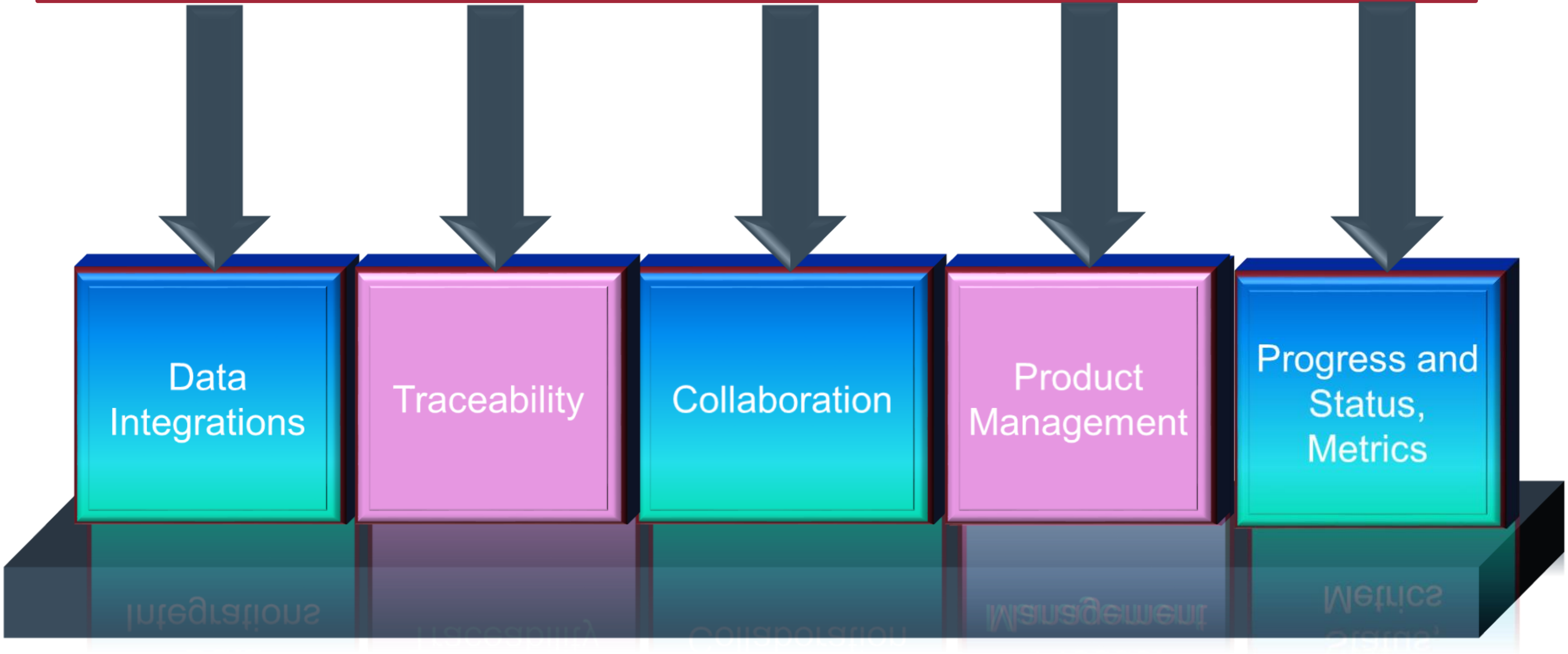
Solving Product Configuration and Traceability Challenges by Leveraging OSLC and the Jazz Platform

Agenda

- Background – What does OSLC and the Jazz platform provide for development activities
  - **http://boeingnews.web.boeing.com/archive/2013/5885.html**
- Motivations for Apache Systems Engineering
- Platform Enablers
- Change Management Process
- Lessons Learned and Next Steps
- *Summary*

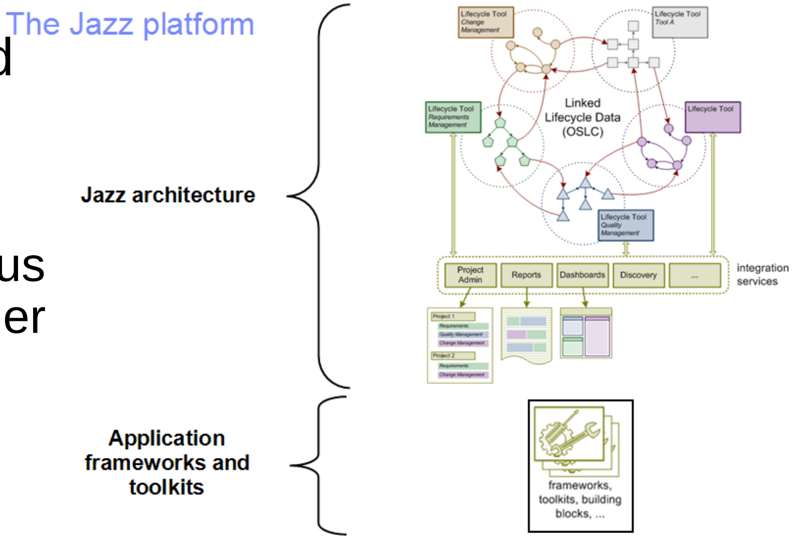
Traditional Boeing Challenges with Systems/Software Development

No unified systems/software development platform leads to significant challenges with the following …



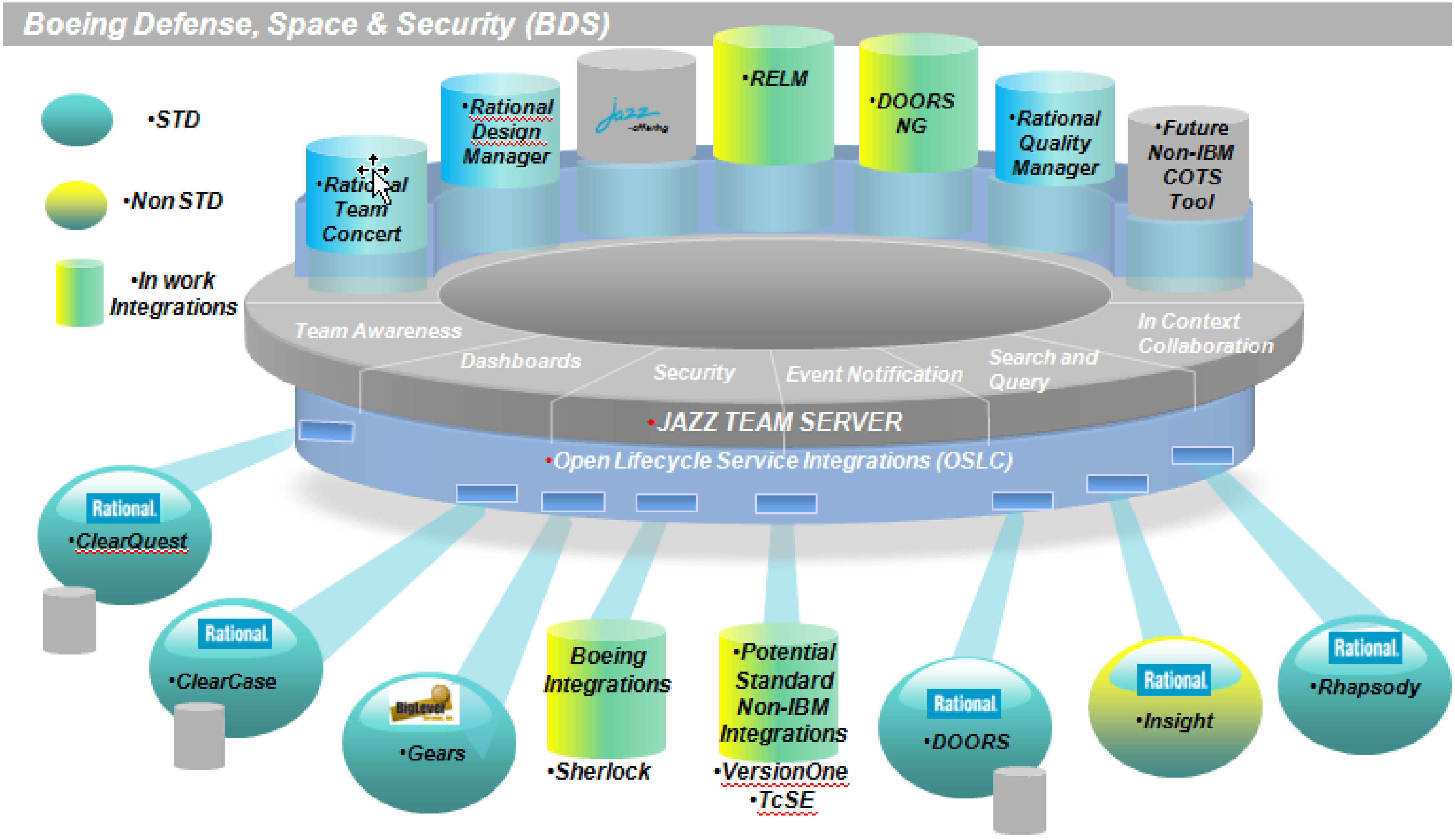
Open Services for Lifecycle Collaboration (OSLC) and Jazz

- **OSLC**
  - a community of software developers and organizations working to standardize the way that software lifecycle tools share data (for example, requirements, defects, test cases, plans, or code) with one another.
  - provides open specifications for service descriptions and cross-tool communication (http://open-services.net/bin/view/Main/WebHome)
    - **Under formal control by OASIS stds group**.
- **Jazz**
  - An open collaboration platform designed to support systems and software tool integrations
    - Integration architecture that allows various tool components to be configured together
  - A set of products that support this platform.

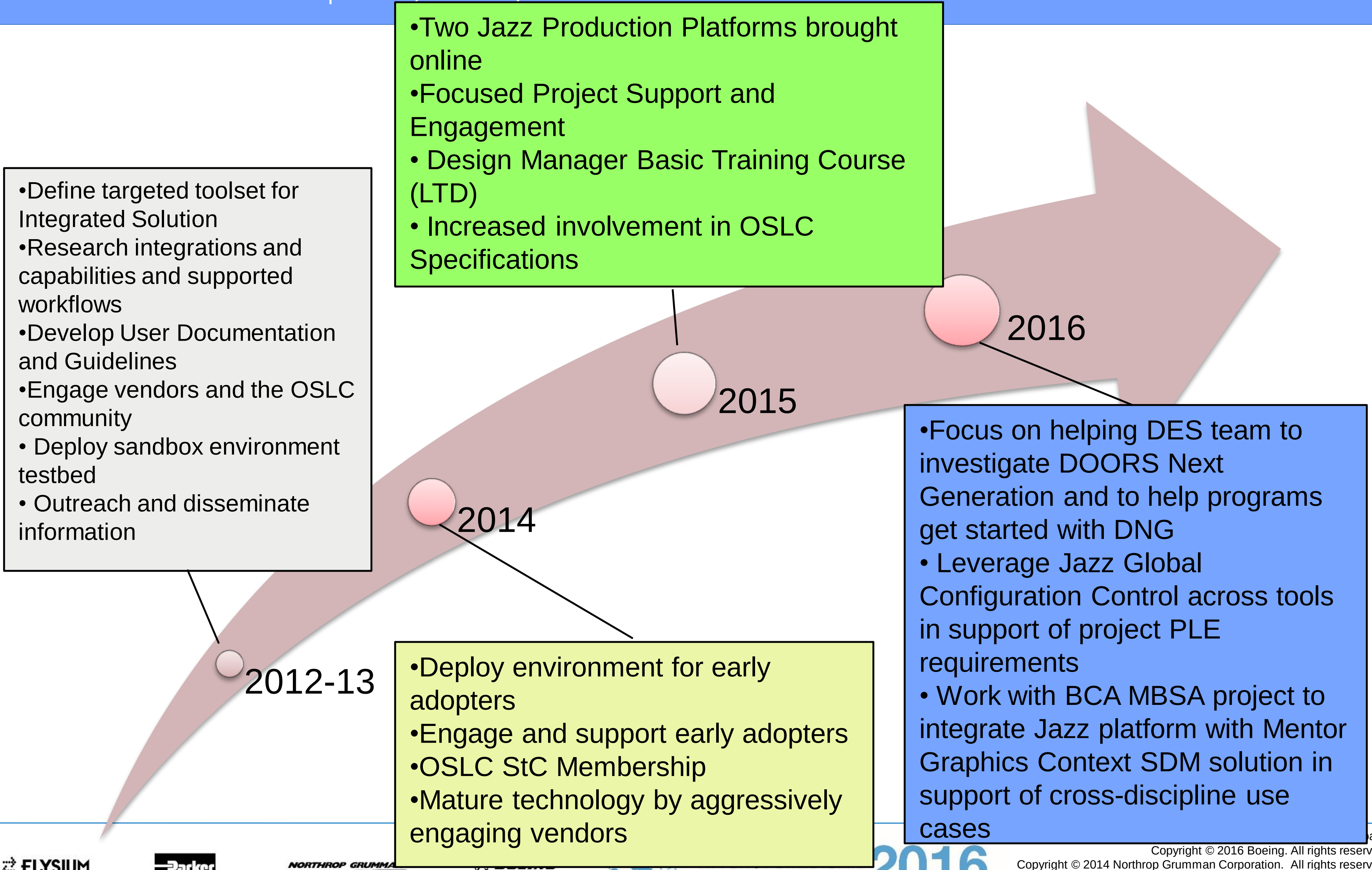


Tool Environment Leveraging the Jazz Framework

Global Product Data Interoperability Summit | **2016**



Jazz Investigation and Deployment Roadmap…



2012-13

- •Define targeted toolset for Integrated Solution
- •Research integrations and capabilities and supported workflows
- •Develop User Documentation and Guidelines
- •Engage vendors and the OSLC community
- • Deploy sandbox environment testbed
- • Outreach and disseminate information

2014

- •Deploy environment for early adopters
- •Engage and support early adopters
- •OSLC StC Membership
- •Mature technology by aggressively engaging vendors

2015

- •Two Jazz Production Platforms brought online
- •Focused Project Support and Engagement
- • Design Manager Basic Training Course (LTD)
- • Increased involvement in OSLC Specifications

2016

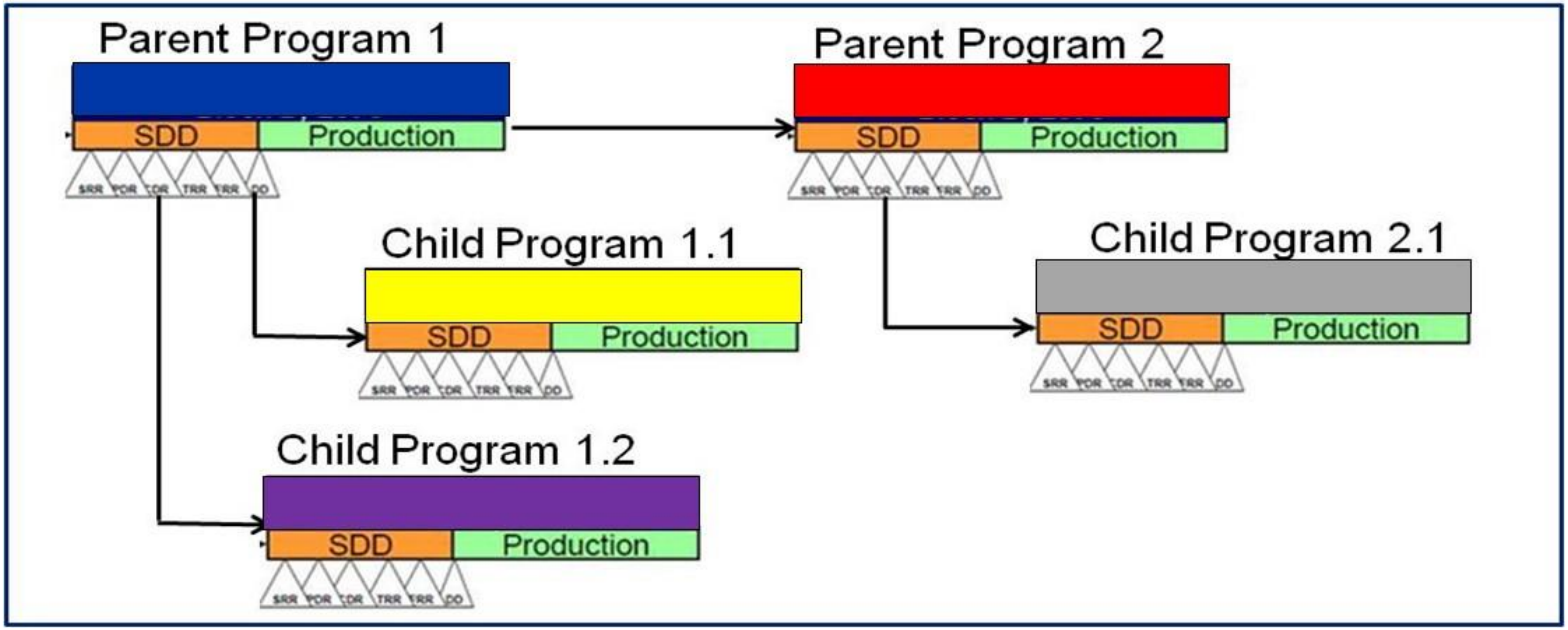
- •Focus on helping DES team to investigate DOORS Next Generation and to help programs get started with DNG
- • Leverage Jazz Global Configuration Control across tools in support of project PLE requirements
- • Work with BCA MBSA project to integrate Jazz platform with Mentor Graphics Context SDM solution in support of cross-discipline use cases

Agenda

- Background – What does OSLC and the Jazz platform provide for development activities
- Motivations for Apache Systems Engineering
- Platform Enablers
- Change Management Process
- Lessons Learned and Next Steps
- *Summary*

Apache Program Relationships

- Child program(s) spawn from a single **parent program** baseline
  - **Designs are reused from parent to child programs where possible**
- **Child Programs may require a subset of content from the parent baseline**
  - Include unique content from parent **program**
  - Require different access controls than parent program



Need for Change Management

Problem:

- Reuse of functional architecture models across programs requires change management process, including traceability.

Plan:

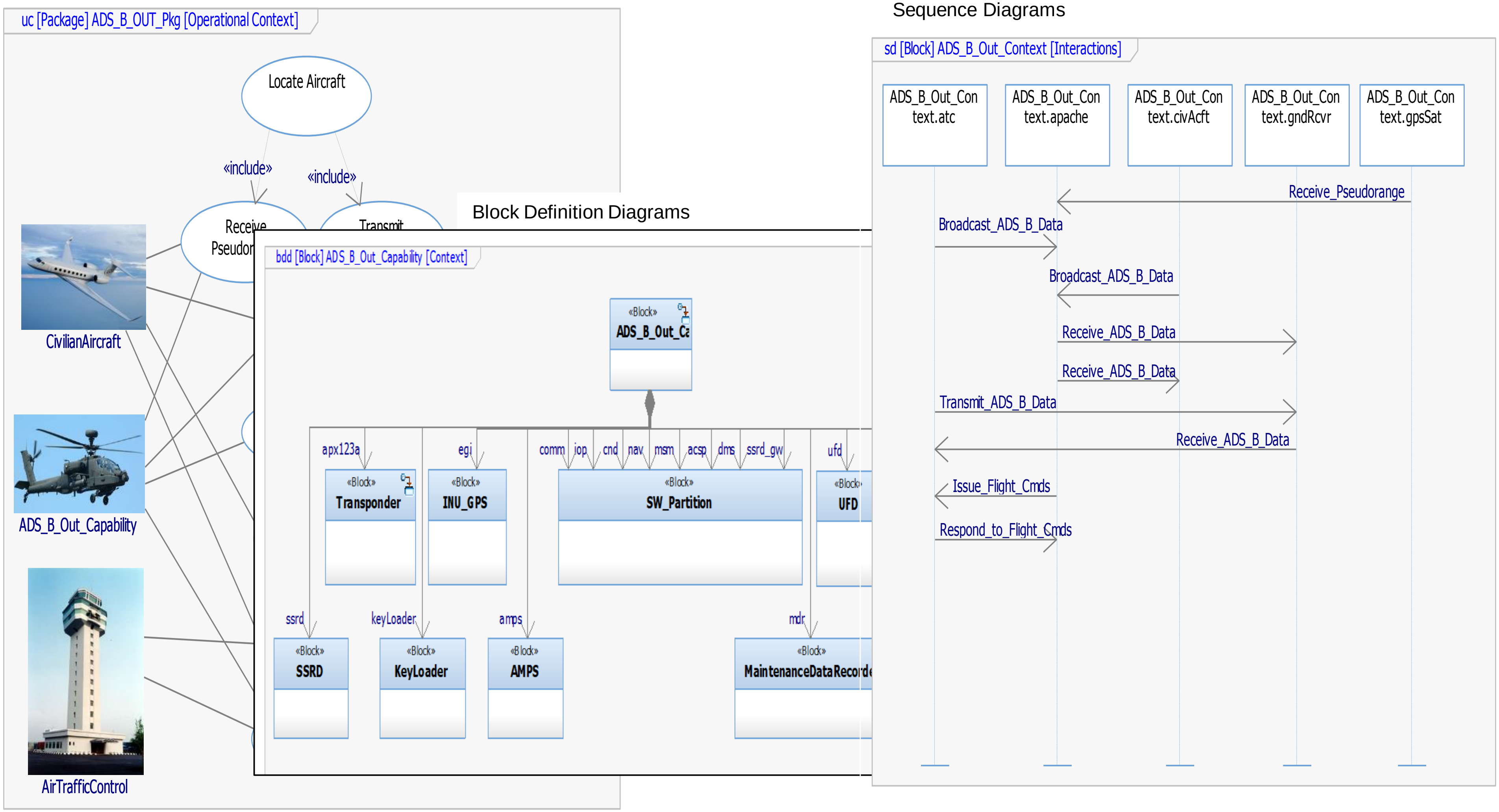
- Leverage established tools/processes
- Develop documented process to authorize architecture changes
- Trace changes to impacted model elements

Functional Architecture Elements

Use Cases

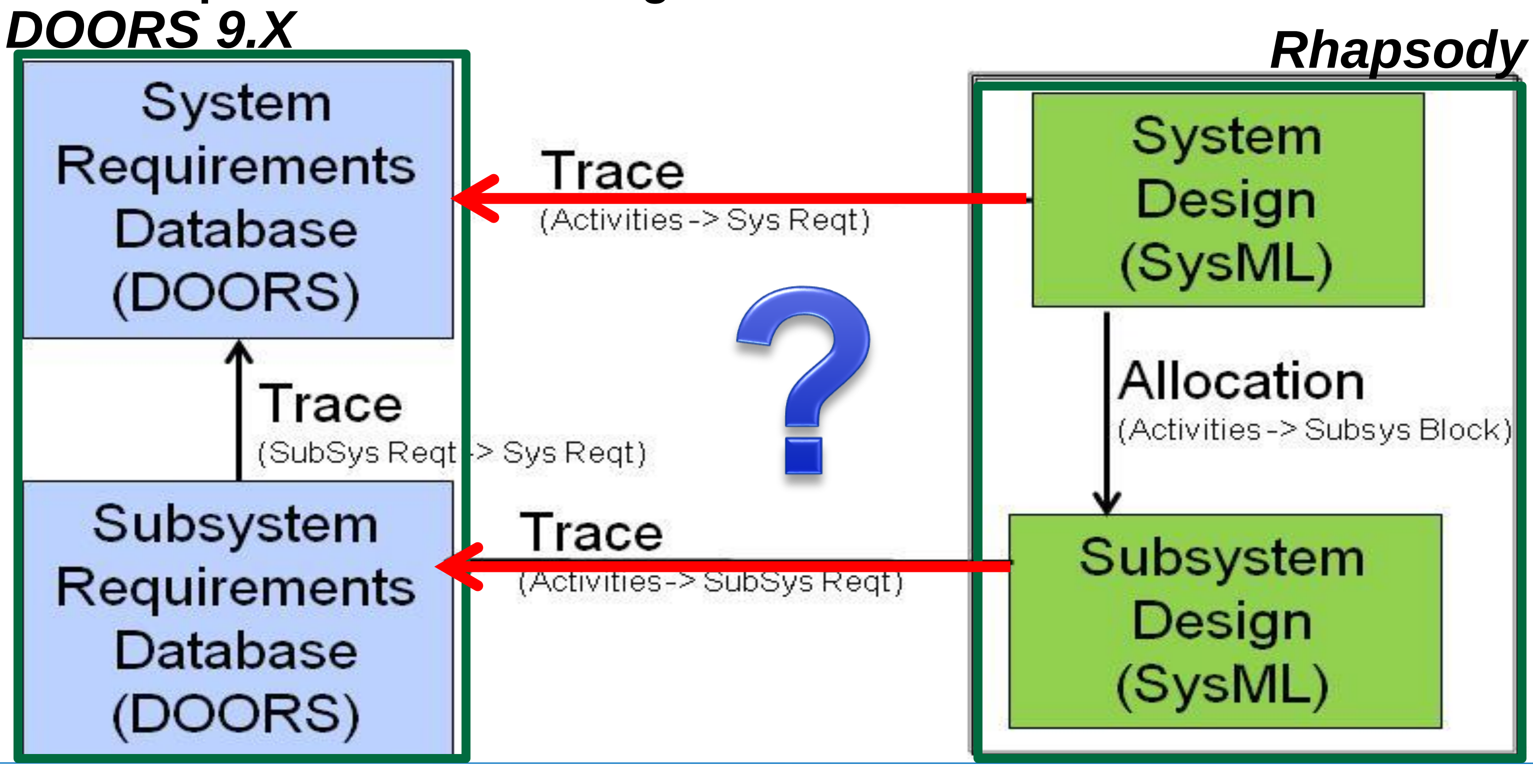
Sequence Diagrams

Block Definition Diagrams



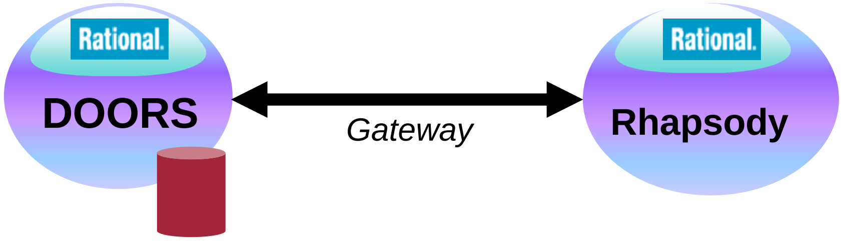
How to Trace Design to Requirements?

- **Traceability between subsystem and system level solved:**
  - **Requirements: managed in DOORS**
  - **Design: Managed in Rhapsody**
  - **Requirements <-> Design: ???**


DOORS 9.X
Rhapsody
System Requirements Database (DOORS)
Trace
(Activities -> Sys Reqt)
System Design (SysML)
Allocation
(Activities -> Subsys Block)
Trace
(SubSys Reqt -> Sys Reqt)
?
Subsystem Requirements Database (DOORS)
Trace
(Activities -> SubSys Reqt)
Subsystem Design (SysML)

Goal is to Leverage Existing Tools…..

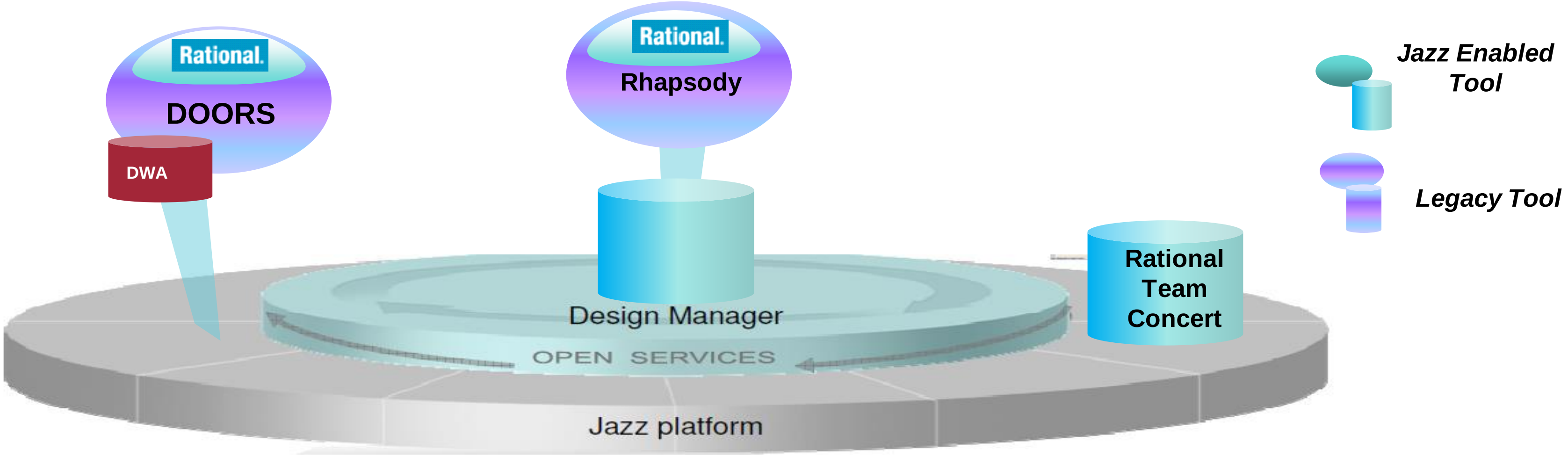
- **DOORS**
  - Stand-alone Requirements Management tool
- **Rhapsody**
  - Client Architecture Development app



Legacy BDS StdTool

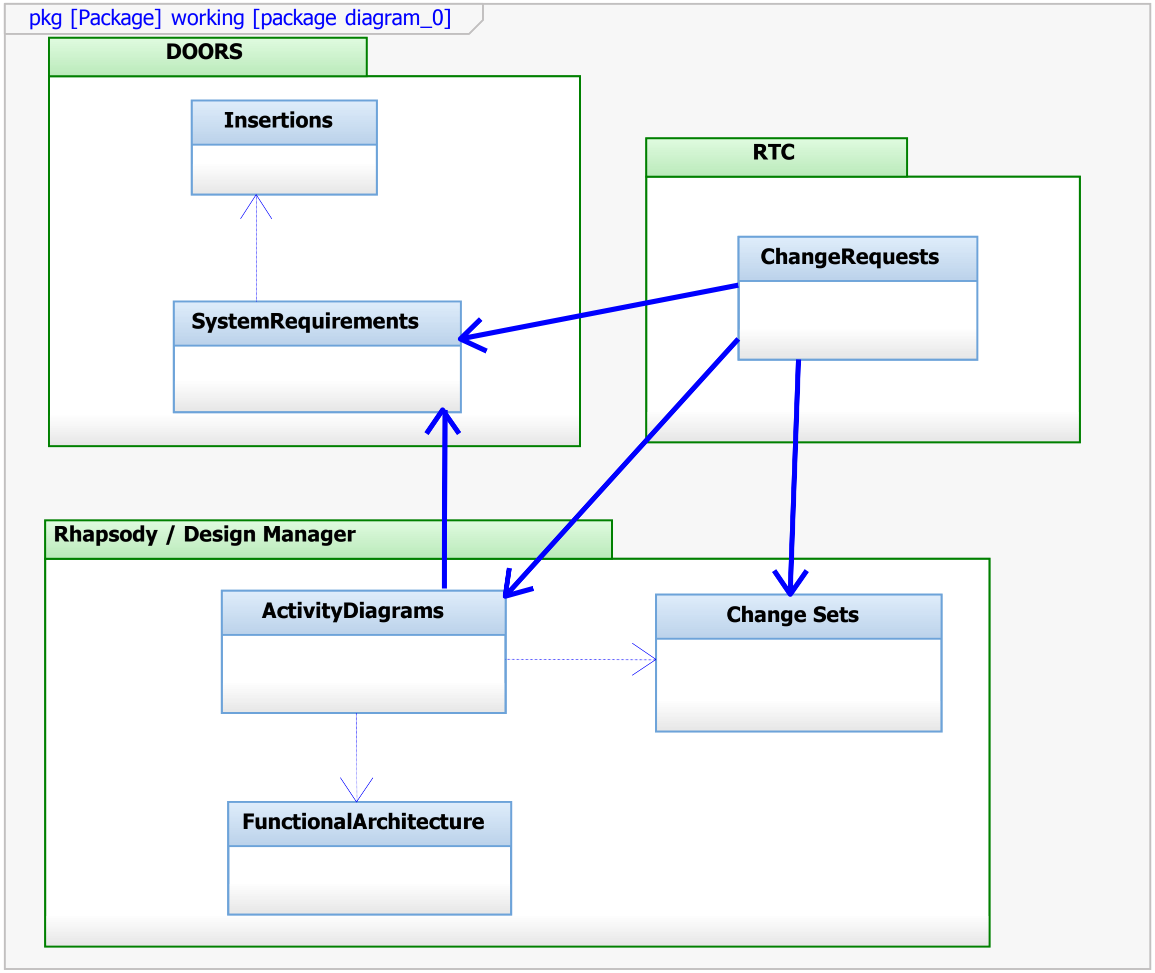
…..and New Capabilities to meet needs

- **DOORS**
  - Stand-alone Requirements Management tool
- **Rhapsody**
  - Client Architecture Development app
- **DOORS Web Access (DWA)**
  - Web based server add-on for DOORS. Supports OSLC
- **Design Manager**
  - Rhapsody Add-On providing Jazz-based Configuration Management, Collaboration, Access Control. Web Interface.
- **Team Concert (RTC)**
  - Jazz offering providing Change Management, workflow planning, interface to other Rational tools.



Resulting Integrated Tool Environment Benefits

- **Data Connectivity between Tools**
  - **Enabled by OSLC linking**
  - **Improved traceability**
  - **Access controls managed by tool per project**
- **Jazz platform is scalable – can add tools for new capabilities**



Agenda

- Background – What does OSLC and the Jazz platform provide for development activities
- Motivations for Apache Systems Engineering
- Platform Enablers
- Change Management Process
- Lessons Learned and Next Steps
- *Summary*

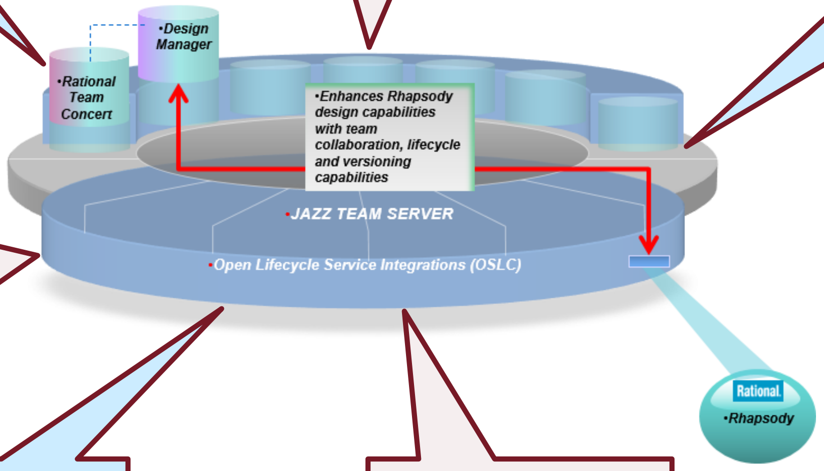
Rhapsody Design Manager Features

System-wide Repository for Design Reuse and Cross-Domain Querying

Manage Design Configuration directly on DM Server

Document Generation and Reporting across all OSLC assets

Linking and Tracing of modeling artifacts to other lifecycle artifacts and also between systems and software architecture/design


•Design Manager
•Rational Team Concert
•Enhances Rhapsody design capabilities with team collaboration, lifecycle and versioning capabilities
•JAZZ TEAM SERVER
•Open Lifecycle Service Integrations (OSLC)
Rational
•Rhapsody

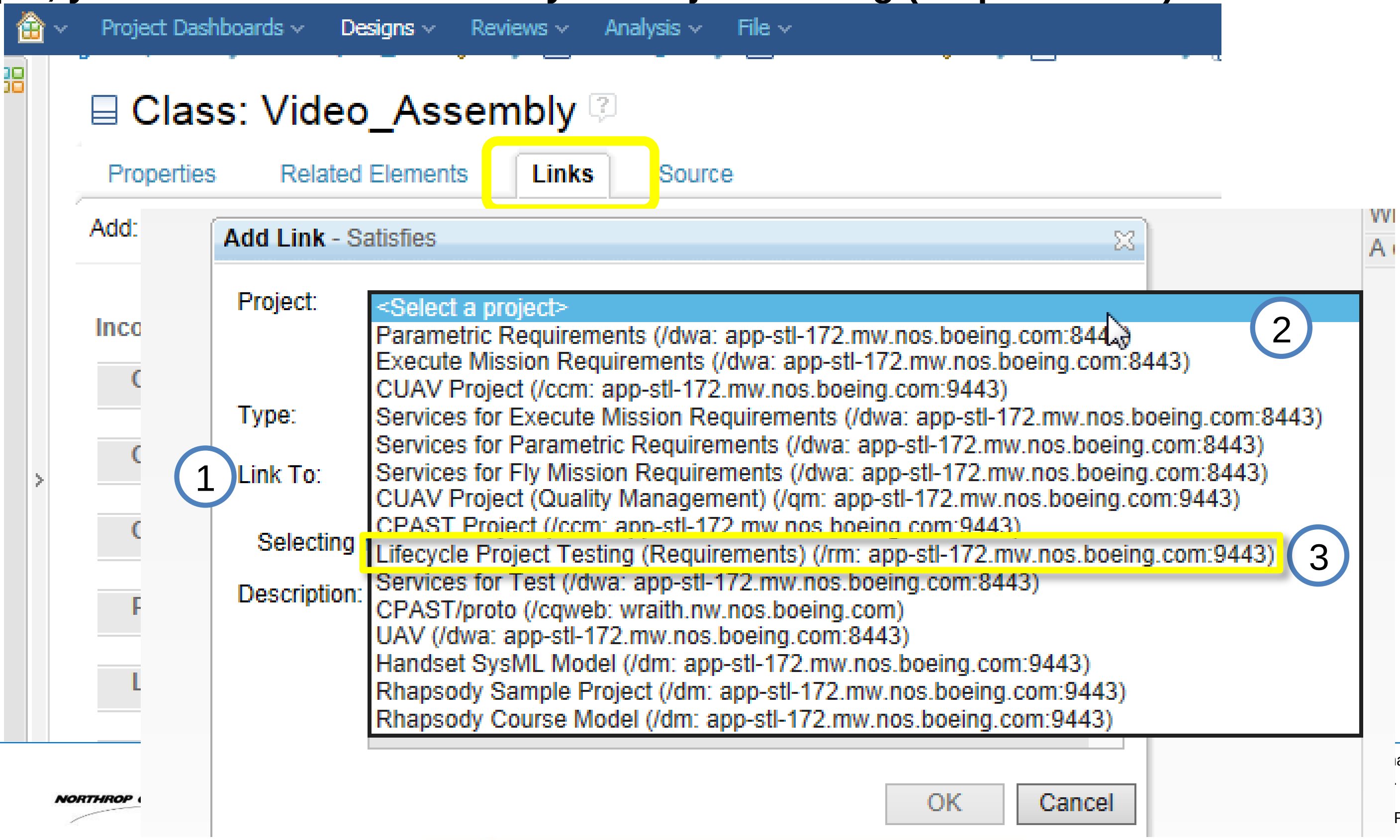
Analyze Data (relationships and dependencies)

Collaboration (including reviews and approvals)

Traceability

You can create links between artifacts from Rhapsody or from the DM web interface (shown here).

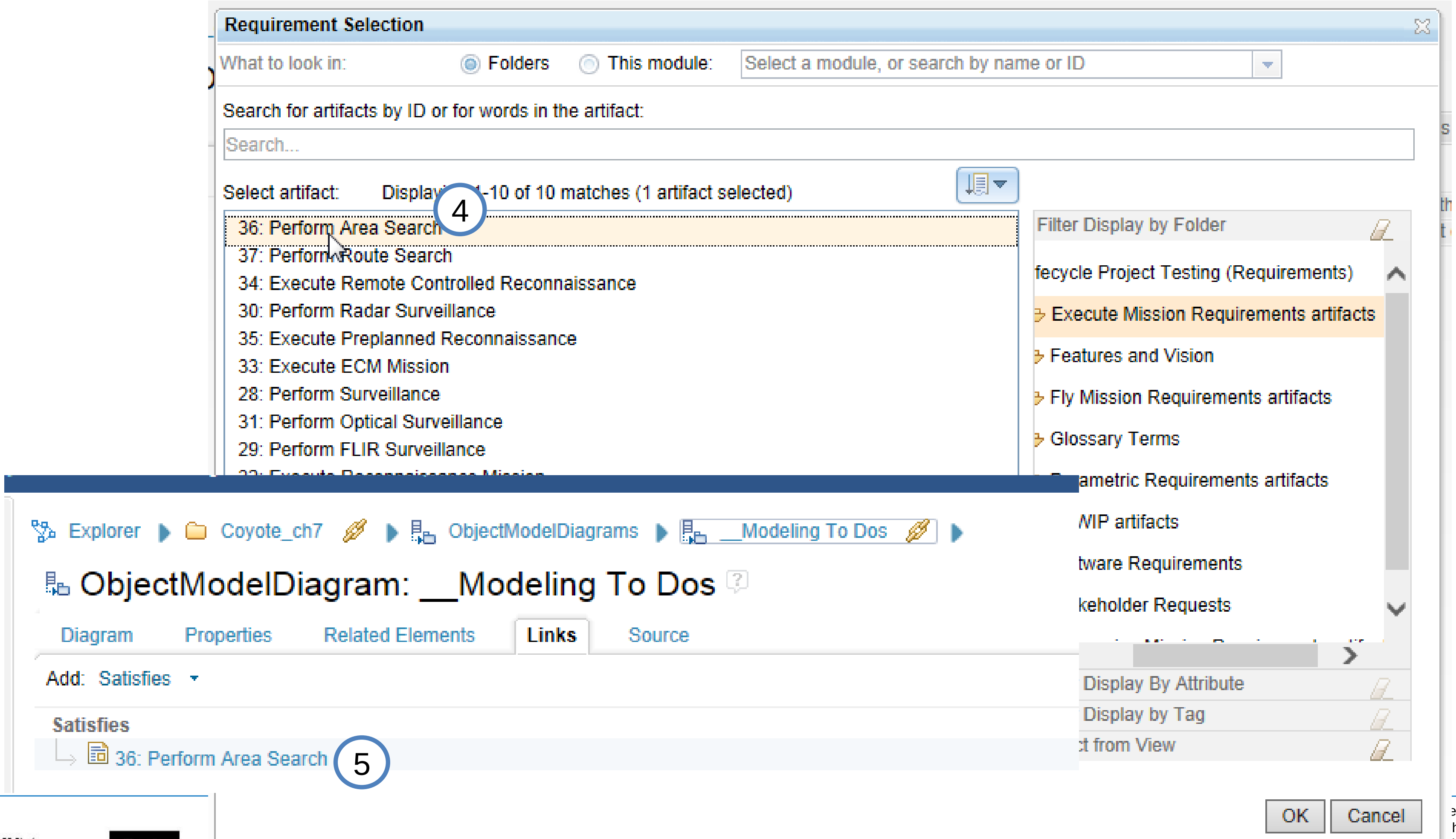
1. **You choose the appropriate link type from the available links which have been defined by the Project Administrator on your Project. The example is representative.**
2. **From the pulldown you will find a number of possible source repositories that you can use as the target.**
3. **In this example, you could select from Lifecycle Project Testing (Requirements).**



<Select a project>
Parametric Requirements (/dwa: app-stl-172.mw.nos.boeing.com:844
Execute Mission Requirements (/dwa: app-stl-172.mw.nos.boeing.com:8443)
CUAV Project (/ccm: app-stl-172.mw.nos.boeing.com:9443)
Services for Execute Mission Requirements (/dwa: app-stl-172.mw.nos.boeing.com:8443)
Services for Parametric Requirements (/dwa: app-stl-172.mw.nos.boeing.com:8443)
Services for Fly Mission Requirements (/dwa: app-stl-172.mw.nos.boeing.com:8443)
CUAV Project (Quality Management) (/qm: app-stl-172.mw.nos.boeing.com:9443)
CPAST Project (/ccm: app-stl-172.mw.nos.boeing.com:9443)
Lifecycle Project Testing (Requirements) (/rm: app-stl-172.mw.nos.boeing.com:9443)
Services for Test (/dwa: app-stl-172.mw.nos.boeing.com:8443)
CPAST/proto (/cqweb: wraith.nw.nos.boeing.com)
UAV (/dwa: app-stl-172.mw.nos.boeing.com:8443)
Handset SysML Model (/dm: app-stl-172.mw.nos.boeing.com:9443)
Rhapsody Sample Project (/dm: app-stl-172.mw.nos.boeing.com:9443)
Rhapsody Course Model (/dm: app-stl-172.mw.nos.boeing.com:9443)

Traceability (cont.)

4. **Then using the interface, navigate and find the requirement(s) you wish to select for the link. Select them and click OK.**
5. **The resulting hyperlink will be reflected on the Links tab.**



Requirement Selection

What to look in: ● Folders ○ This module: Select a module, or search by name or ID

Search for artifacts by ID or for words in the artifact:

Search...

Select artifact: Displaying 1-10 of 10 matches (1 artifact selected)

- 36: Perform Area Search
- 37: Perform Route Search
- 34: Execute Remote Controlled Reconnaissance
- 30: Perform Radar Surveillance
- 35: Execute Preplanned Reconnaissance
- 33: Execute ECM Mission
- 28: Perform Surveillance
- 31: Perform Optical Surveillance
- 29: Perform FLIR Surveillance

Filter Display by Folder

- fecycle Project Testing (Requirements)
- Execute Mission Requirements artifacts
- Features and Vision
- Fly Mission Requirements artifacts
- Glossary Terms
- ametric Requirements artifacts
- WIP artifacts
- tware Requirements
- keholder Requests

Display By Attribute

Display by Tag

t from View

OK Cancel

Explorer ▸ Coyote_ch7 ▸ ObjectModelDiagrams ▸ __Modeling To Dos ▸

ObjectModelDiagram: __Modeling To Dos

Diagram | Properties | Related Elements | **Links** | Source

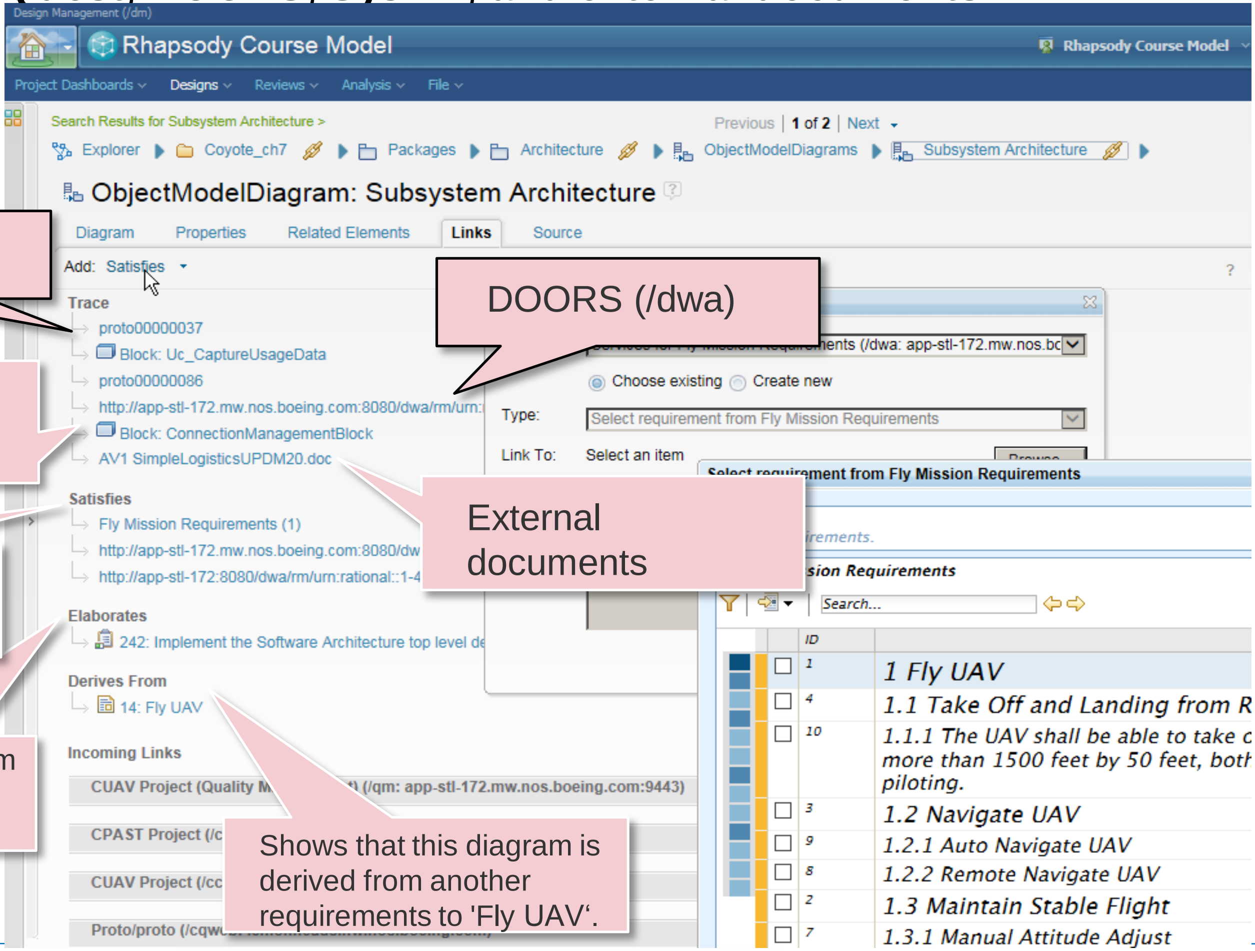
Add: Satisfies

Satisfies

36: Perform Area Search

Linking Possibilities

Traceability can be established by linking model artifacts to many different artifacts in other lifecycle tools or in other Design Management models. Some examples of linked data include IBM ClearQuest, DOORS, SysML, and external documents.



Design Management (/dm)

Rhapsody Course Model

ObjectModelDiagram: Subsystem Architecture

Add: Satisfies

Trace
- proto00000037
- Block: Uc_CaptureUsageData
- proto00000086
- http://app-stl-172.mw.nos.boeing.com:8080/dwa/rm/urn:
- Block: ConnectionManagementBlock
- AV1 SimpleLogisticsUPDM20.doc

Satisfies
- Fly Mission Requirements (1)
- http://app-stl-172.mw.nos.boeing.com:8080/dw
- http://app-stl-172:8080/dwa/rm/urn:rational::1-4

Elaborates
- 242: Implement the Software Architecture top level de

Derives From
- 14: Fly UAV

Incoming Links
- CUAV Project (Quality M) (/qm: app-stl-172.mw.nos.boeing.com:9443)
- CPAST Project (/c
- CUAV Project (/cc
- Proto/proto (/cqw

ents (/dwa: app-stl-172.mw.nos.bc

○ Choose existing ○ Create new

Type: Select requirement from Fly Mission Requirements

Link To: Select an item

Select requirement from Fly Mission Requirements

irements.

sion Requirements

Search...

ID	
1	1 Fly UAV
4	1.1 Take Off and Landing from R
10	1.1.1 The UAV shall be able to take c more than 1500 feet by 50 feet, both piloting.
3	1.2 Navigate UAV
9	1.2.1 Auto Navigate UAV
8	1.2.2 Remote Navigate UAV
2	1.3 Maintain Stable Flight
7	1.3.1 Manual Attitude Adjust

IBM ClearQuest

SysML model blocks

Shows Satisfies relationships (i.e. this object model diagram satisfies the following requirements) to DOORS.

Shows that this diagram elaborates an RTC work item task,

DOORS (/dwa)

External documents

Shows that this diagram is derived from another requirements to 'Fly UAV‘.

BOEING is a trademark of Boeing Management Company

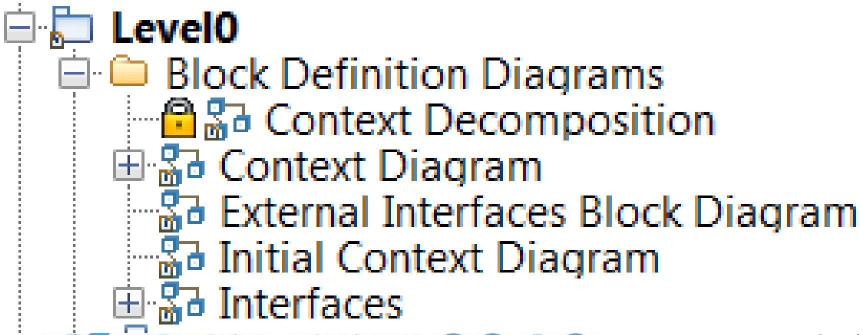
Model Management on the Jazz Platform

- ***Actively Managed Model***

When your Rhapsody/DM project is actively managed, you can move Rhapsody models to the Design Management Server or create new models directly on the server and open them from Rational Rhapsody.

A Rhapsody user can open a model on Design Manager, make changes to the model in Rhapsody, and save the changes back to the Design Management Server.

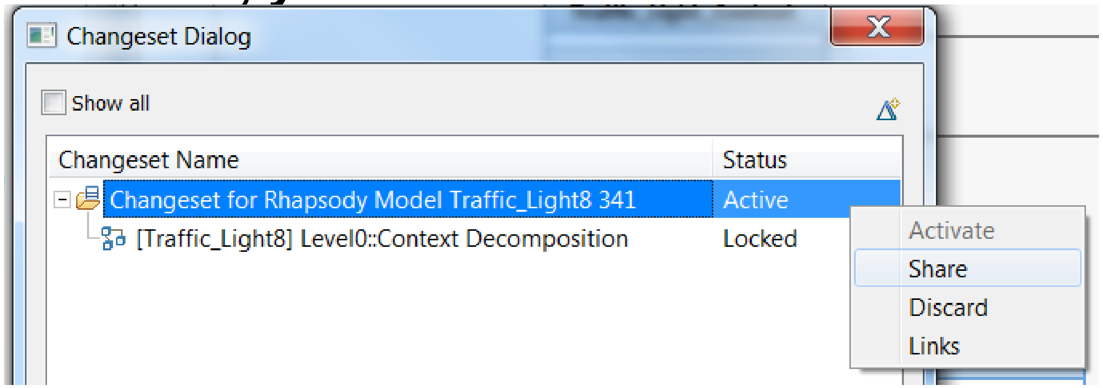
Other users, reviewers, or customers who may access the model through the DM web client, will get the latest updates to the model that have been shared by the development team.



Level0
Block Definition Diagrams
Context Decomposition
Context Diagram
External Interfaces Block Diagram
Initial Context Diagram
Interfaces

Change Management

- ***Change Sets***
  - **Use change sets to group related sets of changes. To make the changes in the change set visible to other users, you must *share* them.**

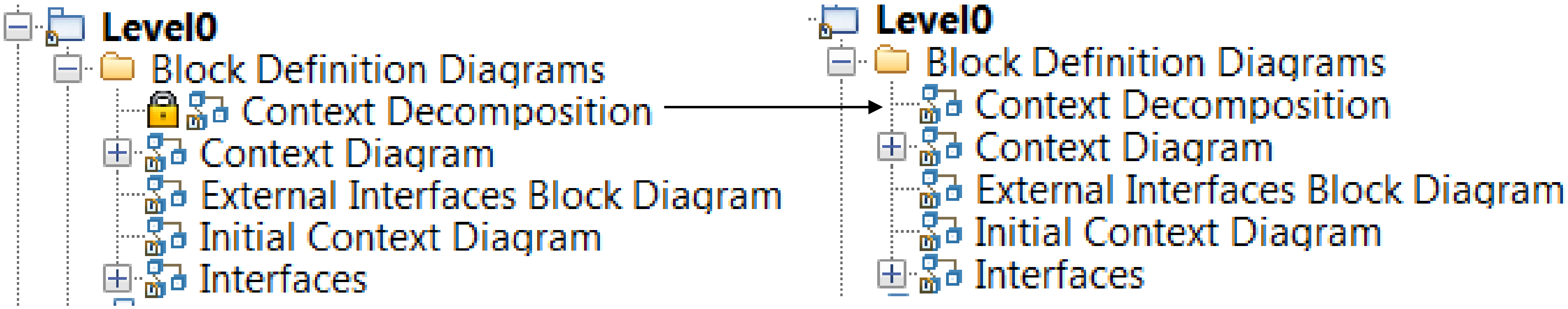


Changeset Dialog

☐ Show all

Changeset Name	Status
Changeset for Rhapsody Model Traffic_Light8 341	Active
[Traffic_Light8] Level0::Context Decomposition	Locked

1. The changes to the Context Decomposition Diagram are now part of the Project Area
2. All Users can see the modified diagram.
3. Any of the users can further modify it.

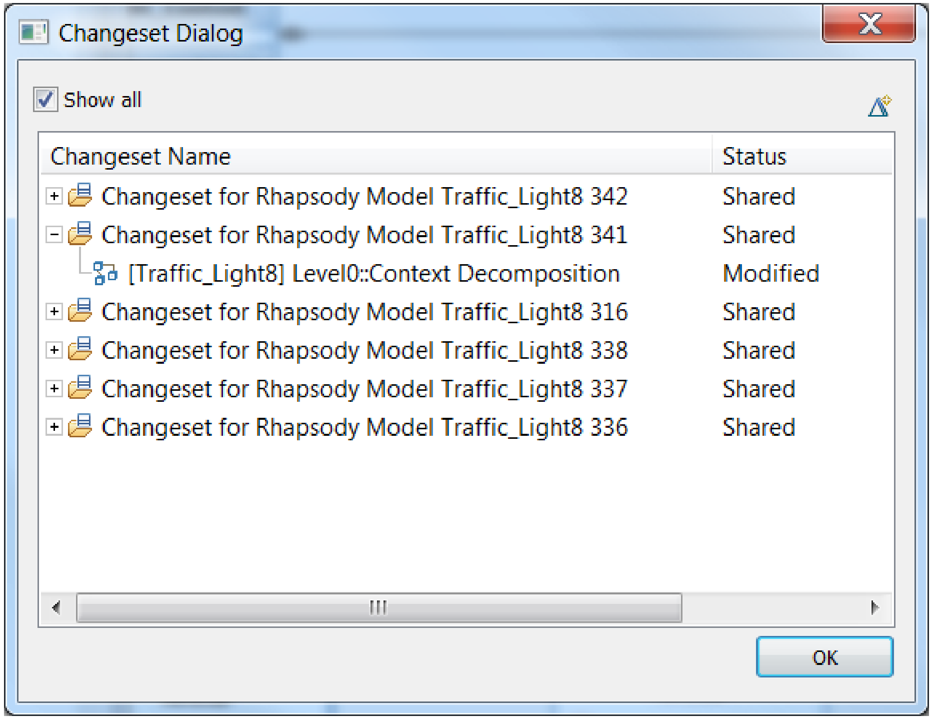


Level0
Block Definition Diagrams
Context Decomposition
Context Diagram
External Interfaces Block Diagram
Initial Context Diagram
Interfaces

Level0
Block Definition Diagrams
Context Decomposition
Context Diagram
External Interfaces Block Diagram
Initial Context Diagram
Interfaces

Change Management – Multiple Change Sets

- ***Multiple change sets can be active at once***



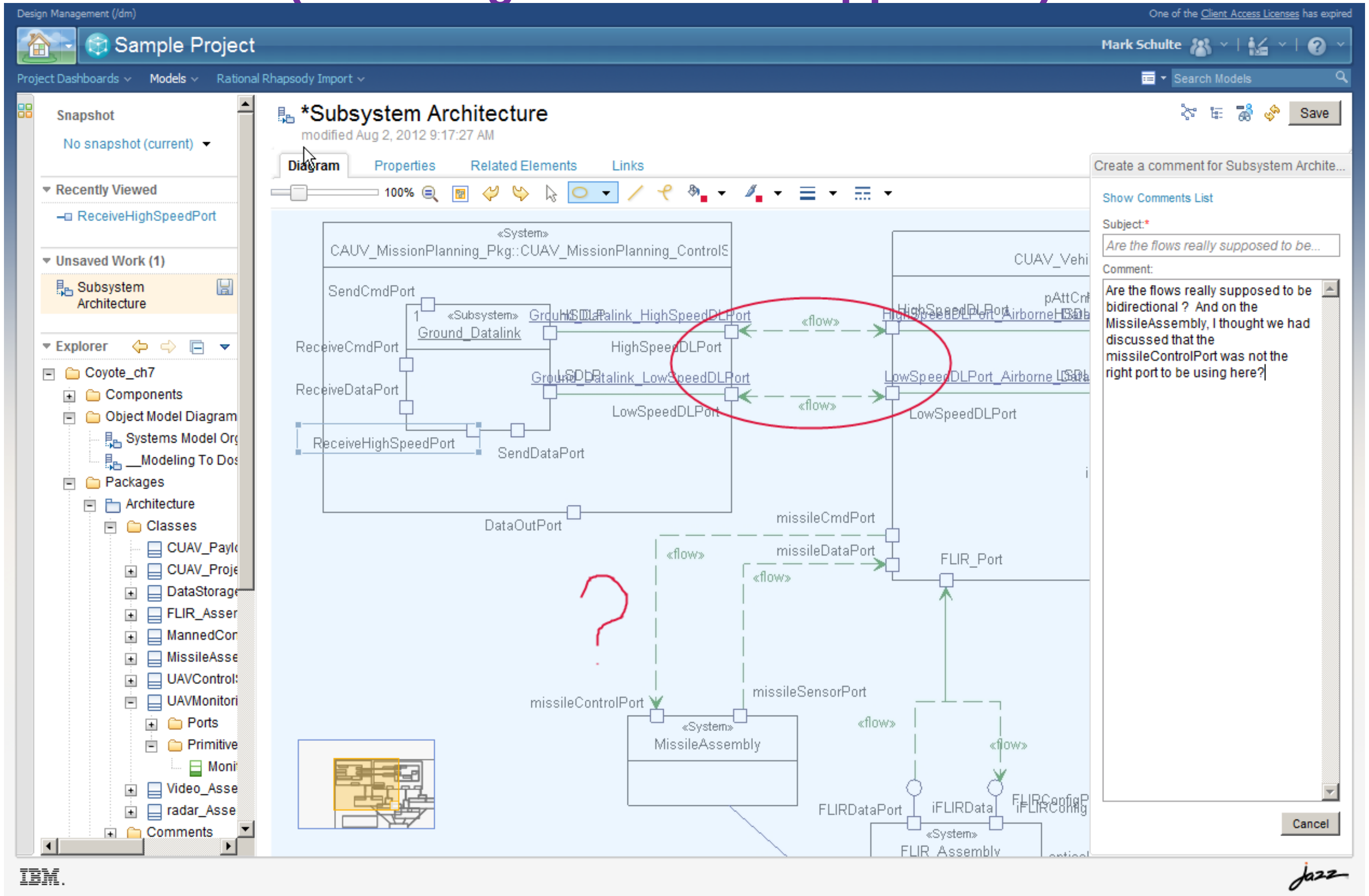
☑ Show all

Changeset Name	Status
⊞ Changeset for Rhapsody Model Traffic_Light8 342	Shared
⊟ Changeset for Rhapsody Model Traffic_Light8 341	Shared
[Traffic_Light8] Level0::Context Decomposition	Modified
⊞ Changeset for Rhapsody Model Traffic_Light8 316	Shared
⊞ Changeset for Rhapsody Model Traffic_Light8 338	Shared
⊞ Changeset for Rhapsody Model Traffic_Light8 337	Shared
⊞ Changeset for Rhapsody Model Traffic_Light8 336	Shared

1. Change set 341 has been shared with the Project Area
2. That change set modified the Context Decomposition Diagram.

Collaboration

- **Improved workflows and lifecycle management**
  - **Collaboration (including reviews and approvals)**



Create a comment for Subsystem Archite...

Show Comments List

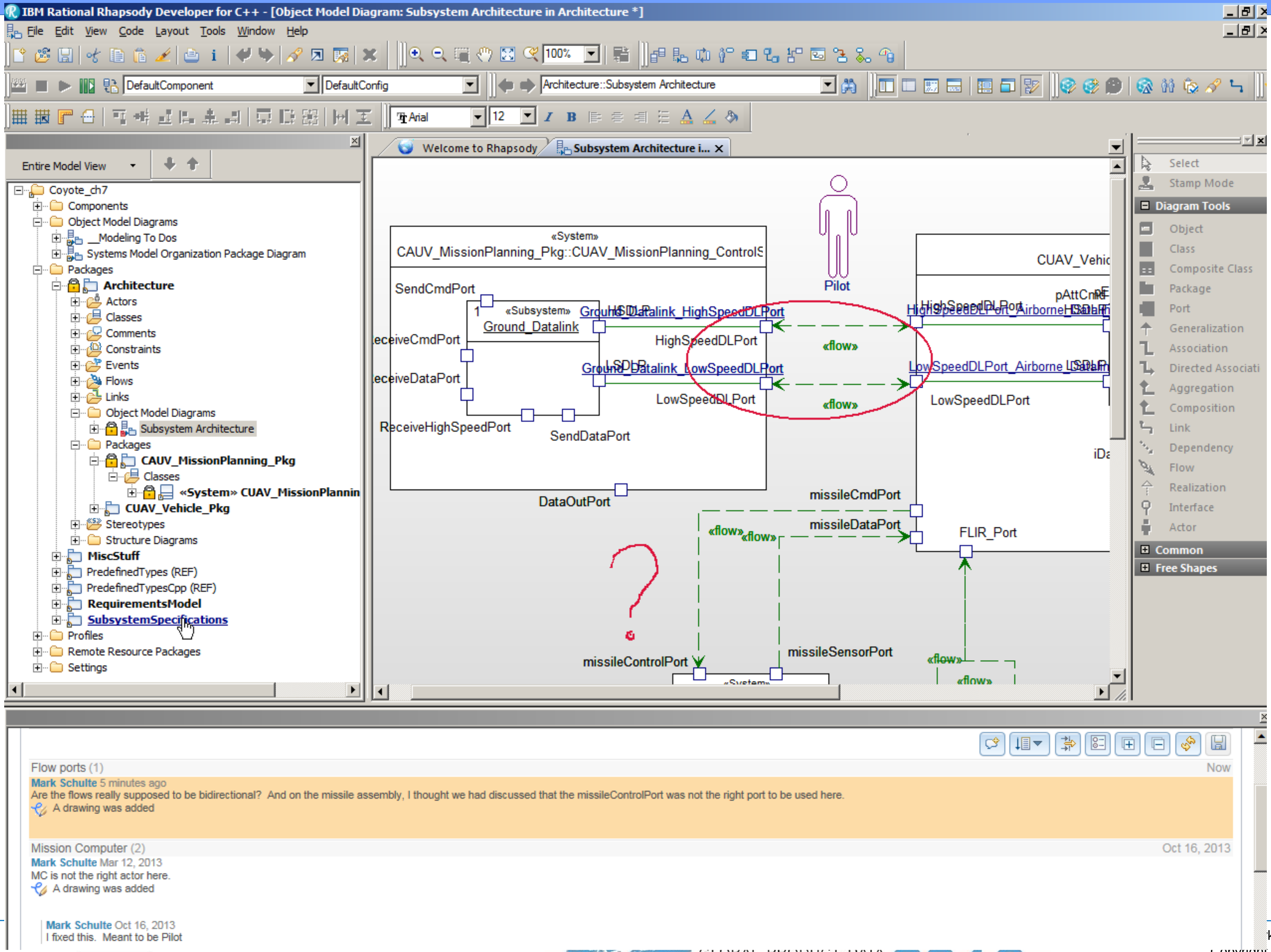
Subject:* Are the flows really supposed to be...

Comment:

Are the flows really supposed to be bidirectional ? And on the MissileAssembly, I thought we had discussed that the missileControlPort was not the right port to be using here?

Cancel

Comments are viewable from the Rhapsody Client



Flow ports (1) Now

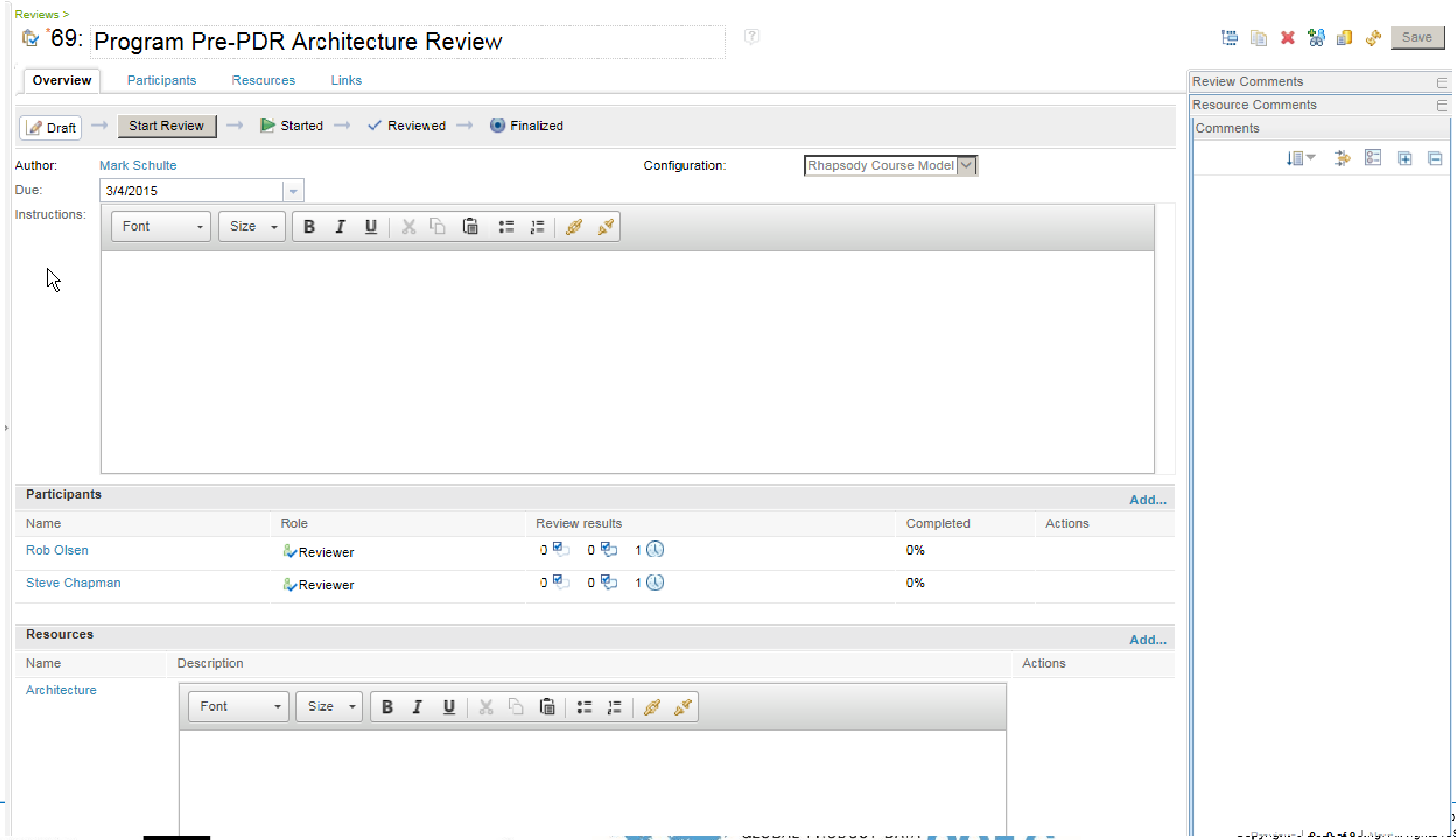
Mark Schulte 5 minutes ago
Are the flows really supposed to be bidirectional? And on the missile assembly, I thought we had discussed that the missileControlPort was not the right port to be used here.
A drawing was added

Mission Computer (2) Oct 16, 2013

Mark Schulte Mar 12, 2013
MC is not the right actor here.
A drawing was added

Mark Schulte Oct 16, 2013
I fixed this. Meant to be Pilot

Online Reviews



*69: Program Pre-PDR Architecture Review

Overview | Participants | Resources | Links

Author: Mark Schulte

Configuration: Rhapsody Course Model

Due: 3/4/2015

Instructions:

Participants

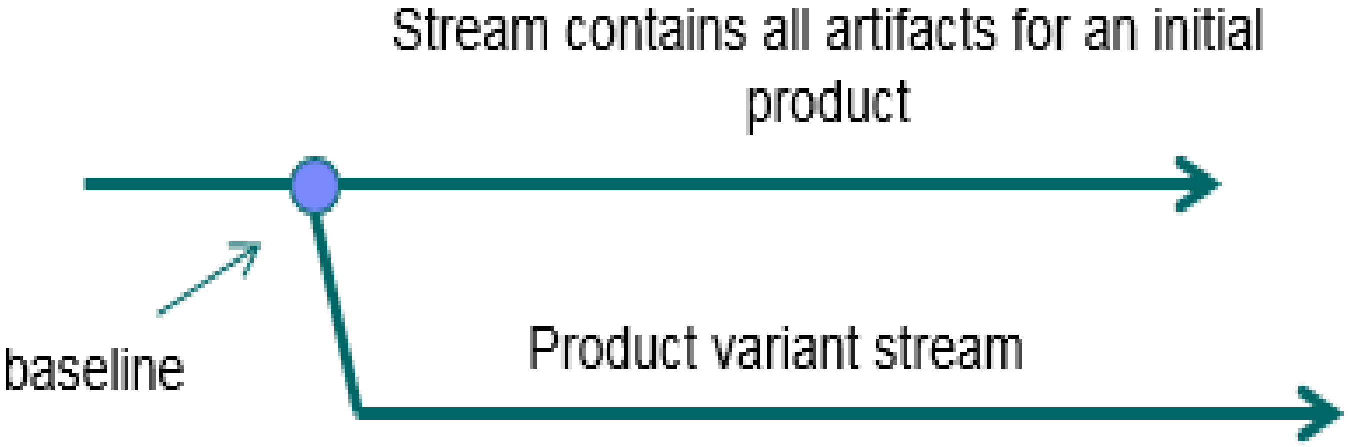
Name	Role	Review results	Completed	Actions
Rob Olsen	Reviewer	0 0 1	0%	
Steve Chapman	Reviewer	0 0 1	0%	

Resources

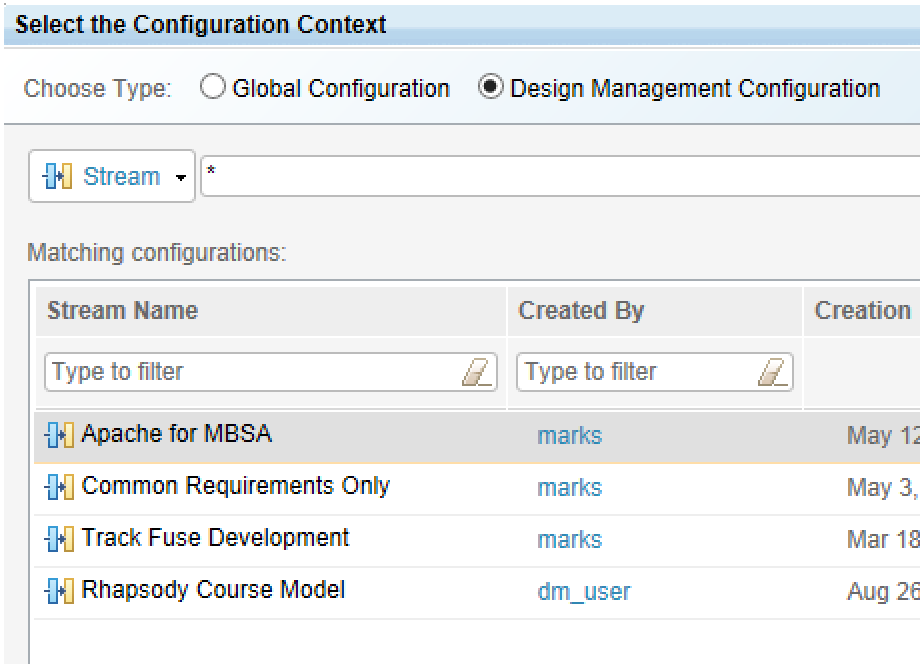
Name	Description	Actions
Architecture		

Configuration Management on the Jazz Platform

- ***Configurations (Streams and Baselines)***
  - **A *stream* is a mutable working set of artifacts that make up a particular version of the model**
  - **A baseline is an immutable (snapshot) set of the artifacts that made up the model at a particular point in time.**



Configuration Management on the Jazz Platform - Examples



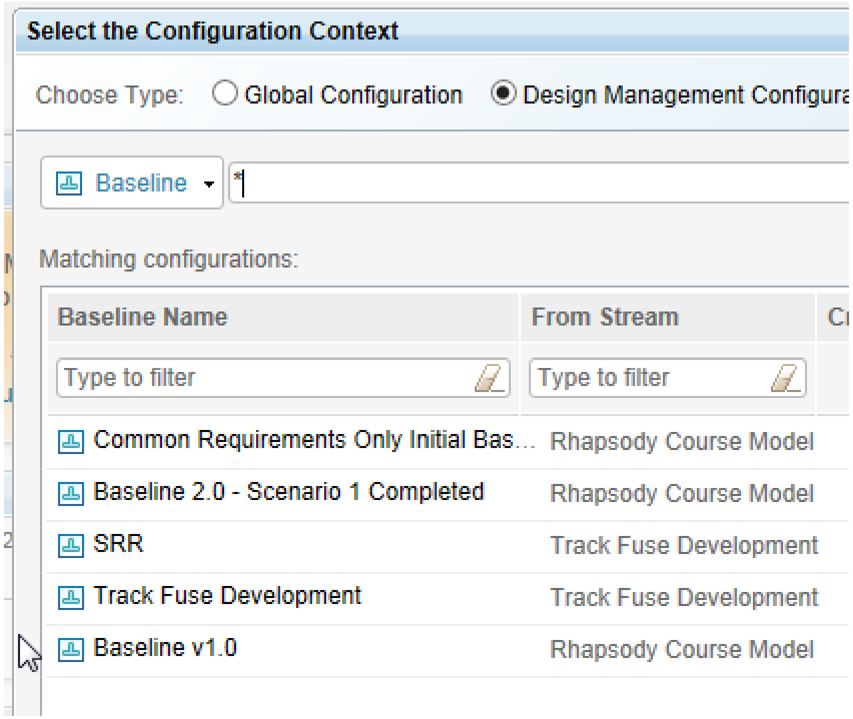
Select the Configuration Context

Choose Type: ○ Global Configuration ● Design Management Configuration

Stream ▾ *

Matching configurations:

Stream Name	Created By	Creation
Type to filter	Type to filter	
Apache for MBSA	marks	May 12
Common Requirements Only	marks	May 3,
Track Fuse Development	marks	Mar 18
Rhapsody Course Model	dm_user	Aug 26



Select the Configuration Context

Choose Type: ○ Global Configuration ● Design Management Configura

Baseline ▾ *

Matching configurations:

Baseline Name	From Stream	Cr
Type to filter	Type to filter	
Common Requirements Only Initial Bas...	Rhapsody Course Model	
Baseline 2.0 - Scenario 1 Completed	Rhapsody Course Model	
SRR	Track Fuse Development	
Track Fuse Development	Track Fuse Development	
Baseline v1.0	Rhapsody Course Model	

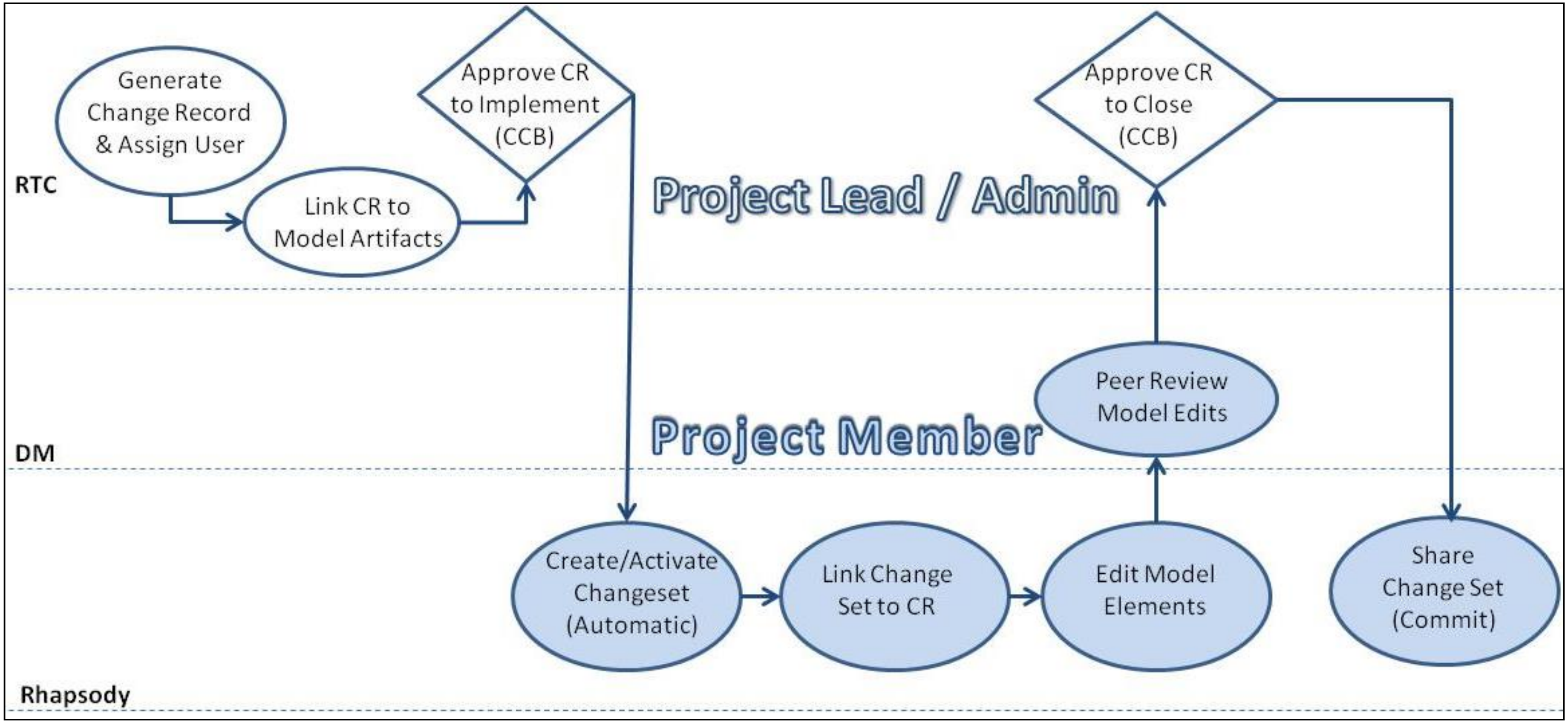
Agenda

- Background – What does OSLC and the Jazz platform provide for development activities
- Motivations for Apache Systems Engineering
- Platform Enablers
- Change Management Process
- Lessons Learned and Next Steps
- *Summary*

Change Management Process

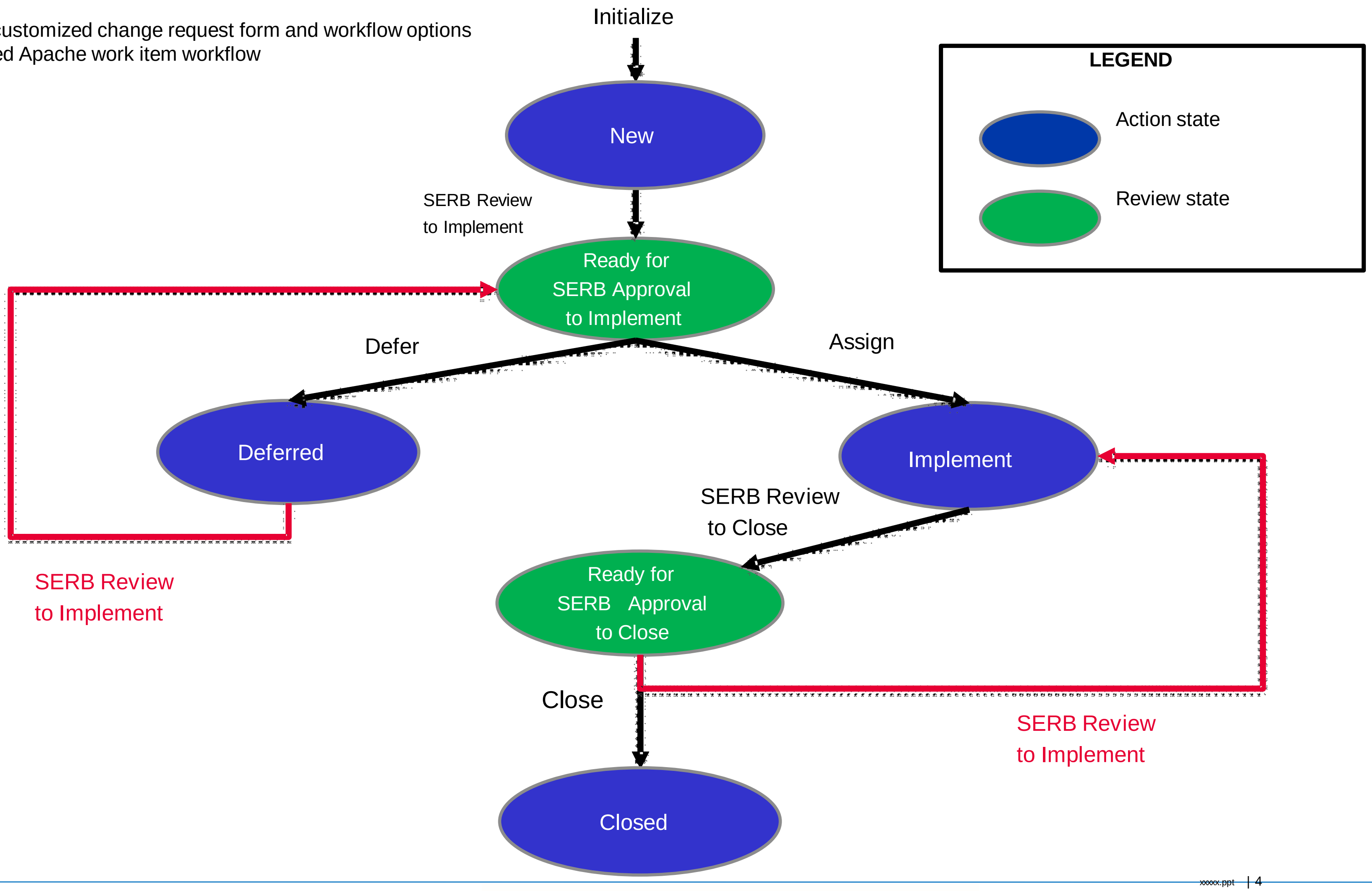
Managing Changes to the Functional Architecture

- **Developed change management process flow**
  - Leverages selected tools
  - Manages FA changes using records and approvals

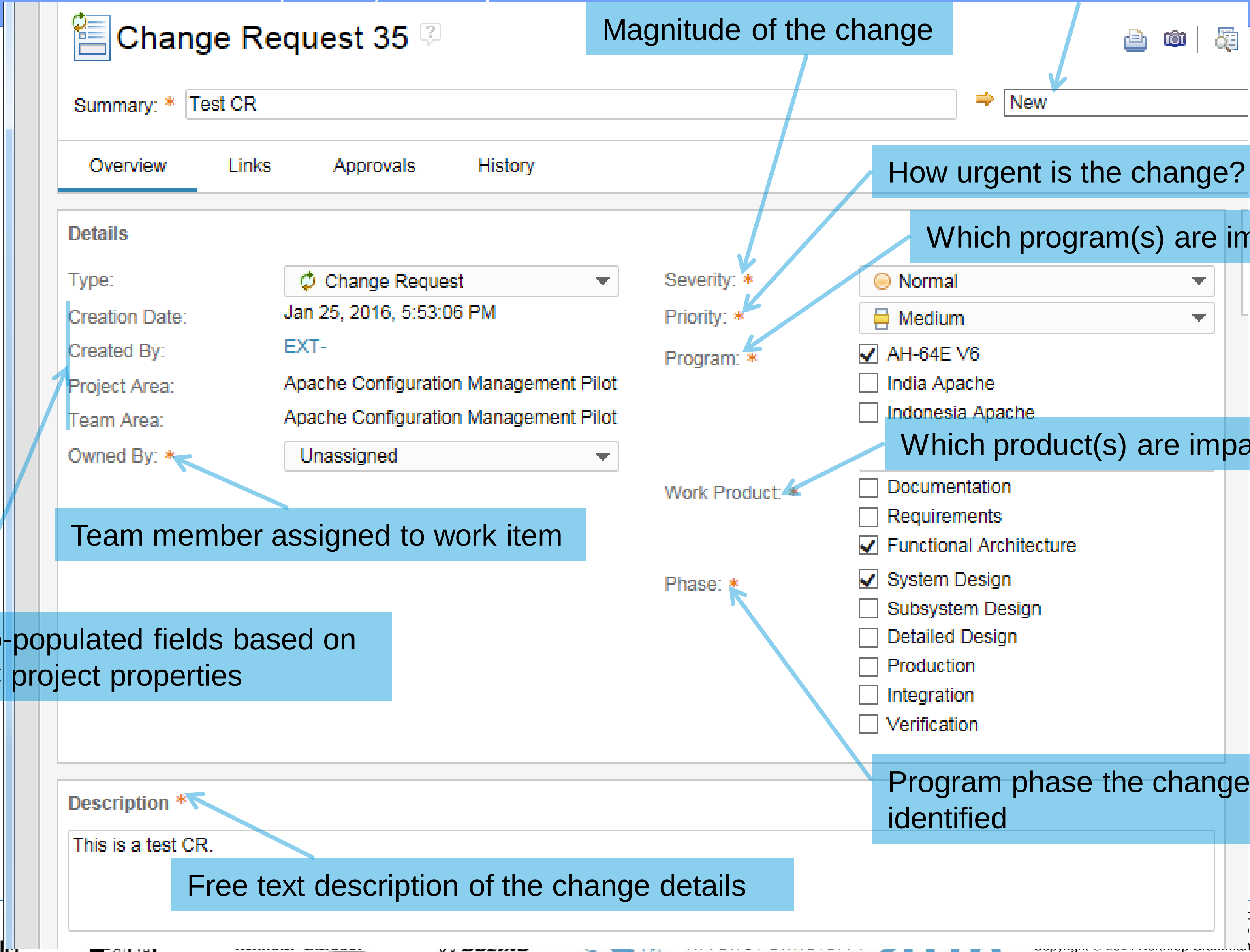

RTC
Generate Change Record & Assign User
Link CR to Model Artifacts
Approve CR to Implement (CCB)
Approve CR to Close (CCB)
Project Lead / Admin
DM
Project Member
Peer Review Model Edits
Rhapsody
Create/Activate Changeset (Automatic)
Link Change Set to CR
Edit Model Elements
Share Change Set (Commit)

RTC for Change Management

- Provides customized change request form and workflow options
- Customized Apache work item workflow



RTC: Change Request Form



Change Request 35

Summary: * Test CR

New

Overview | Links | Approvals | History

Details

Type: Change Request

Creation Date: Jan 25, 2016, 5:53:06 PM

Created By: EXT-

Project Area: Apache Configuration Management Pilot

Team Area: Apache Configuration Management Pilot

Owned By: * Unassigned

Severity: * Normal

Priority: * Medium

Program: *
☑ AH-64E V6
☐ India Apache
☐ Indonesia Apache

Work Product: *
☐ Documentation
☐ Requirements
☑ Functional Architecture

Phase: *
☑ System Design
☐ Subsystem Design
☐ Detailed Design
☐ Production
☐ Integration
☐ Verification

Description *

This is a test CR.

Change Request State

Magnitude of the change

How urgent is the change?

Which program(s) are impacted?

Which product(s) are impacted?

Team member assigned to work item

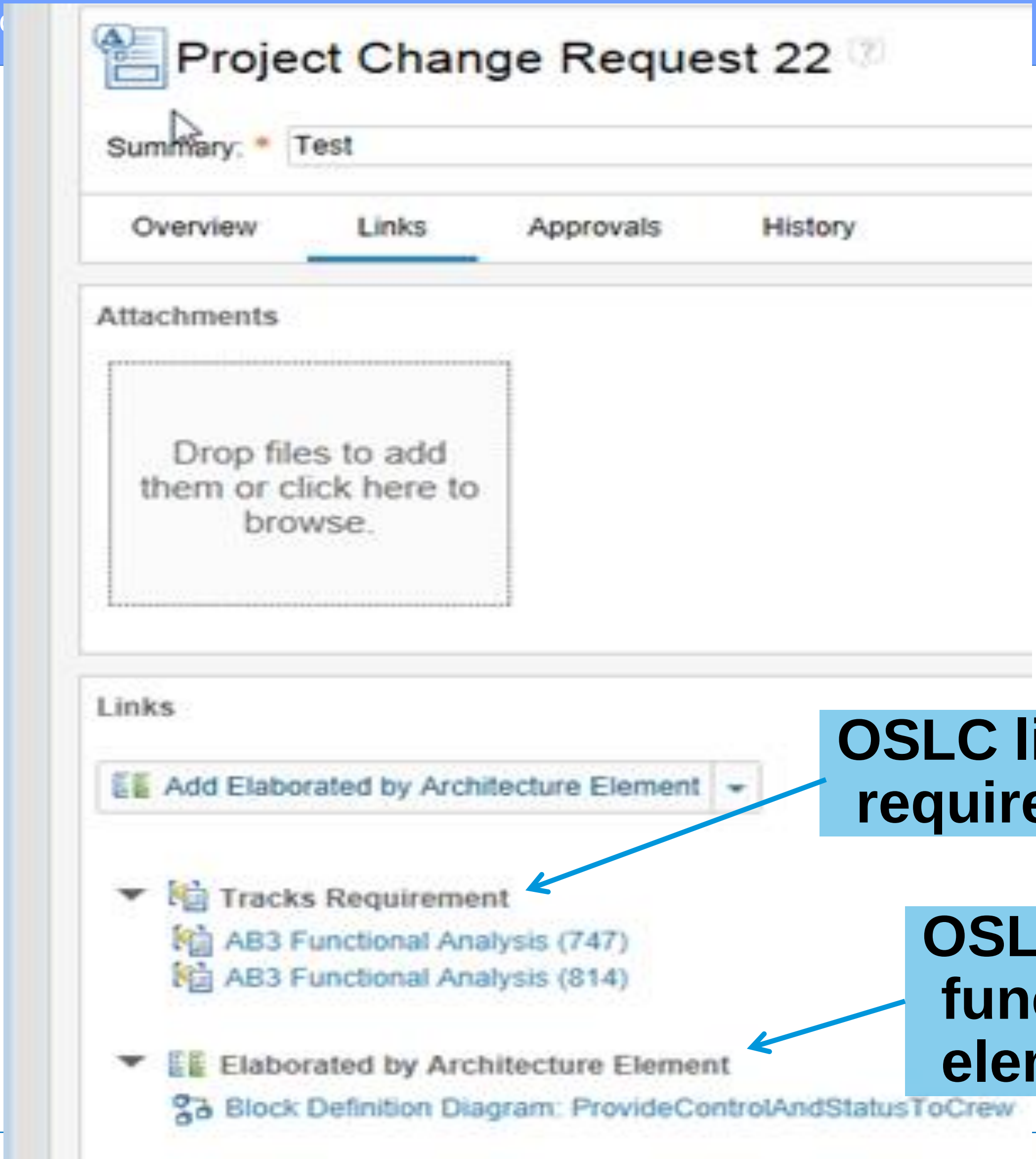
Auto-populated fields based on RTC project properties

Program phase the change was identified

Free text description of the change details

RTC: Change Request Form - Linking

Gl



Project Change Request 22

Summary: * Test

Attachments

Drop files to add them or click here to browse.

Links

- Tracks Requirement
  - AB3 Functional Analysis (747)
  - AB3 Functional Analysis (814)
- Elaborated by Architecture Element
  - Block Definition Diagram: ProvideControlAndStatusToCrew

OSLC linked impacted requirements

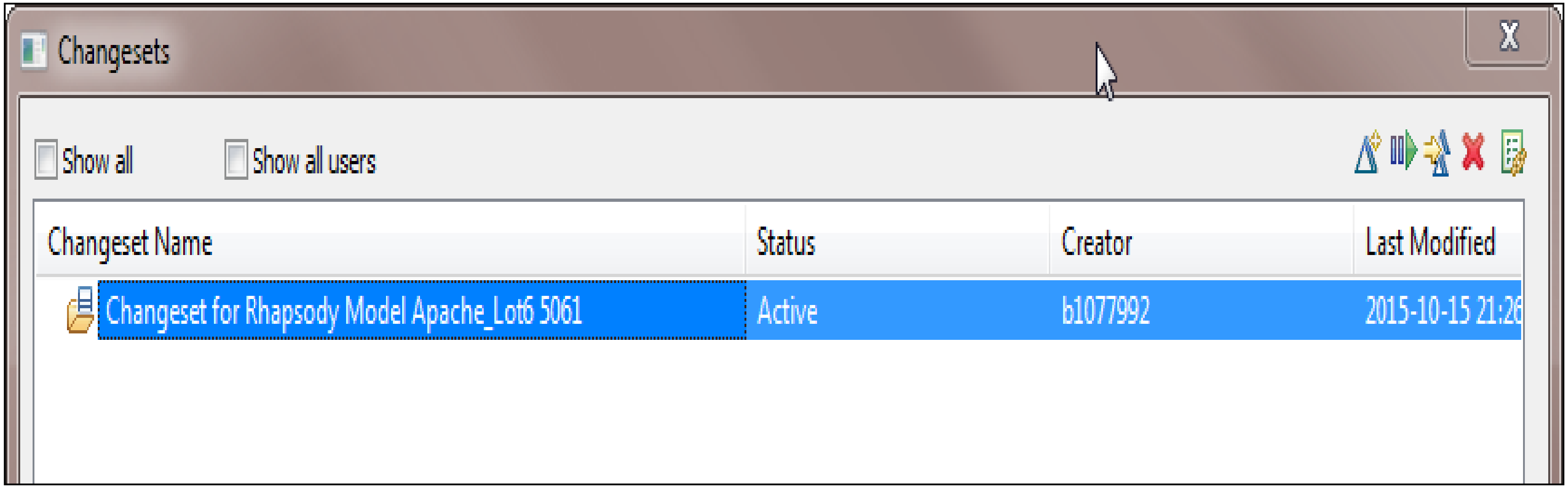
OSLC linked impacted functional architecture elements

RTC: Change Request Form Fields

- **Type**, **Creation Date**, **Created By**, **Project Area**, and **Team Area**:
  These fields are predefined and will already be populated when the CR is initiated. The **Project Area** is the RTC project containing the change.
- **Owned By**:
  Select the name of the person who is responsible for implementing the requested change.
- **Severity:**
  Select Minor, Normal, or Major to indicate the importance of the change.
- **Priority:**
  Select Low, Medium, or High to indicate how urgent the change needs to be incorporated.
- **Program:**
  Check the box of the program that this requested change will impact. Since each CR is written for a specific program, only one box should be checked here.
- **Work Product**:
  This is the area against which the change is being recorded. Check only one box here since each CR is written for one specific work product.
- **Phase:**
  Select System Design, Subsystem Design, Detailed Design, Production, Integration, or Verification. This is the phase in the lifecycle of the project, and only one box should be checked here.
- **Description:**
  Enter a detailed description of the change being requested.

Design Manager: Changesets

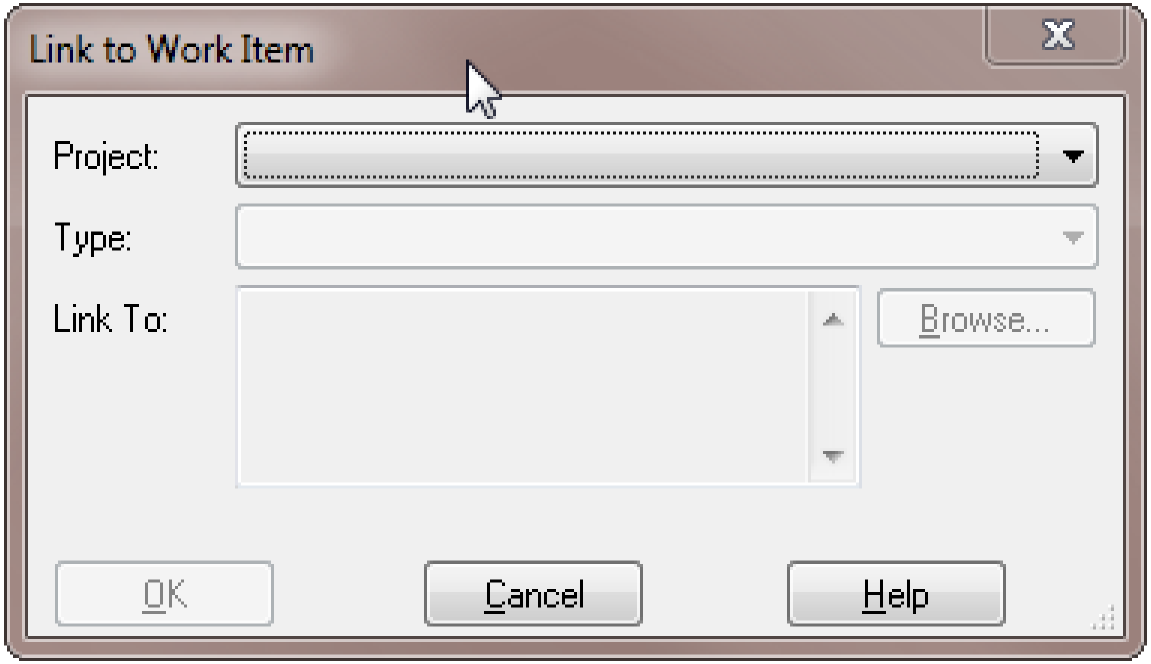
- **Automatically created upon editing the model**
- **Accumulate changes over multiple sessions**
- **Each Changeset can have exactly one user (owner)**
- **Users may own one or more Changesets**



Changeset Name	Status	Creator	Last Modified
Changeset for Rhapsody Model Apache_Lot6 5061	Active	b1077992	2015-10-15 21:26

Design Manager: Changeset Linking

- **Design Manager can link Changesets to Change Request**
  - **Provides Traceability**
  - **'Link to Work Item': changeset option in Rhapsody model**

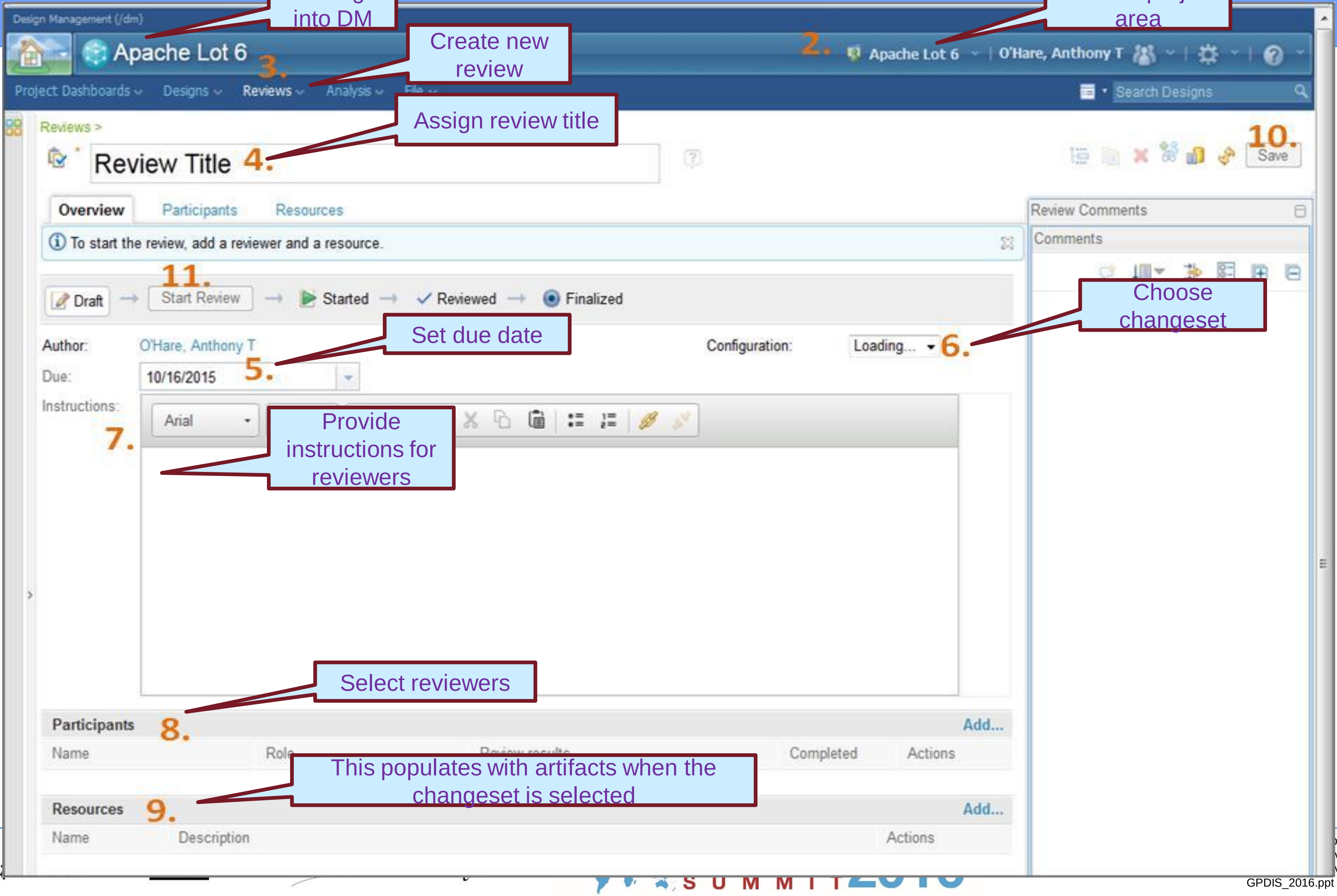


Design Manager: Peer Reviews

- **Peer Review of Model Edits**
  - **Requires Change Request to be Approved to Implement**
  - **Model Elements have been updated**

- **Design Manager allows the peer review owner to specify a Changeset to review**
  - **Automatically populates model elements**
  - **Allows other users to view pending changes**

Design Manager: Peer Review Form



1. Log into DM

2. Choose project area

3. Create new review

4. Assign review title

5. Set due date

6. Choose changeset

7. Provide instructions for reviewers

8. Select reviewers

9. This populates with artifacts when the changeset is selected

10.

11.

Review Title

To start the review, add a reviewer and a resource.

Author: O'Hare, Anthony T

Due: 10/16/2015

Configuration: Loading...

Instructions:

Participants

Name	Role	Review results	Completed	Actions

Resources

Name	Description	Actions

RTC: Change Request Closure

- **Two possible outcomes for Change Request submitted to SERB for closure:**

Approved: CR state advanced to “CLOSED”

- Changeset is incorporated and model is updated

Rejected: CR state is reverted to “Implement”

- The CR owner must complete the Implementation process correctly
- make any adjustments requested
- Resubmit CR to SERB for closure review

Agenda

Global Product Data Interoperability Summit | **2016**

- Background – What does OSLC and the Jazz platform provide for development activities
- Motivations for Apache Systems Engineering
- Platform Enablers
- Change Management Process
- Lessons Learned and Next Steps
- *Summary*

Advantages of Change Process & Tools

- **Process provides consistency for changes made to the functional architecture model**
- **Electronic history of changes made to functional architecture database**
- **OSLC enables creating traceability from a change request record to impacted model elements/requirements**
- **Opportunity for proposed changes to be reviewed and authorized before implementing**
- **In-Process changes are hidden from other users to avoid using incorrect or incomplete data**
- **Web based tools support geographically diverse teams working on common model**

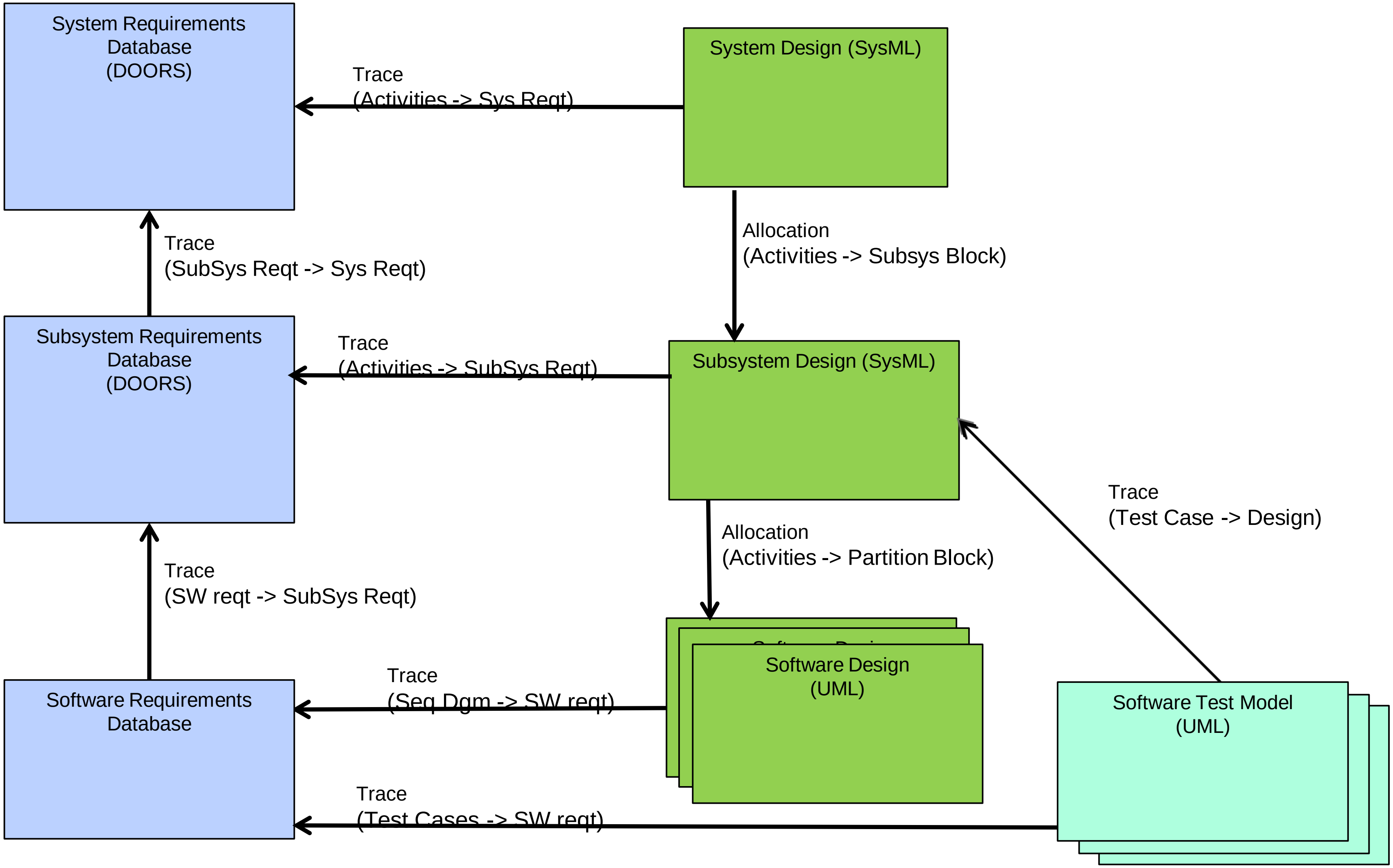
Lessons Learned

- **Process**
  - Keep Change Request form and process steps minimal
  - Leverage existing processes for managing changes to other work products to help team adapt to new process for managing functional architecture
- **Tools**
  - Need to deliver changes from Rhapsody rather than the DM client in order to avoid inconsistencies and to take advantage of the DiffMerge capability for conflicting changes.
  - Access Control Consideration
    - Each project area is completely readable by all users
    - In 6.0.2
      - Will be able to define teams with different write access to different parts of the model
      - Will be able to assign specific permissions to users to deliver changes with and across streams.
  - **Consider separation of project data (can you use one RTC project to support all DM projects?)**

Next Steps

- **Transition from DOORS 9.X + DWA to DOORS Next generation (DNG)**
  - Seamless requirements change process transition when DNG switchover occurs
  - Addresses limitations when linking from DOORS object to Rhapsody model currently experienced in DOORS + DWA
- **Design Manager 6.0.2**
  - Will be able to define teams with different write access to different parts of the model (addresses access control lesson learned on previous slide)
  - Will be able to assign specific permissions to users to deliver changes with and across streams.
  - Expand use of parallel configurations (streams and baselines) to manage work and develop PL assets
- **Organize multi-discipline models (i.e. testing and software engineering) to expand change process and traceability (next slide)**

Next Steps (Cont'd): Model Management for Multiple Disciplines



System Requirements Database (DOORS)

System Design (SysML)

Trace (Activities -> Sys Reqt)

Trace (SubSys Reqt -> Sys Reqt)

Allocation (Activities -> Subsys Block)

Subsystem Requirements Database (DOORS)

Trace (Activities -> SubSys Reqt)

Subsystem Design (SysML)

Trace (Test Case -> Design)

Allocation (Activities -> Partition Block)

Trace (SW reqt -> SubSys Reqt)

Software Design (UML)

Software Requirements Database

Trace (Seq Dgm -> SW reqt)

Software Test Model (UML)

Trace (Test Cases -> SW reqt)

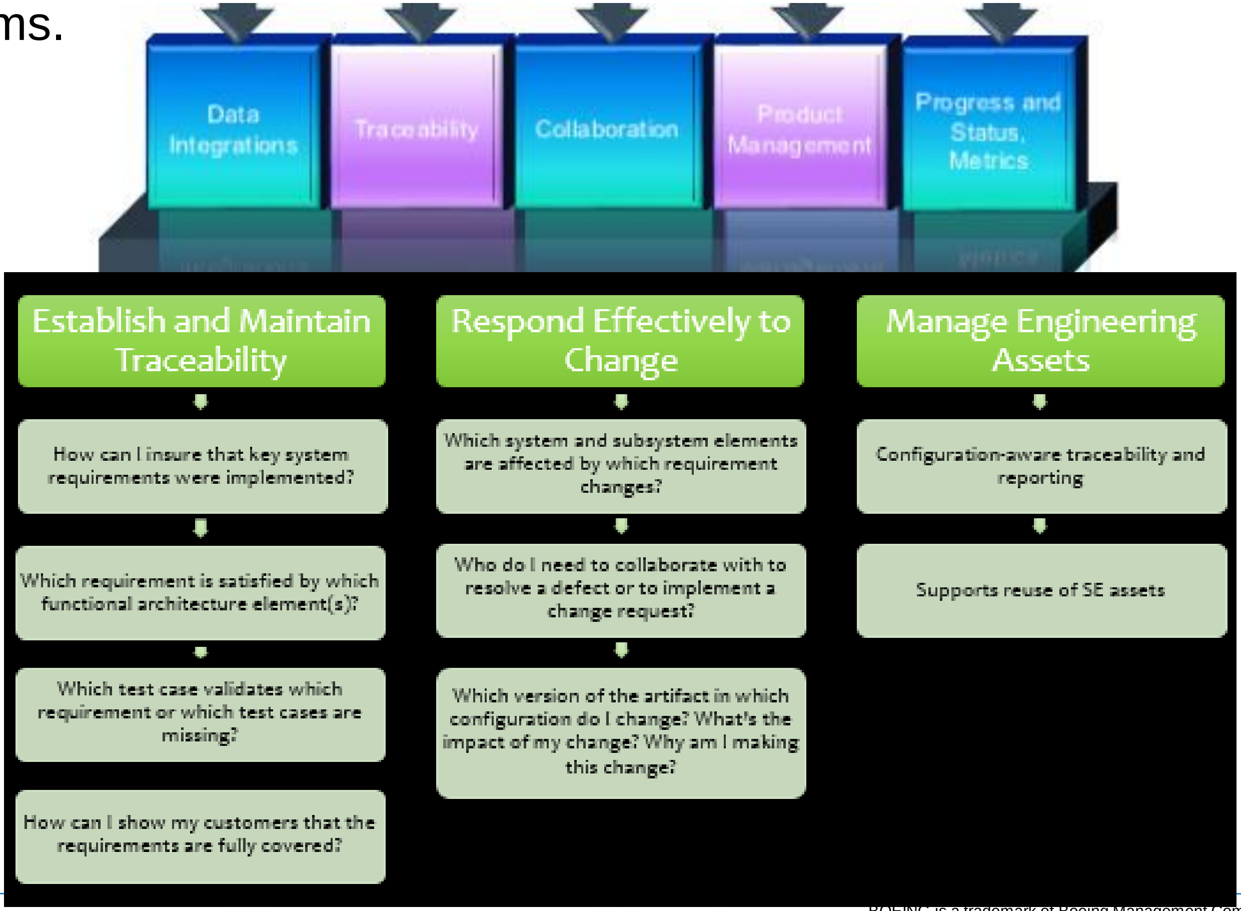
Agenda

Global Product Data Interoperability Summit | **2016**

- Background – What does OSLC and the Jazz platform provide for development activities
- Motivations for Apache Systems Engineering
- Platform Enablers
- Change Management Process
- Lessons Learned and Next Steps
- *Summary*

Summary

- **Projects adopting this platform and approach** are addressing some of the key integration challenges mentioned at the outset using open interoperability standards on production Jazz development platforms currently available to Boeing engineering teams.



Establish and Maintain Traceability

How can I insure that key system requirements were implemented?

Which requirement is satisfied by which functional architecture element(s)?

Which test case validates which requirement or which test cases are missing?

How can I show my customers that the requirements are fully covered?

Respond Effectively to Change

Which system and subsystem elements are affected by which requirement changes?

Who do I need to collaborate with to resolve a defect or to implement a change request?

Which version of the artifact in which configuration do I change? What's the impact of my change? Why am I making this change?

Manage Engineering Assets

Configuration-aware traceability and reporting

Supports reuse of SE assets