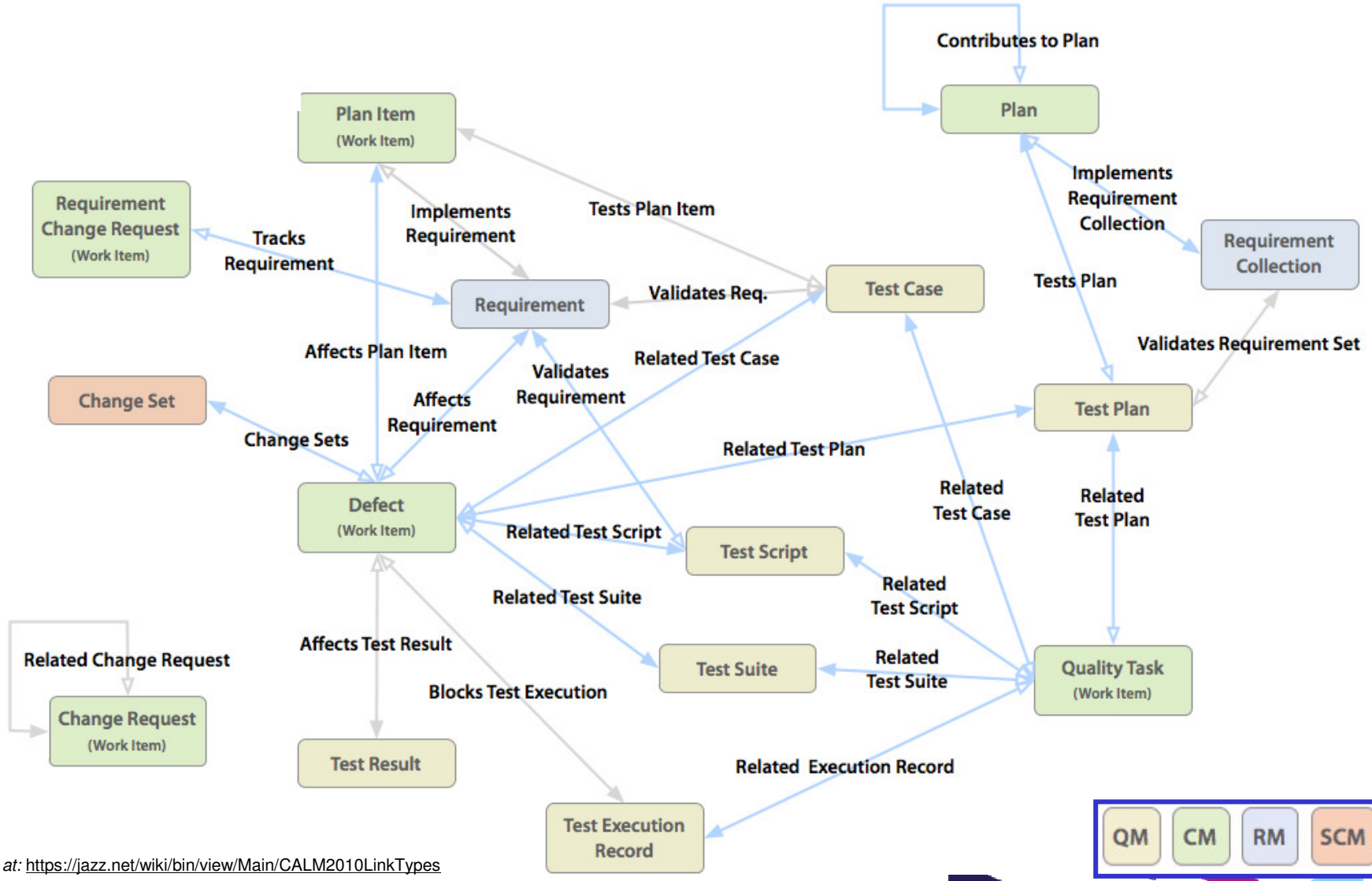
# Collaborative Lifecycle Management

- Collaborative Lifecycle Management (CLM) provides integrations across Jazz-based products to connect the work of analysts with development and test teams
- Cross-product links support traceability, web-like navigation, review, commenting, and status tracking across project repositories
- CLM integrations build on the Jazz™ Foundation to provide a common approach to artifact linking, dashboards, security, and user interface frameworks




Learn more at
http://publib.boulder.ibm.com/infocenter/clmhelp/v3r0m1/index.jsp?topic=/com.ibm.help.common.jazz.calm.doc/topics/c_calm_common.html

# CLM Link Types



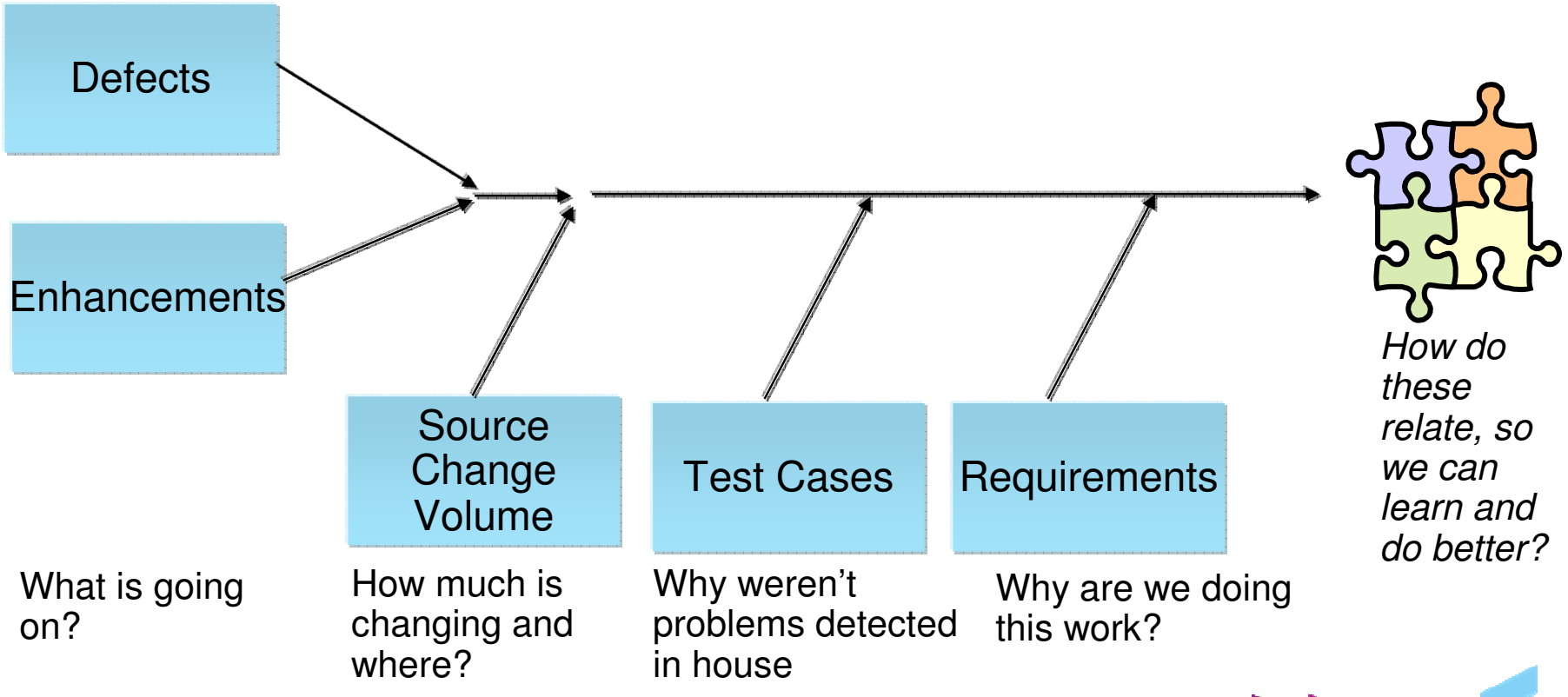

*Learn more at:* https://jazz.net/wiki/bin/view/Main/CALM2010LinkTypes

# Reporting Needs

- Strong reporting capabilities aid successful management
  - All relevant data must be reportable
  - Custom report definition must be supported
  - Report delivery must enable timely action
- Development organizations use multiple tools
  - Tools may be acquired from multiple sources
    - Rational, other vendors, in-house
  - Tools may be geographically distributed
  - Reports must span these tools
  - Managing traceability is a key aspect of Collaborative Lifecycle Management (CLM)
- Standardization of reporting technology reduces the total cost of ownership
  - Reduced server administration costs
  - Reduced user training costs

# Goals: Address Issues Like This

- *Why are we getting so many defects from customers in one specific area? What is going on? Why weren't defects caught in-house? How should our process change so we can do better?*



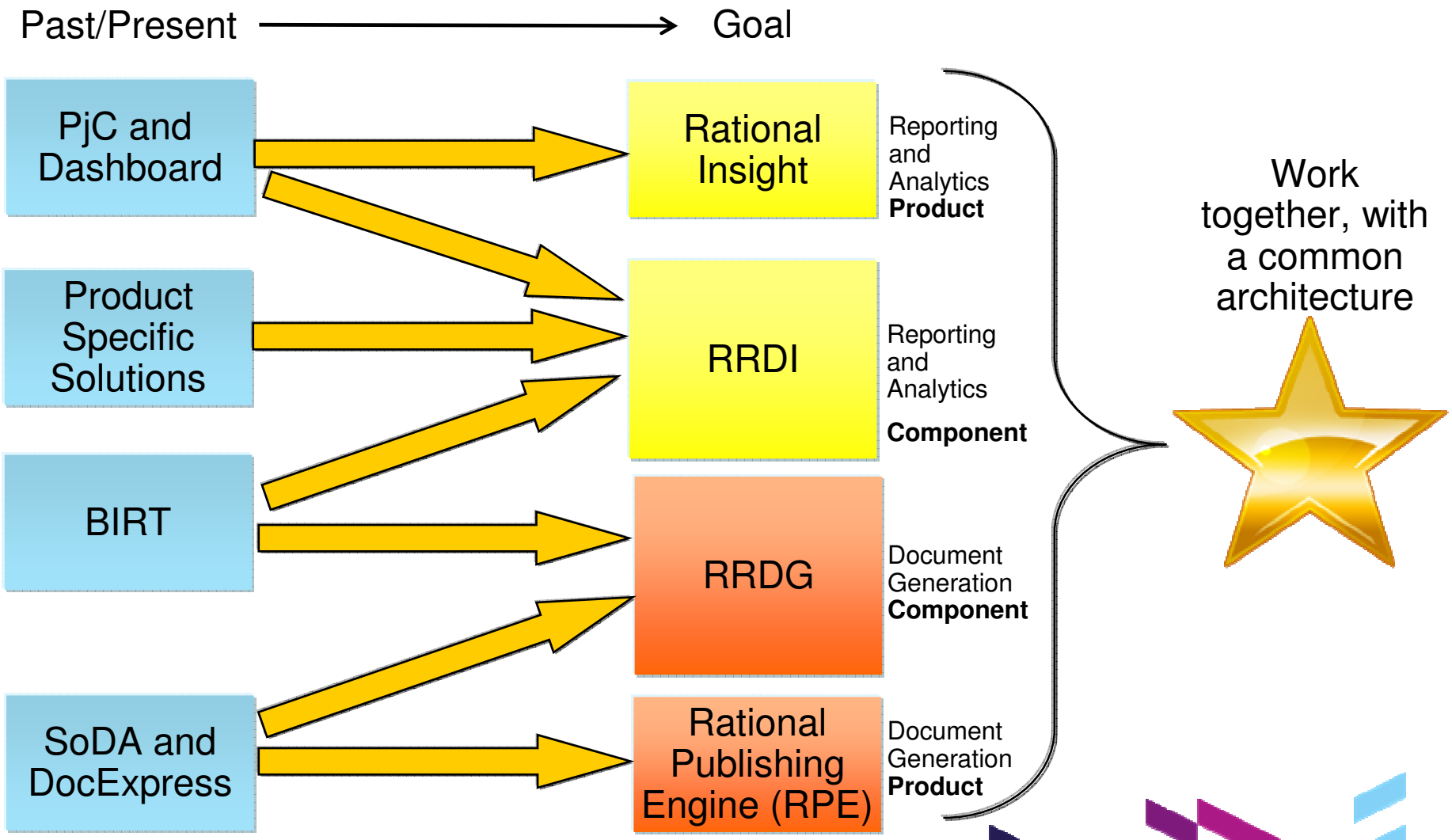
# Some example CLM traceability reporting needs

- **As a development manager,**
  - I need to know if all planned features have corresponding requirements.
    I'll investigate any features that don't implement any requirement.
- **As a quality manager,**
  - I need to know if all requirements planned for this release have corresponding test cases.
    I'll add tasks to develop test cases for requirements that don't have any.
- **As a tester,**
  - I need to know which failed test execution results should be resolved by the latest build.
    I'll retry the test case to verify the fix.
- **As a project manager,**
  - I need to know which requirements have been verified by successful test execution results.
    I'll use this information to assess progress against the cost and schedule.

# Some Reporting Terminology

- *Reporting* is a general term that has several related aspects
- *Query* refers to retrieving information that satisfies certain criteria and presenting it in a simple format
  - A query builder is used to create the retrieval criteria
  - The result is normally displayed as a simple list or table
  - e.g. Rational Team Concert has an integrated Work Item query builder
- *Development Intelligence* refers to the application of Business Intelligence techniques to development data
  - Report authoring tools are used to define the output presentation
  - The result often contains charts, graphs, tables, cross-tabs, gauges, dashboards, etc.
  - e.g. IBM Cognos BI and Rational Insight support development intelligence
- *Document Generation* refers to the automatic creation of documents from development data based on templates
  - The documents are often deliverables, e.g. a Requirements Specification
  - The output is normally Word or PDF
  - e.g. Rational Publishing Engine supports document generation

# Rational Reporting Evolution



# Rational common reporting component strategy

- Rational is transforming technology from Rational Insight and Rational Publishing Engine into a standard set of common reporting components that can be embedded in any Rational product
  - Rational Reporting for Development Intelligence (RRDI)
    - Includes common data warehouse technology from Rational Insight and IBM Cognos BI
  - Rational Reporting for Document Generation (RRDG)
    - Includes document templating technology from Rational Publishing Engine
- Rational Insight and Rational Publishing Engine will continue as stand-alone products
  - They will be built on RRDI and RRDG
  - They will continue to support more open environments

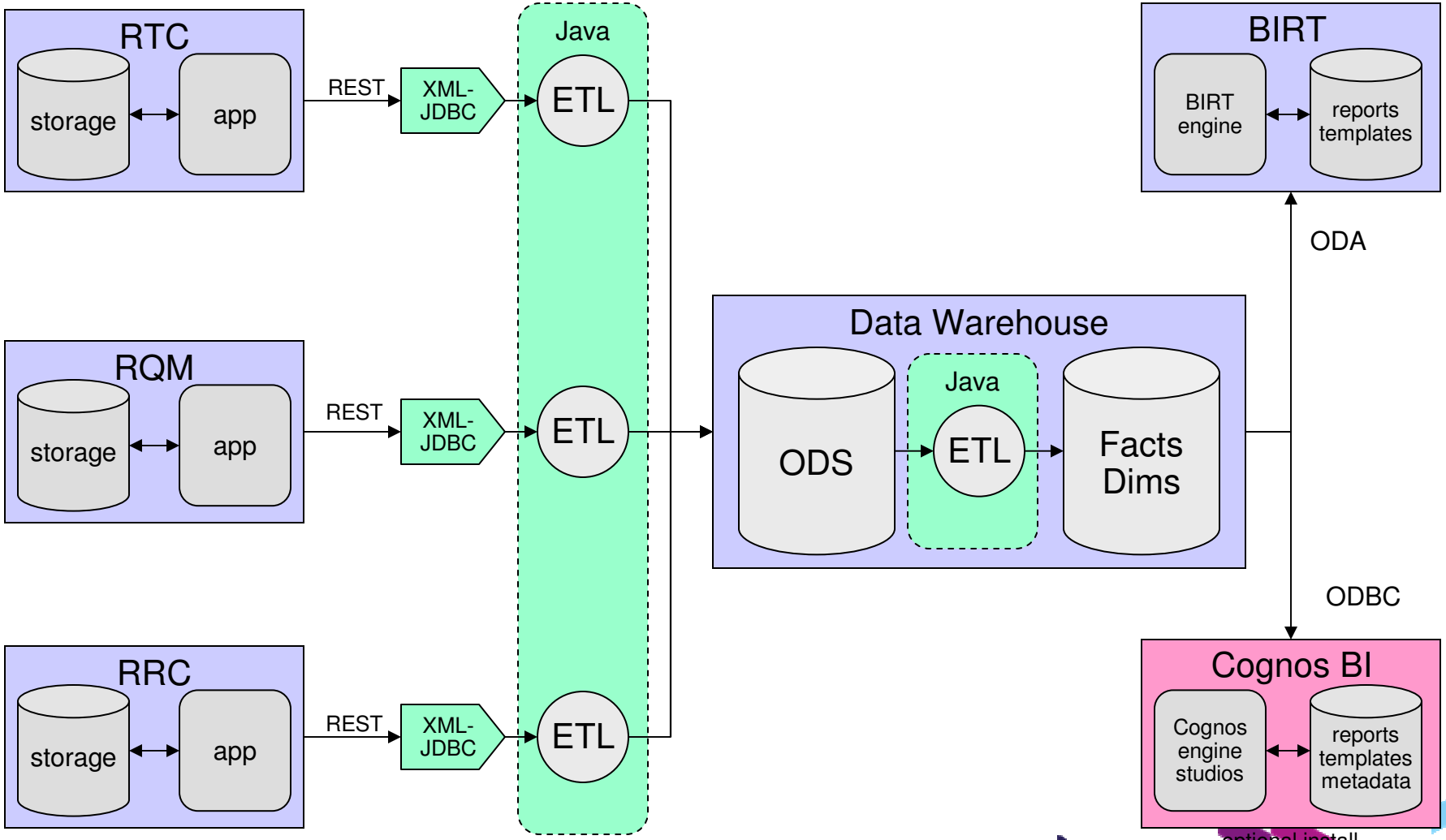
# CLM 2011 Reporting

- Common Data Warehouse (DW)
- Out-of-the-box development intelligence reports (BIRT)
- Out-of-the-box document-style reports (RRDG) for RRC
  - https://jazz.net/wiki/bin/view/Main/RRC3RRDGReports
- (Optional) Custom reporting with Rational Reporting for Development Intelligence (RRDI)
  - Cognos-based authoring
  - Viewing experience integrated into the CLM Web UI
- (Optional) Custom document generation with standalone RPE purchase

# Consumability features of CLM 2011 reporting

- The core installation package has all reporting components except the Cognos BI server
  - Data services and ETLs are integrated into point products
  - Predefined BIRT and RRDG reports are provided out-of-the-box
  - Reporting is configured using the JTS setup wizard
  - Packaged Apache Derby can be used as the common data warehouse for evaluation purposes, or for small teams (BIRT only)
- The optional installation package has the Cognos BI server and tools (RRDI)
  - Predefined Cognos reports provided out-of-the-box
  - Report authoring tools provided for custom reports
  - Must use a supported commercial database (DB2, Oracle, SQL Server)
- CLM 2011 reporting is a closed configuration to simplify administration and support
  - Upgrade to Rational Insight or Rational Publishing Engine for more function and flexibility

# CLM 2011 data warehouse reporting



# CLM 2011 out-of-the box reports

- CLM 2011 includes a large set of out-of-the-box reports
  - Some present data from individual products (RTC, RQM, RRC)
  - Some combine data across products (e.g. traceability reports)
- See Help topic *Using available reports and templates*
  - http://jazz.net/help-dev/clm/topic/com.ibm.rational.rr.usage.doc/topics/t_using_sample_reports_templates.html
- See Jazz Team wiki topic *Out-of-the-box Reports*
  - https://jazz.net/wiki/bin/view/Main/CALMReportingOOTBReports

# Reporting Space

**Development Intelligence**

**Why**:

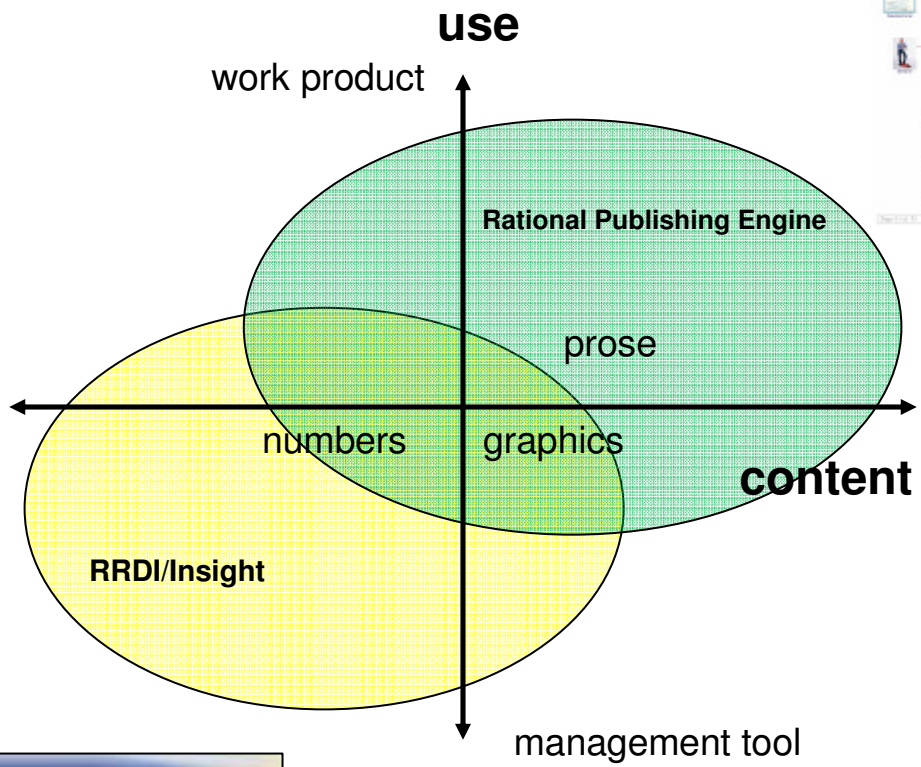
-To view, monitor, and answer questions status and trend.

-Enables deeper analytics to discover root causes and determine corrective actions

**How**:

-Interactive web based reports, charts, graphs often showing aggregated and/or calculated results

-Progressive disclosure via alerts, drill-down, and drill-thru




**Documentation**

**Why**:

-To demonstrate compliance, communicate across contractual boundaries, support offline review of development data, and communicate development data with the extended team

**How**:

-Typically in highly stylized and formatted documents.

# Capability Map in CLM 2011

| | Viewing Reports | | | Authoring New Reports (or customizing existing reports) | | |
|---|---|---|---|---|---|---|
| | **RTC** | **RQM** | **RRC** | **RTC** | **RQM** | **RRC** |
| **Development Intelligence Style Reports (BIRT)** | Within RTC | Within RQM | Not available | Use BIRT Report Designer[4] | Not supported | Not available |
| **Development Intelligence Style Reports (RRDI)[1]** | Within RTC | Within RQM | Within Rational Reporting Server | Within Rational Reporting Server | Within Rational Reporting Server | Within Rational Reporting Server |
| **Document Style Reports (RRDG)[2]** | Use Standalone RPE[3] | Use Standalone RPE[3] | Within RRC | Use Standalone RPE[3] | Use Standalone RPE[3] | Use Standalone RPE[3] |

Capabilities available within CLM application UI

1. Rational Reporting for Development Intelligence (RRDI)
2. Rational Reporting for Document Generation (RRDG)
3. Current limitation: requires RPE (Rational Publishing Engine) License
4. BIRT Report Designer is not supplied with CLM 2011. We recommend the install and use of RRDI for customizing DI reports

**Reporting on data beyond these three CLM products requires separate installation and licensing of Insight and/or RPE**

# The Data Warehouse

- Based on schema from Rational Insight
- Includes artifacts from RM, QM and CCM domains
- Operational Data Store (ODS) stores copies of a subset of operational data from repository
- Star stores pre-computed metrics (facts and dimensions) which include aggregated, historical data drawn from the ODS

# The CLM Data Model

- Built using the Cognos Framework Manager tool (not included with RRDI)
- Provides access to:
  - All metrics and dimension data: star schemas
  - Operational Data Store: detail data in Relational tables
- Model is shown in authoring tools to be used during report construction
- The model is intended to cover Rational's product space
  - Not all fields in the model can be populated by CLM products.
- Provides a "user friendly" view of the data for report authors

# The CLM Data Model: Finding information in RRDI

- Useful Sources of information
  - https://jazz.net/wiki/bin/view/Main/CALMReportingDataModelDetails

Note: fields that are omitted from the FM Model column are those that are irrelevant to the CLM element in question

| UI Element | REST API Resource | REST API Property | DB Table | DB Field | FM Model Table | FM Model Field | Description |
|---|---|---|---|---|---|---|---|
| **Project Area** | projectArea | | PROJECT | | Project | | |
| | projectArea | itemId | PROJECT | EXTERNAL_KEY2 | Project | External Key2 | The UUID representing the item in storage. This is technically an internal detail, and resources should mostly be referred to by their uni |
| | projectArea | uniqueId | PROJECT | EXTERNAL_ID | Project | External ID | An MD5 hash of the URI for this element. |
| | projectArea | name | PROJECT | NAME | Project | Name | The human-readable name of the project area (e.g. "My Project") |
| | projectArea | archived | PROJECT | ISSOFTDELETED | Project | Is Deleted | A boolean indicating whether or not the resource is "archived". Archived resources are typically hidden from the UI and filtered out of que |
| | projectArea | modified | PROJECT | REC_DATETIME | Project | Last Updated | The timestamp of the last modification date of this resource |
| | projectArea | teamMembers | | | | | A list of members of this project |
| | projectArea | teamAreaHierarchy | | | | | A list of records reflecting the team area hierarchy for this project area |
| | projectArea | developmentLines | | | | | A list of development lines for this project area |
| | projectArea | projectDevelopmentLine | | | | | The main development line for this project area |
| | projectArea | href | PROJECT | URL | Project | URL | |
| | | | PROJECT | PROJECT_CLASS_ID | | Classification ID | |
| **Team Area** | teamArea | | TEAM | | Team | | |
| | teamArea | itemId | TEAM | EXTERNAL_KEY2 | Team | External Key2 | The UUID representing the item in storage. This is technically an internal detail, and resources should mostly be referred to by their uni |
| | teamArea | uniqueId | | | | | An MD5 hash of the URI for this element. |
| | teamArea | name | TEAM | NAME | Team | Name | The human-readable name of the team area (e.g. "My Team") |
| | teamArea | archived | TEAM | ISSOFTDELETED | Team | Is Deleted | A boolean indicating whether or not the resource is "archived". Archived resources are typically hidden from the UI and filtered out of que |
| | teamArea | modified | TEAM | REC_DATETIME | Team | Last Updated | The timestamp of the last modification date of this resource |
| | teamArea | projectArea/name | PROJECT | NAME | Team | Project Name | |
| **Contributor** | contributor | | RESOURCE | | Resource | | |
| | contributor | itemId | RESOURCE | EXTERNAL_KEY2 | Resource | External Key2 | The UUID representing the item in storage. This is technically an internal detail, and resources should mostly be referred to by their uni |
| | contributor | uniqueId | RESOURCE | EXTERNAL_ID | Resource | External ID | An MD5 hash of the URI for this element. |
| | contributor | modified | RESOURCE | REC_DATETIME | Resource | Last Updated | The timestamp of the last modification date of this resource |
| | contributor | archived | RESOURCE | ISSOFTDELETED | Resource | Is Deleted | A boolean indicating whether or not the resource is "archived". Archived resources are typically hidden from the UI and filtered out of que |
| | contributor | name | RESOURCE | FULL_NAME | Resource | Full Name | The human-readable name of the contributor (e.g. "James Moody") |
| | contributor | emailAddress | RESOURCE | REFERENCE_ID | | | The email address of the contributor |
| | contributor | userId | RESOURCE | NAME | Resource | Name | The userId of the contributor, unique in this application (e.g. "jmoody") |
| | contributor | href | RESOURCE | URL | Resource | URL | |
| **Iteration** | iteration | | ITERATION | | Iteration | | |
| | iteration | itemId | ITERATION | EXTERNAL_KEY2 | Iteration | External Key2 | The UUID representing the item in storage. This is technically an internal detail, and resources should mostly be referred to by their uni |
| | iteration | uniqueId | | | | | An MD5 hash of the URI for this element. |
| | iteration | modified | ITERATION | REC_DATETIME | Iteration | Last Updated | The timestamp of the last modification date of this resource |
| | iteration | archived | ITERATION | ISSOFTDELETED | Iteration | Is Deleted | A boolean indicating whether or not the resource is "archived". Archived resources are typically hidden from the UI and filtered out of que |
| | iteration | name | ITERATION | NAME | Iteration | Name | The human-readable name of this iteration (e.g. "M1") |
| | iteration | id | ITERATION | EXTERNAL_ID | Iteration | External ID | The identifier of this iteration (e.g. "3.0M1") |
| | iteration | startDate | ITERATION | START_DATE | Iteration | Expected Start Date | The start date of this iteration |
| | iteration | endDate | ITERATION | END_DATE | Iteration | Expected End Date | The end date of this iteration |
| | iteration | parent | | | | | The parent iteration of this iteration, if any |
| | iteration | children | | | | | The immediate child iterations of this iteration, if any |
| | iteration | developmentLines | | | | | The development line in which this iteration appears |
| | iteration | hasDeliverable | | | | | Whether or not this iteration is marked as having deliverables associated with it |
| | iteration | projectAreaItemId | ITERATION | PROJECT_ID | Iteration | Project ID | |
| | iteration | href | ITERATION | URL | Iteration | URL | |
| **Workspace/Stream** | workspace | | FILE_STREAM | | File Stream | | |
| | workspace | itemId | FILE_STREAM | EXTERNAL_KEY2 | File Stream | External Key2 | The UUID representing the item in storage. This is technically an internal detail, and resources should mostly be referred to by their uni |
| | workspace | uniqueId | | | | | An MD5 hash of the URI for this element. |
| | workspace | modified | FILE_STREAM | REC_DATETIME | File Stream | Last Updated | The timestamp of the last modification date of this resource |
| | workspace | archived | FILE_STREAM | ISSOFTDELETED | File Stream | Is Deleted | A boolean indicating whether or not the resource is "archived". Archived resources are typically hidden from the UI and filtered out of que |
| | workspace | name | FILE_STREAM | NAME | File Stream | Name | The name of the workspace or stream |
| | workspace | stream | | | | | True if this is a stream, false if this is a workspace |
| | workspace | description | | | | | A description of the workspace or stream |
| | workspace | collectData | | | | | Whether or not ETL data collection is configured for this stream |

Sheet1 Sheet2 Sheet3

# Reportable REST data services

- Reportable REST Data Services hide details of proprietary point product APIs
- Supports Web scale distribution using HTTP GET requests
- Provides XML representations of the data
- Defines query strings for data filtering to improve performance
  - Field selection
  - Conditional tests
  - “Modified Since” parameter for delta ETLs
- Clients are:
  - Rational Insight XML ODBC driver for live reporting and ETL
  - Rational Publishing Engine for live document generation
- Pain points:
  - Live reporting performance depends on the design of the Data Service
  - In general, only simple live reports perform well

# Common data warehouse

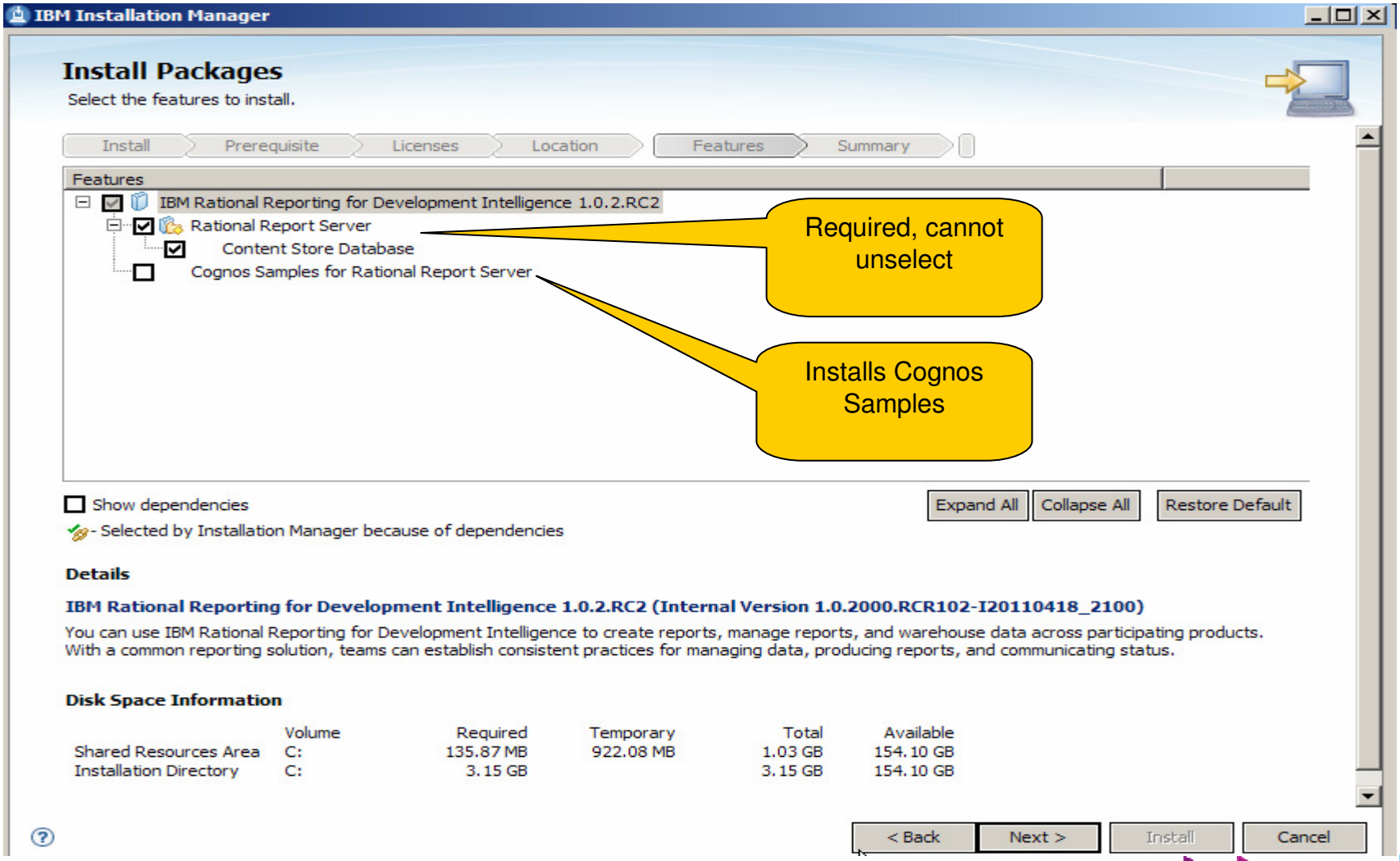
- Data from point products is copied into a common data warehouse using Extract-Transform-Load (ETL) processes
  - Cognos Data Manager (DM)
  - Jazz Foundation Java ETL Framework
- Data warehouse contains:
  - an Operational Data Store (ODS) and
  - a set of Datamarts (aka Fact Tables)
- This enables:
  - complex point product reports,
  - cross-product reports, and
  - analytical reports (OLAP)
- Clients are:
  - IBM Cognos BI
  - Eclipse BIRT
- Pain points:
  - Data warehouse configuration and customization is complex and requires skills
  - ETL execution time may be significant, which limits report currency

# RRDI Installation

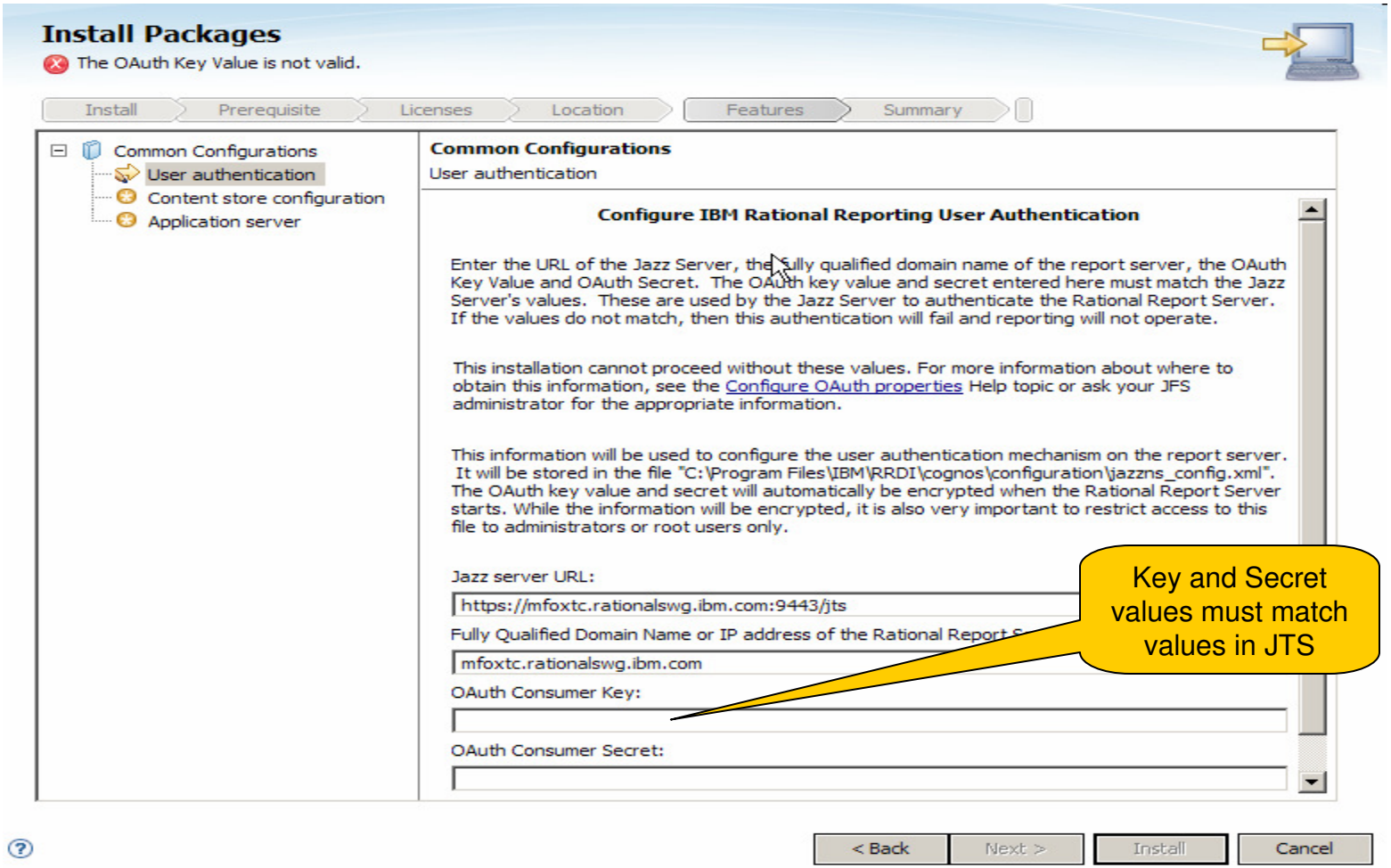
RRDI is a separate install offering from JTS

1. Install JTS
   - Create common Data Warehouse
   - Configure Reporting
   - Specify “OAuth” Key and Secret for server authentication
2. Install RRDI
   - Available as a download from jazz.net
   - No additional licensing or cost
   - Standard Install Manager offering
   - Only option is installing Cognos examples, everything else is installed by default
   - Some required manual steps are required
   - Automated more of the process from the “Common Reporting” shipped with RQM V2.0 and 2.0.1.
3. Perform post install configuration

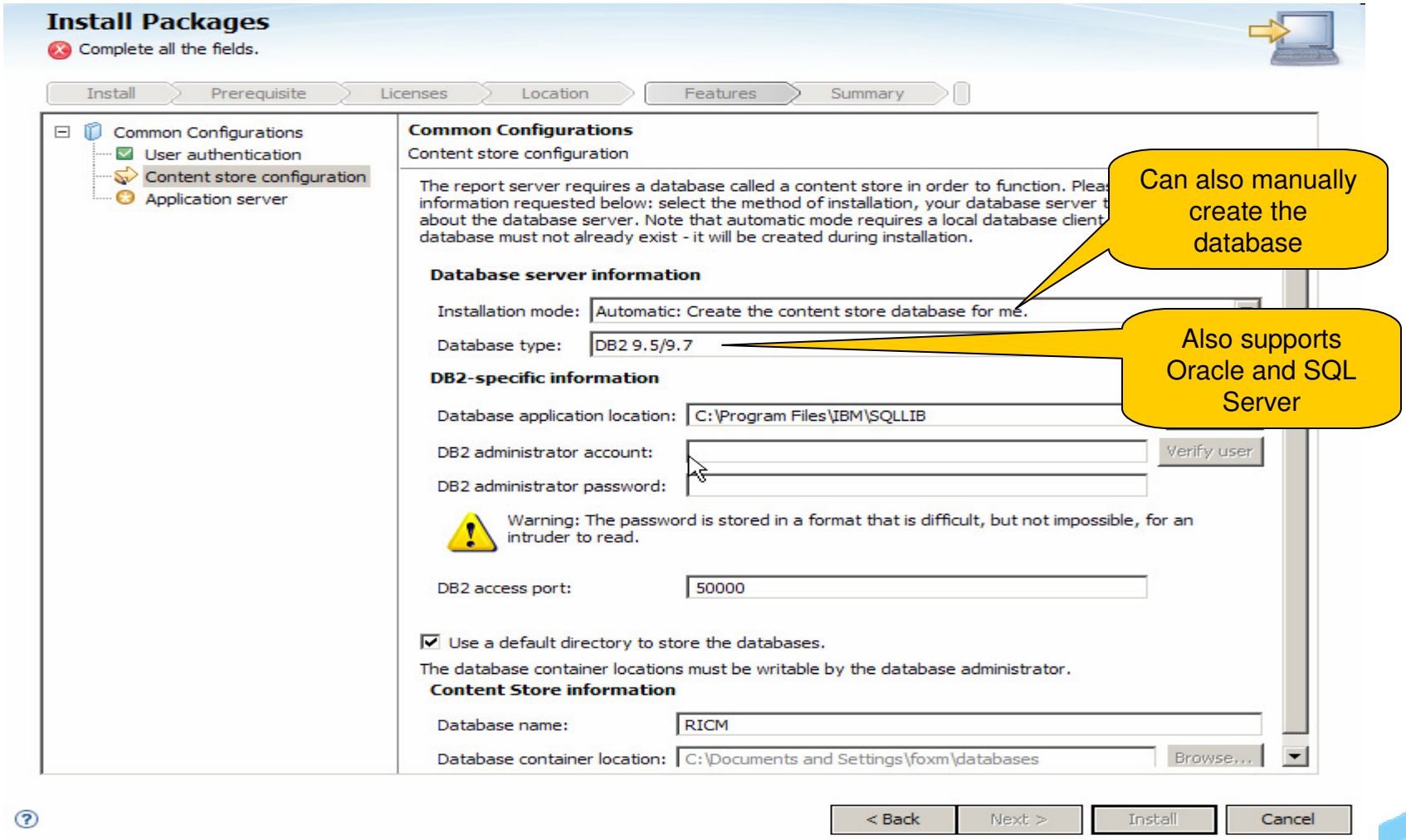
# Select Install Options



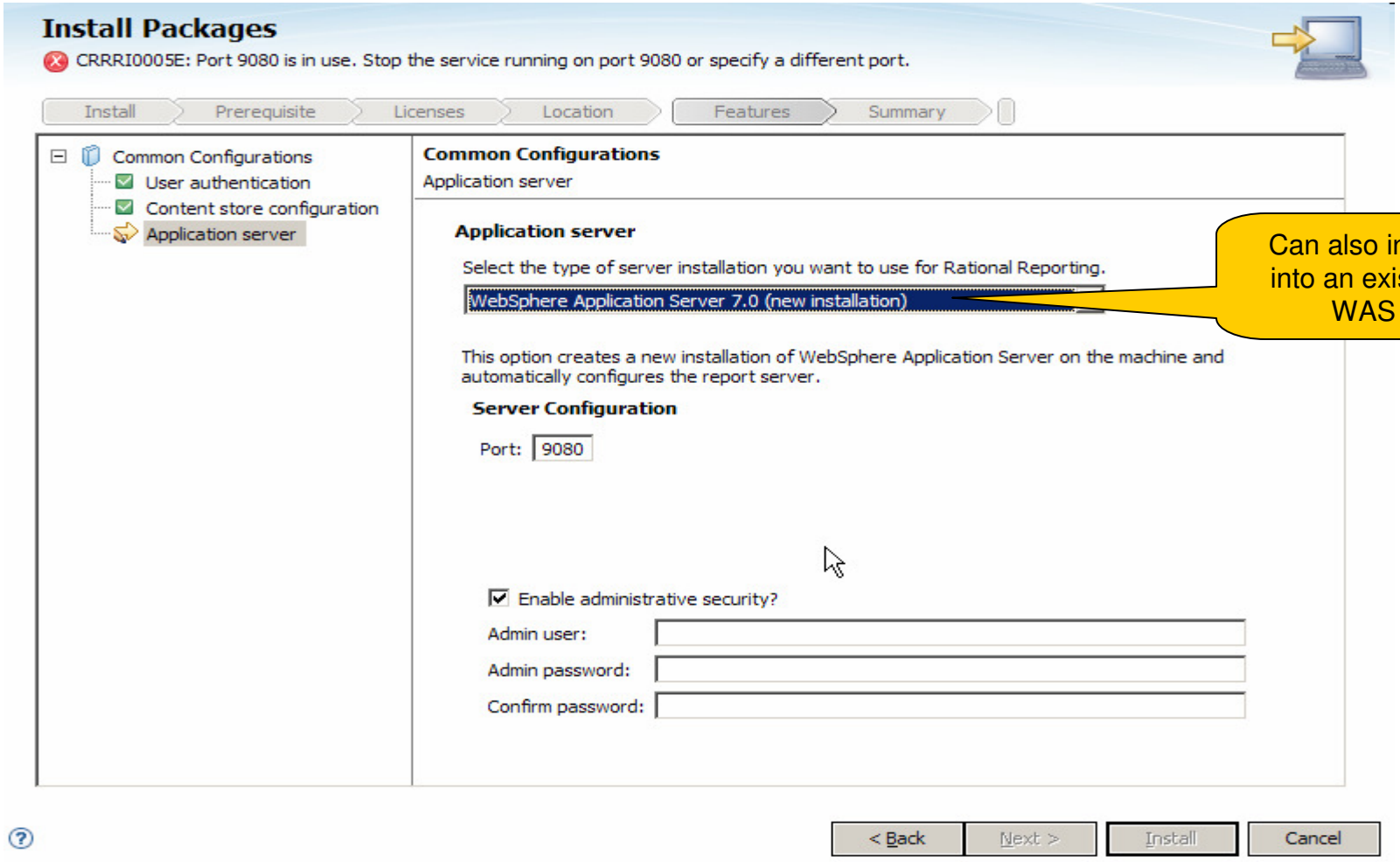
# Configure OAuth Authentication



# Configure Cognos Content Store Database



# Configure WebSphere



# Post Install Configuration

- Some manual steps required on Linux
- Adjust OAuth Key/Secret if necessary
- Collaboration with JTS (security certificates for Internet Explorer 8)
- Download and import Cognos report archives
  - Data Models
  - “Out of the Box” reports for each product
- Set up the Rational Style
- Configure Cognos data source and Signon (if necessary)
- Live Reporting setup for RQM

# Cognos Overview

- Industry leading reporting and analytics product suite acquired by IBM
- Powers *Rational Reporting for Development Intelligence* and *Rational Insight*
- Totally web based
  - Report Server
  - Report Authoring
  - Dashboard creation and display
- Data Abstraction to simplify report authoring
  - Report authors do not need to know the database schema
  - No SQL in reports
  - Isolates reports from database schema changes
- Many advanced reporting features built in
- Many add on products available
  - OLAP Analysis and data cubes
  - Metrics / scorecarding tools
  - Microsoft integration
  - Mobile access
- And much more!

# The CLM Data Model

- Provides a “user friendly” view of the data for report authors
- Built using the Cognos Framework Manager tool (not included with RRDI)
- Provides access to:
  - All metrics and dimension data: star schemas
  - Operational Data Store: detail data in Relational tables
- Model is shown in authoring tools to be used during report construction
- The model is intended to cover Rational’s product space
  - Not all fields in the model can be populated by CLM products.

# The CLM Data Model

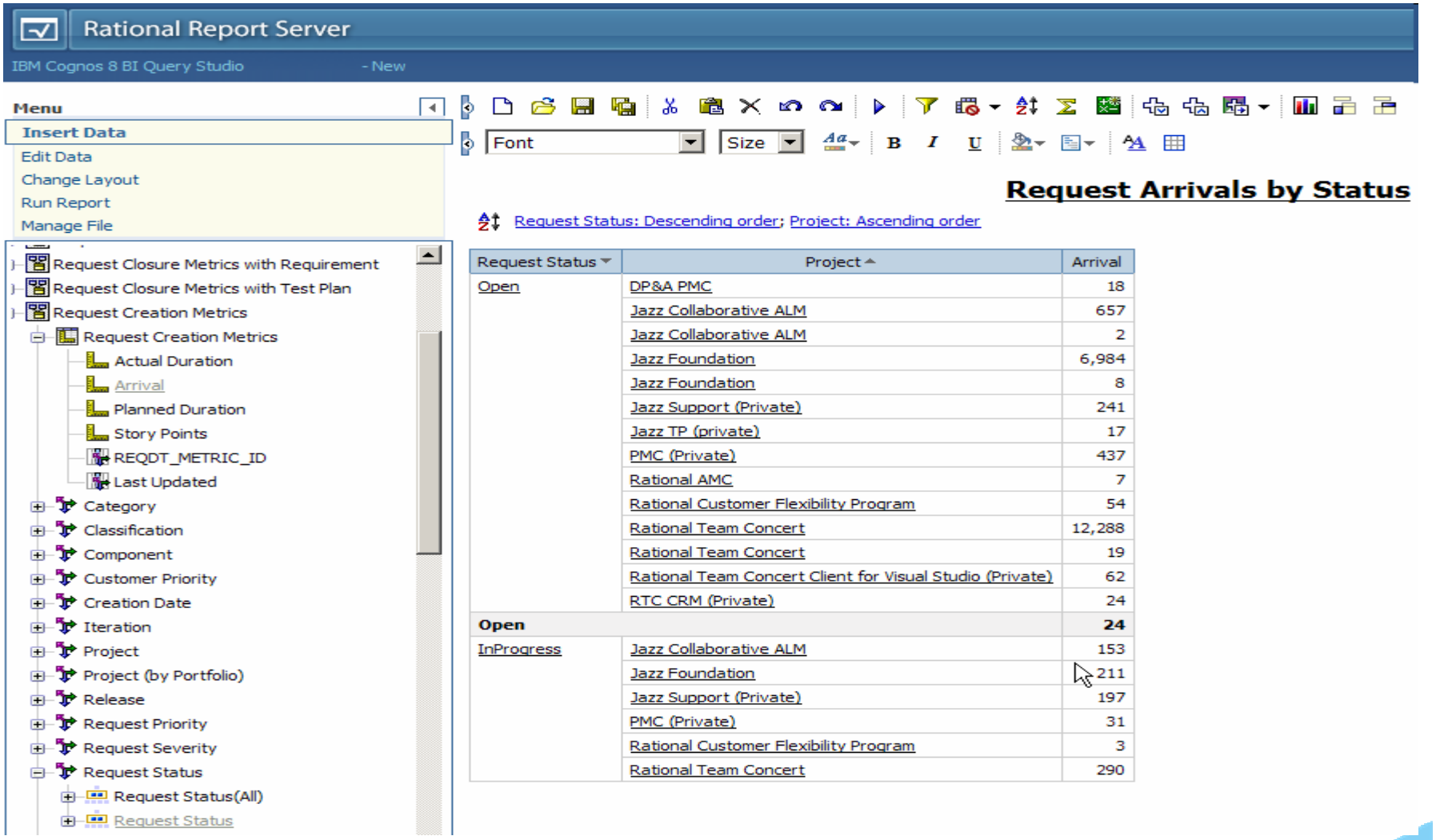
*Areas Covered by the data model*

- Change Management
  - RTC work items
- Configuration Management
  - Activities, Build Metrics
- Project Management
  - Resources, Risk, Time Management
- Quality Management
  - Test cases, jobs, Execution Results
- Requirement Management
  - Requirement Metrics, relationship with Test Cases/Plans
- Operational Data Store
  - Detail data for each area, generally used for tabular reports. Includes cross product linkages

# Cognos Report Authoring Tooling

- Query Studio
  - Used for quick, informal reports
  - Intended to be generally usable with little training
  - Provides “ad hoc” query capability
  - Construct with Drag/Drop
  - Tabular reports: sorting, grouping, filtering, etc
  - Convert to chart: multiple chart types
  - Save report for later use or to open in Report Studio

# Query Studio Example

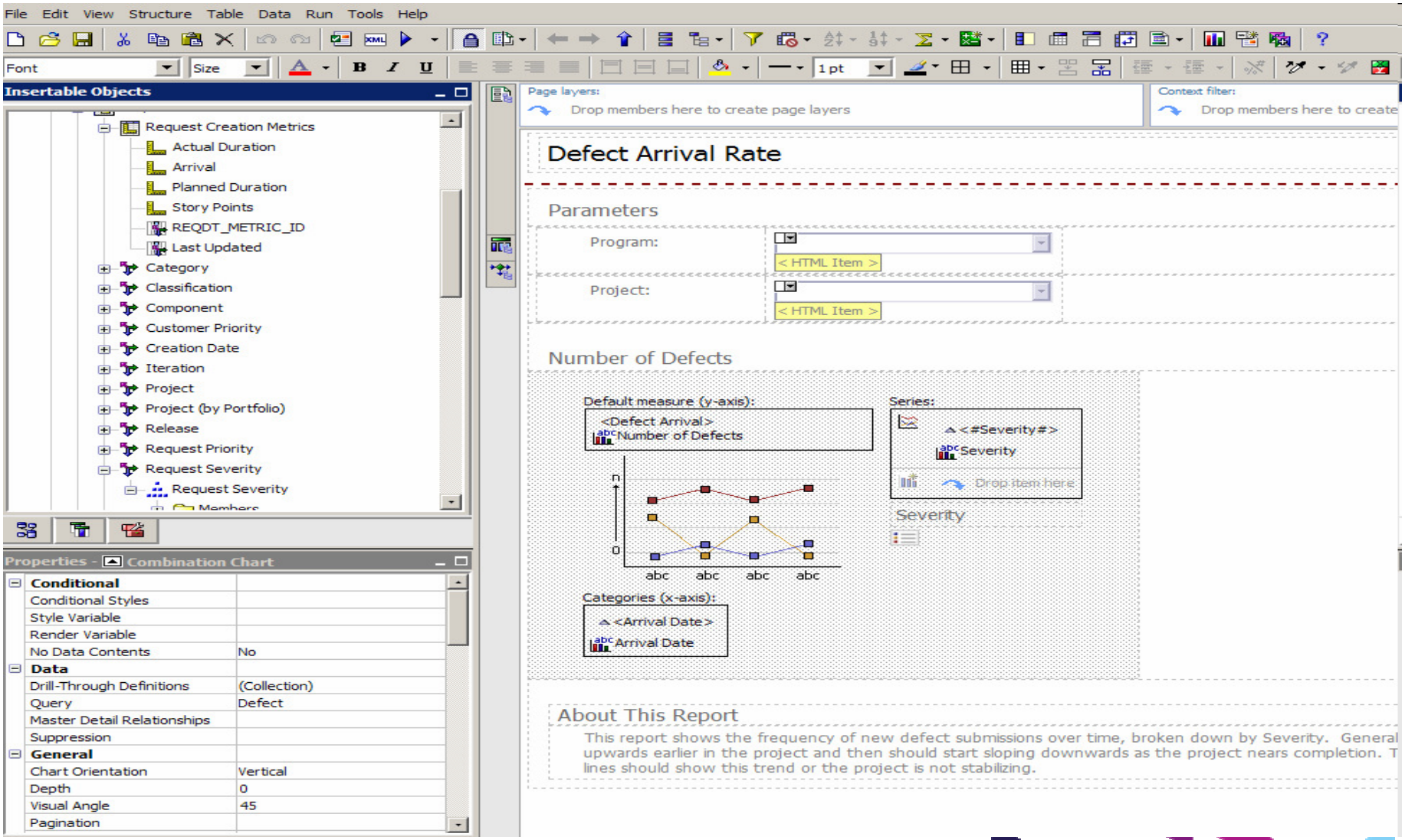



| Request Status | Project | Arrival |
|---|---|---|
| Open | DP&A PMC | 18 |
| | Jazz Collaborative ALM | 657 |
| | Jazz Collaborative ALM | 2 |
| | Jazz Foundation | 6,984 |
| | Jazz Foundation | 8 |
| | Jazz Support (Private) | 241 |
| | Jazz TP (private) | 17 |
| | PMC (Private) | 437 |
| | Rational AMC | 7 |
| | Rational Customer Flexibility Program | 54 |
| | Rational Team Concert | 12,288 |
| | Rational Team Concert | 19 |
| | Rational Team Concert Client for Visual Studio (Private) | 62 |
| | RTC CRM (Private) | 24 |
| **Open** | | **24** |
| InProgress | Jazz Collaborative ALM | 153 |
| | Jazz Foundation | 211 |
| | Jazz Support (Private) | 197 |
| | PMC (Private) | 31 |
| | Rational Customer Flexibility Program | 3 |
| | Rational Team Concert | 290 |

# Report Studio

- Tool used to develop professional reports
- Generally used by Report Authors
- Very extensive web based report authoring capabilities

# Report Studio




**www.ibm.com/software/rational**


**www.ibm.com/software/rational**