From www.eclipse.org/osee to deployment
Tutorial

Ryan Brooks
Donald Dunne
Roberto Escobar

Boeing
Mesa, AZ

 March 22, 2009 |

Agenda

- Installation & Setup
- OSEE
  - What is it?
  - Background Information
- Architecture
  - Network Layout
- T. Rex Software Company
- Data Model
- Version Control
- Variant Management
- Break

Before we begin…

- This is an interactive session: feel free to ask questions
- Tell us about yourself
  - Background Info
  - Operating Systems
- Tutorial based on OSEE 0.7.0

Requirements

- System Requirements (non-eclipse)
  - System with at least 1GB of RAM
  - Java Runtime Environment (JRE) 1.6 or higher
  - Microsoft Office (For Demo Only)

- Eclipse Dependencies
  - Eclipse 3.4.2 SDK
  - org.eclipse.gef
  - org.eclipse.draw2d
  - org.eclipse.birt
  - org.eclipse.datatools
  - The easiest solution is the Ganymede Eclipse install Eclipse IDE for Java and Report Developers

Installation

- Database
  - Run the PostgreSql installer, located under the “PostgreSql” folder, for your OS using the following settings
    - Default install path
    - Default data directory
    - Password “Postgre1”
    - Port 5432
    - Default Locale
    - On last screen uncheck “Launch Stack Builder”
  - Setup database accounts and schemas by executing the bash/bat script for your OS located under the “PostgreSql” folder
    - Save db password using pgadmin
    - osee_db_setup.bat – Windows
    - osee_db_setup.sh – Others
  - For more information or for the files mentioned above visit
    - http://www.eclipse.org/osee/documentation/installation/postgresql_install.php

Installation - continued

- JRE
  - Ensure the JRE 1.6 is in the path by typing java –version at a command prompt
- OSEE Client
  - Extract the Eclipse base zip for your OS located under “Eclipse Base” to a short path and then launch eclipse
  - From the Eclipse update manager, install the update sites located under “OseeClient”
    - org.ecilpse.osee_integration_build_incubation.zip
    - osee.add.ons.updatesite.zip

Initialization

- Launch Application Server
  - Execute the launch script for your OS located under “OseeApplicationServer”
    - osee_app_server.bat – Windows
    - osee_app_server.sh - Others
  - Wait until the server finishes the start up procedure
    - Do not close the console
- Database Initialization
  - In a command prompt change to the eclipse install dir

```
eclipsec -application org.eclipse.osee.framework.database.configClient -vmargs -Xmx512m
-Dosee.log.default=INFO -Dosee.application.server=http://localhost:8089
-Dosee.authentication.protocol=trustAll -Dosee.prompt.on.db.init=false
-Dosee.choice.on.db.init="OSEE Demo Database"
```

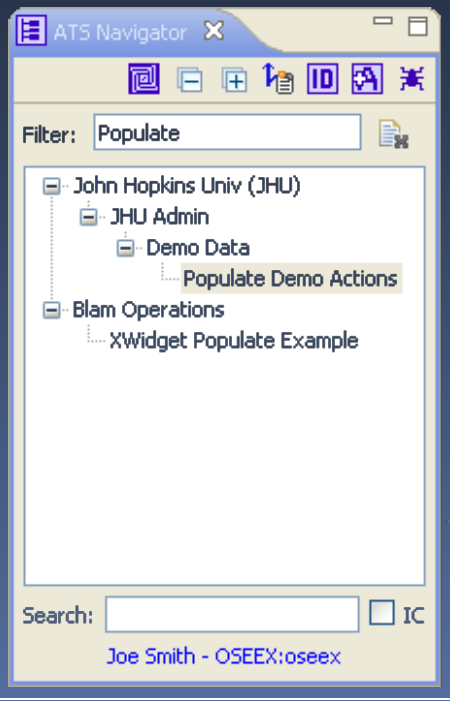
  - Once db init completes, type exit in the server console

Populate Demo Data

- Launch Application Server
- Launch OSEE Client

```
eclipse -vmargs -Xmx512m
-Dosee.application.server=http://localhost:8089
-Dosee.authentication.protocol=demo
```

- Switch to the ATS Perspective
  1. Window
  2. Open Perspective
  3. ATS
     - In the ATS Navigator Window
       - Type **“Populate”** in the Filter text box
       - Press **‘Enter’** to add filter
       - Double-click on the **“Populate Demo Actions” item**
       - **Wait for operation to complete**



What is OSEE?

- OSEE is a tightly integrated environment designed to support lean engineering principles across a product's full life-cycle in the context of an overall systems engineering approach.

Background

- OSEE began was first deployed to develop Boeing's next generation Apache Helicopter
- It provides
  - An integrated tool set
  - End-to-end traceability
  - Variant configuration management
  - Integrated workflows and processes
  - A Comprehensive issue tracking system
  - Deliverable document generation
  - Real-time project tracking and reporting
  - Validation and verification of mission software

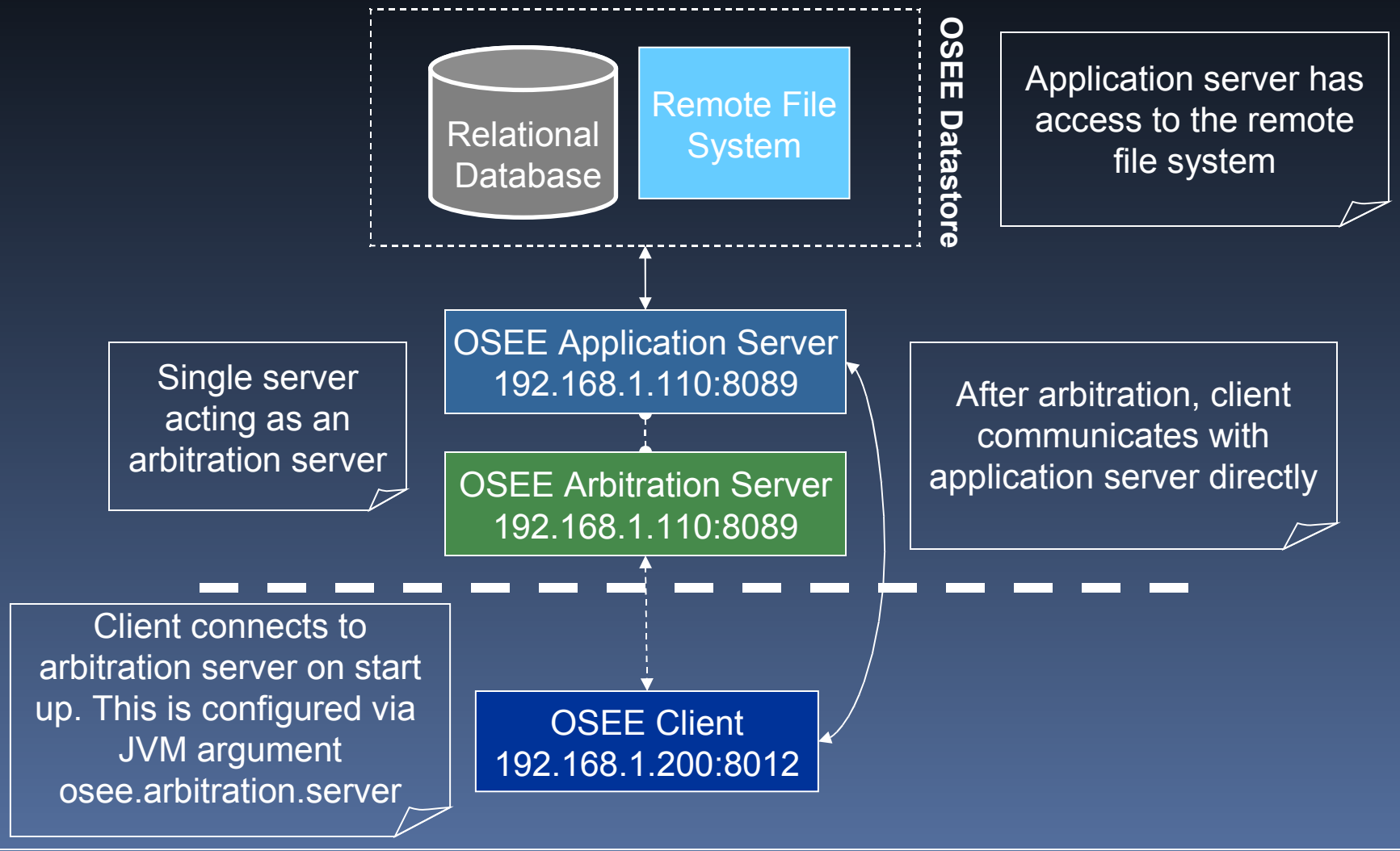
Background - continued

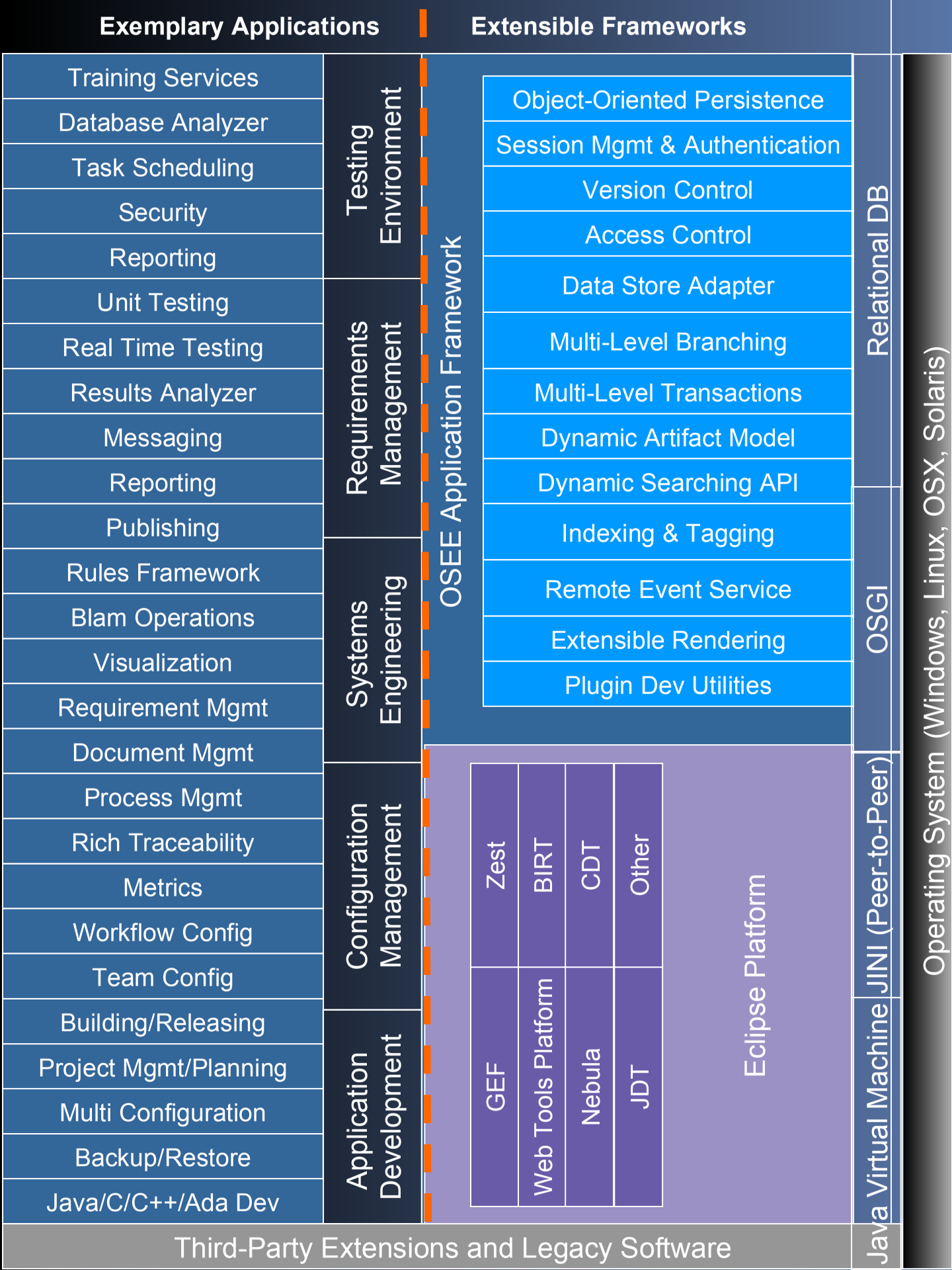
- As an eclipse project
  - Milestones
    - Initial source committed on Dec 8, 2007 (~140K LOC)
    - Project proposal approved on July 10, 2007
    - Test environment framework submitted Spring 2009 (37K LOC)
- OSEE is used to engineer itself

Architecture

- Initially OSEE was architected as a heavy client
  - Direct client-to-database interactions
- Migrating into a Thin-Client/Server architecture
  - Utilizing OSGI on server-side
  - Flexible deployment and maintenance
  - Address scalability and load management

Simple Network Layout

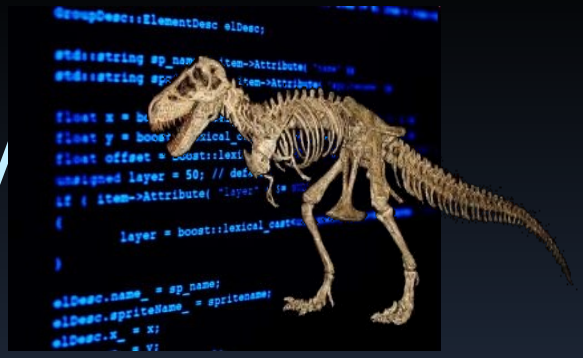



Exemplary Applications
Extensible Frameworks
Training Services
Database Analyzer
Task Scheduling
Security
Reporting
Unit Testing
Real Time Testing
Results Analyzer
Messaging
Reporting
Publishing
Rules Framework
Blam Operations
Visualization
Requirement Mgmt
Document Mgmt
Process Mgmt
Rich Traceability
Metrics
Workflow Config
Team Config
Building/Releasing
Project Mgmt/Planning
Multi Configuration
Backup/Restore
Java/C/C++/Ada Dev
Testing Environment
Requirements Management
Systems Engineering
Configuration Management
Application Development
OSEE Application Framework
Object-Oriented Persistence
Session Mgmt & Authentication
Version Control
Access Control
Data Store Adapter
Multi-Level Branching
Multi-Level Transactions
Dynamic Artifact Model
Dynamic Searching API
Indexing & Tagging
Remote Event Service
Extensible Rendering
Plugin Dev Utilities
Zest
BIRT
CDT
Other
GEF
Web Tools Platform
Nebula
JDT
Eclipse Platform
Relational DB
OSGI
JINI (Peer-to-Peer)
Java Virtual Machine
Operating System (Windows, Linux, OSX, Solaris)
Third-Party Extensions and Legacy Software

T. Rex Software Company

- 15 years in the business
- Developing software for medical applications
- Waterfall development cycle
- Isolated teams using disconnected tools to track issues
  - Requirements - using spreadsheets
  - Code – problem change report database
  - Test – spreadsheets and emails
- Status is reported weekly via emails to team leads who then flow information to project managers

T. Rex Software Company



- Inefficiencies/Cost
  - Software license fees
  - Steep learning curve
  - Data redundancies
  - Weak knowledge management
  - Issues fall through the cracks
  - Poor Planning
  - Lagging metrics

SAW-TSR Project

- Project with many challenges
- Must develop a Surgical Assistant Workstation for Tele-operated Surgical Robots (SAWTSR)
- Project was 40% under funded
- Stringent requirements on software quality
  - Medical application
  - Severe consequences of faulty software
- Project must meet a company wide initiative to streamline processes
- If T. Rex does not make a change, the contract will be lost

Why should T. Rex use OSEE?

- Full life-cycle engineering environment
- Open source extensible platform
- Benefits of eclipse community
- Tightly integrated
  - Common data model
  - Version control
  - Change management
  - Workflows and processes
  - Supports multiple databases
- Zero license cost

Company is ready to evolve!

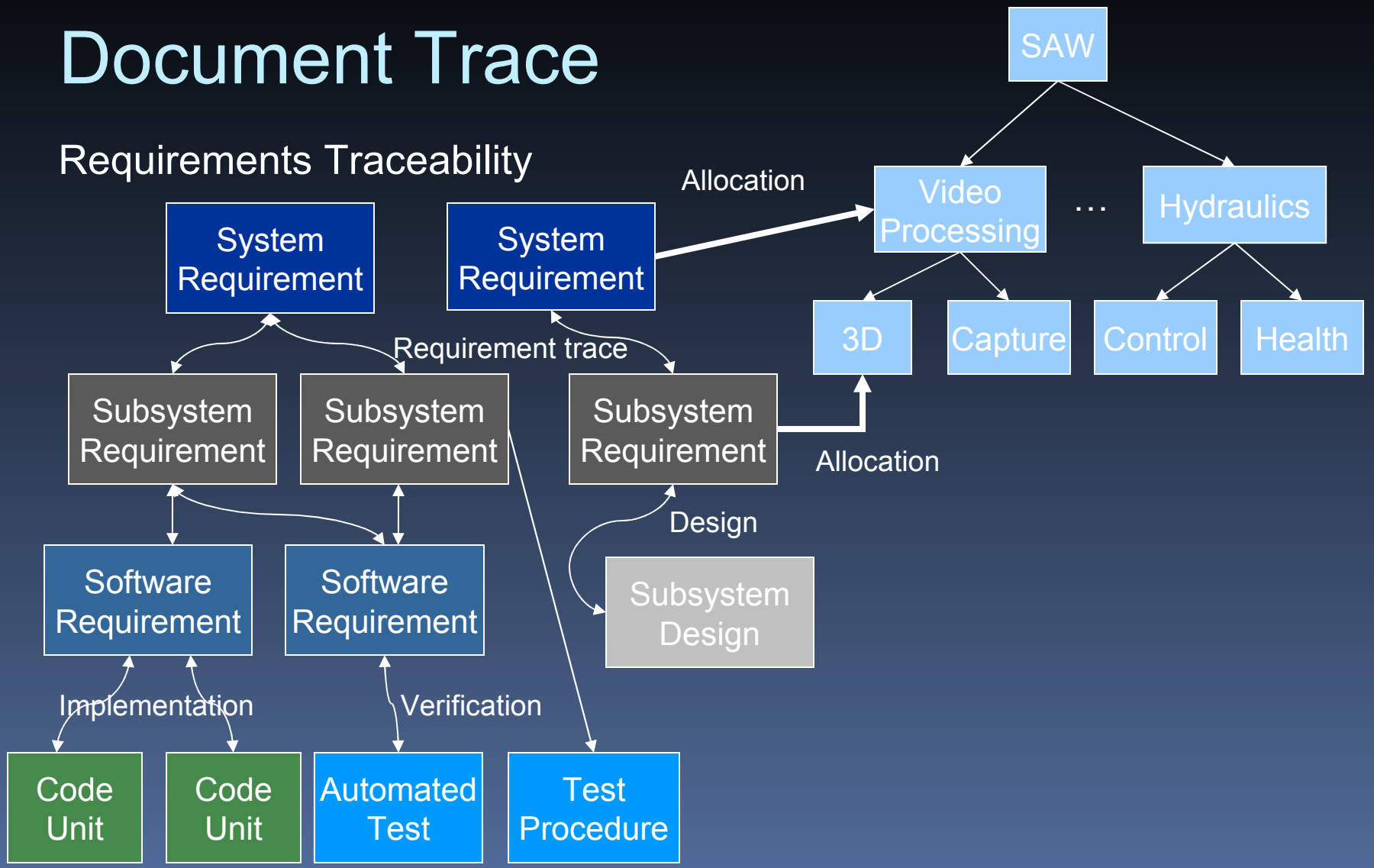
Management decides to use OSEE

Where do we begin?

- Analyze data produced during life-cycle
  - Project produces various files
    - System Level Requirements
    - Software Level Requirements
    - System Requirements
    - Source Code
    - Test Source
    - Test Results
- Determine how data relates to one another

Document Trace

Requirements Traceability



Understanding the OSEE Data Model

- Artifact
  - Main OSEE data object consisting of attributes
  - Artifact type - blue print for instance creation
  - Artifact types can inherit from one another
  - All artifacts inherit from “Artifact” (similar to Object in Java)

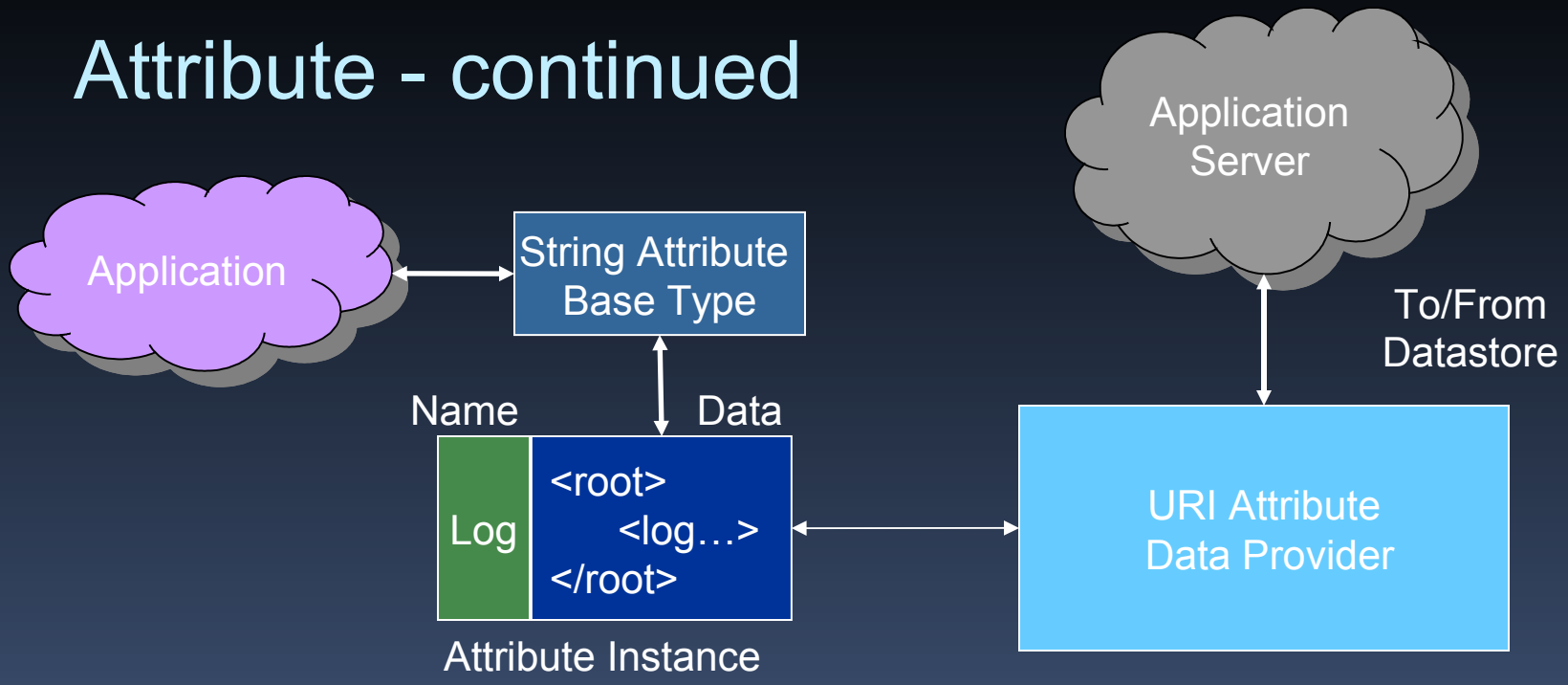
Attribute

- Key / Value pair representing a single data element
- Attribute type - blue print for instance creation
- Attribute base type is used to convert raw data into a native type or other object
- Attribute base types
  - String
    - Word Templated Content
    - Word Whole Document
  - Date
  - Boolean
  - Integer
  - Floating Point
  - Enumeration

Attribute - continued

- Attribute backing data managed by an attribute data provider
  - Transfers data between client/server
  - Can be extended to serve data from outside OSEE data store
- Attribute type also defines whether the attribute is searchable and how to tag the data

Attribute - continued

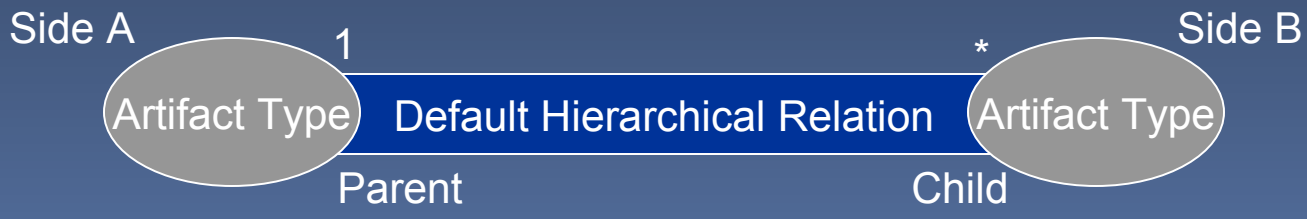


Attribute Name: Log
Attribute Base Type: String Attribute
Attribute Data Provider: URI Attribute Data Provider
File Extension: xml
Default Value: <root></root>
Validity XML:
Min Occurrence: 1
Max Occurrence: 1
Tip Text: Contains log information

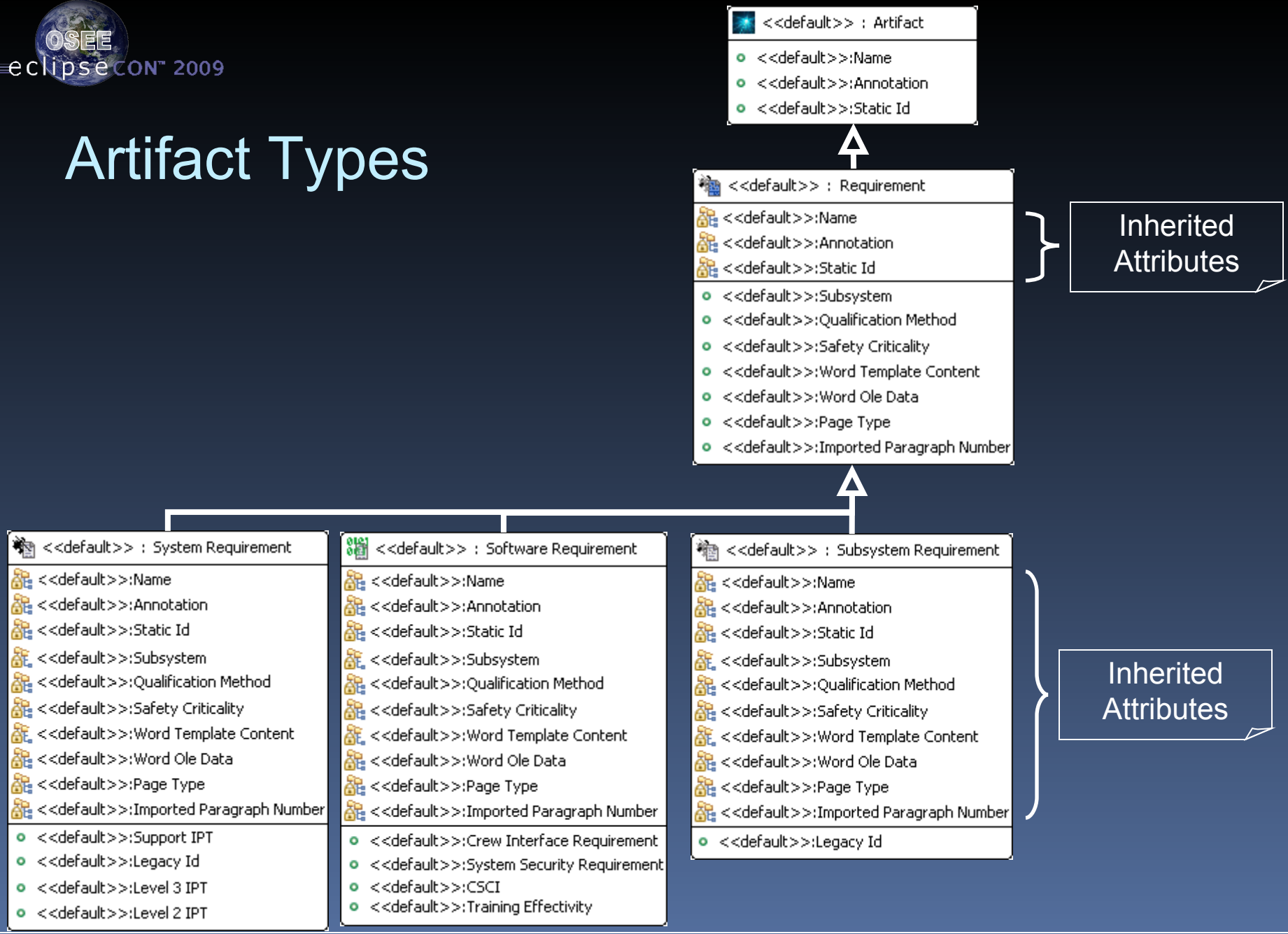
Attribute Type

Relation

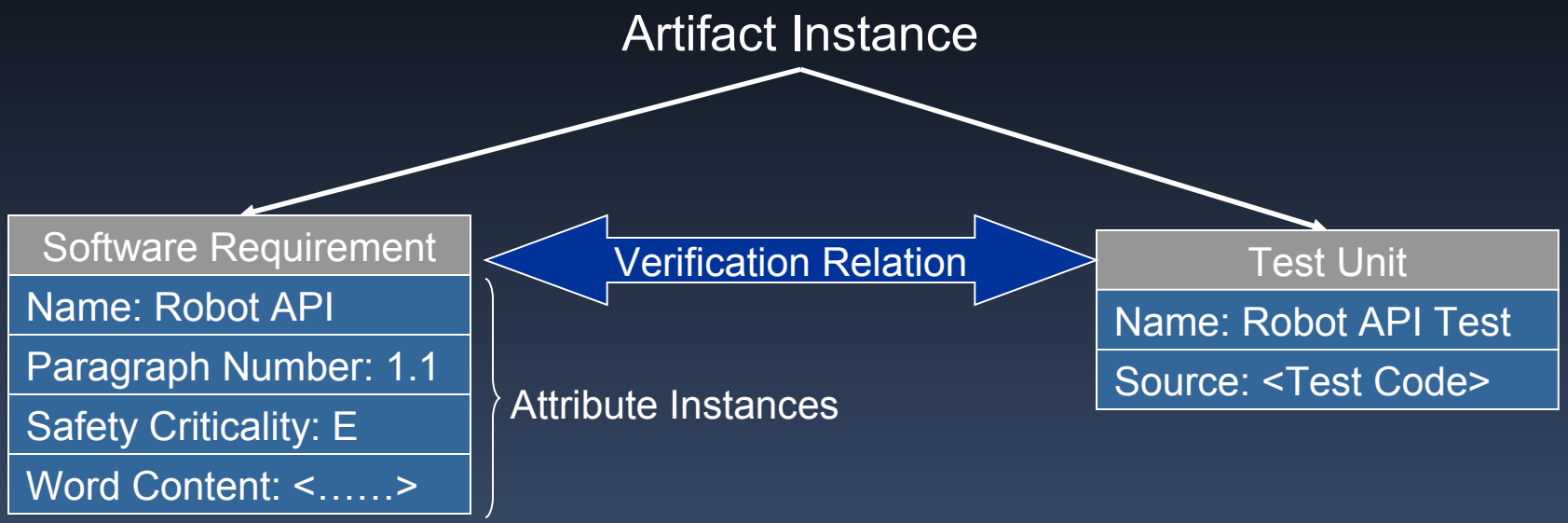
- Relates two or more artifact types
- Relation type - blue print for instance creation
  - Relation type
    - What type allowed on side A
    - What type allowed on side B
    - Multiplicity (1 to 1, 1 to Many, Many to Many)



Artifact Types



Artifacts



Version Control

- Changes to artifacts, attributes, and relations are tracked by the system
- Changes are managed by a transaction based version control system using fine grained change identification
- Data managed under branches

Transaction

Transaction: 1000019
Timestamp: Mar 2, 2008
Author: Joe Smith

My Requirement

Transaction: 1000020
Timestamp: Mar 3, 2008
Author: Alex Kay

My Requirement

My Code unit

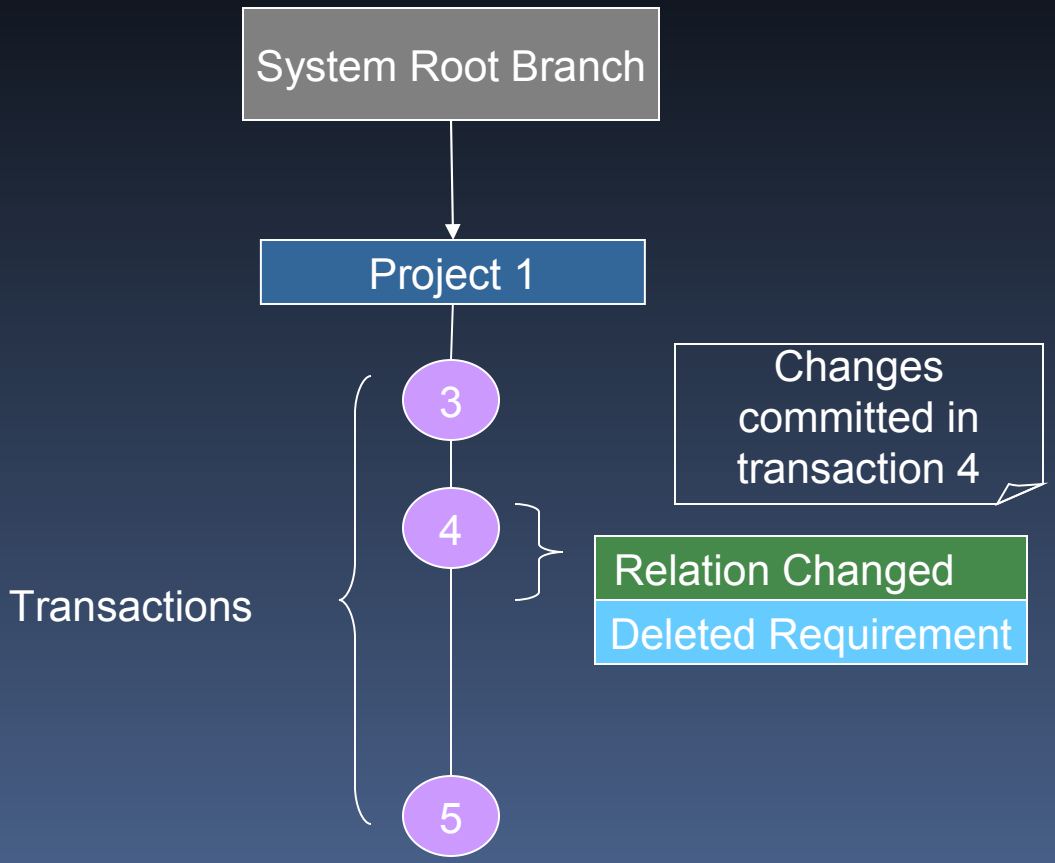
My Test unit

Transaction: 1000021
Timestamp: Mar 4, 2008
Author: Joe Smith

My Requirement

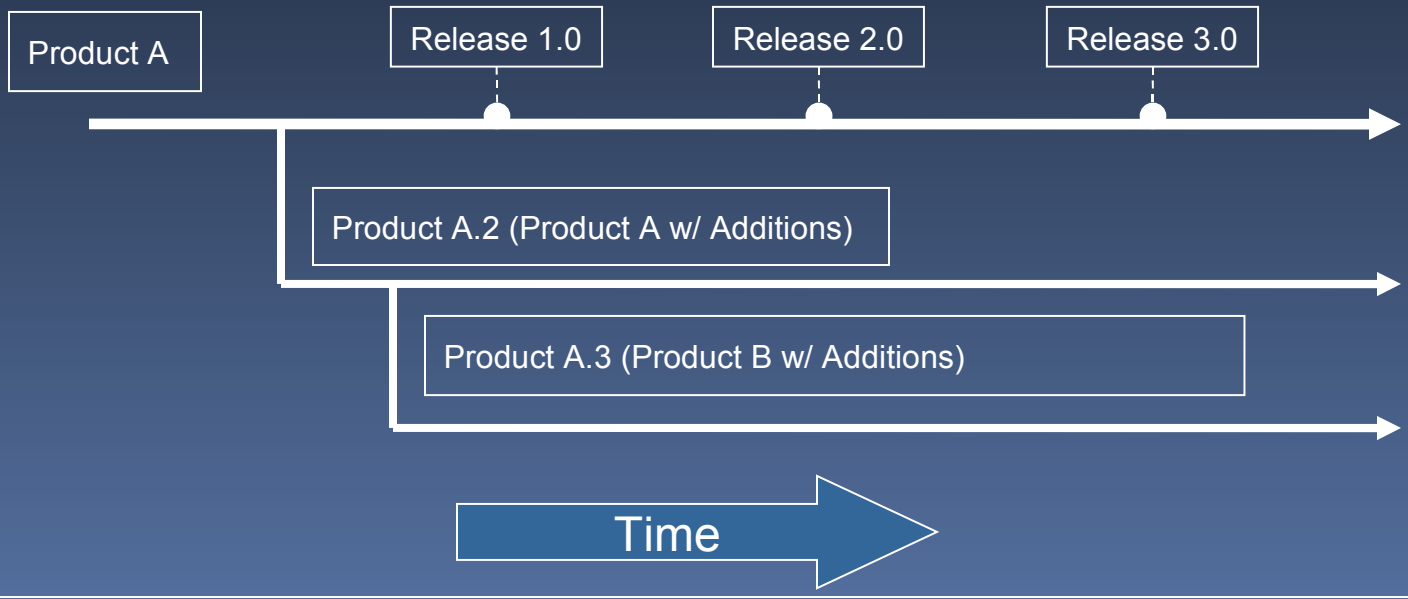
My Code unit

Branch



Variant Management

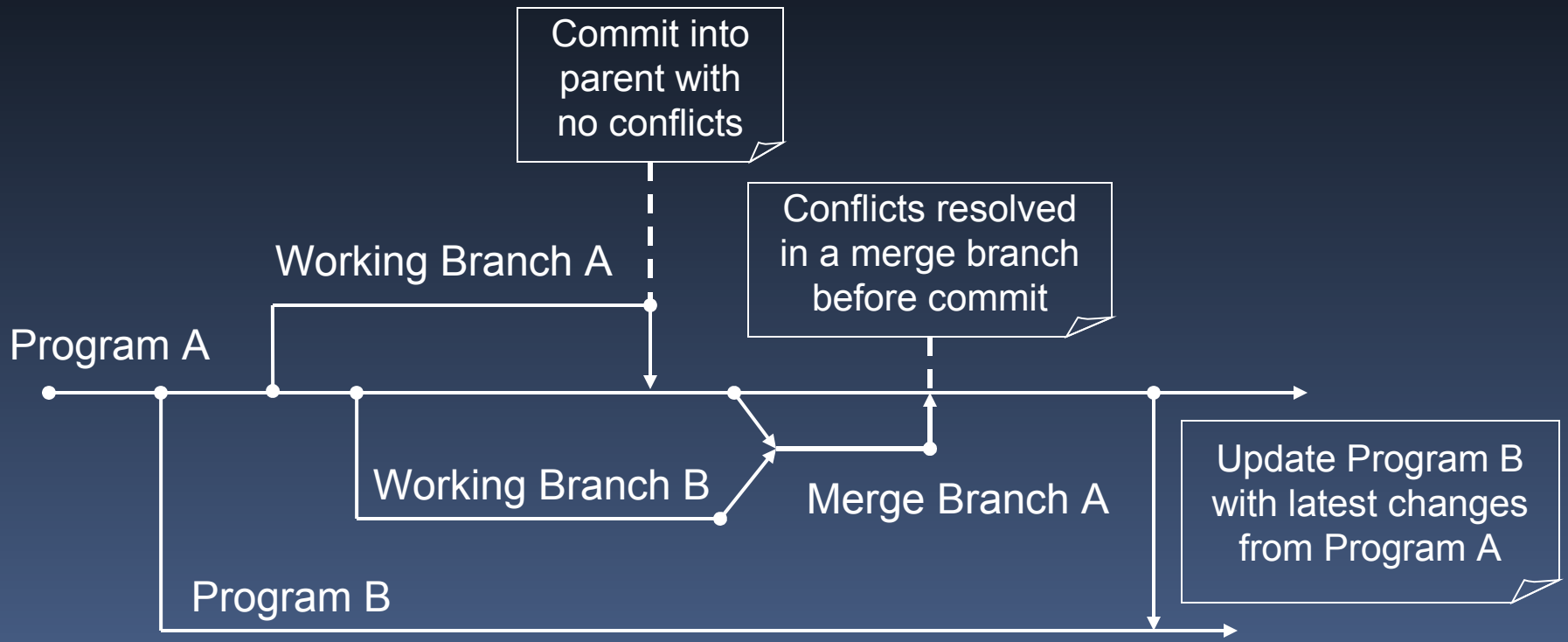
- Product lines share common baseline data throughout all variants
- Changes to the baseline product can be easily merged to the variant product lines



Branching

- Branching
  - Used to create a variant of the parent branch
  - Updates can be performed to obtain changes from the parent branch into the child
- A variant branch can be committed back into its parent
  - Change conflicts are resolved via merging

Branching - continued



BREAK

From www.eclipse.org/osee to deployment
Tutorial (Part II: All in a days work)

Ryan Brooks
Donald Dunne
Roberto Escobar

Boeing
Mesa, AZ

 March 22, 2009 |

Agenda

- T. Rex with OSEE
  - Products & Teams
  - Variants
  - Scenario Roles
  - Project Workflows
- Scenario: All in a days work
  - Search Requirements
  - Create Action
  - Requirement Team Workflow
    - Add Decision Review
    - Change Implementation
  - Code Team Workflow
    - Add Tasks
  - Test Team Workflow
    - Privileged Edit
    - Status
  - Create Peer-To-Peer Review
  - Add new Workflow

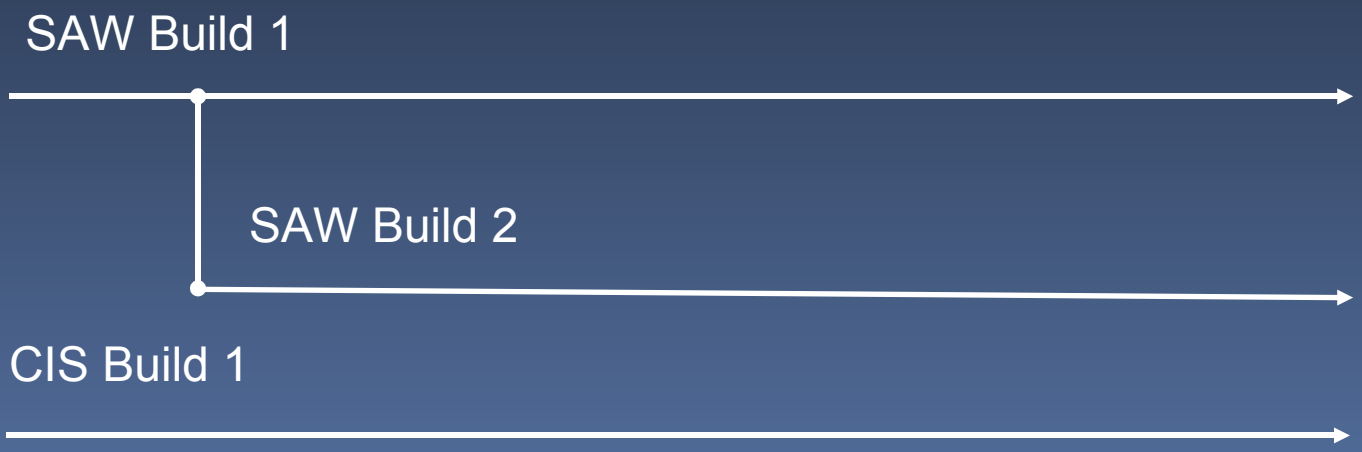
Project has been using OSEE for a couple of months now

Products & Teams

- SAW – Surgical Assistant Workstation –
  - **Teams:** Requirements, Code, Test, HW, SW Design
- CIS – Dummy Project
  - **Teams:** Requirements, Code, Test, SW Design
- Facilities Team
- IT Team
  - Computers, Backups, Network
- Tools Team
- Website Team
- Processes Team

Project Variants

- Surgical Assistant Workstation - SAW
  - Build 1
  - Build 2
- CIS Build 1 – Dummy Project
- Work is performed in working branches which are then committed back into their respective parent branch

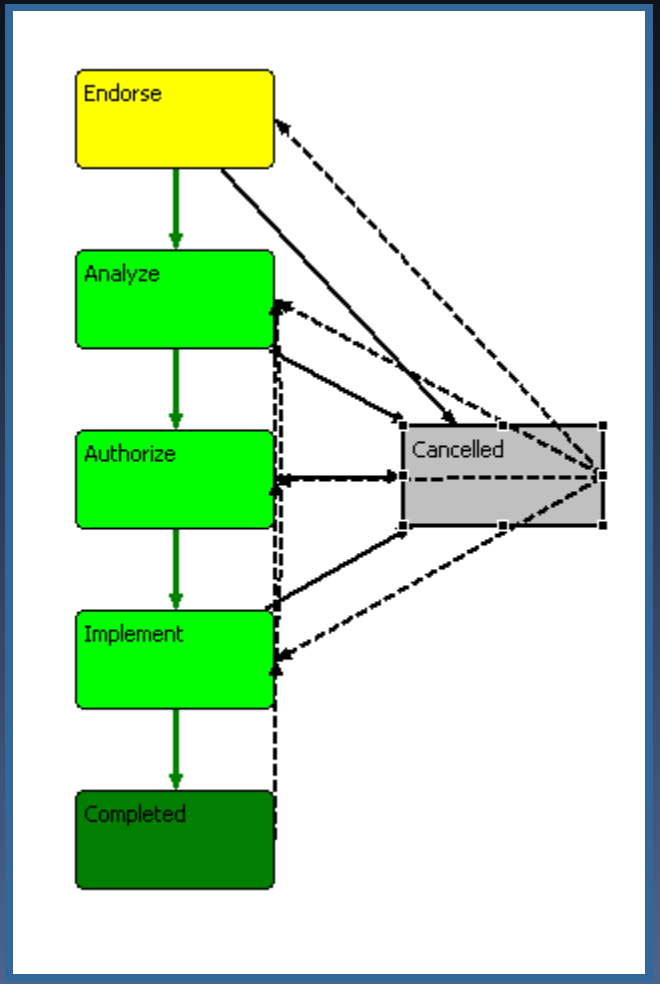


Roles

- Project Manager
  - Requirement
    - Lead
    - Developer
  - Code
    - Lead
    - Developer
  - Test
    - Lead
    - Developer

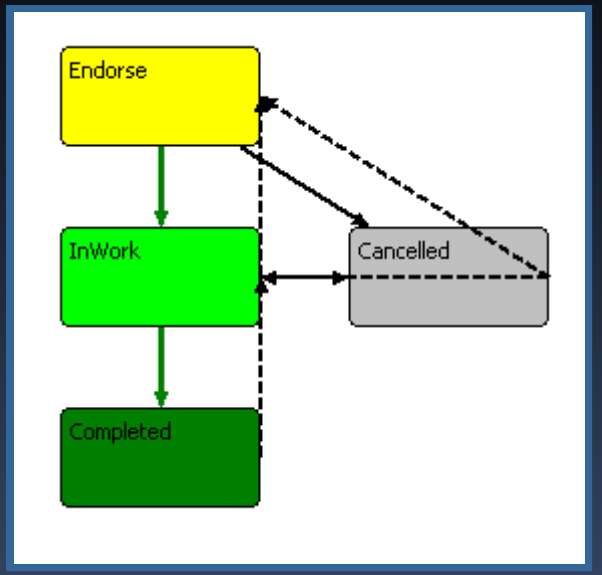
Workflow 1

- Requirements
- Code
- Test
- SW Design
- Tools
- Process



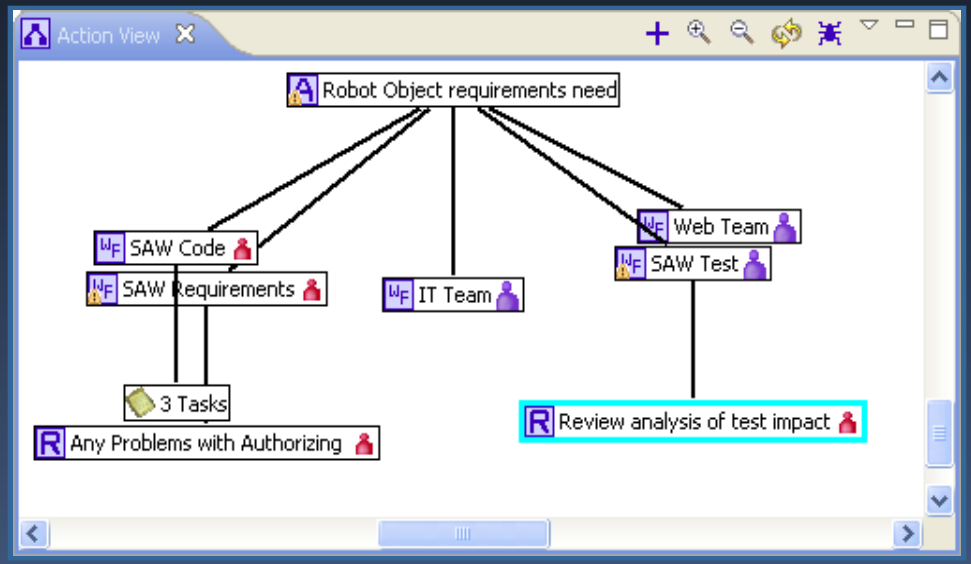
Workflow 2

- Web
- Facilities



Scenario

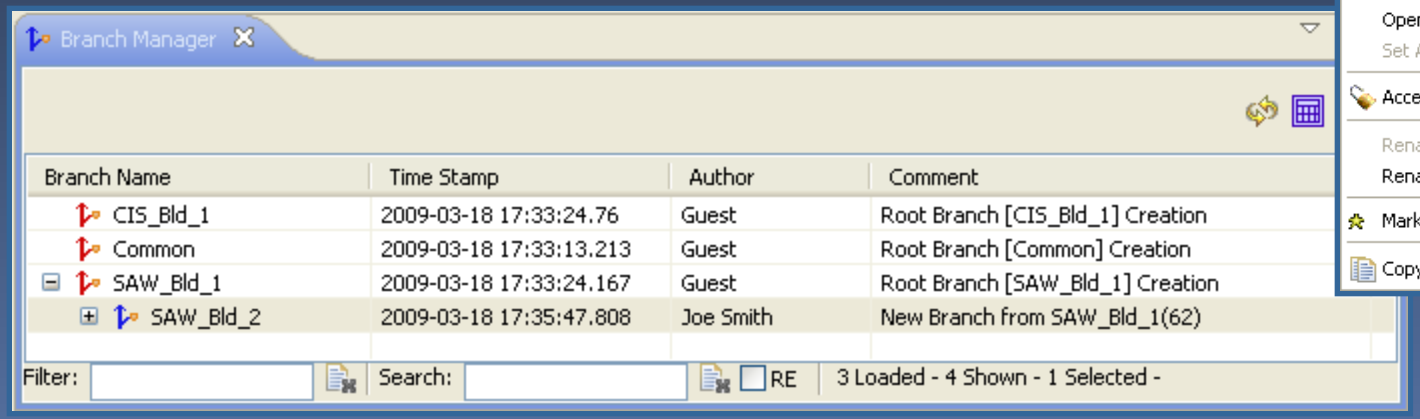
- Joe Smith finds a problem in a requirement impacting the following teams:
  - Code
  - Test
  - Website
  - IT

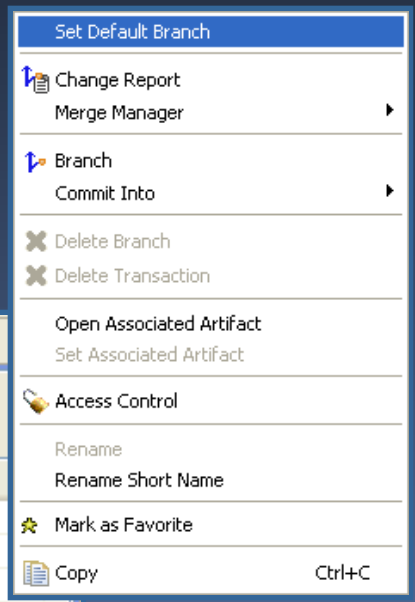


- Reviews needed:
  - Decision Review off Code Team Workflow
  - Peer Review off Test Team Workflow
- Perform tasks off Code Team Workflow

User searches for a requirement

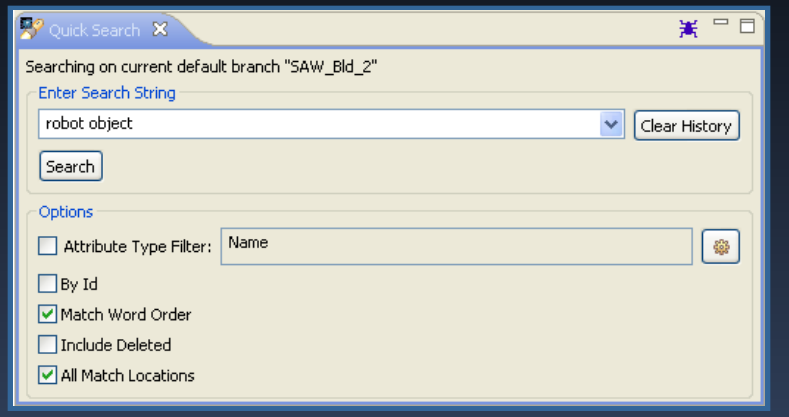
- Switch to the Define perspective
  - Window->Open Perspective->Define
- Set “SAW_Bld_2” as the default branch
  - Click on the Branch Manager View
    - If the view is not open
      - Select Window->Show View->Branch Manager
  - Right-Click on “SAW_Bld_2”
  - Select “Set Default Branch”

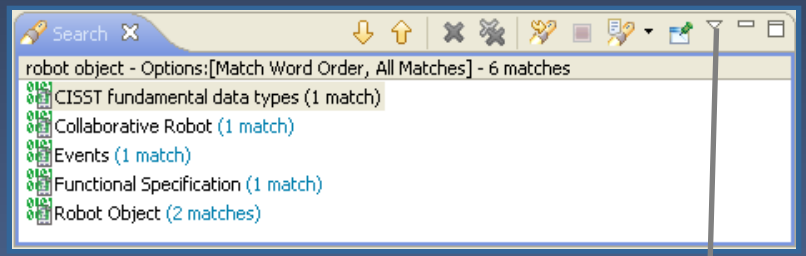




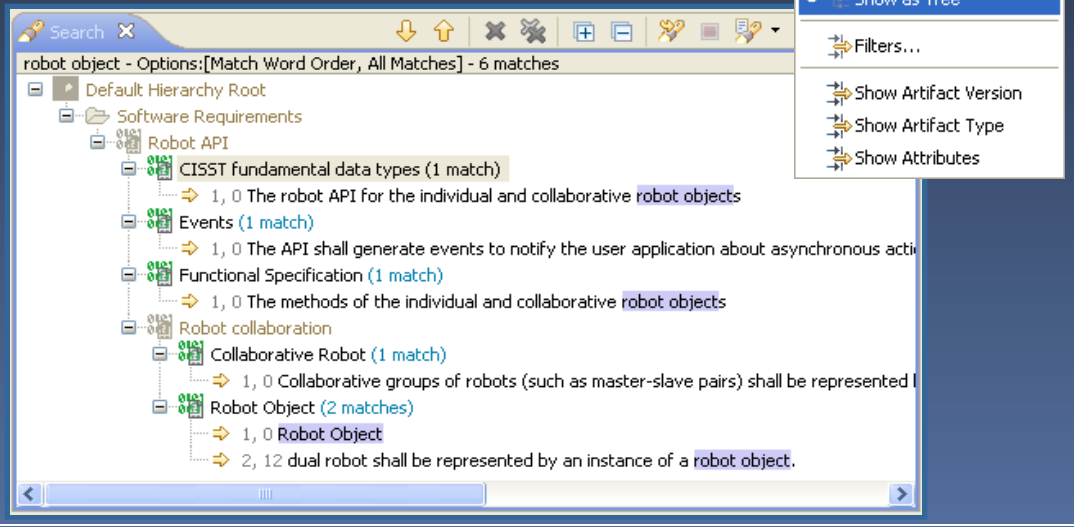
User searches for a requirement

- Search for Item
  - Click on the Quick Search View
  - Enter “robot object” in the search string text box
  - Check the “Match Word Order” option
  - Check the “All Match Locations” option
  - Click the search button
- Search results as list or tree view with match locations





Select View

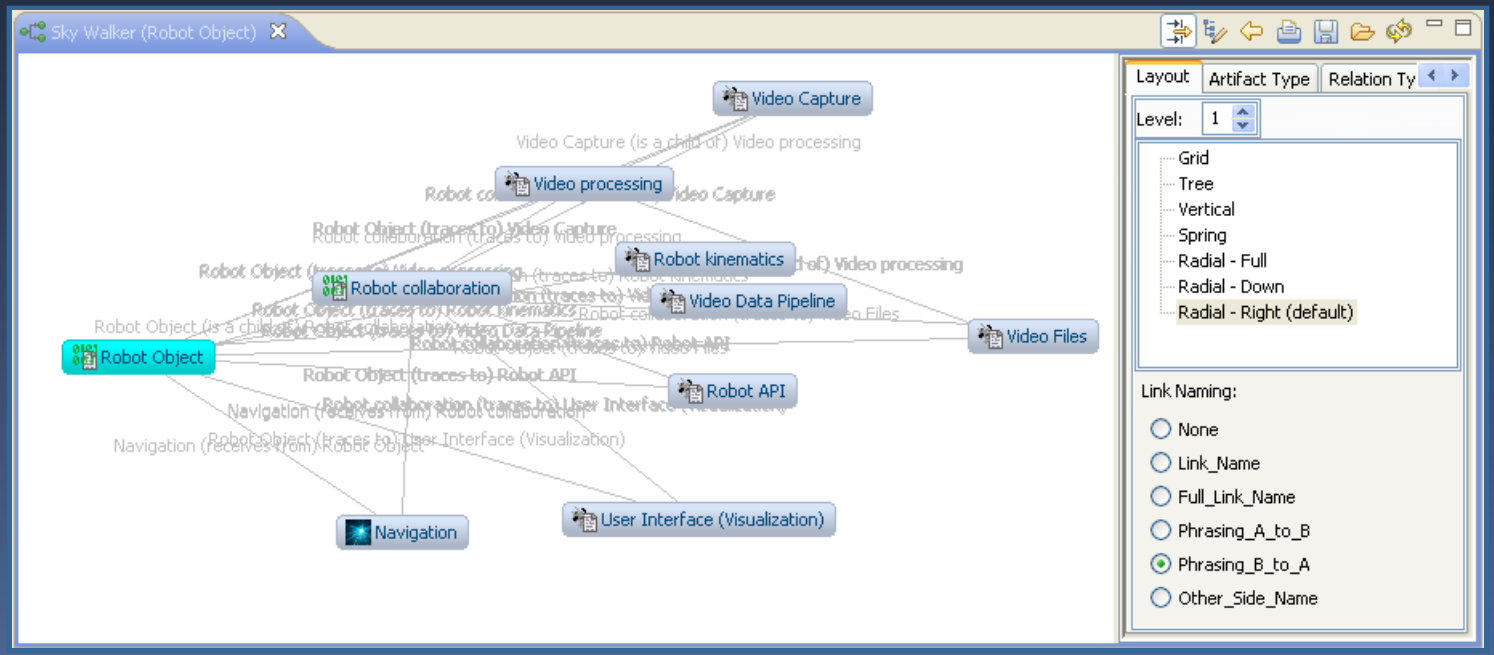


User explores the robot object requirement

- From the search results view
  - Right-Click on the “Robot Object” software requirement
  - Select each of the following from the pop-up menu individually
    - Reveal Artifact in Explorer
    - Resource History
    - Open With Artifact Editor
      - Click on the attribute tab
      - Click on the relation tab
    - Open With MS Word Preview
    - Sky Walker

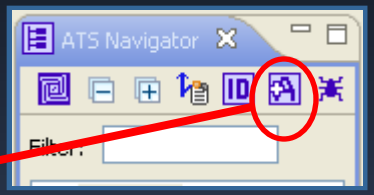
User explores the robot object requirement

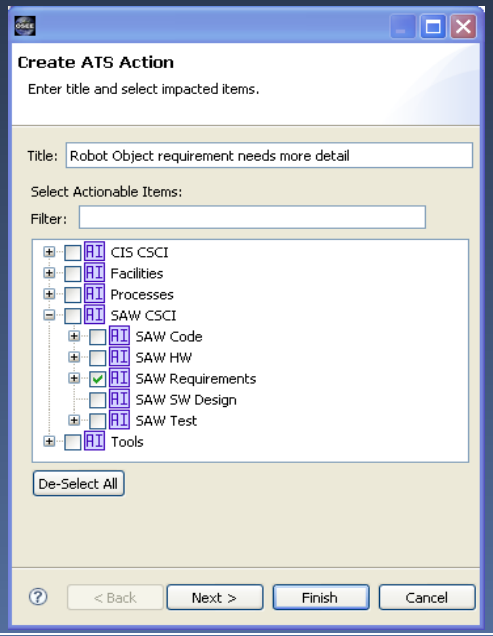
- Robot Object (software requirement) relates to other artifacts



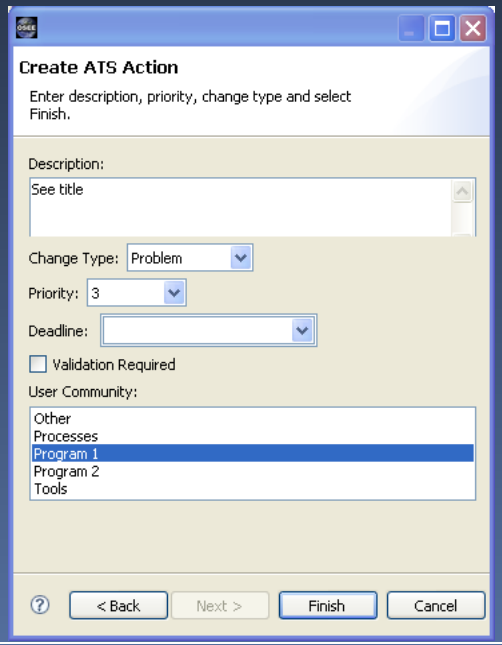
User creates an action

- Create an action against the “Robot Object Requirement”
- Switch to the ATS Perspective
  - Window->Open Perspective->ATS
- Click on the New Action icon in ATS Navigator
  - Fill-in the Create ATS Action Dialog:



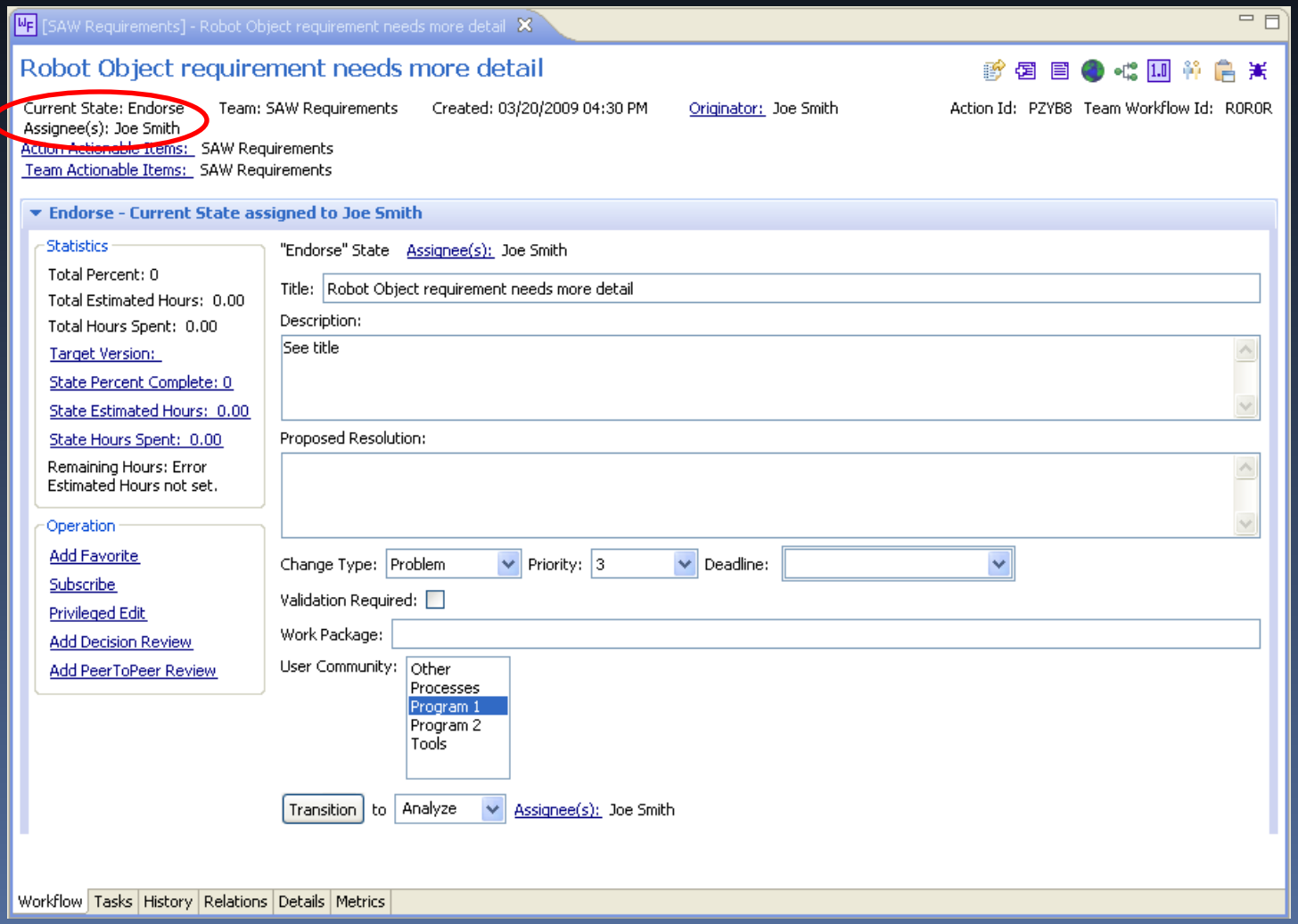


Next



User creates an action

Automatically assigned to team lead

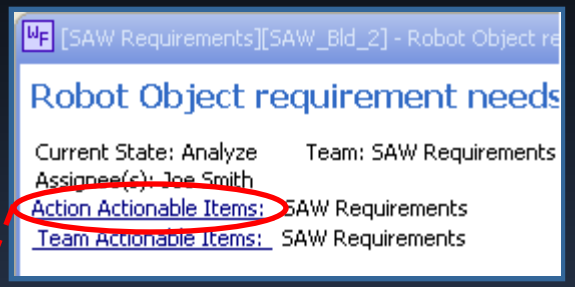


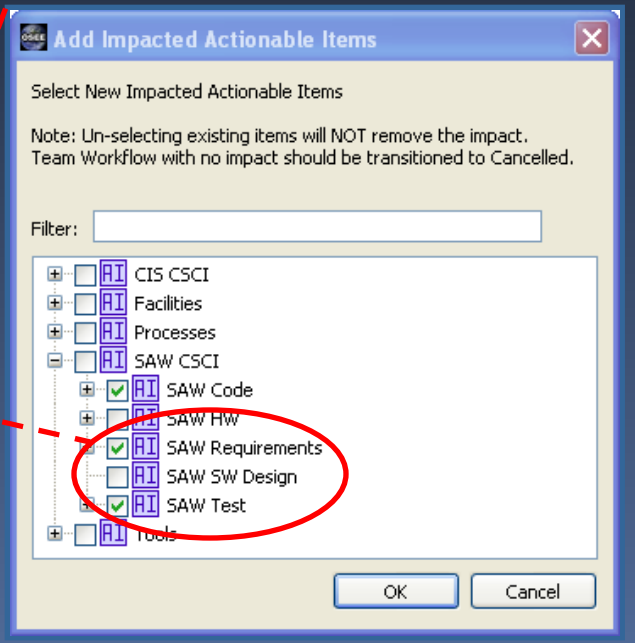
Requirements lead approves requirement team workflow

- Requirements Team Lead - Joe
  - Endorses Action
- Sets Target Version
  - SAW Build 2 (the next version)
- Changes priority to 2
- Transitions state to “Analyze”
  - This is where the lead would normally assign another user to complete the work
  - NOTE: We will not change the assigned user for the demo

Requirements developer analyzes requirement team workflow

- Requirements Developer - Joe
  - Analyzes Action
- Sets proposed resolution to “Fix It”
- Change will impact code and test
- Add code and test workflows
  - Select “Actionable Items” hyperlink
  - Add SAW Code
  - Add SAW Test
- Action View shows new workflows
- Email has been sent to leads
- Transitions to Authorize



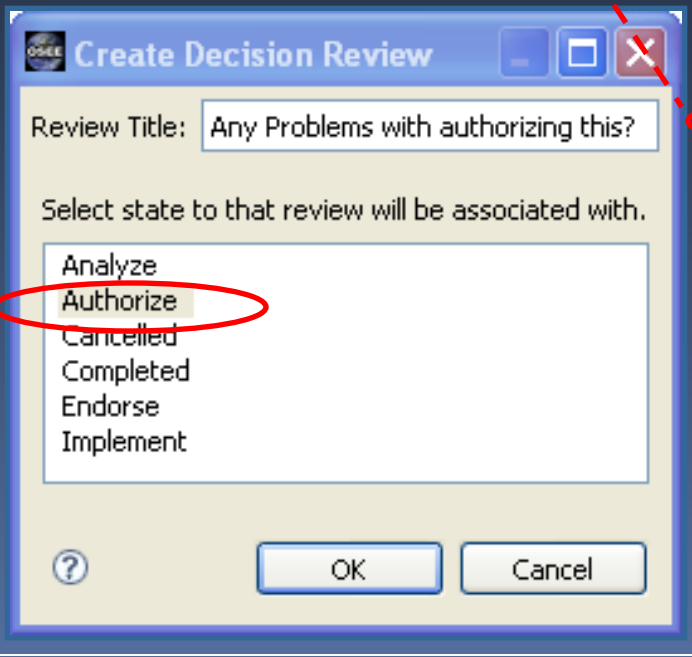


Requirements lead begins to authorize the requirement team workflow

- Requirements Team Lead - Joe
  - Authorizes Action
- Sets Work Package to A324324A
- Team lead needs concurrence from Kay (The Manager)
  - A decision review is needed

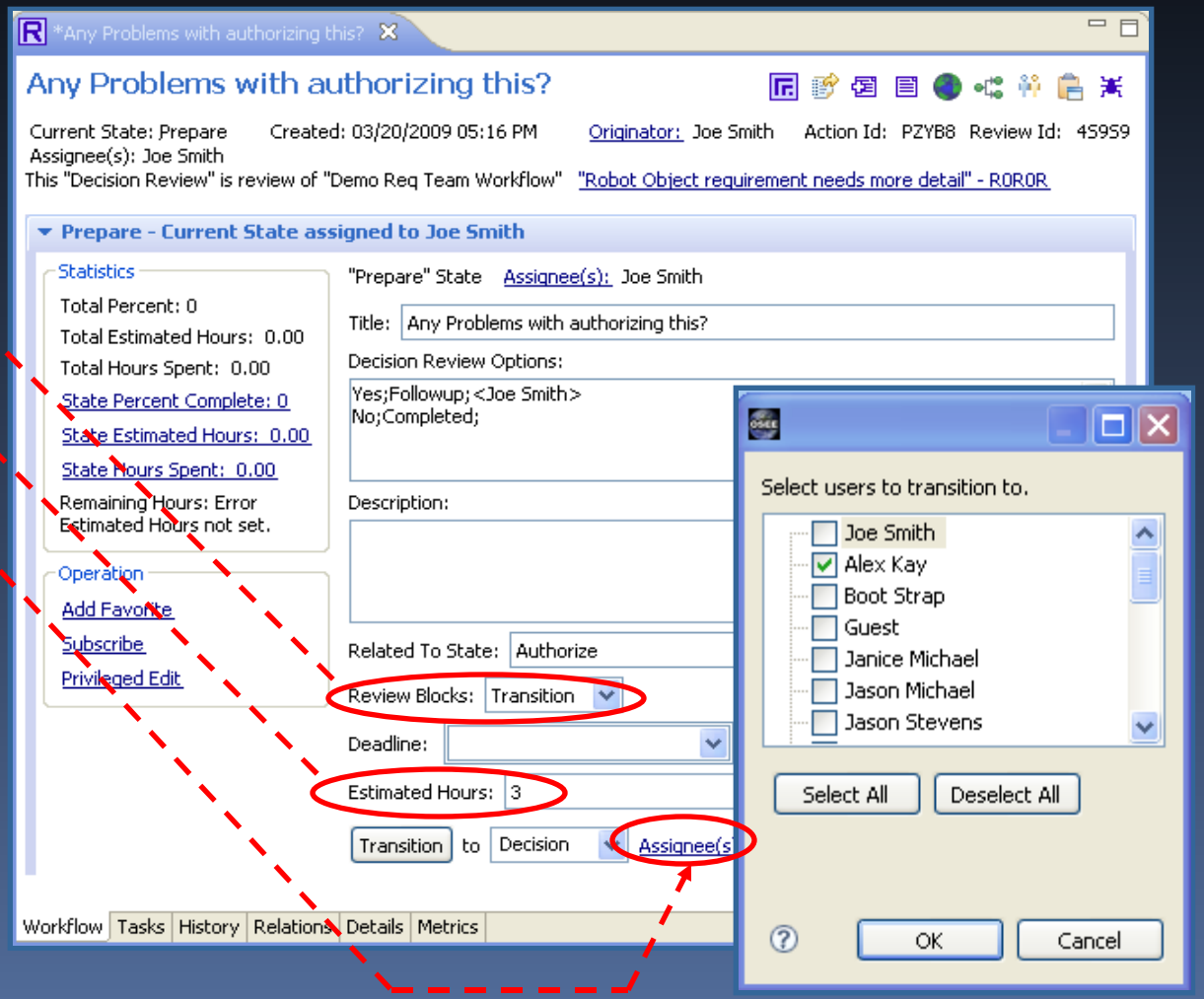
Requirements lead creates a decision review

- Create Decision Review
  - Select Add Decision Review
  - Fill-in the Create Decision review Dialog:



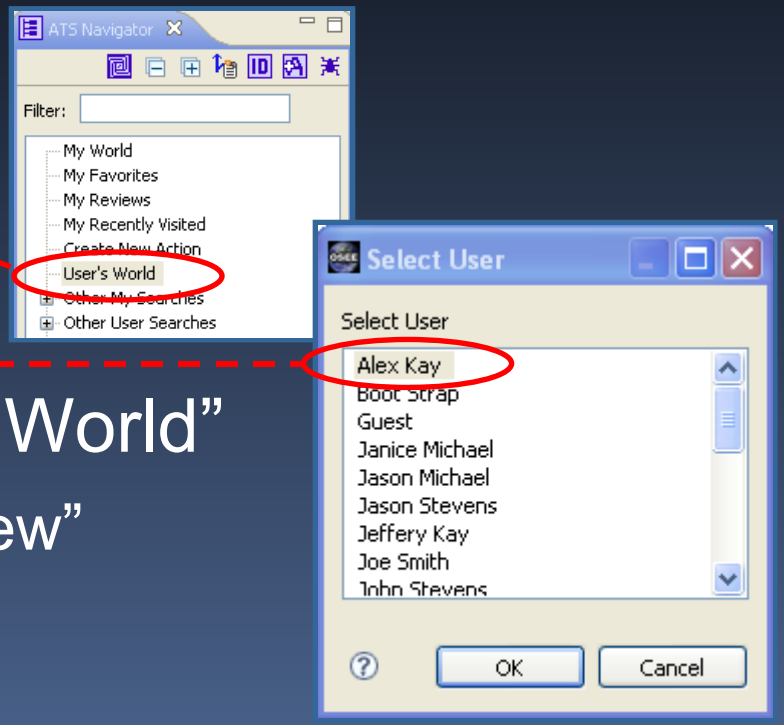
Requirements lead prepares the decision review

- Prepare Review
  - Set Review Blocks to Transition
  - Set Estimated Hours to 3
  - Assign to Alex Kay
  - Transition the review



Manager approves the change

- Alex Kay checks his assigned work
- Runs “My World”
  - Select “User’s World”
  - Select Alex Kay
- From Alex Kay’s “User’s World”
  - Select the “Decision Review”
  - Kay decides Yes
  - Transitions the Review to Completed



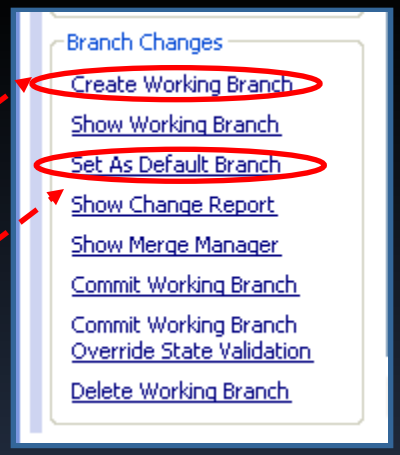
Requirements lead completes authorization

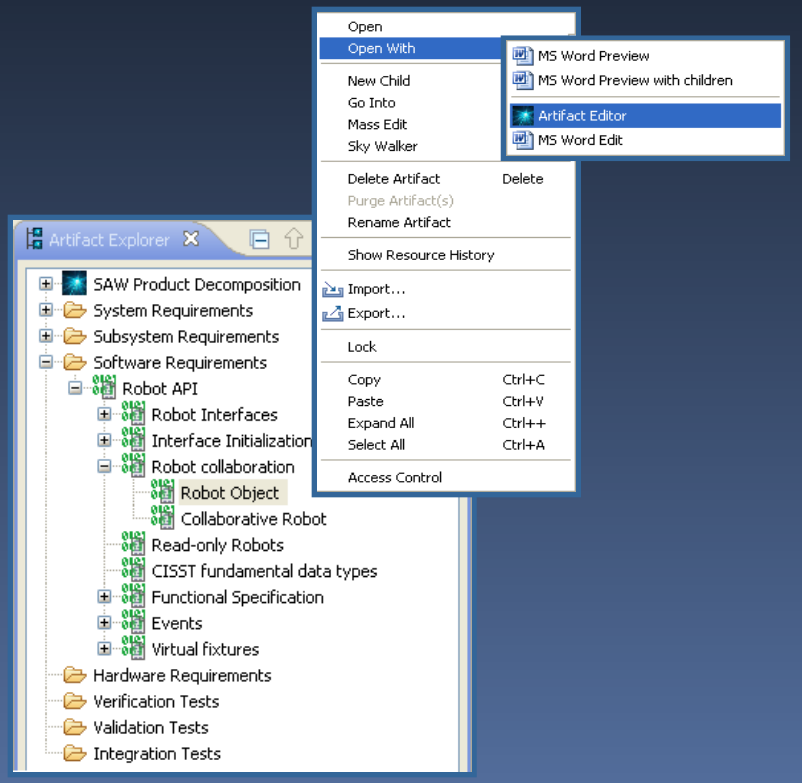
- Sets Estimated Hours to 2.5
- Transition to Implement

Notice that before Alex Kay had completed the decision review, Joe was not able to transition to the next state

Requirements developer implements a change

- From the Implement State
  - Select “Create Working Branch”
  - Set the working branch as the default branch
- Using Artifact Explorer, navigate to the “Robot Object” software requirement
- Right-Click on “Robot Object”
- From the pop-up menu, select “Open With”
- Select “Artifact Editor”

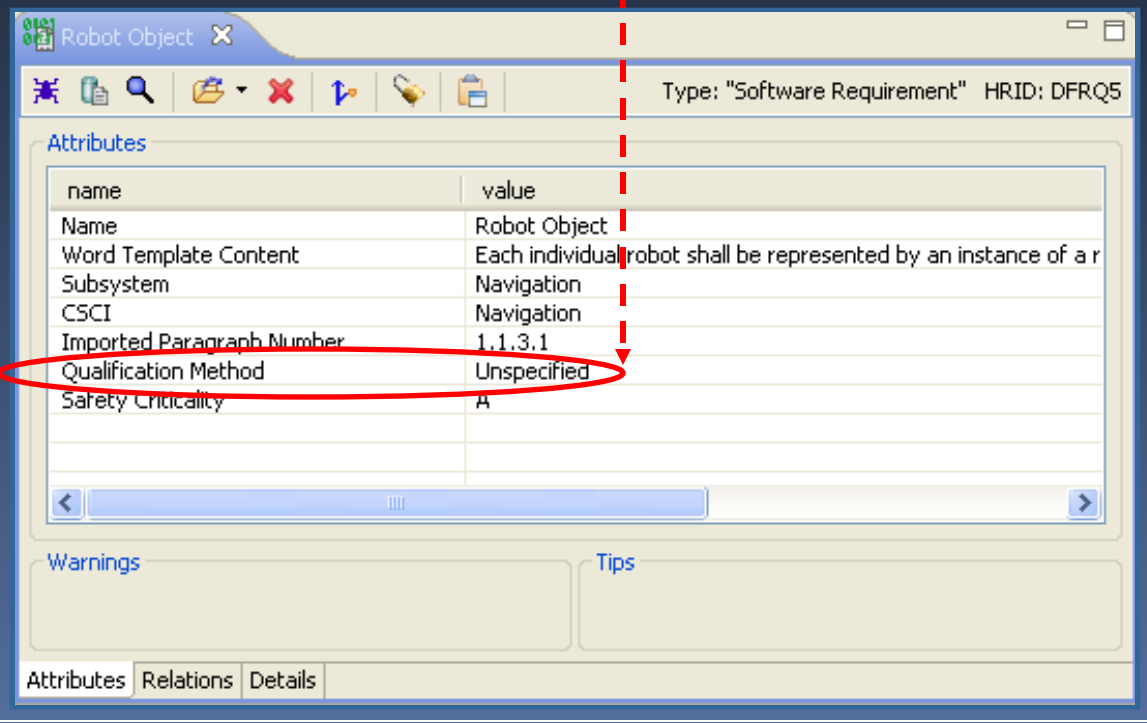




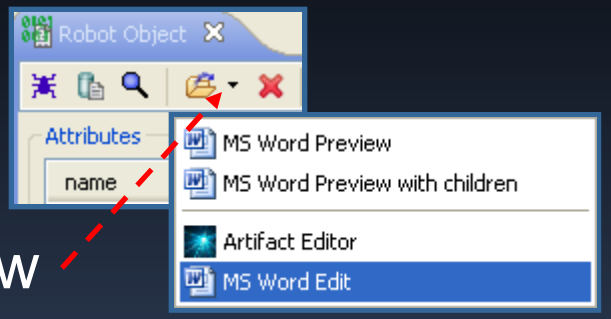
Requirements developer implements a change

- Select the attributes tab
- Change Qualification Method to Inspection
- Save by clicking

  File->Save



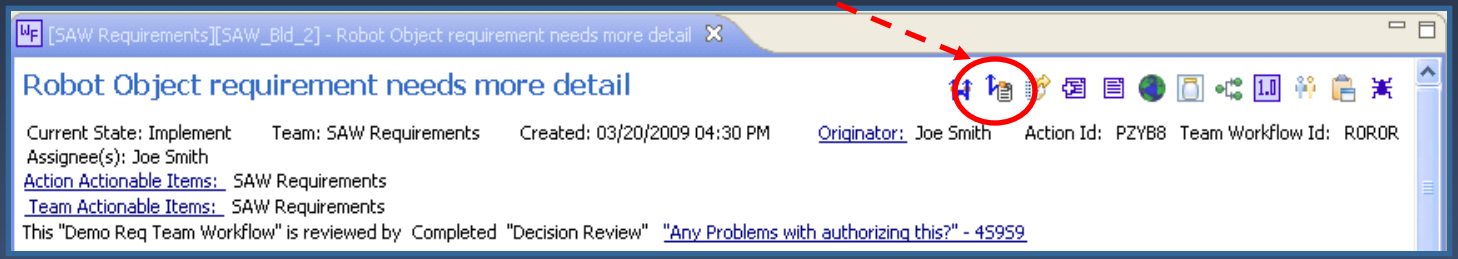
Requirements developer implements a change

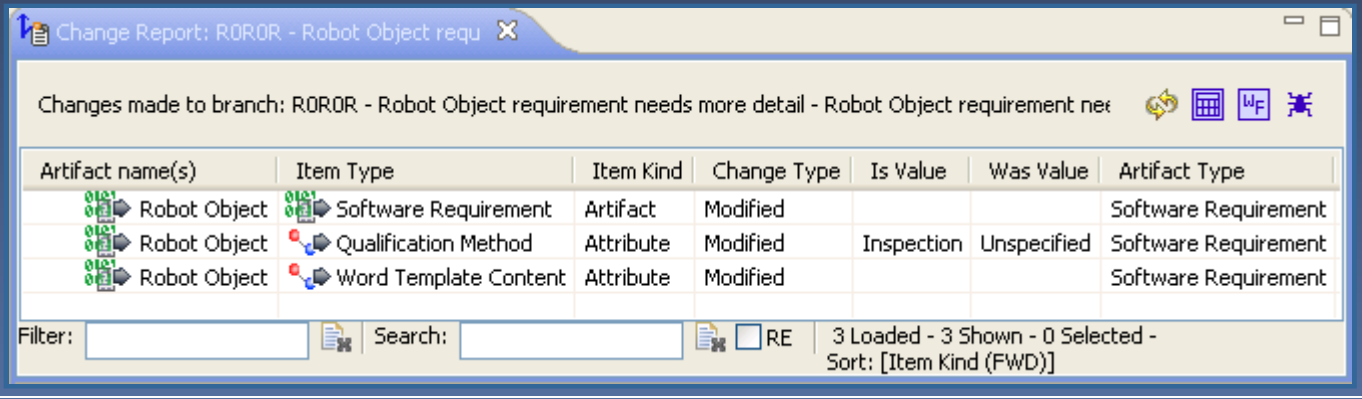


- From the “Artifact Editor’s” toolbar
  - Click on “Open With” down arrow
  - Select MS Word Edit
    - NOTE: If you don't have MS Word, just watch
- Insert into word document
  - “Need more information here.”
- Save document and close
  - “Artifact Editor’s” Word Template Content Attribute should update accordingly

Requirements developer implements a change

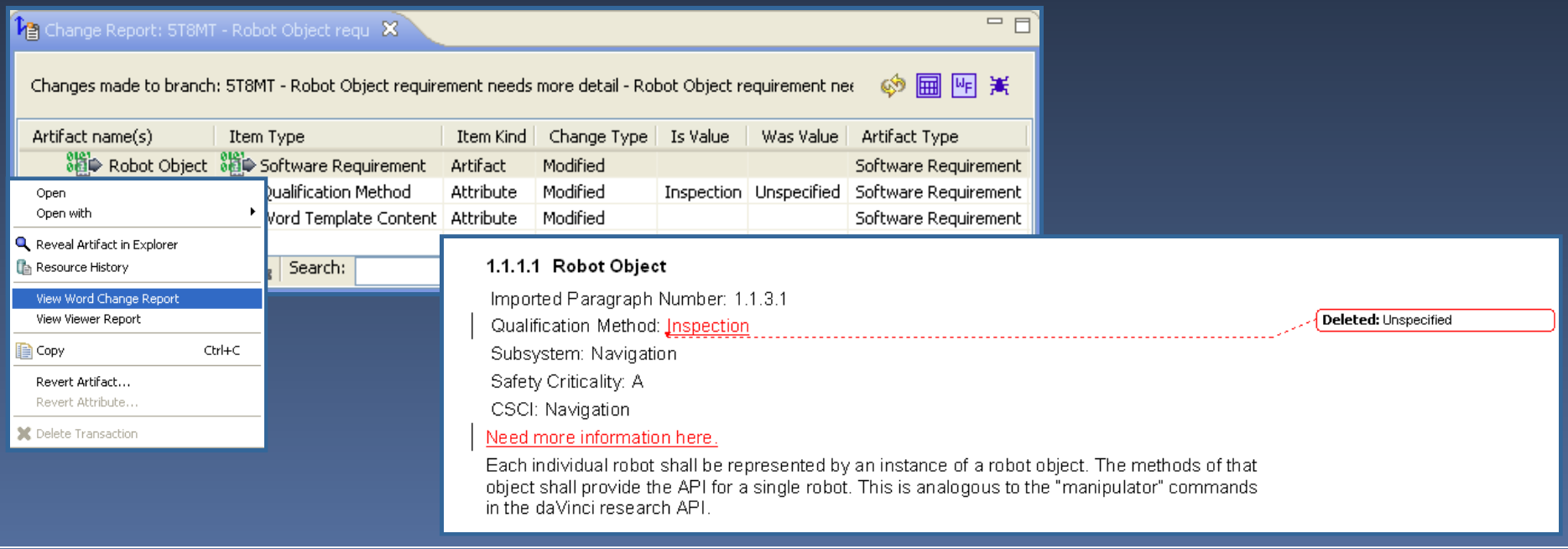
- Switch to the Workflow Editor
- Select “Show Change Report” from the tool bar item





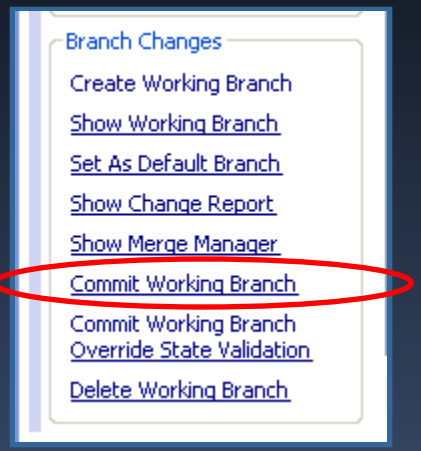
Requirements developer implements a change

- From the Change Report View
- Right-click on the “Robot Object” software requirement
- Select “View Word Change Report” from the pop-up menu



Requirements developer implements a change

- Select "Commit Working Branch“ to apply changes to the parent branch “SAW_Bld_2”
- Transition to Complete

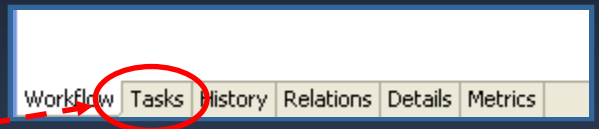


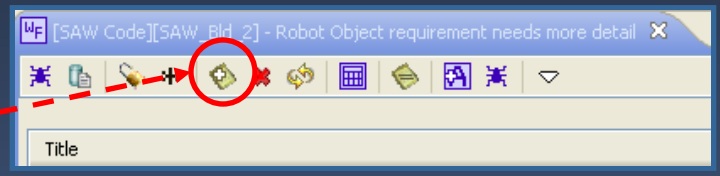
Code lead endorses code team workflow

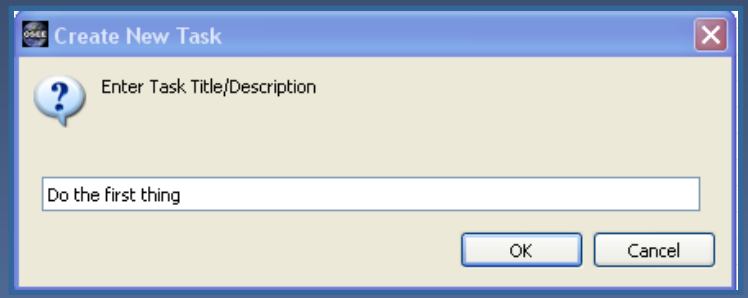
- Sets Work Package to A234532
- Transitions to Analyze

Code developer analyzes team workflow

- Sets Estimated Hours to 10
- Analysis Requires Tasks
  - Switch to the Tasks Page by clicking on the Task Tab
  - Add a Task by clicking on the “New Task” tool bar item
  - In the “Create New Task” dialog enter “Do the first thing”
  - Click “OK” to close the dialog

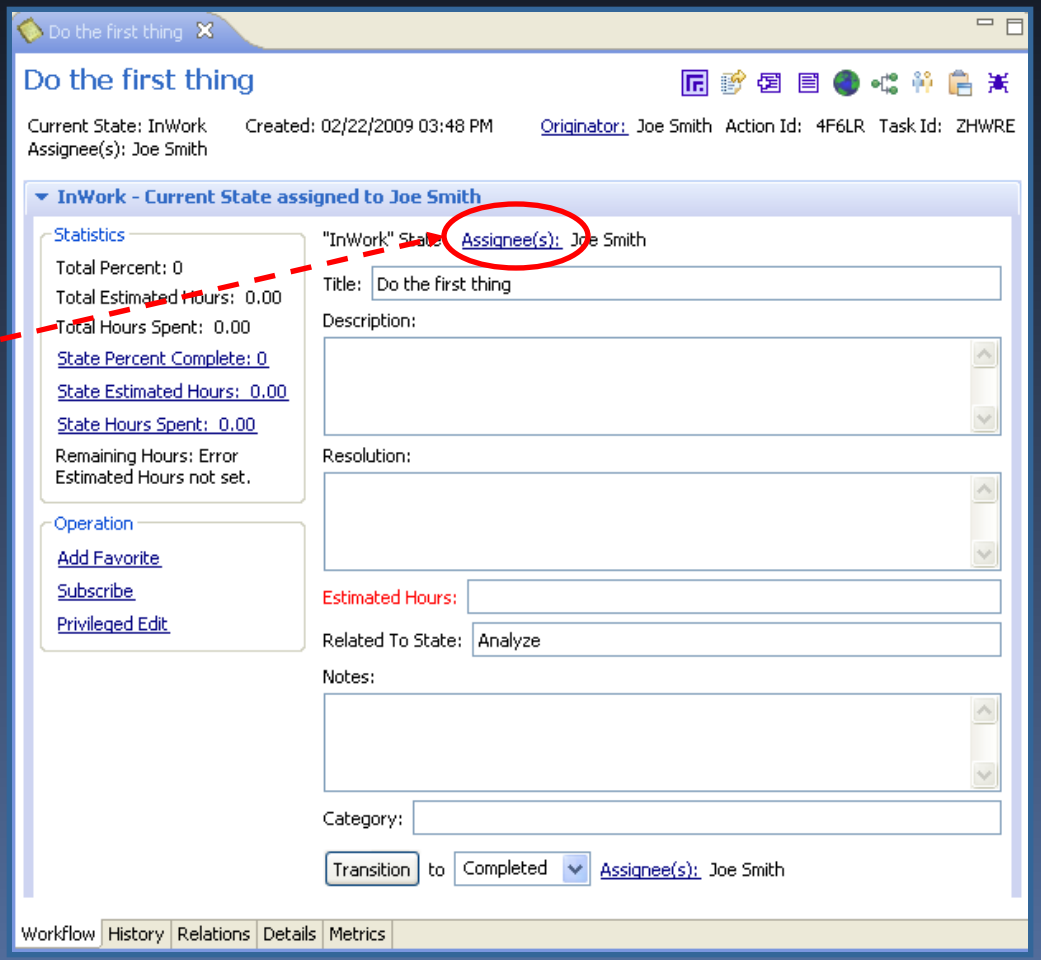






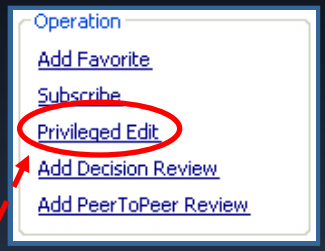
Code developer analyzes team workflow

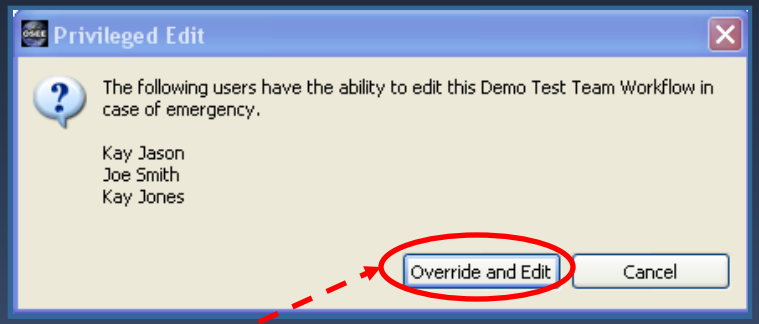
- Double-click on the new task to open the Task Editor
- Click on Assignee(s) to assign a different user
- Close Task Editor
- Add two more tasks
- Transition code team workflow to Authorize

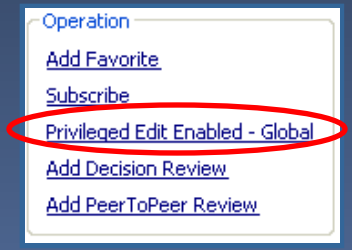


Test lead endorses test team workflow

- Test Lead – Joe
  - Estimates the work
- Kay Jones is not in today and estimates are due...
- Joe will need to get edit privileges to transition the workflow
- Click on “Privileged Edit” to display the “Privileged Edit” dialog
- Click on “Override and Edit”
- Transition To Analyze
- Set Estimated Hours to “25”

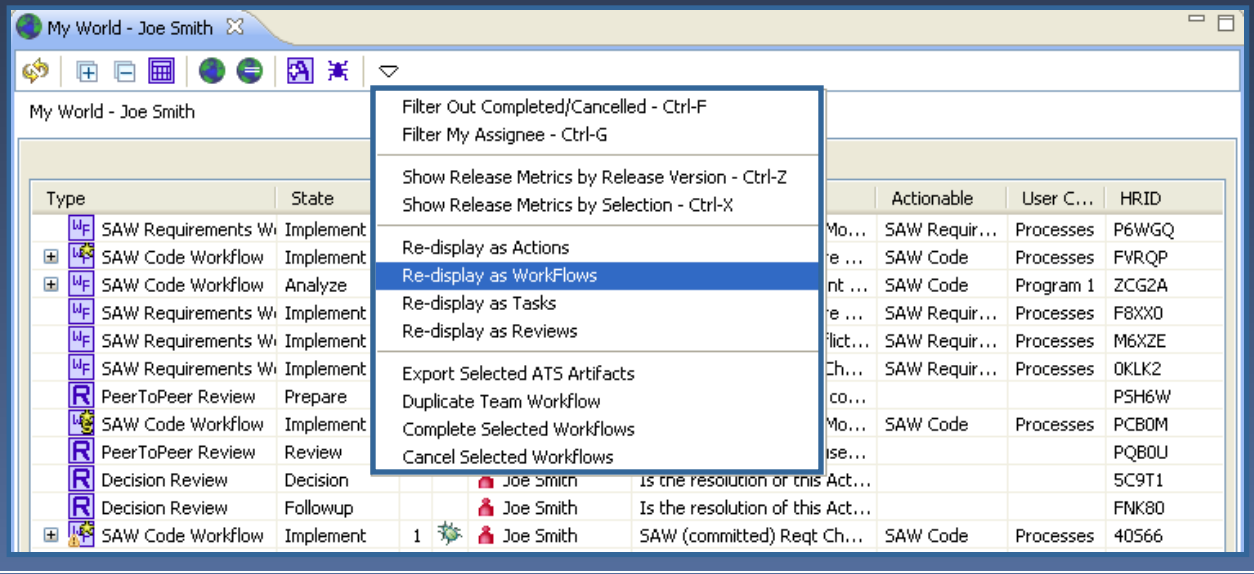






Manager wants status

- Manager – Alex Kay
  - Needs status report
- Select “World”
- Redisplay as workflows

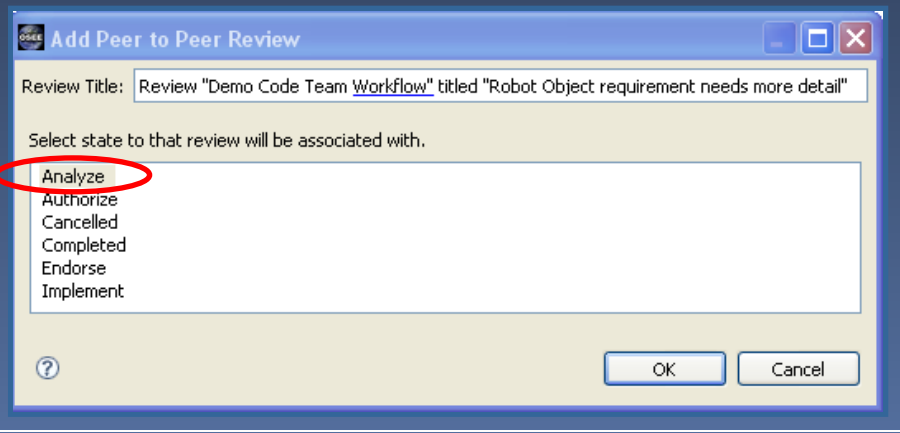


Manager wants status

- Click on “Metrics Tab” to open the Metrics Page
- Set “Estimated Completion Date” to 2 Days from the current date
- Kay won't make it (slacker); Joe will

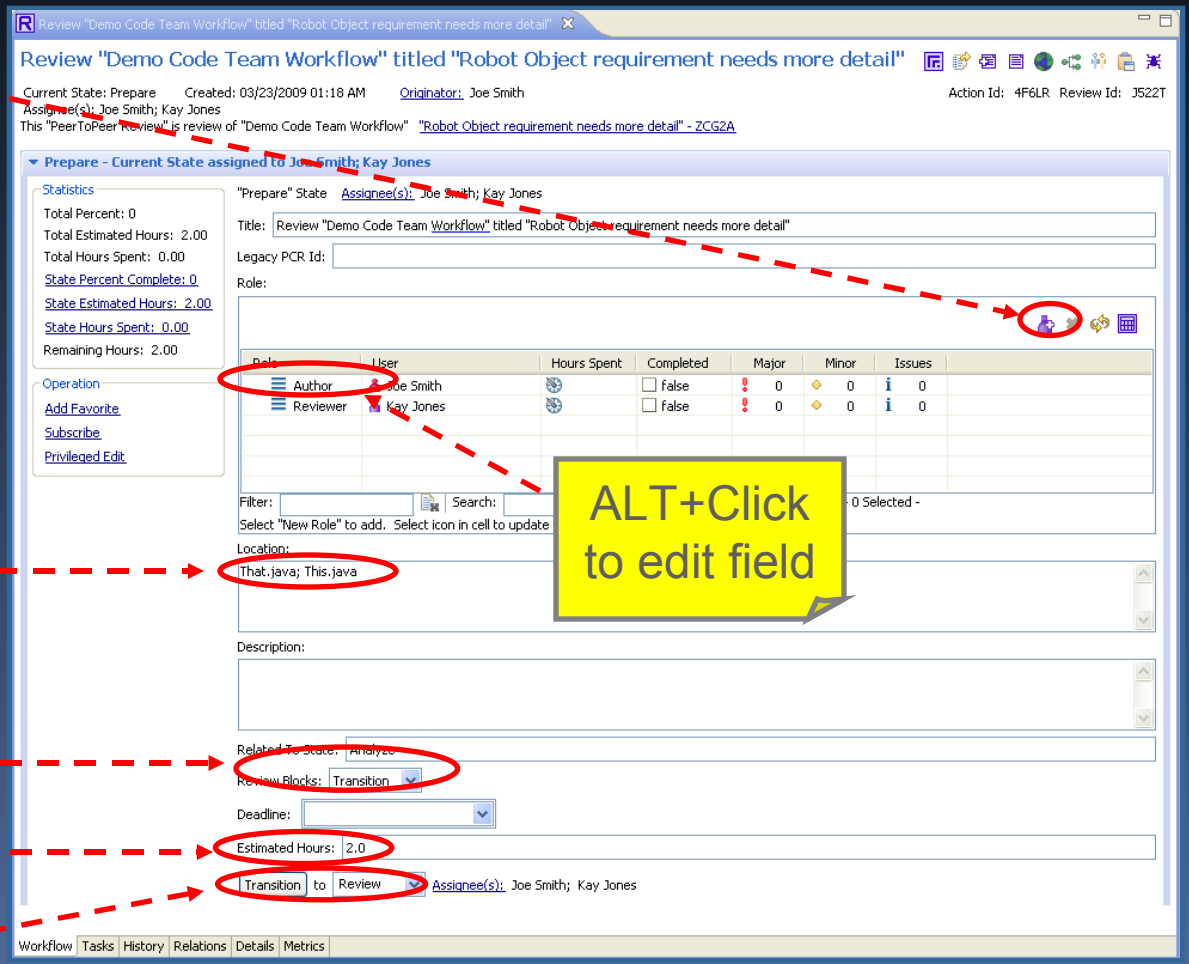
Code developer adds a peer review

- Code Developer – Joe
  - Realizes he needs a peer review for Analysis
- Add Peer-To-Peer Review
  - Click on the “Add PeerToPeer Review” hyperlink
  - In the “Add Peer to Peer Review” dialog, select “Analyze” state
  - Click “Ok”



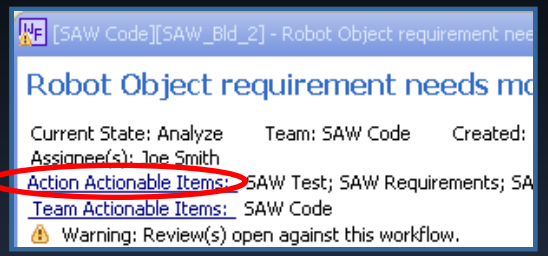
Code developer adds a peer review

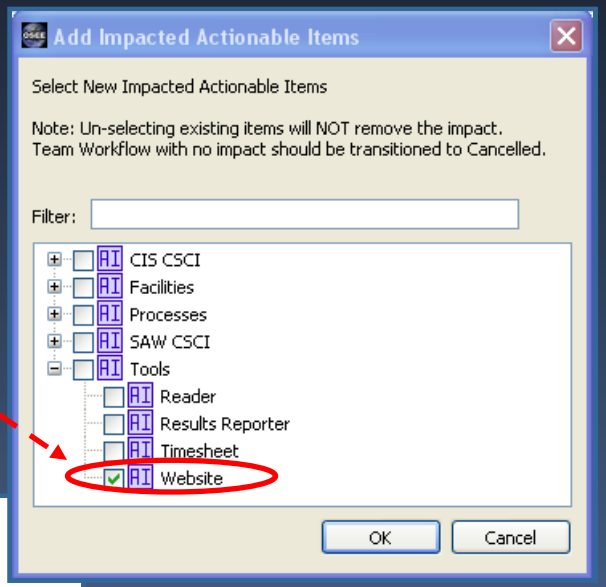
- Click on the New Role tool item to add 2 reviewers
- Set one of the roles to “Author” and the other to “Reviewer”
  - Edit the “Role” field so by clicking on the field while pressing the “ALT” key
- Set Location to “That.java; This.java”
- Set Blocking to “Transition”
- Set Estimated Hours to “2”
- Transition to Review

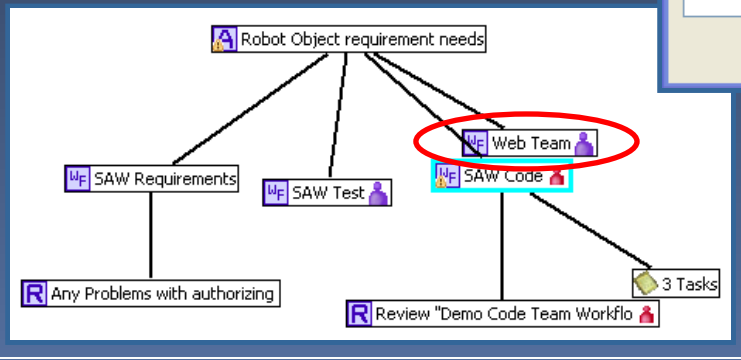


Code developer creates a tool team workflow

- Code Developer – Joe
  - Realizes the change will affect the Tools Team
- Select the “Action Actionable Items” hyperlink from the SAW Code Workflow
- Select “Website” from the “Add Impacted Actionable Items” dialog
- Workflows are created for the Website Team







BREAK

From www.eclipse.org/osee to deployment
Tutorial (Part III)

Ryan Brooks
Donald Dunne
Roberto Escobar

Boeing
Mesa, AZ

 March 22, 2009 |

Agenda

- Deployment: Things to consider
- Administrator Access
- Creating a Baseline Branch
- Customizing the data model
- Importing the data model
- ATS Configuration
- Extensibility

Deployment: Items to Consider

- Remote access to data store
- Hardware considerations
  - Database server
  - Application server
- Requirements
  - Microsoft Office
  - Database Server
    - PostgreSql
    - Oracle Server

Granting Administrator Access

- Set Default Branch to Common
- Open the User Groups Folder
- Open “OseeAdmin” with Artifact Editor
- Click on the Relations Tab
- Perform a Quick Search
  - Check the Attribute Type Option
    - Ensure it is set to Name
  - Uncheck all other options
- Drag and Drop the user “Joe Smith” into the Users relation
- You are now an OSEE Admin

Creating a Baseline Branch

- Ensure you have OSEE Admin privileges
- Refresh the branch manager view
- You should now be able to see the “System Root” branch
- Right-click on the “System Root” branch and select “Branch” from the pop-up menu
- A child branch of the “System Root” branch will be created

Data Model Customization

- The data model in OSEE is extensible and user configurable
- Users can define new artifact, attribute, and relation types and their constraints such as multiplicity and applicability
- Type inheritance allows similar types to be defined and modified without tedious redundancy because similar types inherit what is common from their super type

Data Model Customization - Spreadsheet

- The OSEE data model is defined using a tabular format involving 5 tables.
- The full data model can be defined using a single spreadsheet or be divided among multiple spreadsheets that can reference types defined in any spreadsheet.
- See the following xml spreadsheet OseeTypes_ProgramAndCommon.xml for an example.

Data Model Customization – Table 1 - Artifact

- Factory Class – Retired in 0.7.0
- Artifact Type Name - Any valid UTF-8 characters with a max length of 75 bytes
- Super Type Name - The super artifact type from which this type will inherit associated attributes and relations. Another concrete artifact type or an abstract one that exists only in data model definition.

Data Model Customization – Table 2 - Attributes

- Attribute Base Type - Fully qualified java class name of a class that extends org.eclipse.osee.framework.skynet.core.attribute.
  The typical and simplest case is to specify one of the built-in types (StringAttribute, BooleanAttribute, etc…) If a custom java type that extends Attribute is needed, then that type should be specified here.
- Attribute Type Name - Any valid UTF-8 characters with a max length of 500 bytes
- File Extension - Any valid UTF-8 characters with a max length of 50 bytes; only applies when using the org.eclipse.osee.framework.skynet.core.UriAttributeDataProvider
- Tagger ID - If the attrbiute's content is to be included in the search index for the quick search, use DefaultAttributeTaggerProvider, otherwise leave blank.
- Default Value - The initial value given an attribute upon initialization, this may be left blank
- Validity Xml - For the attribute base type org.eclipse.osee.framework.skynet.core.EnumeratedAttribute, specifies the valid enumerations. For example, <Page_Type><Enum>Portrait</Enum><Enum>Landscape</Enum></Page_Type>
- Min Occurrence - The framework with prevent having lest than this number of this attribute type on a single artifact
- Max Occurrence - The framework with prevent adding more than this number of this attribute type to a single artifact
- Tip Text - Text to describe an attribute. Any valid UTF-8 characters with a max length of 4000 bytes

Data Model Customization – Table 3 - Artifact Type / Attribute Type Mapping

- Artifact Type Name - Exact name of an artifact type defined above (or previously)
- Attribute Type Name - Exact name of an attribute type defined above (or previously) to be associated with the corresponding artifact type

Data Model Customization – Table 4 - Relation Type

- Relation Type Name - Any valid UTF-8 characters with a max length of 50 bytes
- Side A Name - Descriptive name for the A side of the relation. Any valid UTF-8 characters with a max length of 50 bytes
- A to B Phrase - An optional phrase that describe the relation between the artifacts from the side A perspective. Any valid UTF-8 characters with a max length of 50 bytes
- Side B Name - Descriptive name for the B side of the relation. Any valid UTF-8 characters with a max length of 50 bytes
- B to A Phrase - An optional phrase that describe the relation between the artifacts from the side B perspective. Any valid UTF-8 characters with a max length of 50 bytes
- Short Name - Five or less characters is typical. This abbreviated name is used in the user interface when space is at a premium
- Ordered - Yes to have artifacts on the same side of this relation type use a user defined ordered, otherwise No

Data Model Customization – Table 5 - Artifact Type / Relation Type Mapping

- Artifact Type - Exact name of an artifact type defined above (or previously)
- Relation Type - Exact name of an relation type defined above (or previously)
- Side A Max - An artifact of type "Artifact Type" can be on side "A", "Side A Max" number of times for relation links of type "Relation Type“
- Side B Max - An artifact of type "Artifact Type" can be on side "B", "Side B Max" number of times for relation links of type "Relation Type"

Data Model - Add Artifact, Attribute and Relation

- Open OseeTypes_ProgramAndCommon_New.xml spreadsheet (5 highlighted lines were added)
- Added Artifact Type: System Function
- Added Attribute Type: Safety Criticality
  - Enumeration:<Criticality><Enum>A</Enum><Enum>B</Enum><Enum>C</Enum><Enum>D</Enum><Enum>E</Enum></Criticality>
- Added Artifact to Attribute Mapping
- Added Relation Type: Design
- Added Relation to Artifact Mapping

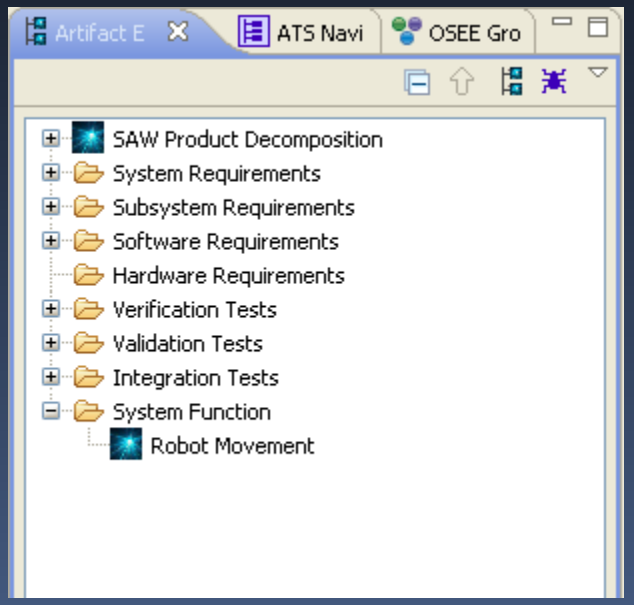
Import OSEE Data Model

- To import changes to the data model for an existing OSEE database:
  - File -> New -> Other -> OSEE -> Osee Types
  - Select File "OseeTypes_ProgramAndCommon_New.xml"
  - Select Branch "Common"
  - Select Finish
- To automatically import the data model during database initialization use the extension point org.eclipse.osee.framework.skynet.core.OseeTypes

Import OSEE Data Model – Try It - 1

Let's create a "System Function" folder and a new "System Function" artifact

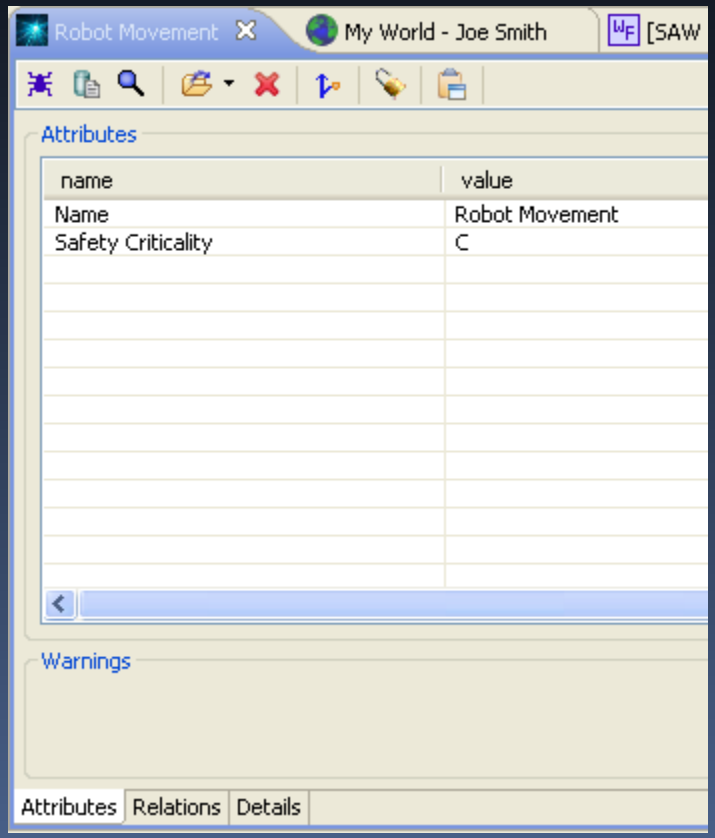
- Set Default Branch -> SAW_Bld_1
- In Artifact Explorer, right-click -> New Folder -> Name: "System Function"
- Select "System Function" -> right-click -> New Child -> "System Function" -> Name: "Robot Movement"



Import OSEE Data Model – Try It - 2

Let's set the "Safety Criticality" attribute of our new "Robot Movement" system function.

- In Artifact Explorer, Double-click to open new "System Function" artifact called "Robot Movement"
- Switch to the attributes tab of the "Robot Movement" artifact
- Modify the "Safety Criticality" value to "C"
- Save

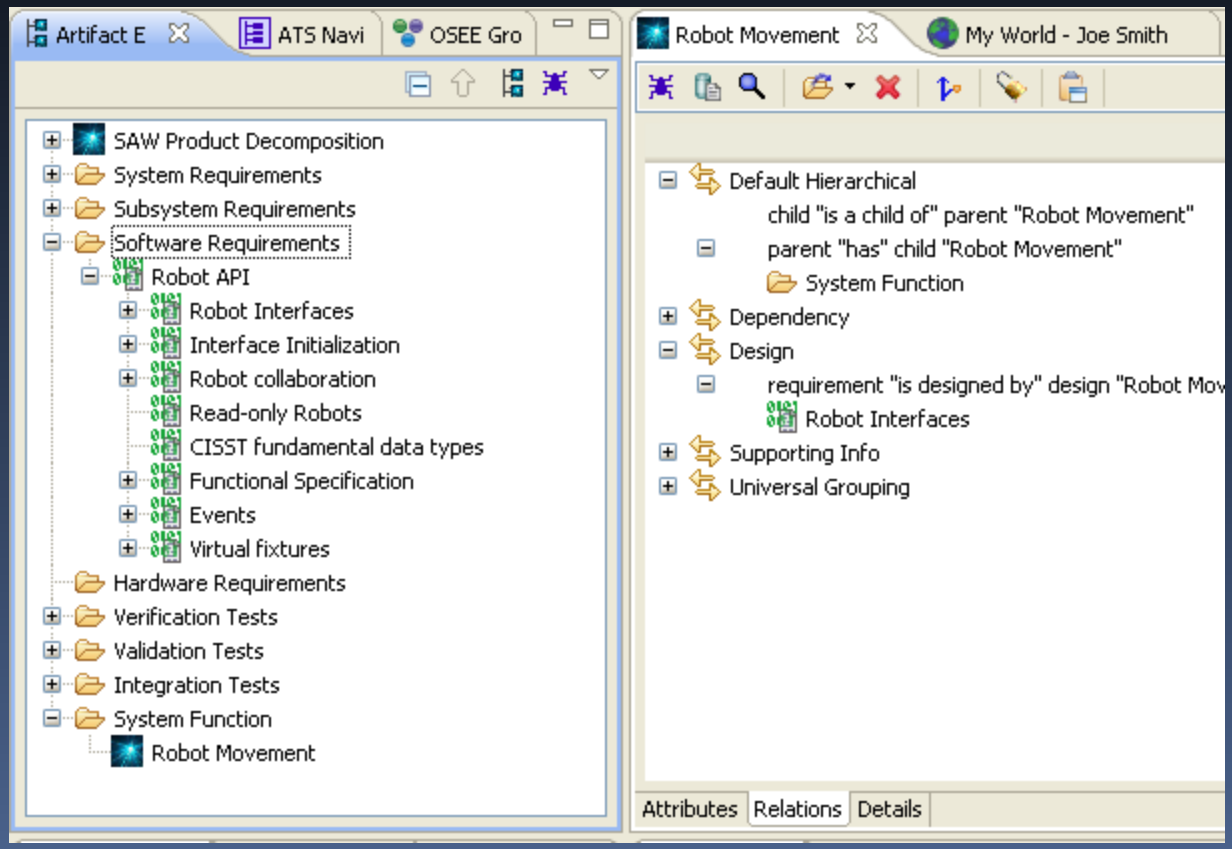


Import OSEE Data Model – Try It - 3

Let's drag an existing Software Requirement into the new "Robot Movement" "System Function" artifact.

- In Artifact Explorer, Double-click to open new "System Function" artifact called "Robot Movement"
- Switch to the relations tab of the "Robot Movement" artifact
- Under "Software Requirements" -> drag any requirement into "Design" relation
- Save

Import OSEE Data Model – Success!

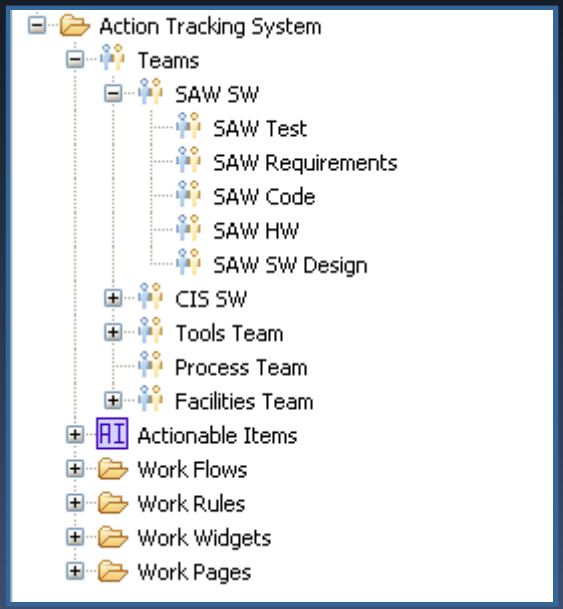


ATS Configuration

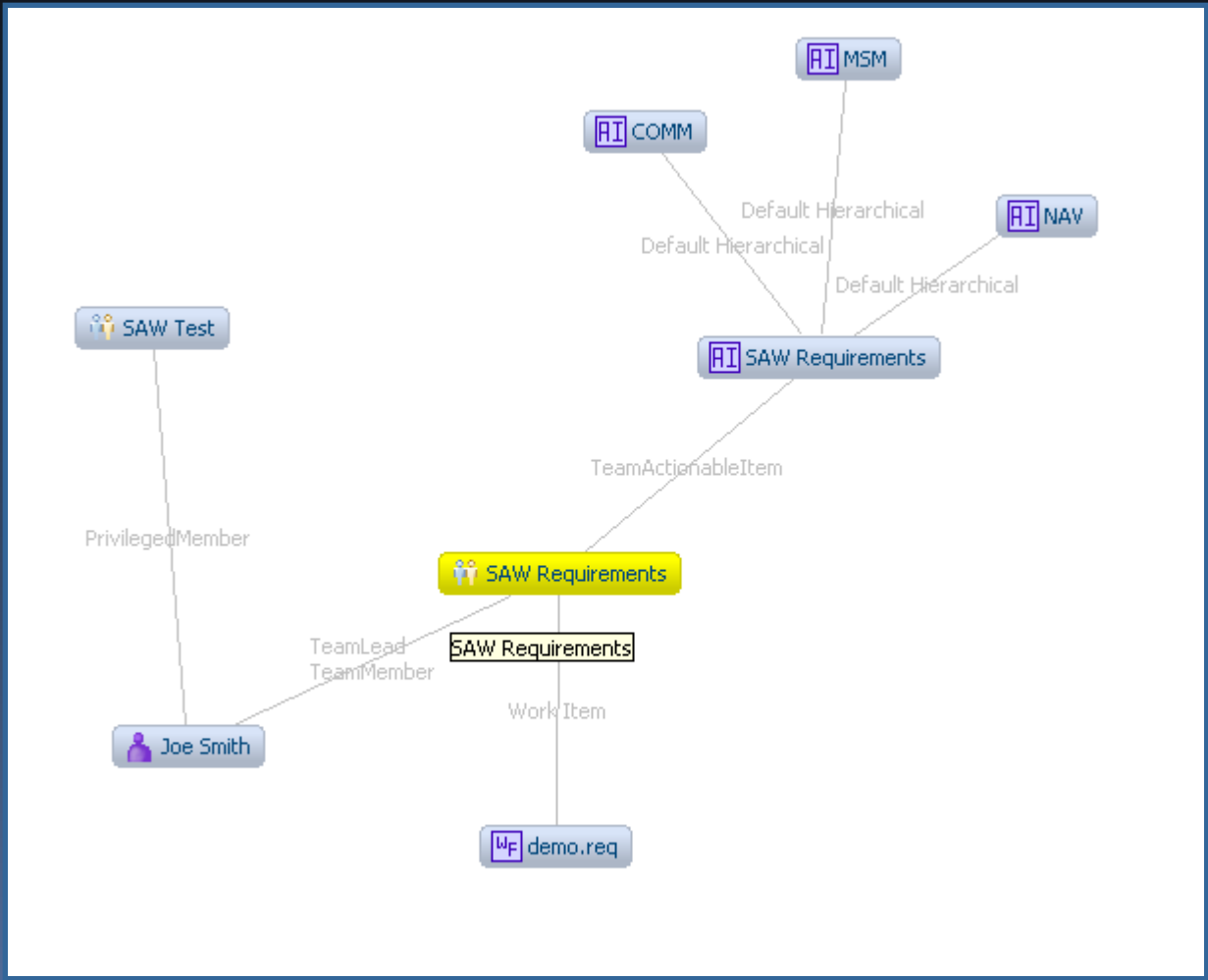
- ATS uses OSEE framework Artifacts, Attributes and Relations to store configuration information
- Configuration changes can be done using framework editors and views
- Configurations can be done dynamically in OSEE without need for code release
- Major changes such as new widget types, advanced algorithms for assignment/routing, customized searching can be done through Eclipse extension points

ATS Configuration – Team Definition

- Artifact in OSEE represents a team that is responsible for performing work
- Configured with users that perform roles
  - Team Lead – Endorses Team Workflow, Assigns Work
  - Team Member – Performs Work on Team
  - Privileged Member – Able to override assignee and edit any field in Team Workflow
- Related to Actionable Items that they are responsible for
- Related to Workflow Configuration that defines how this team does it's work

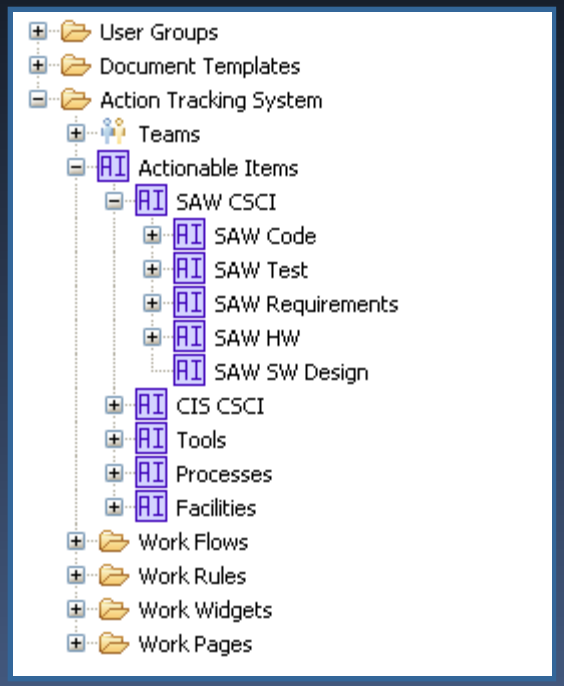


ATS Configuration – Team Definition



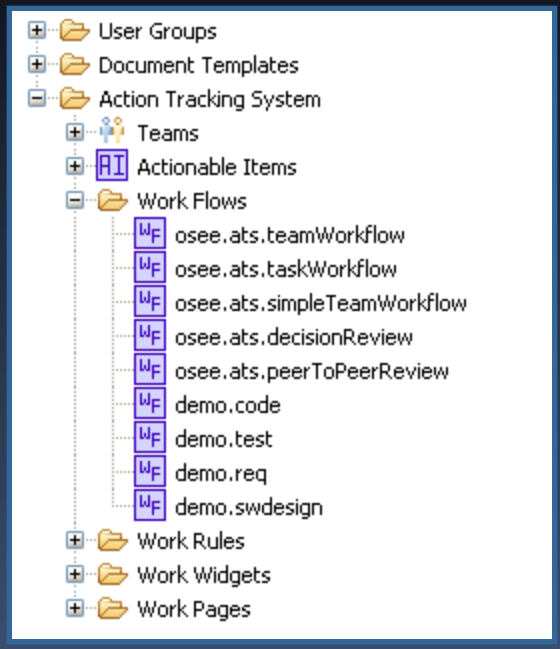
ATS Configuration – Actionable Item

- Artifact in OSEE represents a real or conceptual object that the user would write an Action against
- Active Flag enables items to be retired
- Related to Team Definitions that are responsible for them

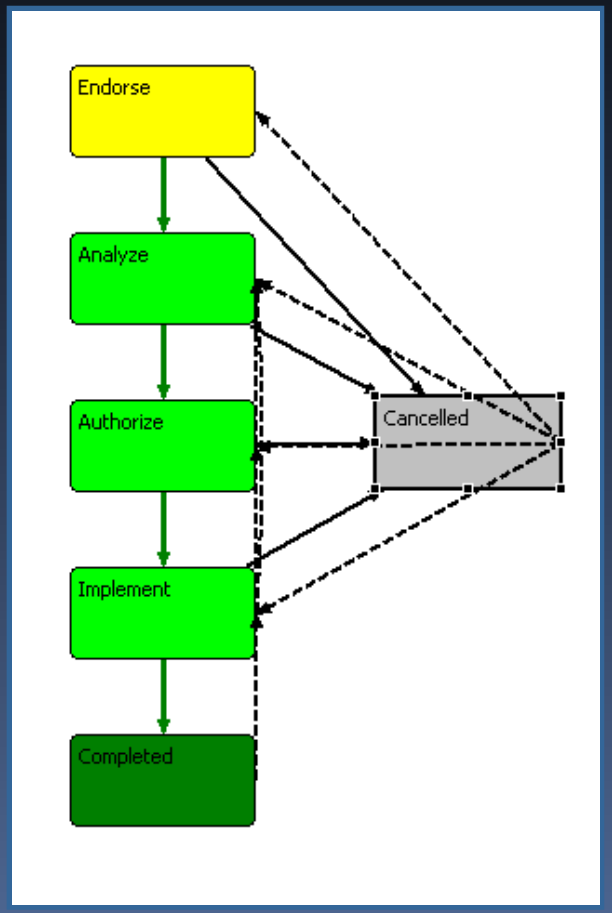


ATS Configuration – Work Flows

- Artifact in OSEE represents a how a team performs it's work
- Represented by a state machine with Work Pages being states
- Related to Team Definitions uses them
- Related to Work Pages and Work Rules that apply

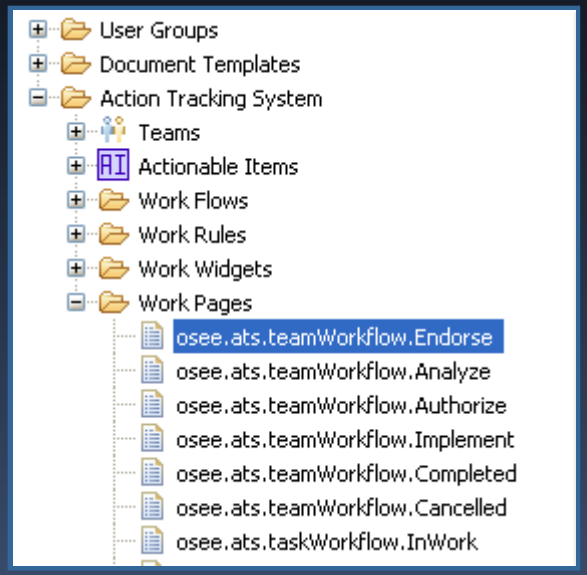


ATS Configuration – Work Flows



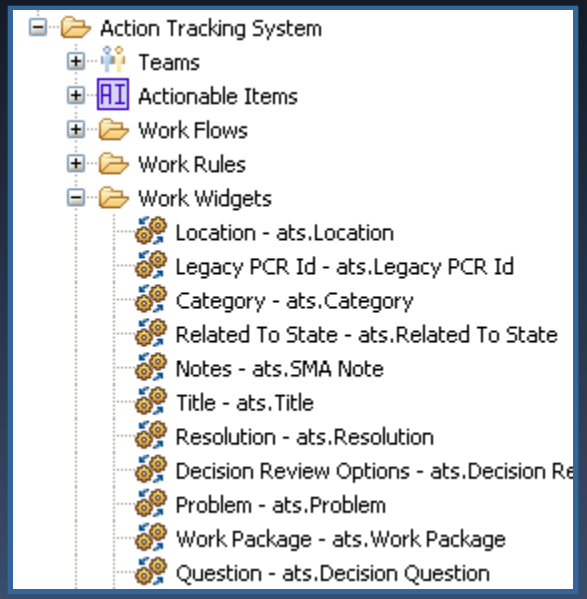
ATS Configuration – Work Pages

- Artifact in OSEE that represents a a state in a Work Flow
- Represented by a state machine with Work Pages being states
- Related to Work Flow they belong to
- Related to Work Rules and Work Widgets



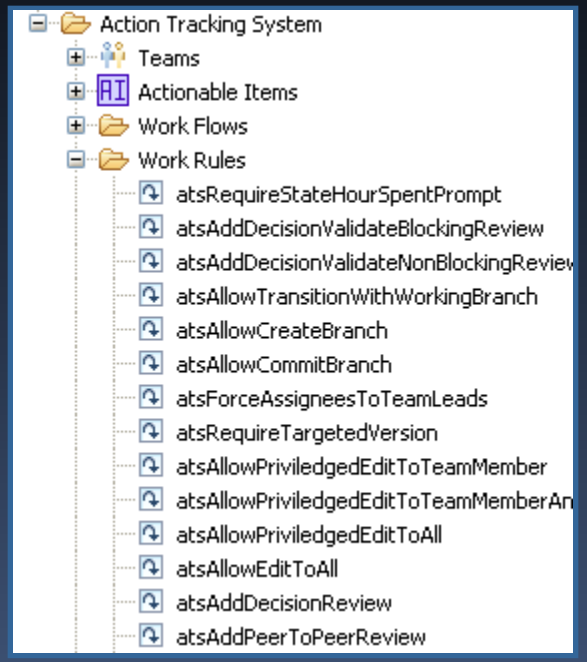
ATS Configuration – Work Widgets

- Artifact in OSEE that represents a a single Widget to display for a Work Page
- Contains some formatting information
- Includes information relating widget contents to storage attribute
- Related to Work Page they belong to

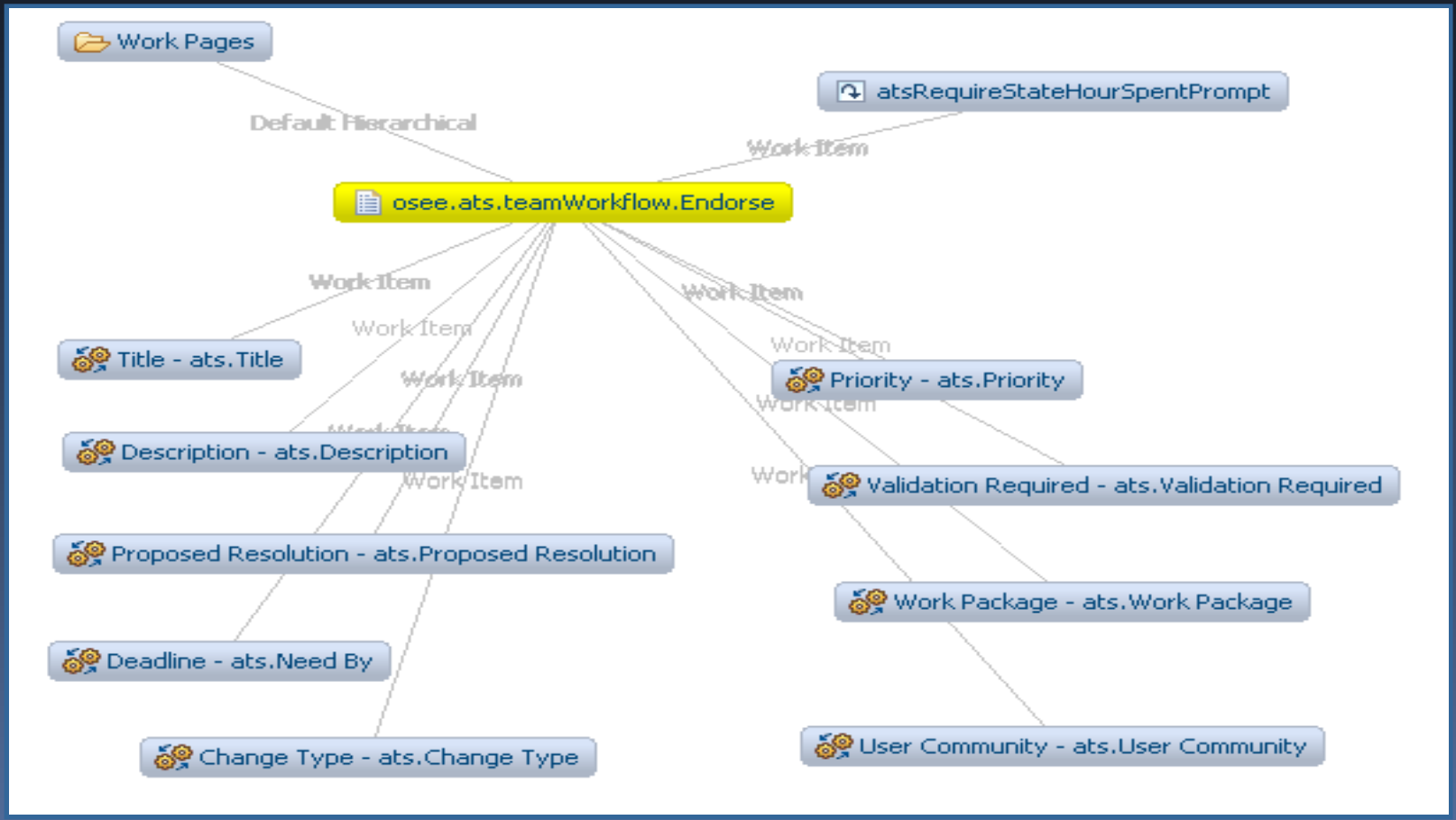


ATS Configuration – Work Rules

- Artifact in OSEE that represents a rule to be applied to Work Flow or Work Page
- Backed by code that provides the functionality described
- ATS provides built in rules for use by work flows
- New rules can provided through extension points

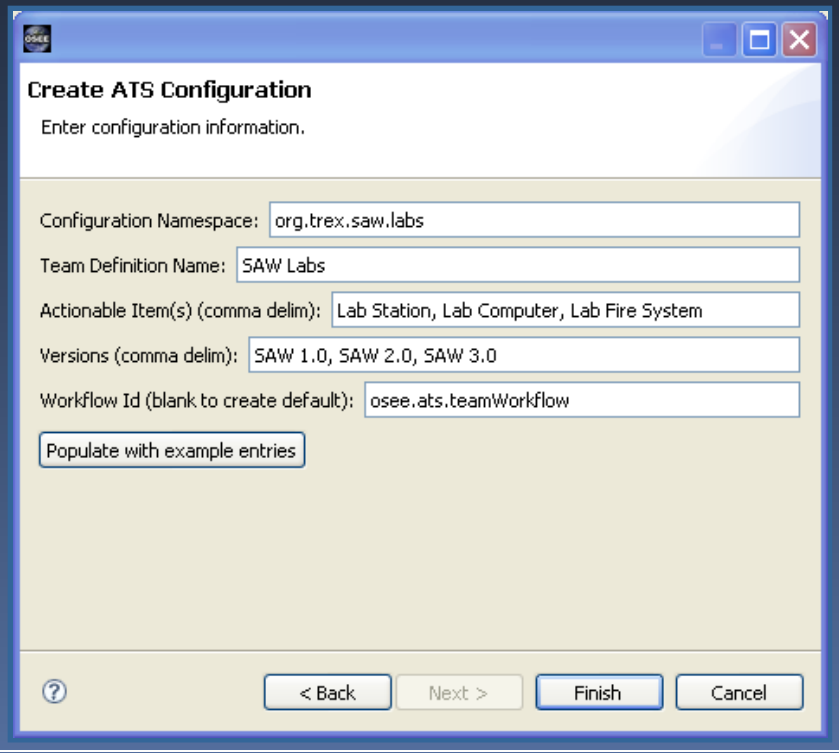


ATS Configuration – Work Pages, Widgets, Rules

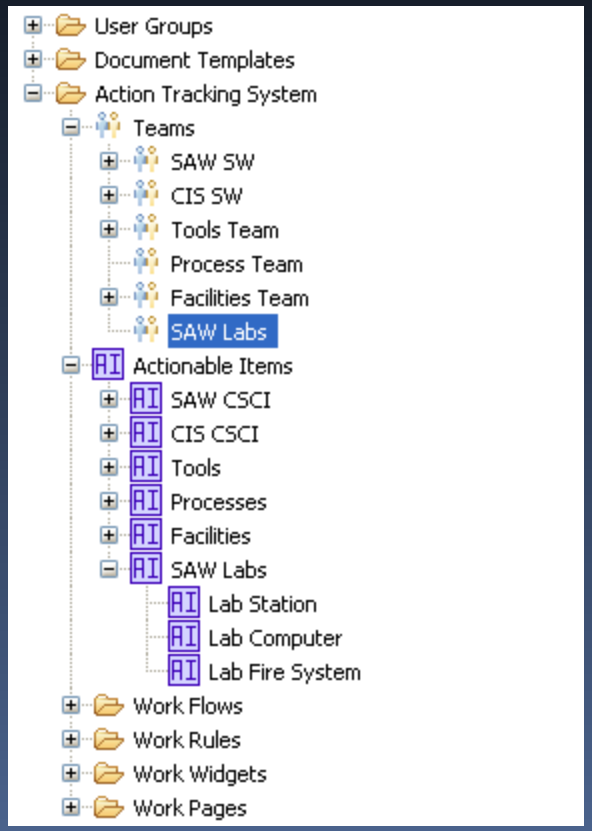


T. Rex – Let's track a new SAW Lab

- File -> New -> Other -> OSEE ATS -> Create ATS Configuration
- Select "Populate with example entries" button



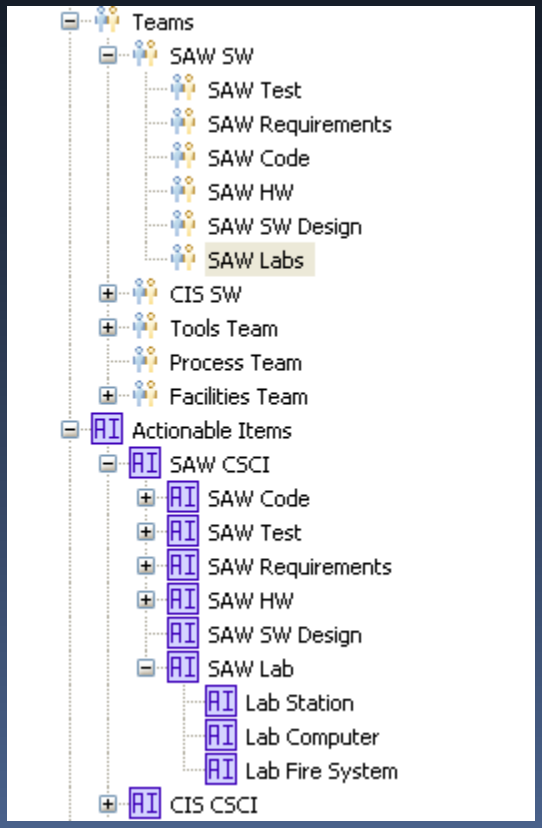
Re-Organize Team Definitions and Actionable Items



In Artifact Explorer:

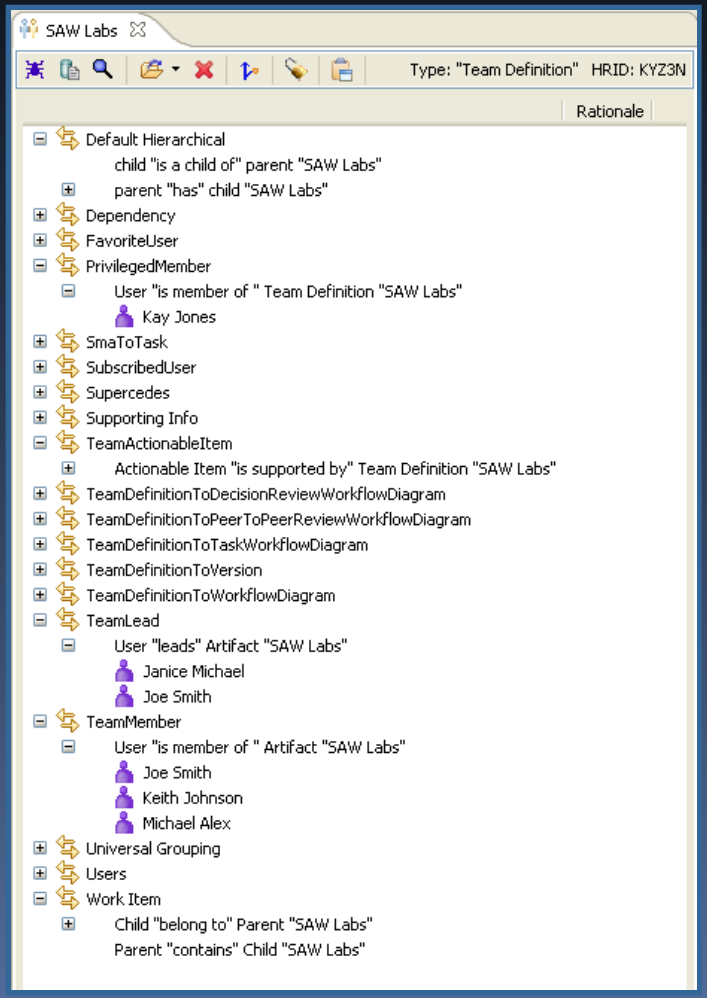
- Move New “SAW Labs” Team Definition under “SAW SW” Team
- Move New “SAW Lab” Actionable Items under “SAW CSCI” Item

becomes



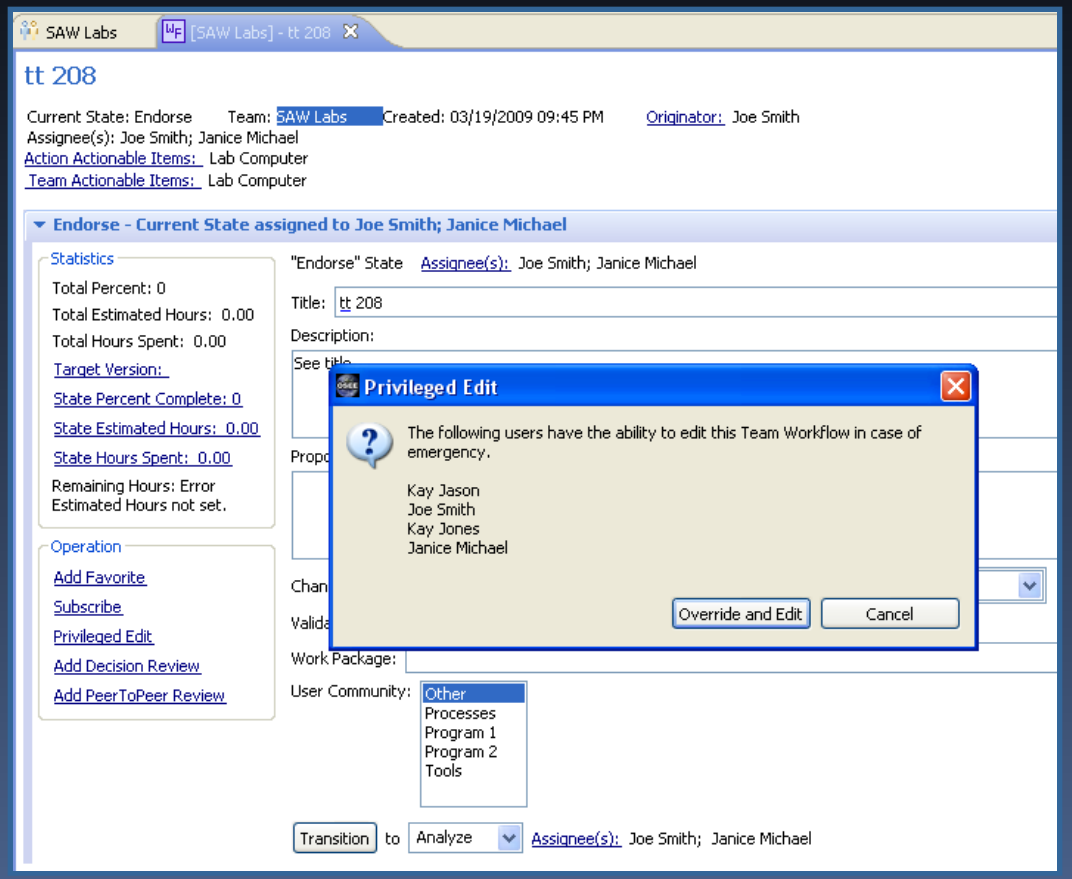
Configure Team Leads and Members

- Search Pulldown – Artifact Search – Artifact Type – User – Add Filter - Search
- From Search Results:
  - Drag “Janice Michael” to TeamLead
  - Drag “Keith Johnson” and “Michael Alex”” to TeamMember
  - Drag “Kay Jones” to Privileged Member
  - Save



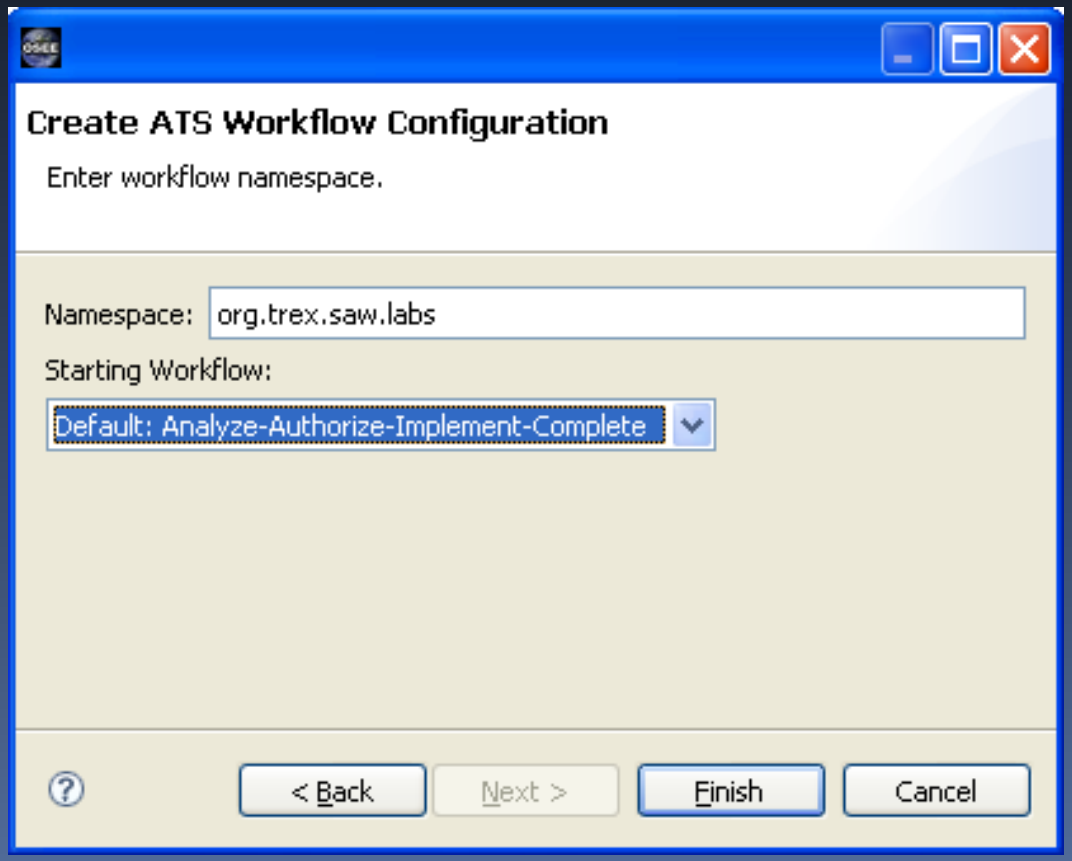
Let's Try It…

- ATS Navigator – New Action – Lab Computer Actionable Item
- Complete Action Wizard
- Notice:
  - Endorse State assigned to both leads "Janice" and "Joe"
  - Privileged Edit allows "Kay Jones" to override



Labs Team wants a different workflow

- File -> New -> Other -> OSEE ATS -> Create ATS Workflow

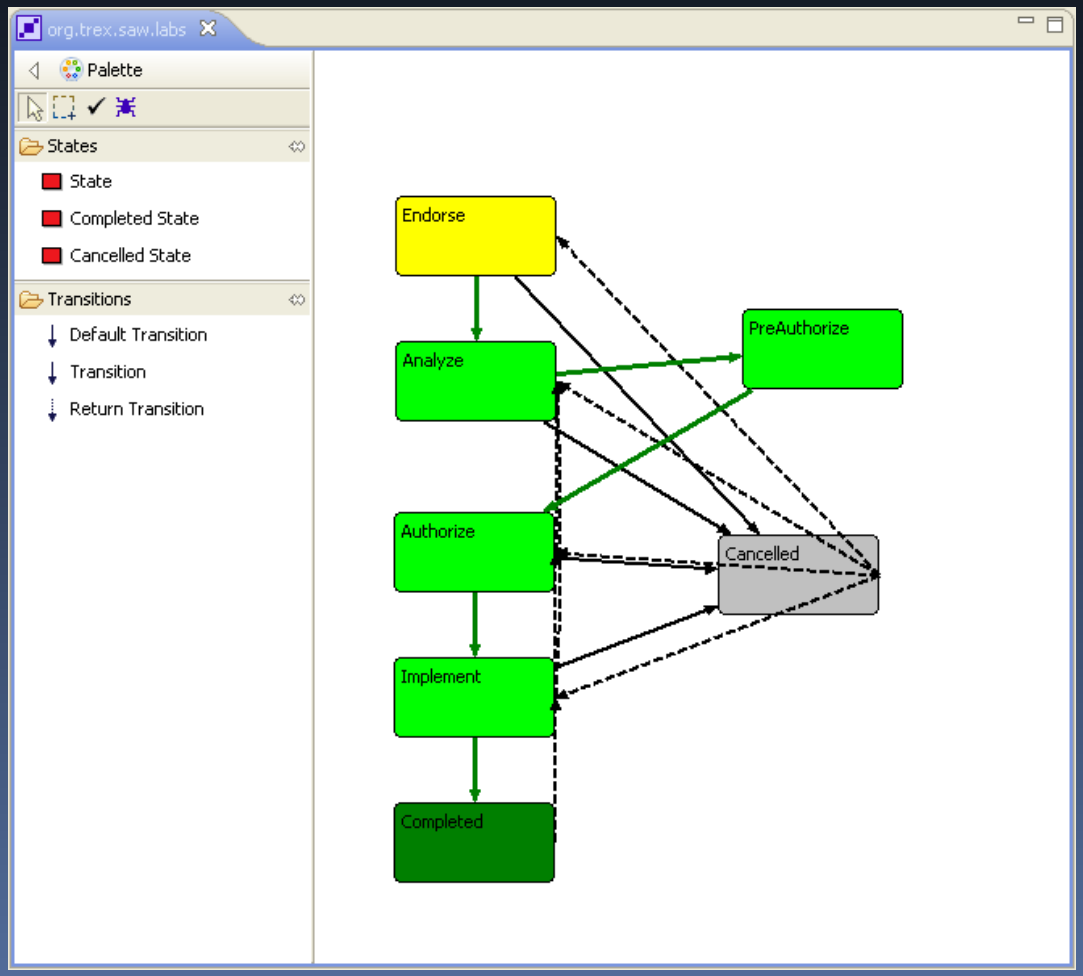


Set new Labs team workflow config

- Double-click “SAW Lab” Team Definition
- Switch to Relations tab
- Expand “Work Item”
- Select item “osee.ats.teamWorkflow” and right-click – “Delete Relation”
- From Artifact Explorer, drag workflow “org.trex.saw.labs” to Work Item – Child
- Save
- Restart OSEE

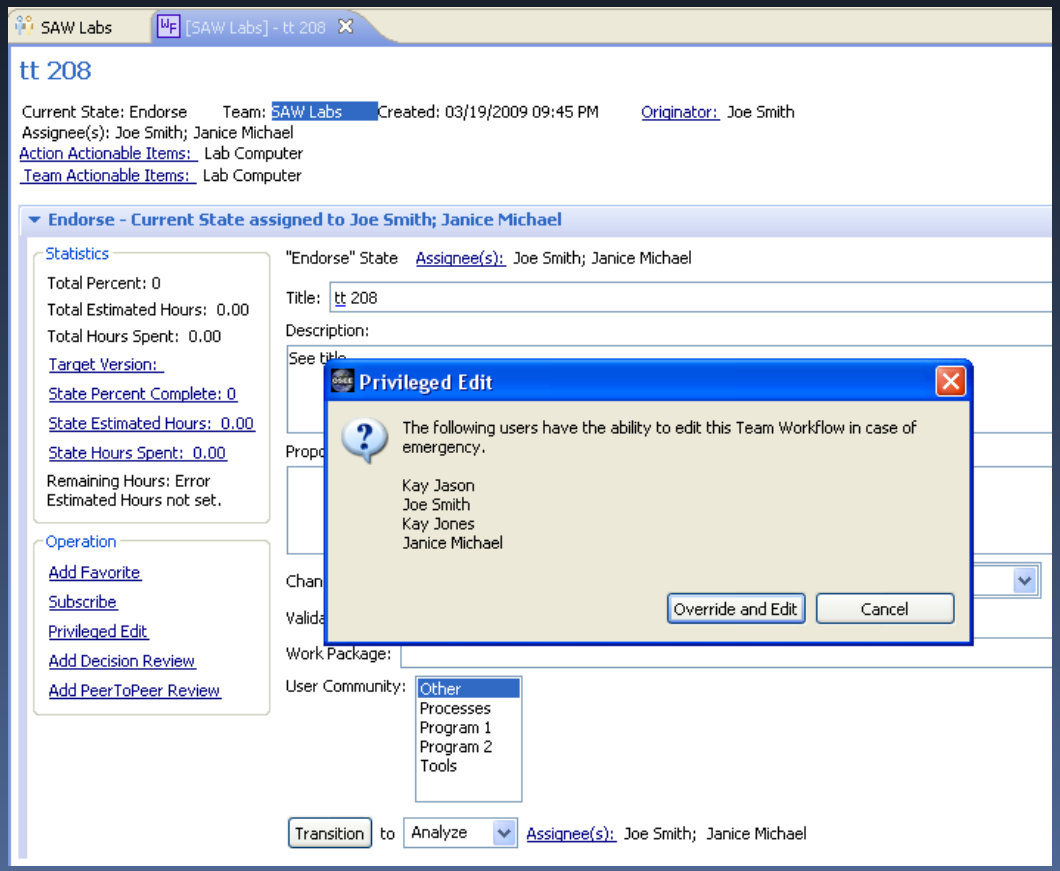
Add a PreAuthorize state…

- Click “State” and click in editor area
- Add “Default Transition” from Analyze to PreAuthorize and from PreAuthorize to Authorize
- In Properties view, change state name to PreAuthorize (Note: id will be updated automatically)
- Save



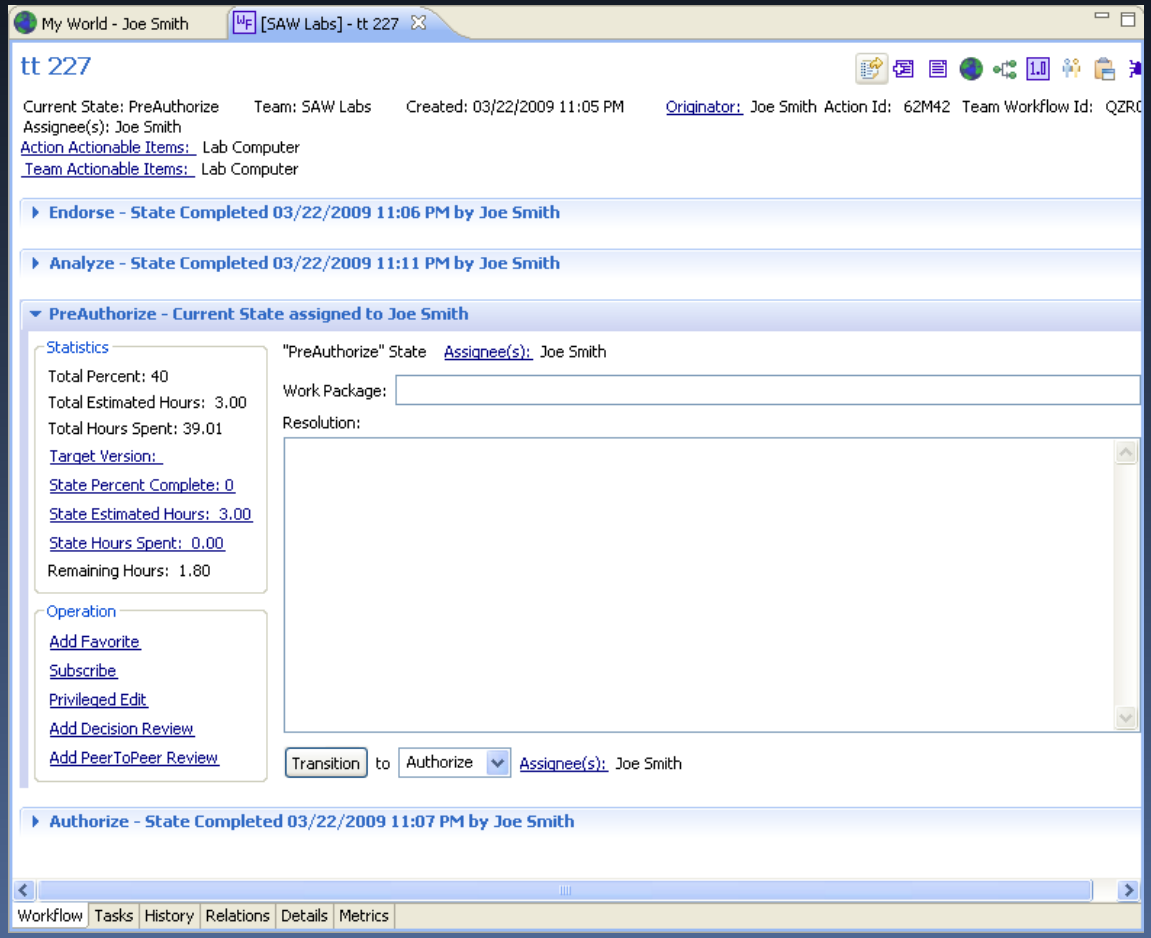
Let's try new PreAuthorize state

- ATS Navigator – New Action – Lab Computer Actionable Item
- Complete Action Wizard
- Transition through to "Implement" state and note that new PreAuthorize state exists



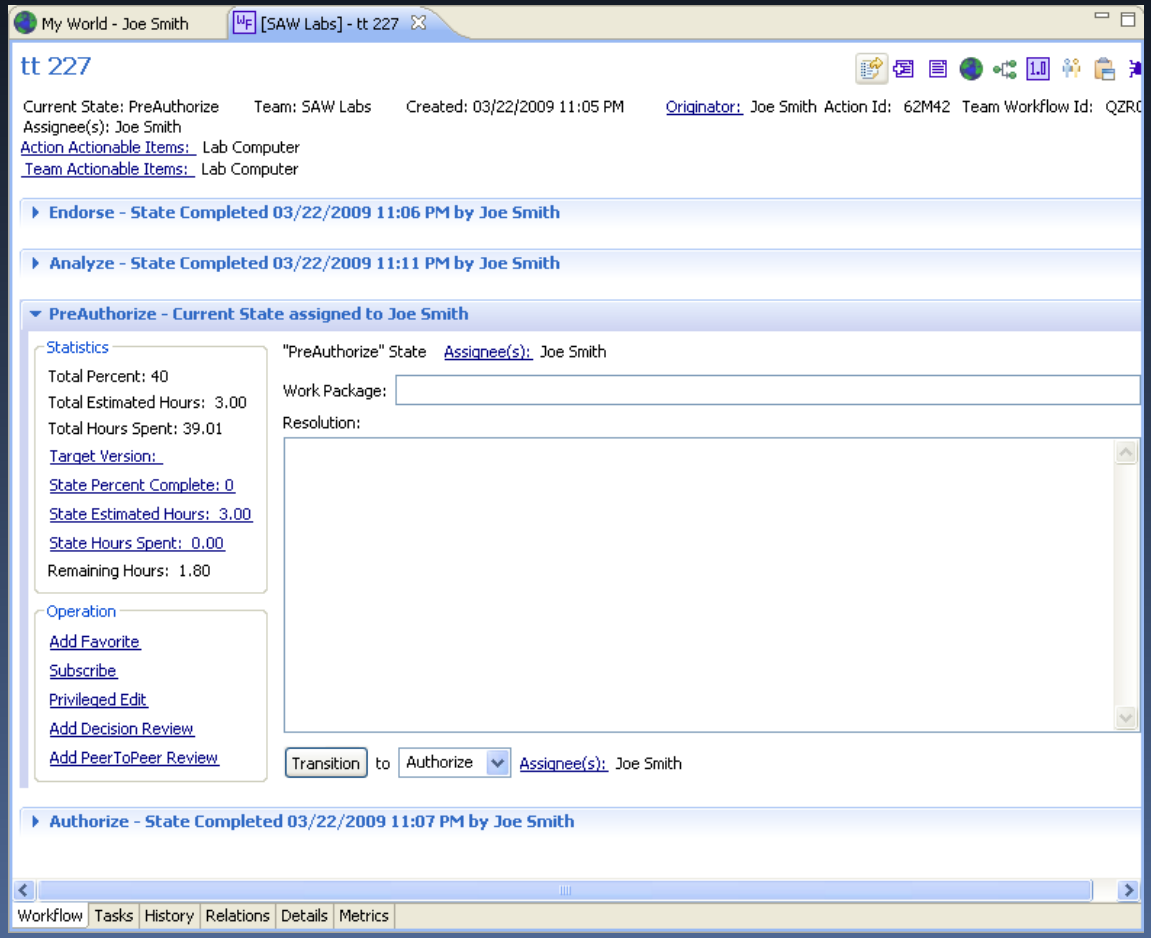
PreAuthorize needs some widgets

- Artifact Explorer
- Double-click “org.trex.saw.labs.PreAuthorize” “Work Page”
- Switch to Relations tab
- Drag in Resolution and Work Package widgets from “Work Widgets”
- Restart OSEE and open Action from “My World”



PreAuthorize needs a Rule

- Artifact Explorer
- Double-click “org.trex.saw.labs.PreAuthorize” “Work Page”
- Switch to Relations tab
- Drag in “atsRequireTargetedVersion” rule from “Work Rules”
- Restart OSEE and open Action from “My World”
- Transition back to Analyze, then PreAuthorize, then Implement – Get stopped until set “Targeted Version”



Advanced customization / extensibility

- New Attributes can be added to Team Workflow artifacts
- New Widgets can be created
  - Simple – attribute with existing XWidget xml
  - Advanced – new XWidget with attribute or other storage (artifacts, relations, etc…)
- New Rules can be created
- Teams and States can have java backed algorithms that enforce or automate tasks (eg: emailing team leads, requiring review if estimated hours > 30, etc…)

Other Extensibility

- Provide customized editors for artifacts
- Attribute Data Providers
- Renderers
- Indexed based taggers
- Authentication Protocols
- Resource Management Protocols
- Artifact Types and Factories
- Customized Dictionaries
- XWidget Providers

We have made it to the end!

- Feedback Questionnaire
- See you tonight at 7:30 pm - Great America 2 for Birds of a Feather
- Other OSEE Talks at EclipseCon 2009
  - "XViewer - An SWT Widget with the power of the spreadsheet"
    - Wednesday Mar. 25th 4:50 pm - Room 203/204
  - "An Integrated Test Environment for Systems Engineering"
    - Wednesday Mar. 25th 11:30 pm - Room 203/204
  - "Unlocking the OSEE Core Framework"
    - Thursday Mar. 26th 10:40 am – Grand Ballroom B

- For further help with OSEE
  - http://www.eclipse.org/osee
    - Newsgroup (Do not use the Mailing List)
    - Documentation
    - FAQs

Legal Notices

- Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

- Other company, product, or service names may be trademarks or service marks of others

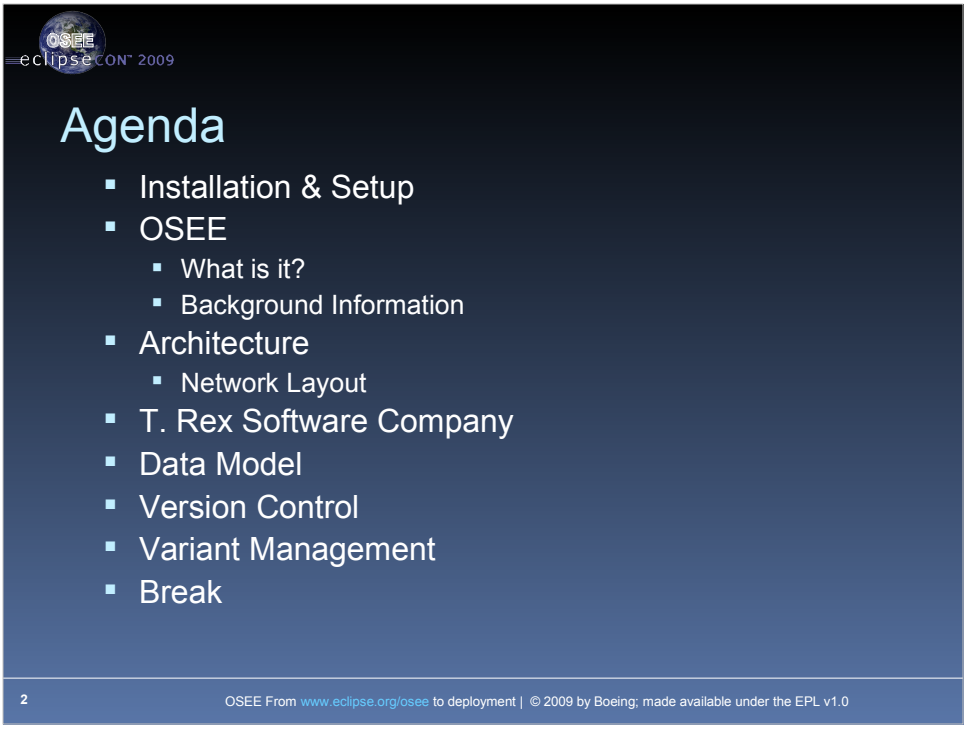
- Show diagram of each team's workflow
- - Diagram of Action, Workflows, Reviews, Tasks
- - Diagram of Team Definitions and Actionable Items
- - Work Flows, Pages, Widgets, Rules
- - Versions
- - Groups
- - Users / User Groups

- - Configuration of ATS
- - ATS Configuration
- - New -> Other -> OSEE ATS -> ATS Configuration
- - Configuration Namespace: org.myCompany.labs
- - Team Definition Name: Labs Team
- - Actionable Item(s): Lab 1, Lab 2, Lab Door, Lab Computer
- - Versions:
- - Workflow Id: osee.ats.defaultTeamWorkflow

From www.eclipse.org/osee to deployment
Tutorial

Ryan Brooks
Donald Dunne
Roberto Escobar

Boeing
Mesa, AZ



Hand-out Flash Drives

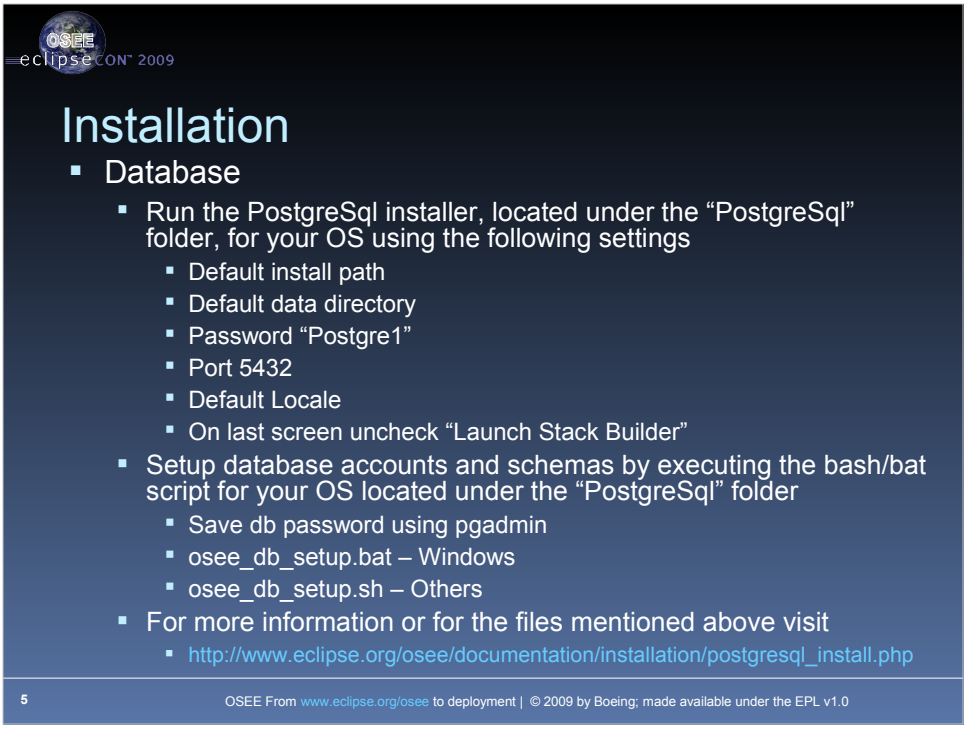
Ask people to follow readme.

Before we begin…

- This is an interactive session: feel free to ask questions
- Tell us about yourself
  - Background Info
  - Operating Systems
- Tutorial based on OSEE 0.7.0

Requirements

- System Requirements (non-eclipse)
  - System with at least 1GB of RAM
  - Java Runtime Environment (JRE) 1.6 or higher
  - Microsoft Office (For Demo Only)
- Eclipse Dependencies
  - Eclipse 3.4.2 SDK
  - org.eclipse.gef
  - org.eclipse.draw2d
  - org.eclipse.birt
  - org.eclipse.datatools
  - The easiest solution is the Ganymede Eclipse install Eclipse IDE for Java and Report Developers



If you already have postgres installed and wish to uninstall it, you will also want to delete the windows user called postgres using the command: net user postgres /delete

Copy pgpass.conf into <user home>\Applicationdata\postgresql

Installation - continued

- JRE
  - Ensure the JRE 1.6 is in the path by typing java –version at a command prompt
- OSEE Client
  - Extract the Eclipse base zip for your OS located under “Eclipse Base” to a short path and then launch eclipse
  - From the Eclipse update manager, install the update sites located under “OseeClient”
    - org.ecilpse.osee_integration_build_incubation.zip
    - osee.add.ons.updatesite.zip

Initialization

- Launch Application Server
  - Execute the launch script for your OS located under "OseeApplicationServer"
    - osee_app_server.bat – Windows
    - osee_app_server.sh - Others
  - Wait until the server finishes the start up procedure
    - Do not close the console
- Database Initialization
  - In a command prompt change to the eclipse install dir

```
eclipsec -application org.eclipse.osee.framework.database.configClient -vmargs -Xmx512m
-Dosee.log.default=INFO -Dosee.application.server=http://localhost:8089
-Dosee.authentication.protocol=trustAll -Dosee.prompt.on.db.init=false
-Dosee.choice.on.db.init="OSEE Demo Database"
```

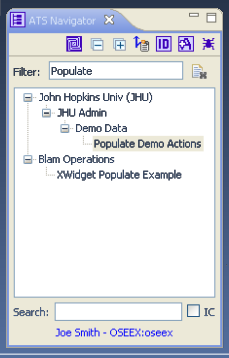
  - Once db init completes, type exit in the server console

Populate Demo Data

- Launch Application Server
- Launch OSEE Client

```
eclipse -vmargs -Xmx512m
-Dosee.application.server=http://localhost:8089
-Dosee.authentication.protocol=demo
```

- Switch to the ATS Perspective
  1. Window
  2. Open Perspective
  3. ATS
     - In the ATS Navigator Window
       - Type **"Populate"** in the Filter text box
       - Press **'Enter'** to add filter
       - Double-click on the **"Populate Demo Actions" item**
       - **Wait for operation to complete**



What is OSEE?

- OSEE is a tightly integrated environment designed to support lean engineering principles across a product's full life-cycle in the context of an overall systems engineering approach.

Background

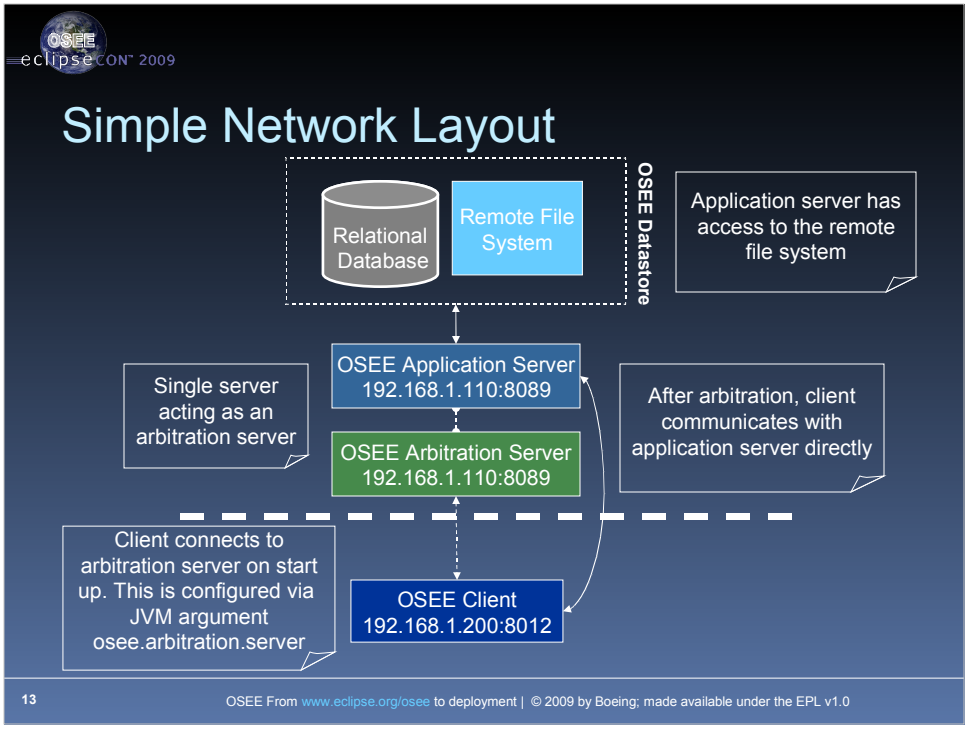
- OSEE began was first deployed to develop Boeing's next generation Apache Helicopter
- It provides
  - An integrated tool set
  - End-to-end traceability
  - Variant configuration management
  - Integrated workflows and processes
  - A Comprehensive issue tracking system
  - Deliverable document generation
  - Real-time project tracking and reporting
  - Validation and verification of mission software

Background - continued

- As an eclipse project
  - Milestones
    - Initial source committed on Dec 8, 2007 (~140K LOC)
    - Project proposal approved on July 10, 2007
    - Test environment framework submitted Spring 2009 (37K LOC)
  - OSEE is used to engineer itself

Architecture

- Initially OSEE was architected as a heavy client
  - Direct client-to-database interactions
- Migrating into a Thin-Client/Server architecture
  - Utilizing OSGI on server-side
  - Flexible deployment and maintenance
  - Address scalability and load management

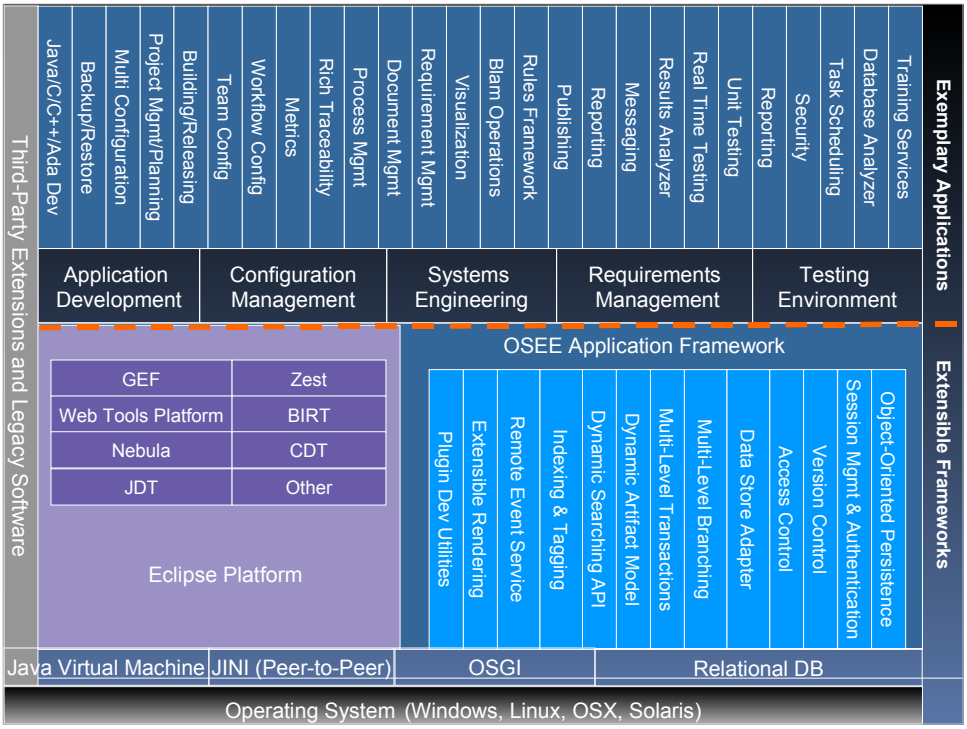


We decided to use a remote file system for binary data storage – relational database performance was a consideration for the decision.

For this demo we are running all components on the same machine.

We can have each system running on a different PC – including the database and the remote file system

Only restriction is that an application server instance must have access to the data store components.



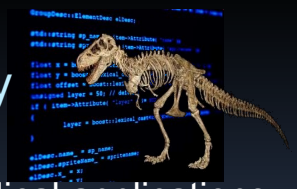
OSEE's Architecture extends the Eclipse Platform and provides it's own extensions for use by the Applications that are built on top.

OSEE Application Framework provides the services necessary for the applications to share the common data model

Custom, Non-Engineering Environment Applications can be built on OSEE Application Framework and delivered to customers

OR Additional Applications can be plugged in to enhance OSEE's already existing applications

T. Rex Software Company

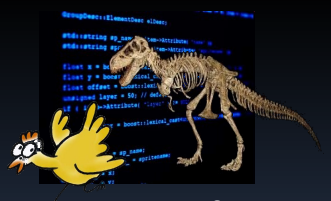


- 15 years in the business
- Developing software for medical applications
- Waterfall development cycle
- Isolated teams using disconnected tools to track issues
  - Requirements - using spreadsheets
  - Code – problem change report database
  - Test – spreadsheets and emails
- Status is reported weekly via emails to team leads who then flow information to project managers

T. Rex Software Company

- Inefficiencies/Cost
  - Software license fees
  - Steep learning curve
  - Data redundancies
  - Weak knowledge management
  - Issues fall through the cracks
  - Poor Planning
  - Lagging metrics

SAW-TSR Project



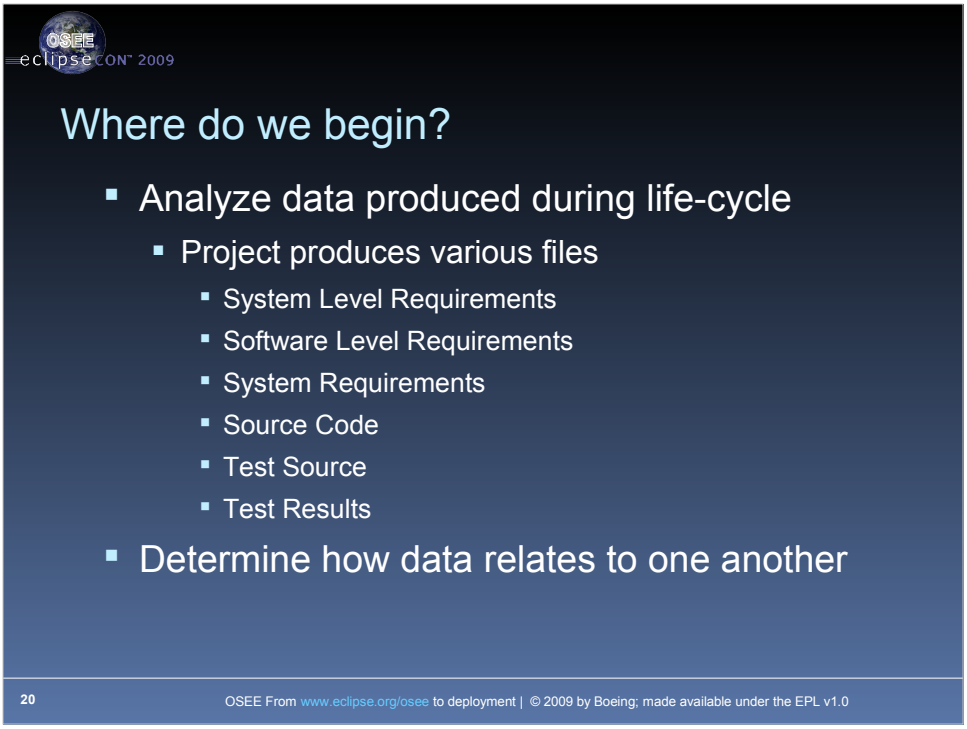
- Project with many challenges
- Must develop a Surgical Assistant Workstation for Tele-operated Surgical Robots (SAWTSR)
- Project was 40% under funded
- Stringent requirements on software quality
  - Medical application
  - Severe consequences of faulty software
- Project must meet a company wide initiative to streamline processes
- If T. Rex does not make a change, the contract will be lost

Why should T. Rex use OSEE?

- Full life-cycle engineering environment
- Open source extensible platform
- Benefits of eclipse community
- Tightly integrated
  - Common data model
  - Version control
  - Change management
  - Workflows and processes
  - Supports multiple databases
- Zero license cost

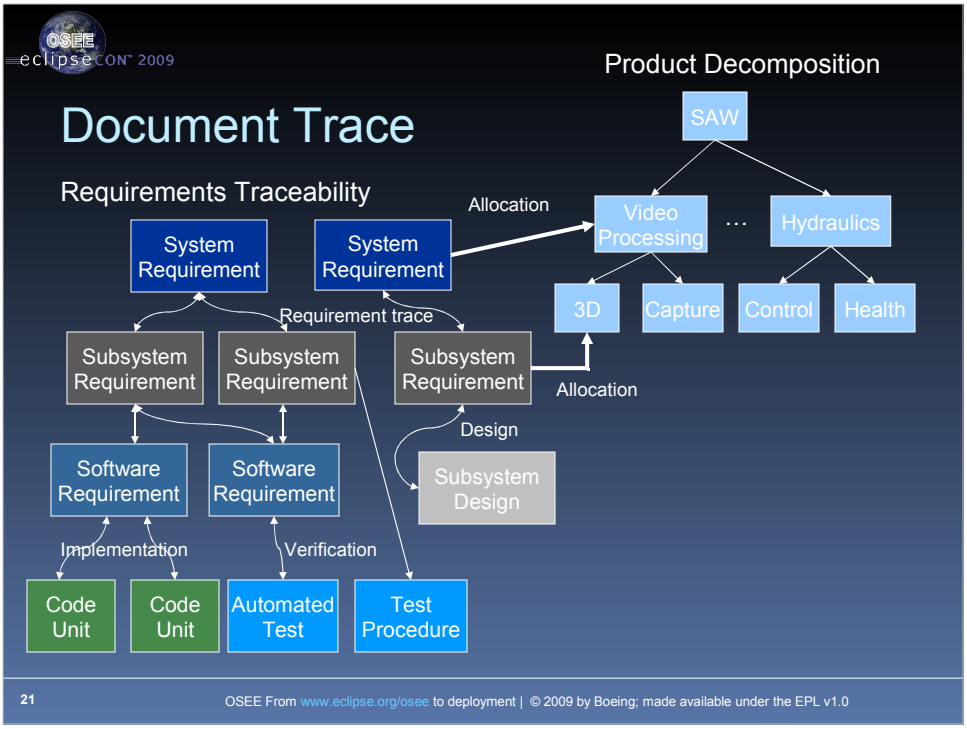
Company is ready to evolve!

Management decides to use OSEE



Windows File Folders

Data Sources



Product Decomposition

Understanding the OSEE Data Model

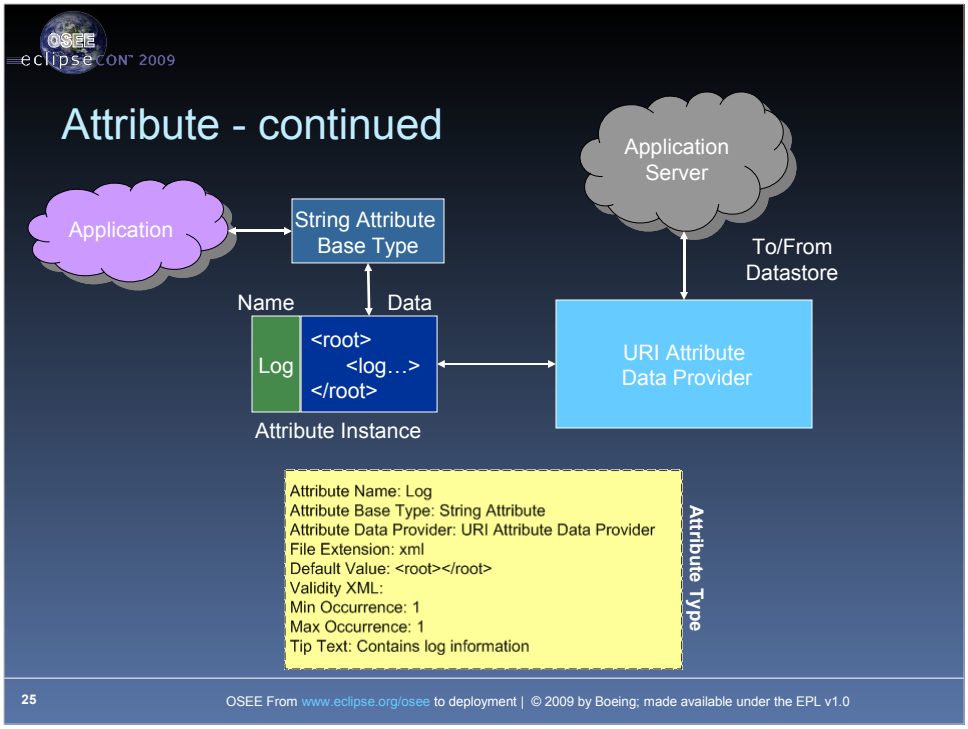
- Artifact
  - Main OSEE data object consisting of attributes
  - Artifact type - blue print for instance creation
  - Artifact types can inherit from one another
  - All artifacts inherit from “Artifact” (similar to Object in Java)

Attribute

- Key / Value pair representing a single data element
- Attribute type - blue print for instance creation
- Attribute base type is used to convert raw data into a native type or other object
- Attribute base types
  - String
    - Word Templated Content
    - Word Whole Document
  - Date
  - Boolean
  - Integer
  - Floating Point
  - Enumeration

Attribute - continued

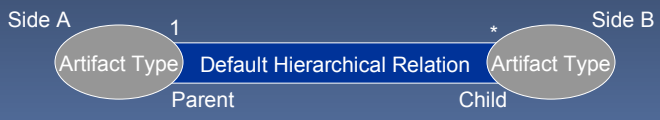
- Attribute backing data managed by an attribute data provider
  - Transfers data between client/server
  - Can be extended to serve data from outside OSEE data store
- Attribute type also defines whether the attribute is searchable and how to tag the data



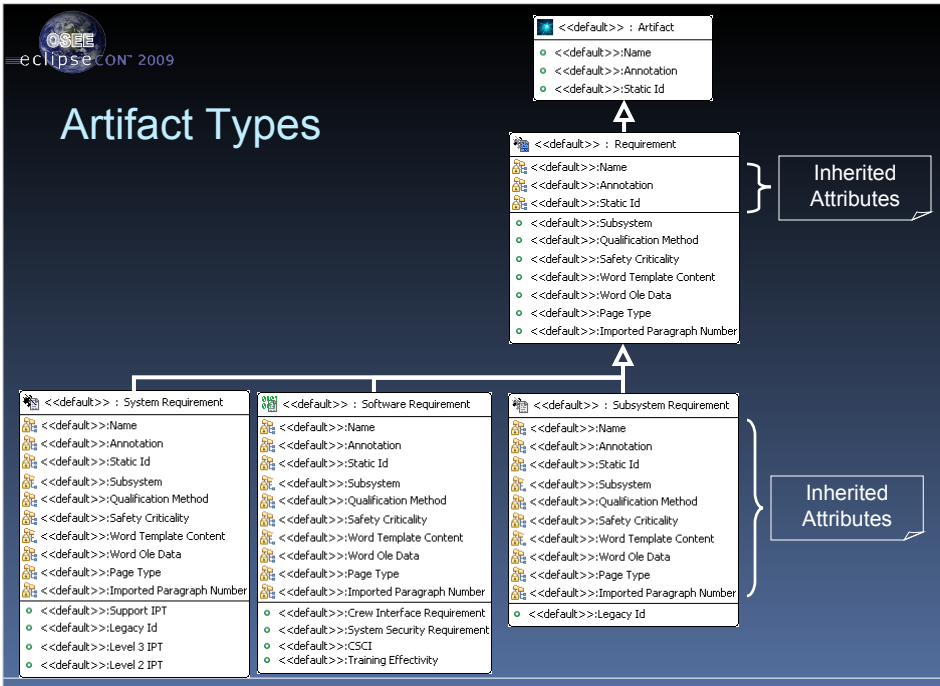
Attribute containing XML data transferred to/from datastore – converted to correct type via base type.

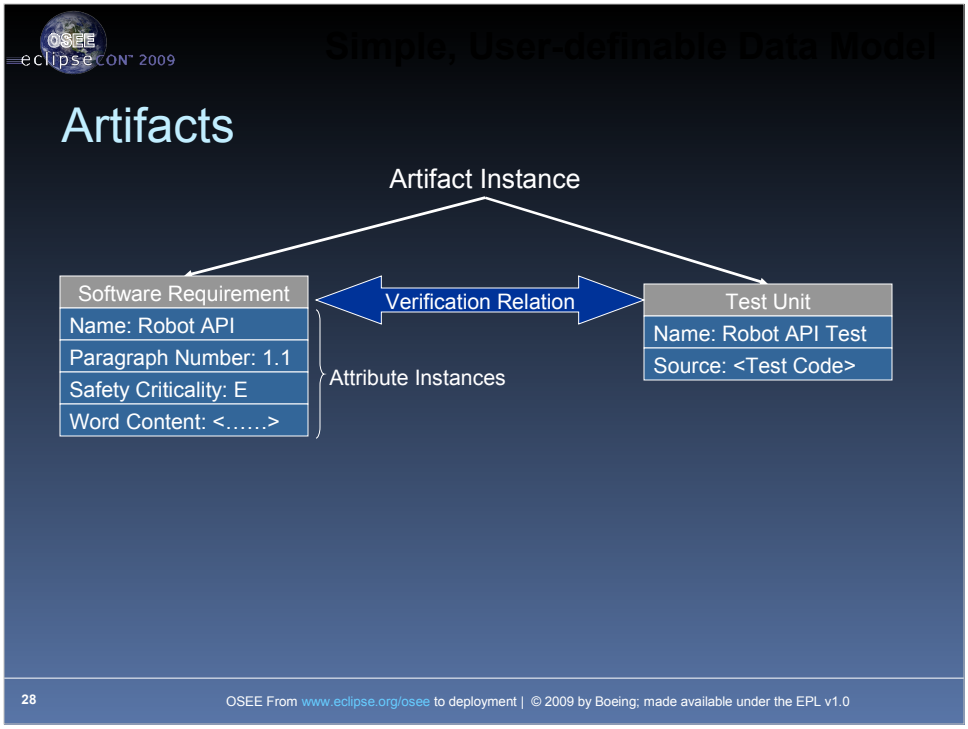
Relation

- Relates two or more artifact types
- Relation type - blue print for instance creation
  - Relation type
    - What type allowed on side A
    - What type allowed on side B
    - Multiplicity (1 to 1, 1 to Many, Many to Many)



Artifact Types





The main components of the data model are:

Artifact – data objects

Attribute – state data on artifacts

Relations – Link artifacts to one another

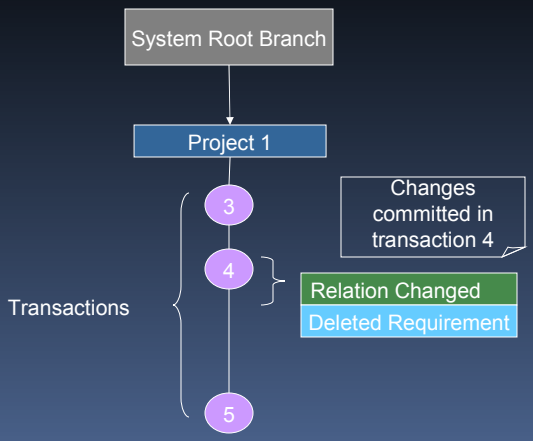
All data types are strongly type and user definable

Version Control

- Changes to artifacts, attributes, and relations are tracked by the system
- Changes are managed by a transaction based version control system using fine grained change identification
- Data managed under branches

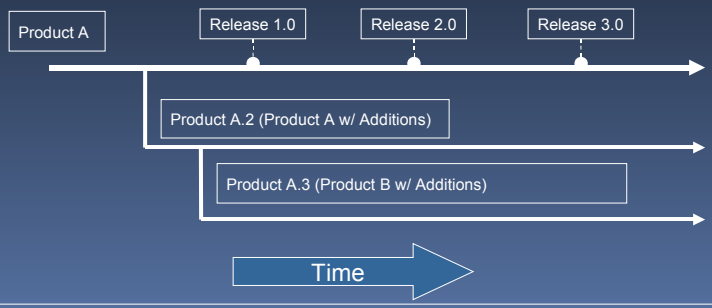
Transaction

Branch



Variant Management

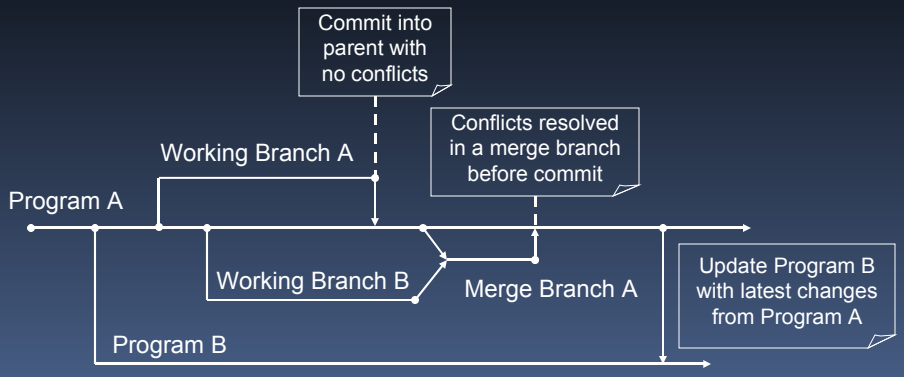
- Product lines share common baseline data throughout all variants
- Changes to the baseline product can be easily merged to the variant product lines

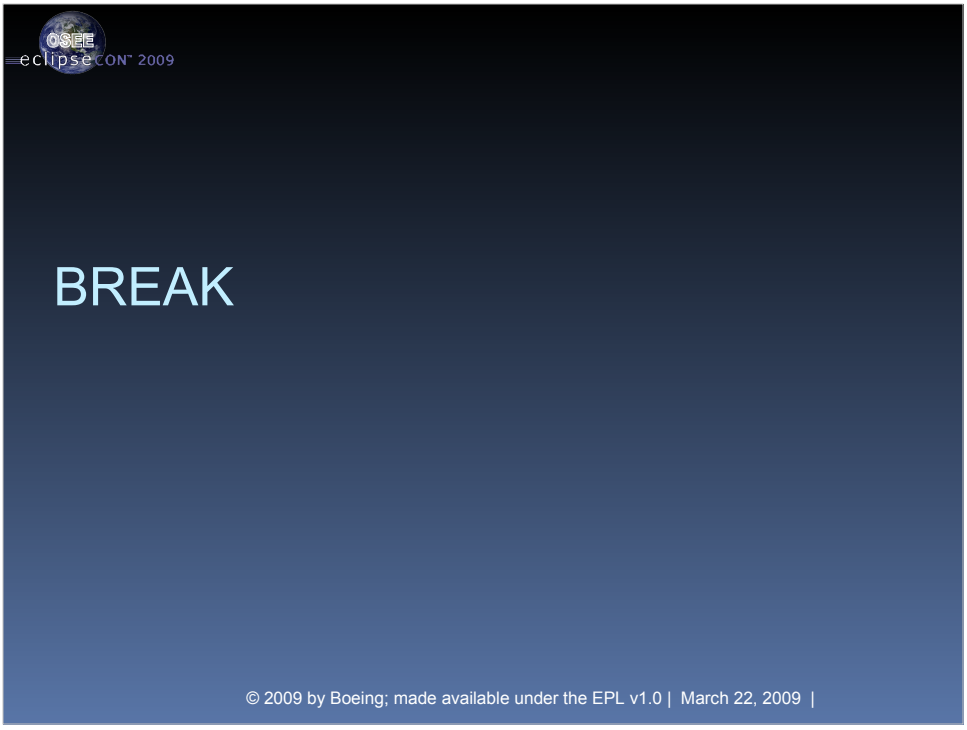


Branching

- Branching
  - Used to create a variant of the parent branch
  - Updates can be performed to obtain changes from the parent branch into the child
- A variant branch can be committed back into its parent
  - Change conflicts are resolved via merging

Branching - continued




OSEE
eclipseCON 2009
BREAK
© 2009 by Boeing; made available under the EPL v1.0 | March 22, 2009 |

From www.eclipse.org/osee to deployment
Tutorial (Part II: All in a days work)

Ryan Brooks
Donald Dunne
Roberto Escobar

Boeing
Mesa, AZ

Agenda

- T. Rex with OSEE
  - Products & Teams
  - Variants
  - Scenario Roles
  - Project Workflows
- Scenario: All in a days work
  - Search Requirements
  - Create Action
  - Requirement Team Workflow
    - Add Decision Review
    - Change Implementation
  - Code Team Workflow
    - Add Tasks
  - Test Team Workflow
    - Privileged Edit
    - Status
  - Create Peer-To-Peer Review
  - Add new Workflow

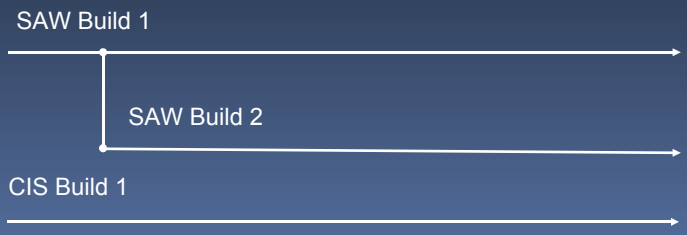
Project has been using OSEE for a couple of months now

Products & Teams

- SAW – Surgical Assistant Workstation –
  - **Teams:** Requirements, Code, Test, HW, SW Design
- CIS – Dummy Project
  - **Teams:** Requirements, Code, Test, SW Design
- Facilities Team
- IT Team
  - Computers, Backups, Network
- Tools Team
- Website Team
- Processes Team

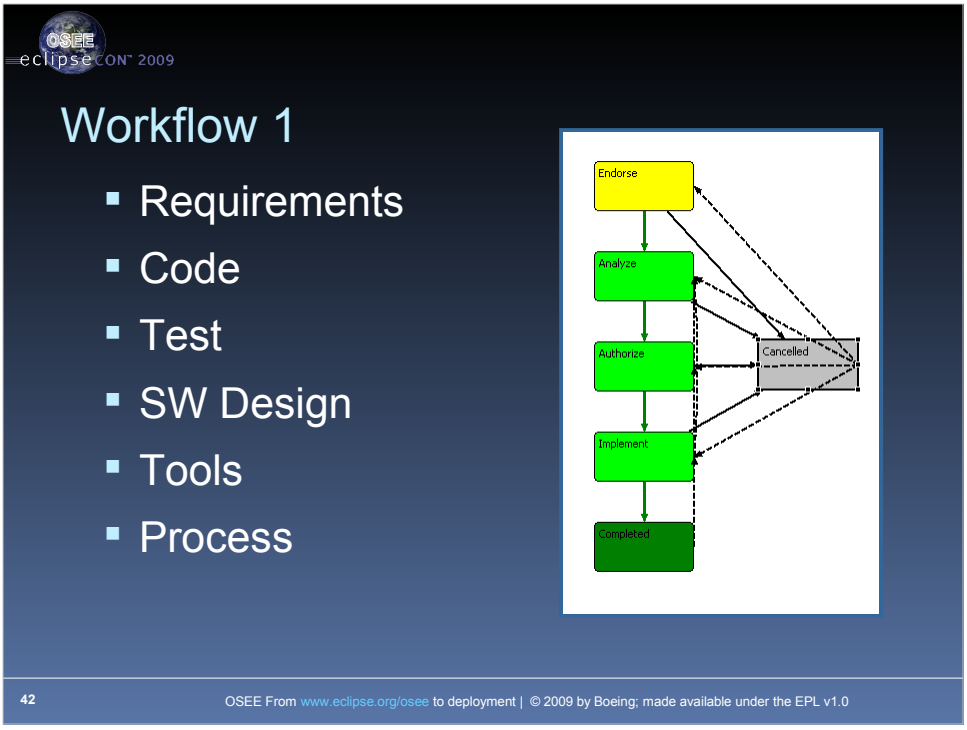
Project Variants

- Surgical Assistant Workstation - SAW
  - Build 1
  - Build 2
- CIS Build 1 – Dummy Project
- Work is performed in working branches which are then committed back into their respective parent branch



Roles

- Project Manager
  - Requirement
    - Lead
    - Developer
  - Code
    - Lead
    - Developer
  - Test
    - Lead
    - Developer

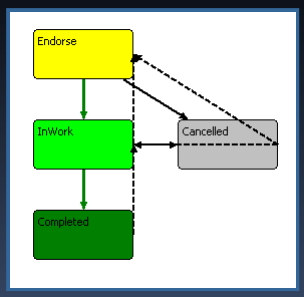


Why move all action, issue, problem tracking to ATS?

1. Everyone has access to all data
2. As issue/problem is analyzed, other teams can be added to Action
3. Each team has it's own independent workflow
4. Workflows can be simple or complex
5. One interface = Low cost of setup, deployment, training and admin

Workflow 2

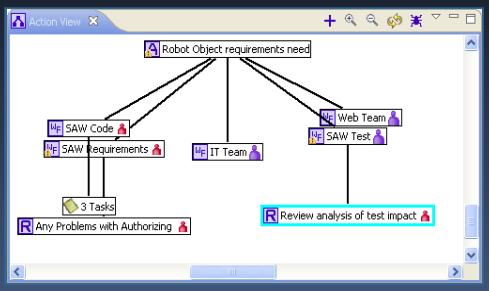
- Web
- Facilities



TODO - Whats CIS stand for?

Scenario

- Joe Smith finds a problem in a requirement impacting the following teams:
  - Code
  - Test
  - Website
  - IT



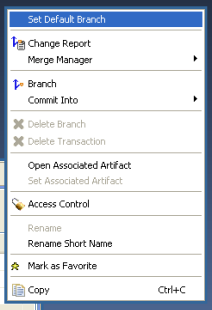
- Reviews needed:
  - Decision Review off Code Team Workflow
  - Peer Review off Test Team Workflow
- Perform tasks off Code Team Workflow

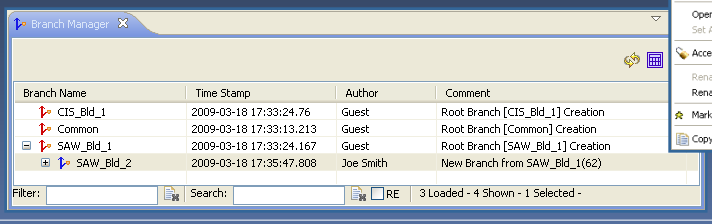
Notes: You're welcome to follow along and perform the steps

- Limited time, so need to keep moving
- Will just explain some of the things going on versus showing due to time
- Will operate as more than one user, but will always say Joe Smith
- Will operate on single computer, this is multi-user/computer system with notifications

User searches for a requirement

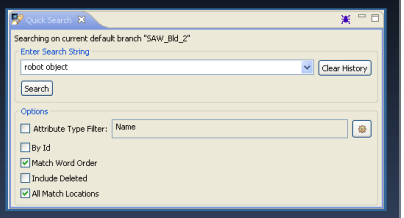
- Switch to the Define perspective
  - Window->Open Perspective->Define
- Set “SAW_Bld_2” as the default branch
  - Click on the Branch Manager View
    - If the view is not open
      - Select Window->Show View->Branch Manager
  - Right-Click on “SAW_Bld_2”
  - Select “Set Default Branch”

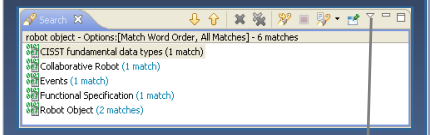




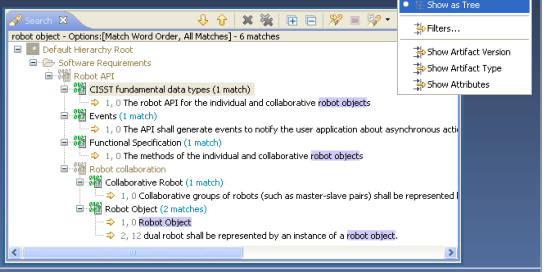
User searches for a requirement

- Search for Item
  - Click on the Quick Search View
  - Enter “robot object” in the search string text box
  - Check the “Match Word Order” option
  - Check the “All Match Locations” option
  - Click the search button
- Search results as list or tree view with match locations





Select View

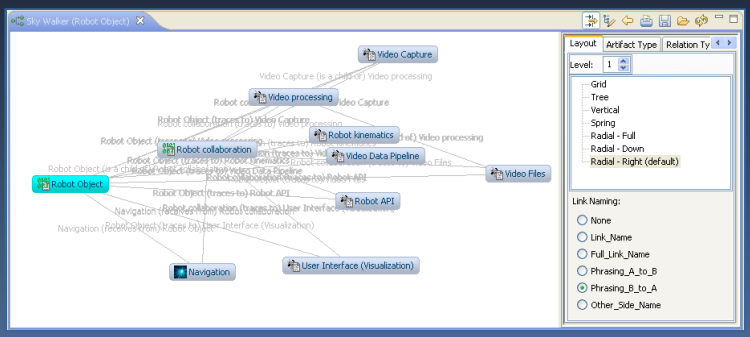


User explores the robot object requirement

- From the search results view
  - Right-Click on the “Robot Object” software requirement
  - Select each of the following from the pop-up menu individually
    - Reveal Artifact in Explorer
    - Resource History
    - Open With Artifact Editor
      - Click on the attribute tab
      - Click on the relation tab
    - Open With MS Word Preview
    - Sky Walker

User explores the robot object requirement

- Robot Object (software requirement) relates to other artifacts

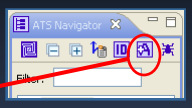


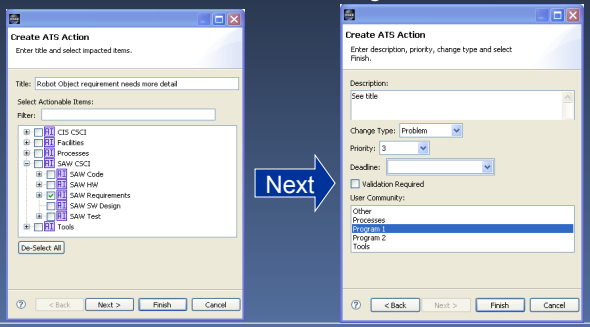
While looking through the robot object requirement, the user notices a problem.

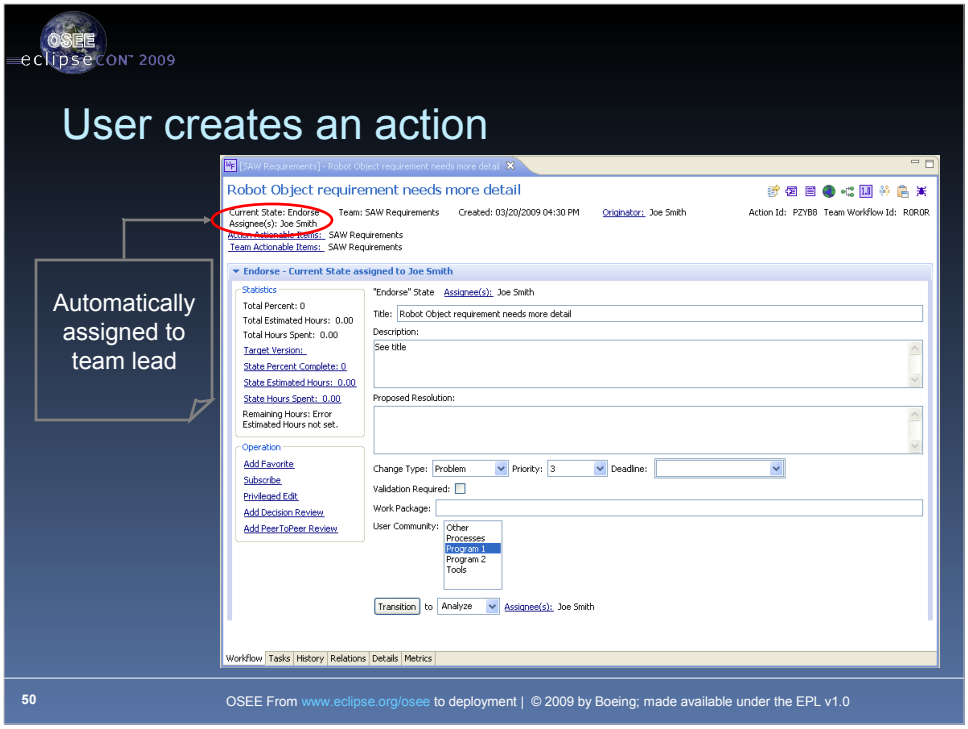
An action will need to be written against it.

User creates an action

- Create an action against the “Robot Object Requirement”
- Switch to the ATS Perspective
  - Window->Open Perspective->ATS
- Click on the New Action icon in ATS Navigator
  - Fill-in the Create ATS Action Dialog:



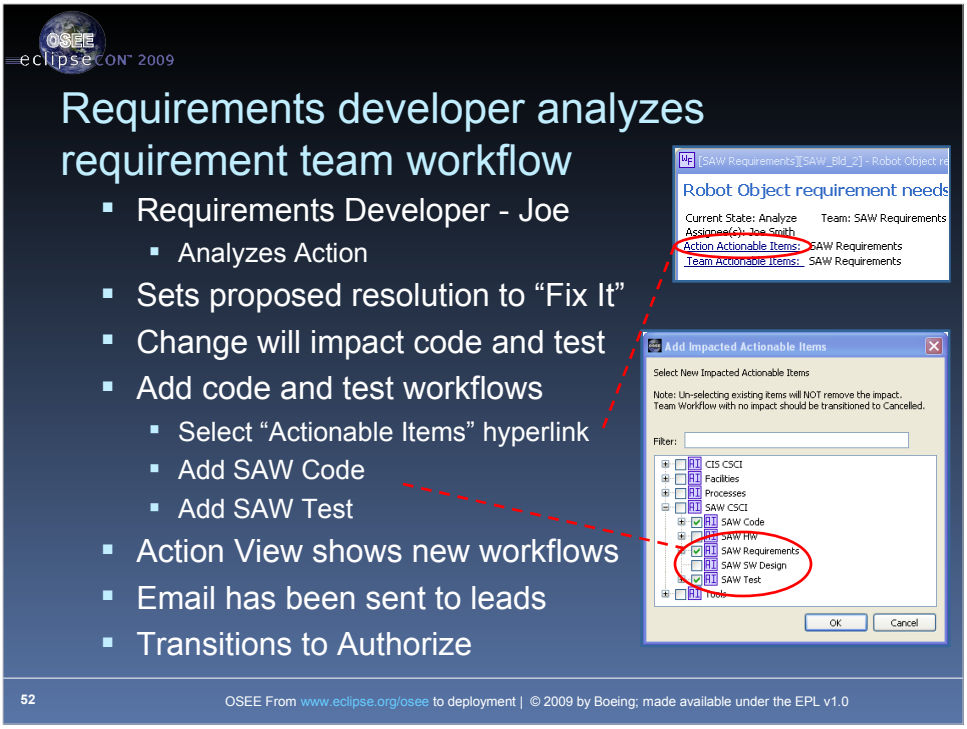




Action created, assigned to Joe Smith (configured Lead for SAW Req Team)

Requirements lead approves requirement team workflow

- Requirements Team Lead - Joe
  - Endorses Action
- Sets Target Version
  - SAW Build 2 (the next version)
- Changes priority to 2
- Transitions state to “Analyze”
  - This is where the lead would normally assign another user to complete the work
  - NOTE: We will not change the assigned user for the demo



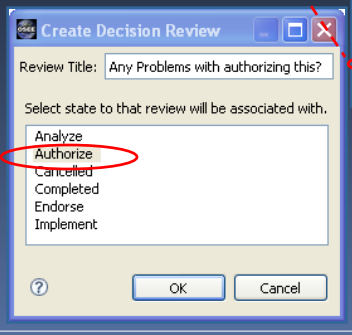
NOTE: Can be reassigned or configured to be auto-assign

Requirements lead begins to authorize the requirement team workflow

- Requirements Team Lead - Joe
  - Authorizes Action
- Sets Work Package to A324324A
- Team lead needs concurrence from Kay (The Manager)
  - A decision review is needed

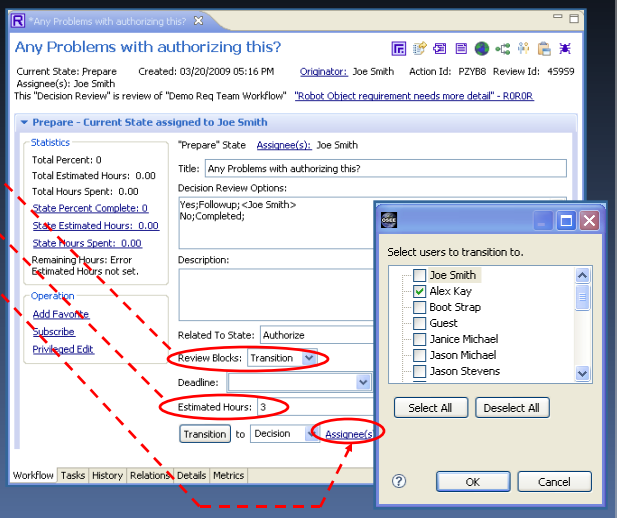
Requirements lead creates a decision review

- Create Decision Review
  - Select Add Decision Review
  - Fill-in the Create Decision review Dialog:



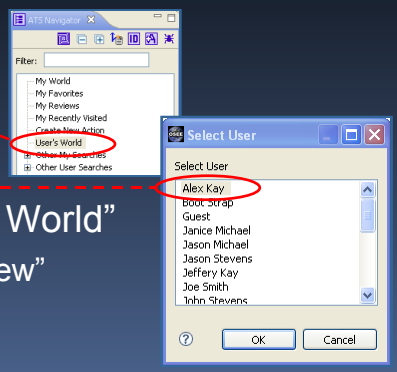
Requirements lead prepares the decision review

- Prepare Review
  - Set Review Blocks to Transition
  - Set Estimated Hours to 3
  - Assign to Alex Kay
  - Transition the review



Manager approves the change

- Alex Kay checks his assigned work
- Runs “My World”
  - Select “User’s World”
  - Select Alex Kay
- From Alex Kay’s “User’s World”
  - Select the “Decision Review”
  - Kay decides Yes
  - Transitions the Review to Completed



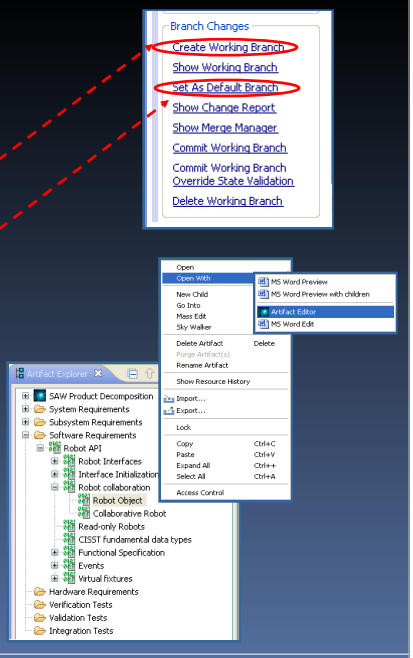
Requirements lead completes authorization

- Sets Estimated Hours to 2.5
- Transition to Implement

Notice that before Alex Kay had completed the decision review, Joe was not able to transition to the next state

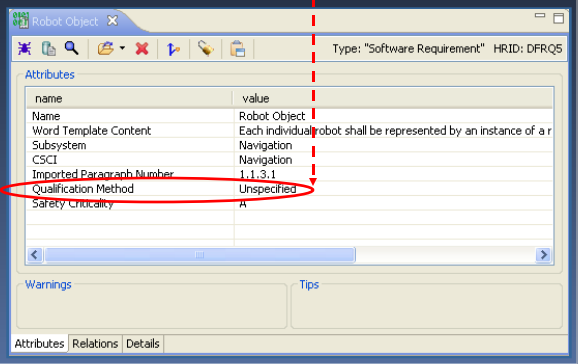
Requirements developer implements a change

- From the Implement State
  - Select “Create Working Branch”
  - Set the working branch as the default branch
- Using Artifact Explorer, navigate to the “Robot Object” software requirement
- Right-Click on “Robot Object”
- From the pop-up menu, select “Open With”
- Select “Artifact Editor”



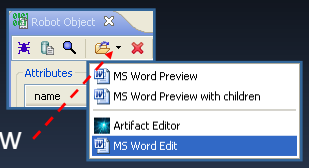
Requirements developer implements a change

- Select the attributes tab
- Change Qualification Method to Inspection
- Save by clicking File->Save



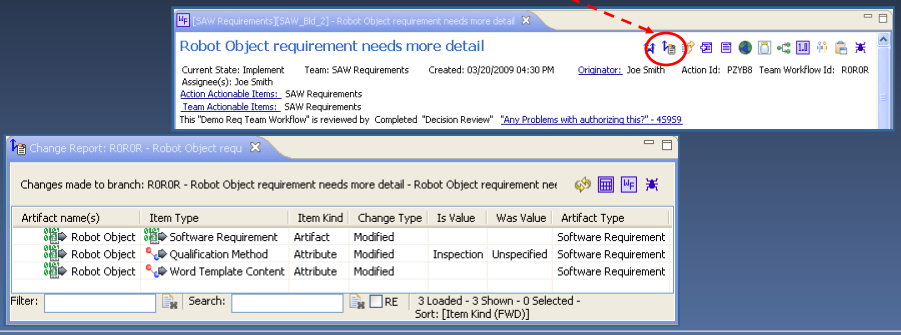
Requirements developer implements a change

- From the “Artifact Editor’s” toolbar
  - Click on “Open With” down arrow
  - Select MS Word Edit
    - NOTE: If you don't have MS Word, just watch
- Insert into word document
  - “Need more information here.”
- Save document and close
  - “Artifact Editor’s” Word Template Content Attribute should update accordingly



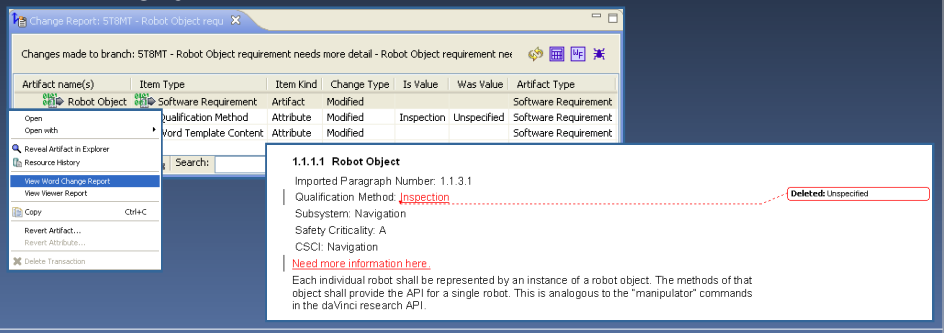
Requirements developer implements a change

- Switch to the Workflow Editor
- Select “Show Change Report” from the tool bar item



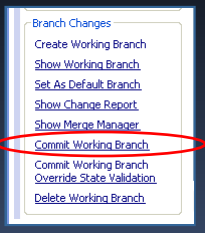
Requirements developer implements a change

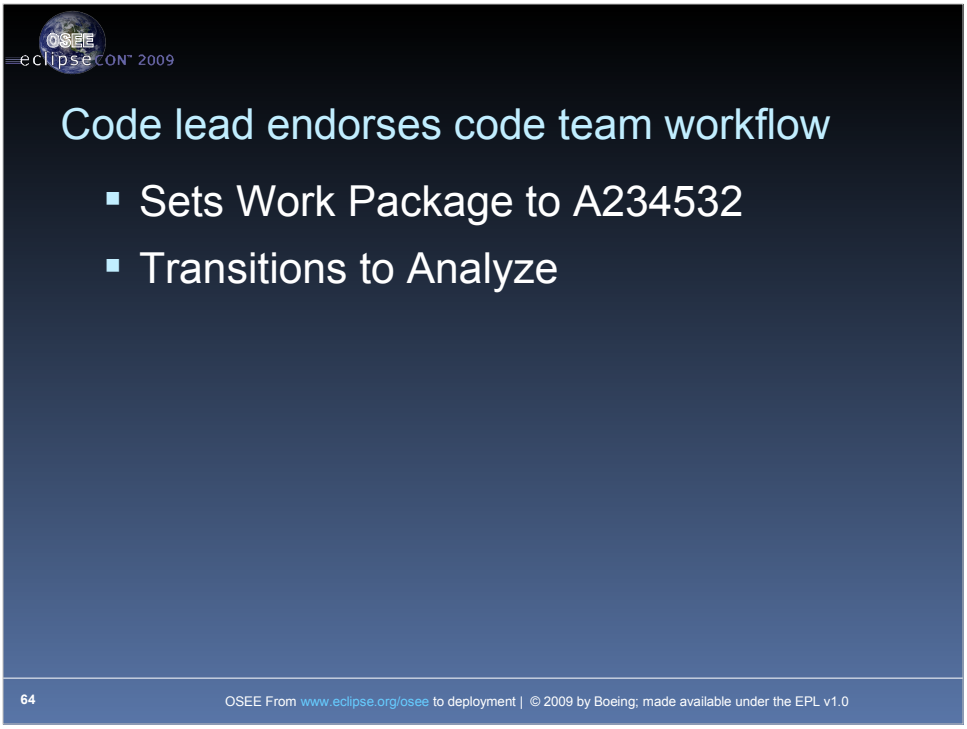
- From the Change Report View
- Right-click on the “Robot Object” software requirement
- Select “View Word Change Report” from the pop-up menu



Requirements developer implements a change

- Select "Commit Working Branch“ to apply changes to the parent branch “SAW_Bld_2”
- Transition to Complete

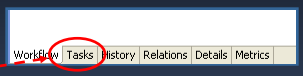


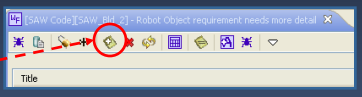


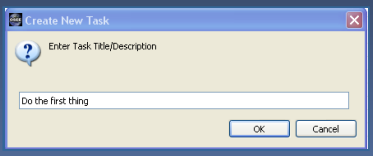
Remind that this was created in the analyze action by the requirements lead

Code developer analyzes team workflow

- Sets Estimated Hours to 10
- Analysis Requires Tasks
  - Switch to the Tasks Page by clicking on the Task Tab
  - Add a Task by clicking on the “New Task” tool bar item
  - In the “Create New Task” dialog enter “Do the first thing”
  - Click “OK” to close the dialog

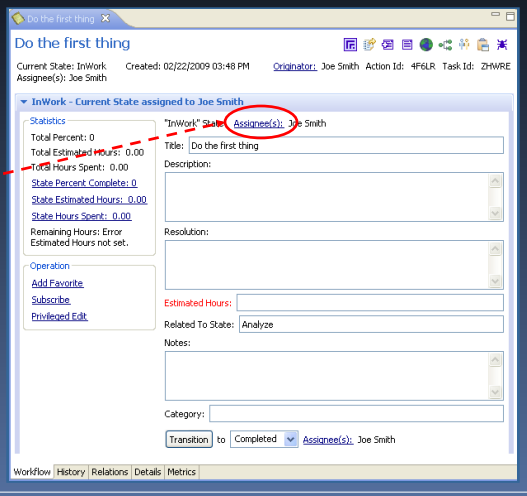


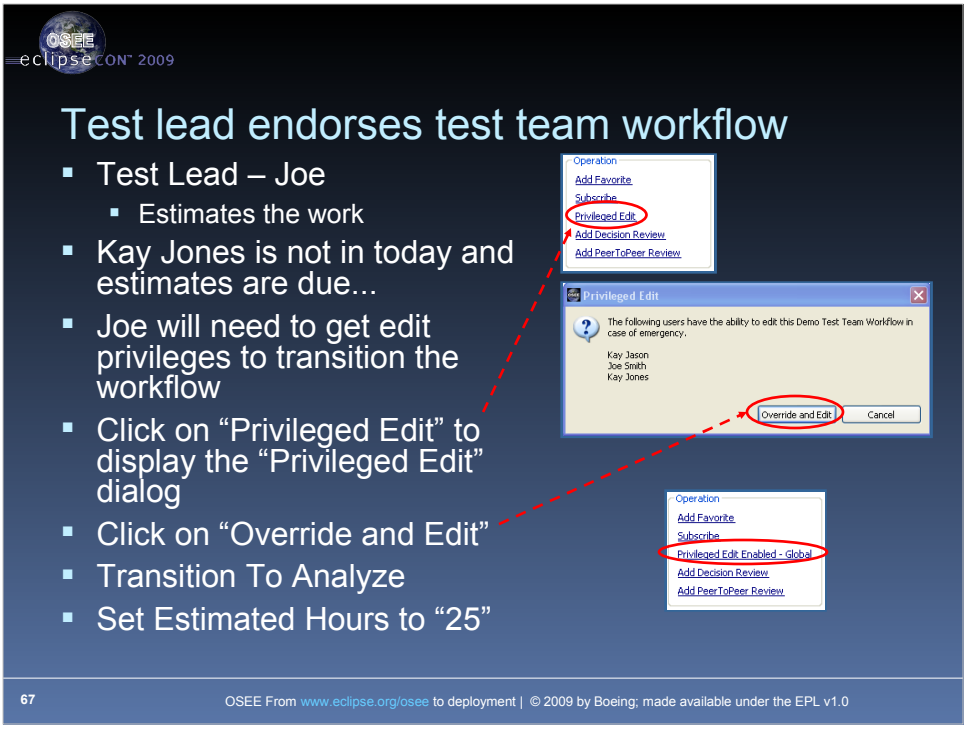




Code developer analyzes team workflow

- Double-click on the new task to open the Task Editor
- Click on Assignee(s) to assign a different user
- Close Task Editor
- Add two more tasks
- Transition code team workflow to Authorize

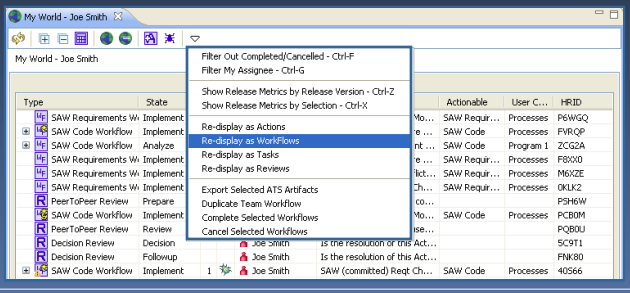




Remind that this was created in the analyze action by the requirements lead

Manager wants status

- Manager – Alex Kay
  - Needs status report
- Select “World”
- Redisplay as workflows

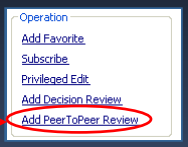


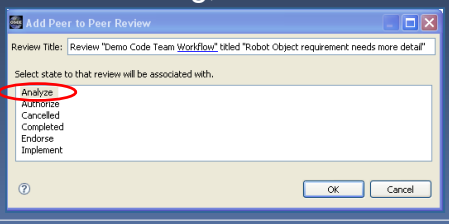
Manager wants status

- Click on “Metrics Tab” to open the Metrics Page
- Set “Estimated Completion Date” to 2 Days from the current date
- Kay won't make it (slacker); Joe will

Code developer adds a peer review

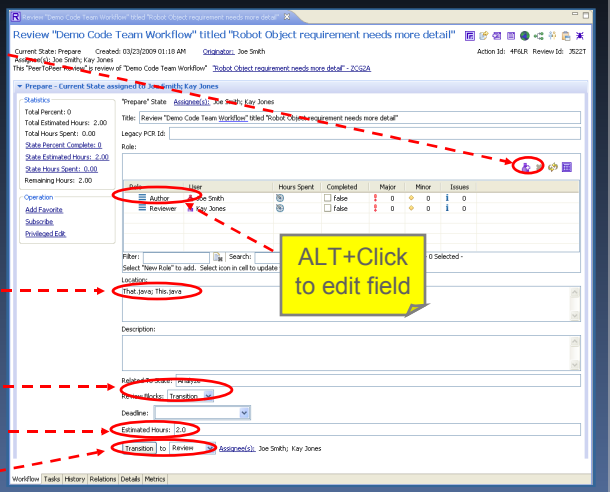
- Code Developer – Joe
  - Realizes he needs a peer review for Analysis
- Add Peer-To-Peer Review
  - Click on the “Add PeerToPeer Review” hyperlink
  - In the “Add Peer to Peer Review” dialog, select “Analyze” state
  - Click “Ok”





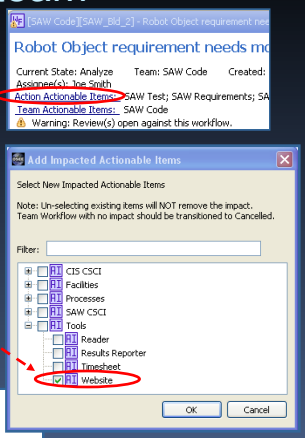
Code developer adds a peer review

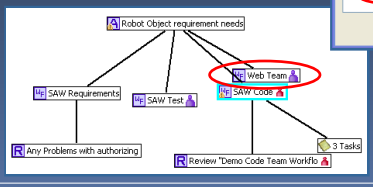
- Click on the New Role tool item to add 2 reviewers
- Set one of the roles to “Author” and the other to “Reviewer”
  - Edit the “Role” field so by clicking on the field while pressing the “ALT” key
- Set Location to “That.java; This.java”
- Set Blocking to “Transition”
- Set Estimated Hours to “2”
- Transition to Review



Code developer creates a tool team workflow

- Code Developer – Joe
  - Realizes the change will affect the Tools Team
- Select the “Action Actionable Items” hyperlink from the SAW Code Workflow
- Select “Website” from the “Add Impacted Actionable Items” dialog
- Workflows are created for the Website Team



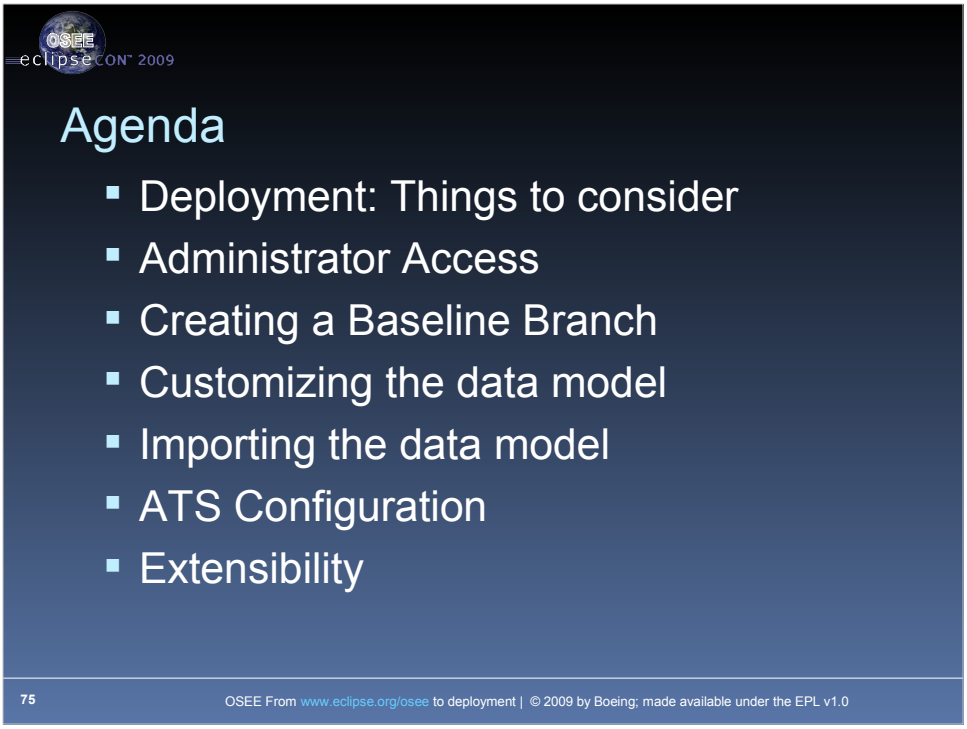


BREAK

From www.eclipse.org/osee to deployment
Tutorial (Part III)

Ryan Brooks
Donald Dunne
Roberto Escobar

Boeing
Mesa, AZ



Hand-out Flash Drives

Ask people to follow readme.

Deployment: Items to Consider

- Remote access to data store
- Hardware considerations
  - Database server
  - Application server
- Requirements
  - Microsoft Office
  - Database Server
    - PostgreSql
    - Oracle Server

Granting Administrator Access

- Set Default Branch to Common
- Open the User Groups Folder
- Open “OseeAdmin” with Artifact Editor
- Click on the Relations Tab
- Perform a Quick Search
  - Check the Attribute Type Option
    - Ensure it is set to Name
  - Uncheck all other options
- Drag and Drop the user “Joe Smith” into the Users relation
- You are now an OSEE Admin

Creating a Baseline Branch

- Ensure you have OSEE Admin privileges
- Refresh the branch manager view
- You should now be able to see the “System Root” branch
- Right-click on the “System Root” branch and select “Branch” from the pop-up menu
- A child branch of the “System Root” branch will be created

Data Model Customization

- The data model in OSEE is extensible and user configurable
- Users can define new artifact, attribute, and relation types and their constraints such as multiplicity and applicability
- Type inheritance allows similar types to be defined and modified without tedious redundancy because similar types inherit what is common from their super type

eclipseCON 2009

Data Model Customization - Spreadsheet

- The OSEE data model is defined using a tabular format involving 5 tables.
- The full data model can be defined using a single spreadsheet or be divided among multiple spreadsheets that can reference types defined in any spreadsheet.
- See the following xml spreadsheet OseeTypes_ProgramAndCommon.xml for an example.

Data Model Customization – Table 1 - Artifact

- Factory Class – Retired in 0.7.0
- Artifact Type Name - Any valid UTF-8 characters with a max length of 75 bytes
- Super Type Name - The super artifact type from which this type will inherit associated attributes and relations. Another concrete artifact type or an abstract one that exists only in data model definition.

Data Model Customization – Table 2 - Attributes

- Attribute Base Type - Fully qualified java class name of a class that extends org.eclipse.osee.framework.skynet.core.attribute.
  The typical and simplest case is to specify one of the built-in types (StringAttribute, BooleanAttribute, etc…) If a custom java type that extends Attribute is needed, then that type should be specified here.
- Attribute Type Name - Any valid UTF-8 characters with a max length of 500 bytes
- File Extension - Any valid UTF-8 characters with a max length of 50 bytes; only applies when using the org.eclipse.osee.framework.skynet.core.UriAttributeDataProvider
- Tagger ID - If the attrbiute's content is to be included in the search index for the quick search, use DefaultAttributeTaggerProvider, otherwise leave blank.
- Default Value - The initial value given an attribute upon initialization, this may be left blank
- Validity Xml - For the attribute base type org.eclipse.osee.framework.skynet.core.EnumeratedAttribute, specifies the valid enumerations. For example, <Page_Type><Enum>Portrait</Enum><Enum>Landscape</Enum></Page_Type>
- Min Occurrence - The framework with prevent having lest than this number of this attribute type on a single artifact
- Max Occurrence - The framework with prevent adding more than this number of this attribute type to a single artifact
- Tip Text - Text to describe an attribute. Any valid UTF-8 characters with a max length of 4000 bytes

Data Model Customization – Table 3 - Artifact Type / Attribute Type Mapping

- Artifact Type Name - Exact name of an artifact type defined above (or previously)
- Attribute Type Name - Exact name of an attribute type defined above (or previously) to be associated with the corresponding artifact type

Data Model Customization – Table 4 - Relation Type

- Relation Type Name - Any valid UTF-8 characters with a max length of 50 bytes
- Side A Name - Descriptive name for the A side of the relation. Any valid UTF-8 characters with a max length of 50 bytes
- A to B Phrase - An optional phrase that describe the relation between the artifacts from the side A perspective. Any valid UTF-8 characters with a max length of 50 bytes
- Side B Name - Descriptive name for the B side of the relation. Any valid UTF-8 characters with a max length of 50 bytes
- B to A Phrase - An optional phrase that describe the relation between the artifacts from the side B perspective. Any valid UTF-8 characters with a max length of 50 bytes
- Short Name - Five or less characters is typical. This abbreviated name is used in the user interface when space is at a premium
- Ordered - Yes to have artifacts on the same side of this relation type use a user defined ordered, otherwise No

Data Model Customization – Table 5 - Artifact Type / Relation Type Mapping

- Artifact Type - Exact name of an artifact type defined above (or previously)
- Relation Type - Exact name of an relation type defined above (or previously)
- Side A Max - An artifact of type "Artifact Type" can be on side "A", "Side A Max" number of times for relation links of type "Relation Type“
- Side B Max - An artifact of type "Artifact Type" can be on side "B", "Side B Max" number of times for relation links of type "Relation Type"

Data Model - Add Artifact, Attribute and Relation

- Open OseeTypes_ProgramAndCommon_New.xml spreadsheet (5 highlighted lines were added)
- Added Artifact Type: System Function
- Added Attribute Type: Safety Criticality
  - Enumeration:<Criticality><Enum>A</Enum><Enum>B</Enum><Enum>C</Enum><Enum>D</Enum><Enum>E</Enum></Criticality>
- Added Artifact to Attribute Mapping
- Added Relation Type: Design
- Added Relation to Artifact Mapping

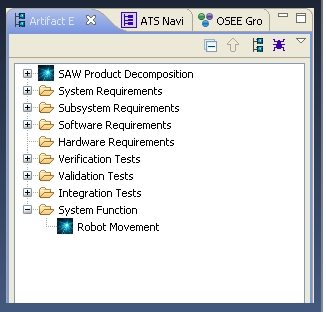
Import OSEE Data Model

- To import changes to the data model for an existing OSEE database:
  - File -> New -> Other -> OSEE -> Osee Types
  - Select File "OseeTypes_ProgramAndCommon_New.xml"
  - Select Branch "Common"
  - Select Finish
- To automatically import the data model during database initialization use the extension point org.eclipse.osee.framework.skynet.core.OseeTypes

Import OSEE Data Model – Try It - 1

Let's create a "System Function" folder and a new "System Function" artifact

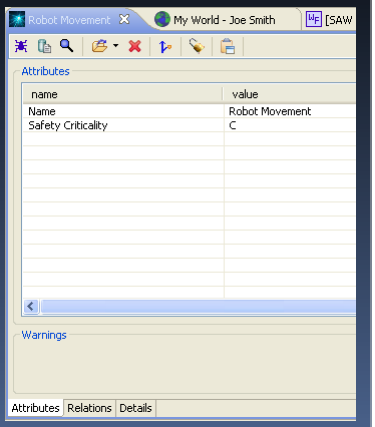
- Set Default Branch -> SAW_Bld_1
- In Artifact Explorer, right-click -> New Folder -> Name: "System Function"
- Select "System Function" -> right-click -> New Child -> "System Function" -> Name: "Robot Movement"



Import OSEE Data Model – Try It - 2

Let's set the "Safety Criticality" attribute of our new "Robot Movement" system function.

- In Artifact Explorer, Double-click to open new "System Function" artifact called "Robot Movement"
- Switch to the attributes tab of the "Robot Movement" artifact
- Modify the "Safety Criticality" value to "C"
- Save

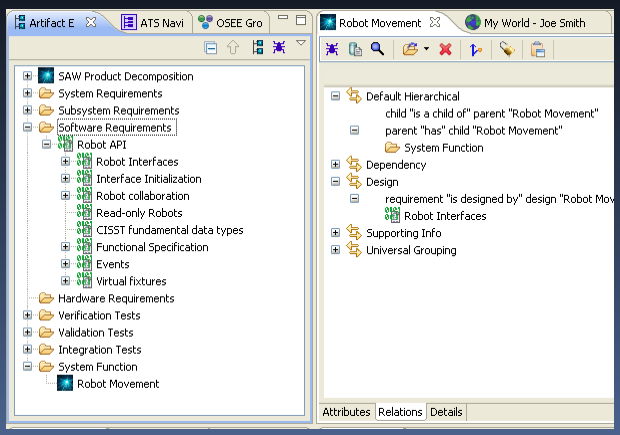


Import OSEE Data Model – Try It - 3

Let's drag an existing Software Requirement into the new "Robot Movement" "System Function" artifact.

- In Artifact Explorer, Double-click to open new "System Function" artifact called "Robot Movement"
- Switch to the relations tab of the "Robot Movement" artifact
- Under "Software Requirements" -> drag any requirement into "Design" relation
- Save

Import OSEE Data Model – Success!

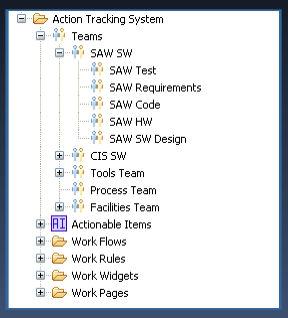


ATS Configuration

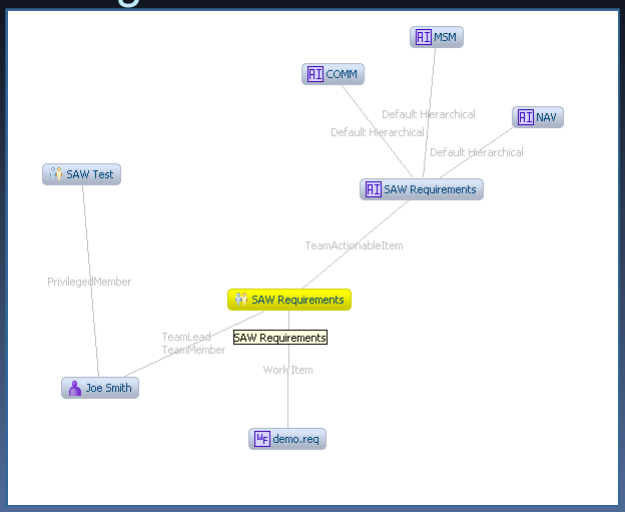
- ATS uses OSEE framework Artifacts, Attributes and Relations to store configuration information
- Configuration changes can be done using framework editors and views
- Configurations can be done dynamically in OSEE without need for code release
- Major changes such as new widget types, advanced algorithms for assignment/routing, customized searching can be done through Eclipse extension points

ATS Configuration – Team Definition

- Artifact in OSEE represents a team that is responsible for performing work
- Configured with users that perform roles
  - Team Lead – Endorses Team Workflow, Assigns Work
  - Team Member – Performs Work on Team
  - Privileged Member – Able to override assignee and edit any field in Team Workflow
- Related to Actionable Items that they are responsible for
- Related to Workflow Configuration that defines how this team does it's work

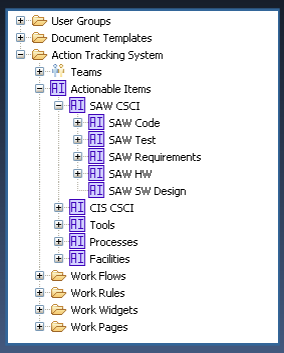


ATS Configuration – Team Definition



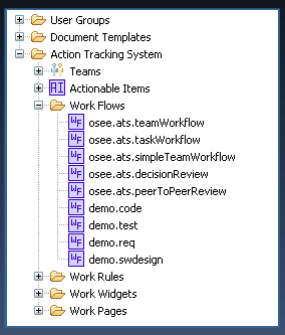
ATS Configuration – Actionable Item

- Artifact in OSEE represents a real or conceptual object that the user would write an Action against
- Active Flag enables items to be retired
- Related to Team Definitions that are responsible for them

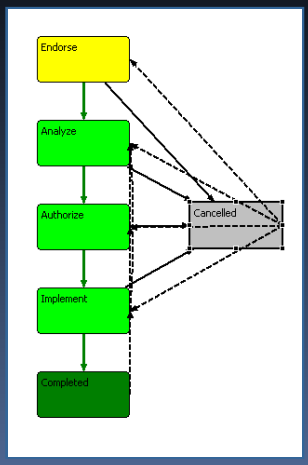


ATS Configuration – Work Flows

- Artifact in OSEE represents a how a team performs it's work
- Represented by a state machine with Work Pages being states
- Related to Team Definitions uses them
- Related to Work Pages and Work Rules that apply

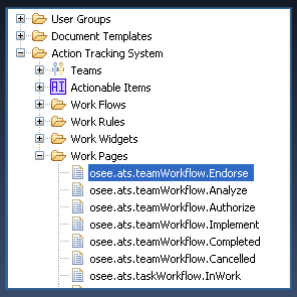


ATS Configuration – Work Flows



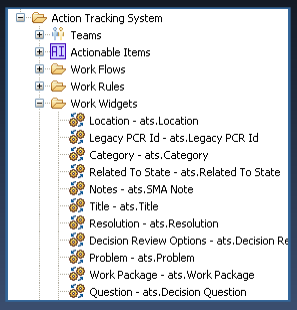
ATS Configuration – Work Pages

- Artifact in OSEE that represents a a state in a Work Flow
- Represented by a state machine with Work Pages being states
- Related to Work Flow they belong to
- Related to Work Rules and Work Widgets



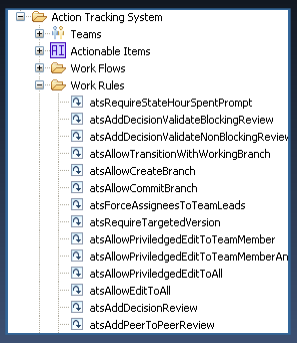
ATS Configuration – Work Widgets

- Artifact in OSEE that represents a a single Widget to display for a Work Page
- Contains some formatting information
- Includes information relating widget contents to storage attribute
- Related to Work Page they belong to

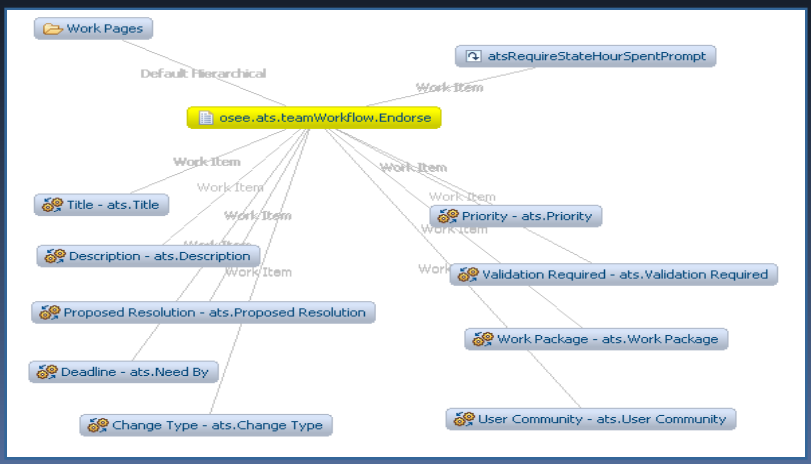


ATS Configuration – Work Rules

- Artifact in OSEE that represents a rule to be applied to Work Flow or Work Page
- Backed by code that provides the functionality described
- ATS provides built in rules for use by work flows
- New rules can provided through extension points

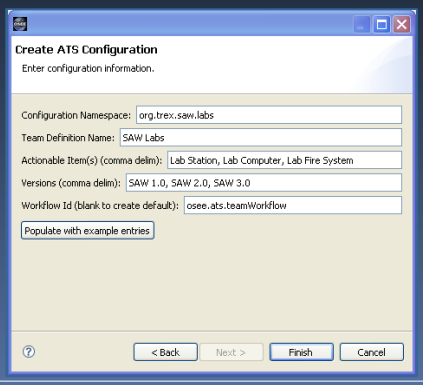


ATS Configuration – Work Pages, Widgets, Rules

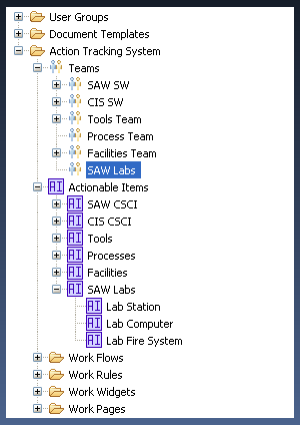


T. Rex – Let's track a new SAW Lab

- File -> New -> Other -> OSEE ATS -> Create ATS Configuration
- Select "Populate with example entries" button



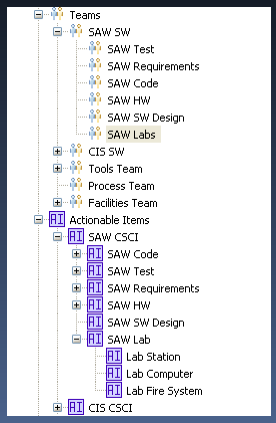
Re-Organize Team Definitions and Actionable Items



In Artifact Explorer:

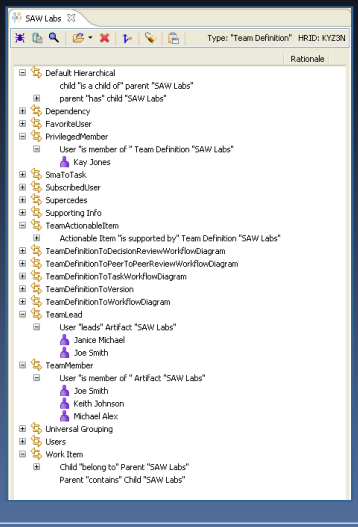
- Move New “SAW Labs” Team Definition under “SAW SW” Team
- Move New “SAW Lab” Actionable Items under “SAW CSCI” Item

becomes



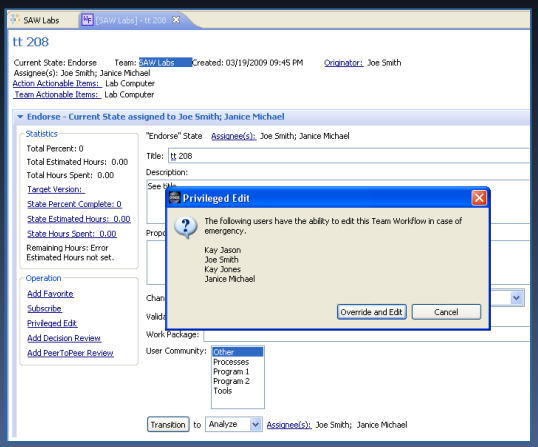
Configure Team Leads and Members

- Search Pulldown – Artifact Search – Artifact Type – User – Add Filter - Search
- From Search Results:
  - Drag “Janice Michael” to TeamLead
  - Drag “Keith Johnson” and “Michael Alex”” to TeamMember
  - Drag “Kay Jones” to Privileged Member
  - Save



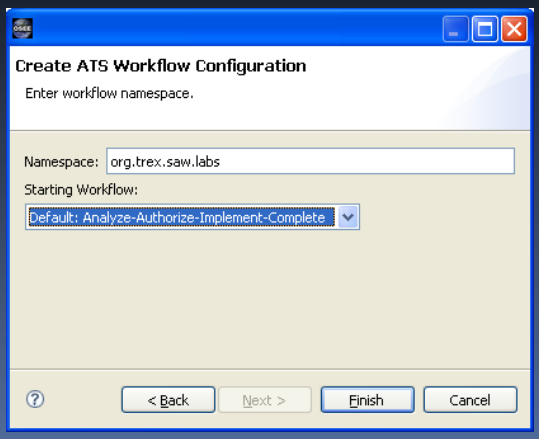
Let's Try It…

- ATS Navigator – New Action – Lab Computer Actionable Item
- Complete Action Wizard
- Notice:
  - Endorse State assigned to both leads "Janice" and "Joe"
  - Privileged Edit allows "Kay Jones" to override



Labs Team wants a different workflow

- File -> New -> Other -> OSEE ATS -> Create ATS Workflow

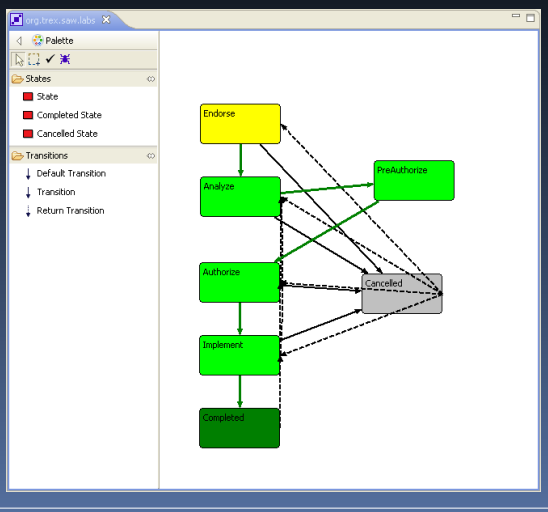


Set new Labs team workflow config

- Double-click “SAW Lab” Team Definition
- Switch to Relations tab
- Expand “Work Item”
- Select item “osee.ats.teamWorkflow” and right-click – “Delete Relation”
- From Artifact Explorer, drag workflow “org.trex.saw.labs” to Work Item – Child
- Save
- Restart OSEE

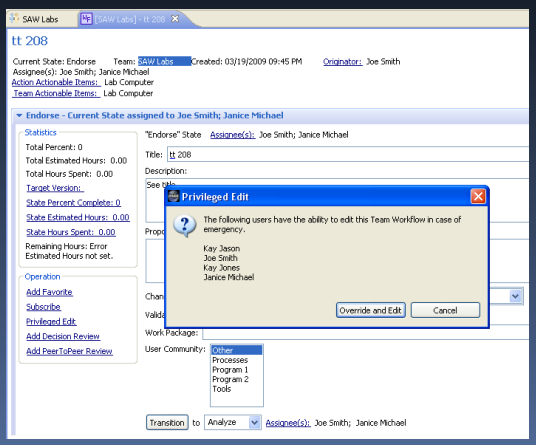
Add a PreAuthorize state…

- Click “State” and click in editor area
- Add “Default Transition” from Analyze to PreAuthorize and from PreAuthorize to Authorize
- In Properties view, change state name to PreAuthorize (Note: id will be updated automatically)
- Save



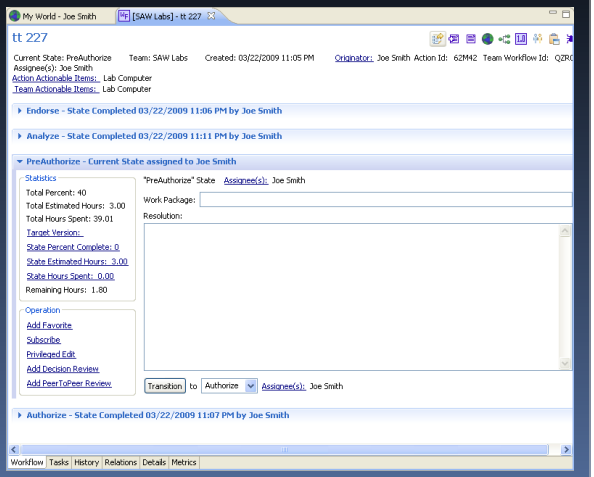
Let's try new PreAuthorize state

- ATS Navigator – New Action – Lab Computer Actionable Item
- Complete Action Wizard
- Transition through to "Implement" state and note that new PreAuthorize state exists



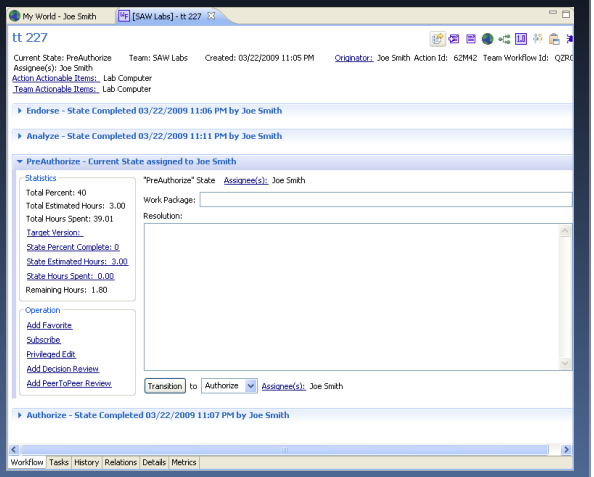
PreAuthorize needs some widgets

- Artifact Explorer
- Double-click “org.trex.saw.labs.PreAuthorize” “Work Page”
- Switch to Relations tab
- Drag in Resolution and Work Package widgets from “Work Widgets”
- Restart OSEE and open Action from “My World”



PreAuthorize needs a Rule

- Artifact Explorer
- Double-click “org.trex.saw.labs.PreAuthorize” “Work Page”
- Switch to Relations tab
- Drag in “atsRequireTargetedVersion” rule from “Work Rules”
- Restart OSEE and open Action from “My World”
- Transition back to Analyze, then PreAuthorize, then Implement – Get stopped until set “Targeted Version”



Advanced customization / extensibility

- New Attributes can be added to Team Workflow artifacts
- New Widgets can be created
  - Simple – attribute with existing XWidget xml
  - Advanced – new XWidget with attribute or other storage (artifacts, relations, etc…)
- New Rules can be created
- Teams and States can have java backed algorithms that enforce or automate tasks (eg: emailing team leads, requiring review if estimated hours > 30, etc…)

Other Extensibility

- Provide customized editors for artifacts
- Attribute Data Providers
- Renderers
- Indexed based taggers
- Authentication Protocols
- Resource Management Protocols
- Artifact Types and Factories
- Customized Dictionaries
- XWidget Providers

We have made it to the end!

- Feedback Questionnaire
- See you tonight at 7:30 pm - Great America 2 for Birds of a Feather
- Other OSEE Talks at EclipseCon 2009
  - "XViewer - An SWT Widget with the power of the spreadsheet"
    - Wednesday Mar. 25th 4:50 pm - Room 203/204
  - "An Integrated Test Environment for Systems Engineering"
    - Wednesday Mar. 25th 11:30 pm - Room 203/204
  - "Unlocking the OSEE Core Framework"
    - Thursday Mar. 26th 10:40 am – Grand Ballroom B

Click to add title

- For further help with OSEE
  - http://www.eclipse.org/osee
    - Newsgroup (Do not use the Mailing List)
    - Documentation
    - FAQs

Legal Notices

- Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- Other company, product, or service names may be trademarks or service marks of others

Click to add title

- Show diagram of each team's workflow
- - Diagram of Action, Workflows, Reviews, Tasks
- - Diagram of Team Definitions and Actionable Items
- - Work Flows, Pages, Widgets, Rules
- - Versions
- - Groups
- - Users / User Groups

- - Configuration of ATS
- - ATS Configuration
- - New -> Other -> OSEE ATS -> ATS Configuration
- - Configuration Namespace: org.myCompany.labs
- - Team Definition Name: Labs Team
- - Actionable Item(s): Lab 1, Lab 2, Lab Door, Lab Computer
- - Versions:
- - Workflow Id: osee.ats.defaultTeamWorkflow