Work Item and Process Customization

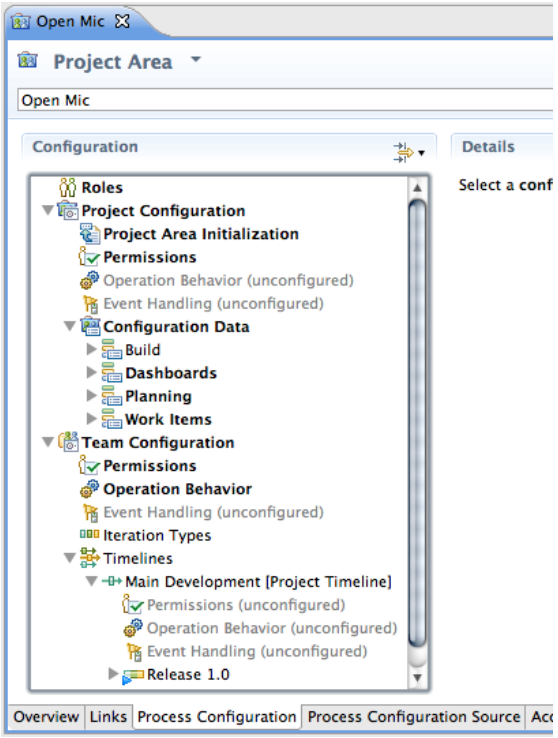
André Weinand
RTC Change Management Architect
andre_weinand@ch.ibm.com

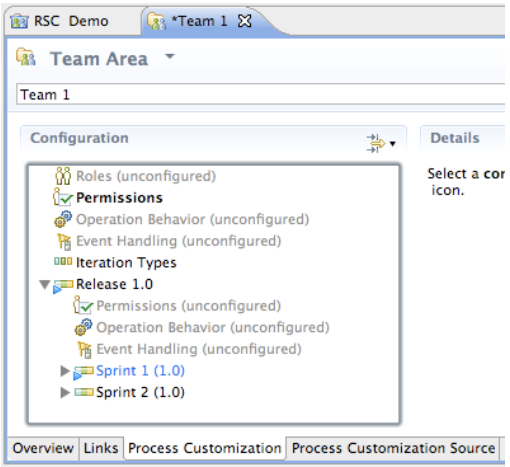
Customizations: Scope

- RTC supports customizations for these capabilities:
  - Process (Jazz Foundation)
  - Work Items
  - Planning
  - Dashboards
  - SCM
  - Build
- Today's focus:
  - Only common customizations
  - Components:
    - Process
    - Work Items

Process Customizations

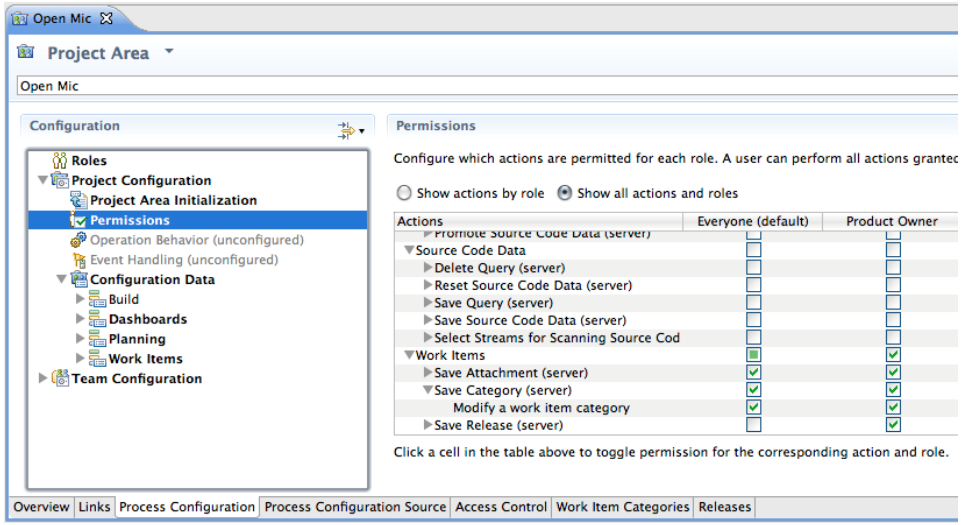
- Project Area:
  - Roles
  - Project configurations
    - Initialization
    - Role based Operations
    - Configurations
  - Team configurations
    - Role based Operations
    - Timeline/Role based Operations
- Team Area (optional):
  - Roles
  - Role based Operations
  - Timeline/Role based Operations

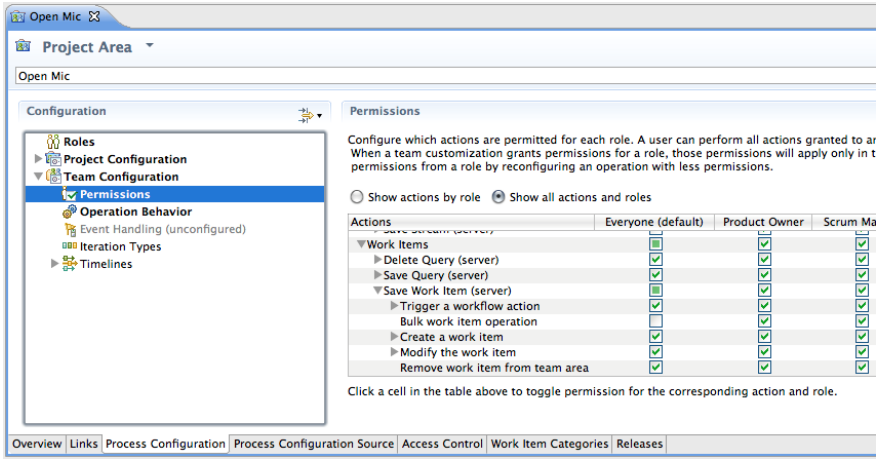




Process Customization: Operations

- **Example**:
  - “Save Category”: project level
  - “Save Work Item”: team level
- **Operation Permissions**:
  - Based on finer grained “actions”:
    - ***Create*** Work Item
    - ***Modify*** Work Item
    - ***Modify Attribute “Priority”*** of Work Item
- **Operation Behavior**:
  - Preconditions, e.g. “Required Properties”
  - Follow-up actions, e.g. “Create Initial Work Items”

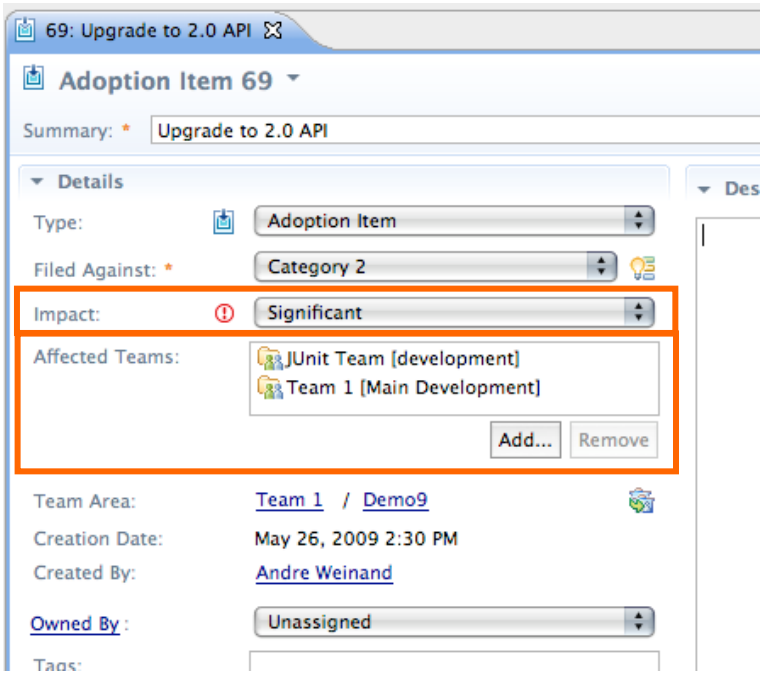




Work Item Types & Custom Attributes

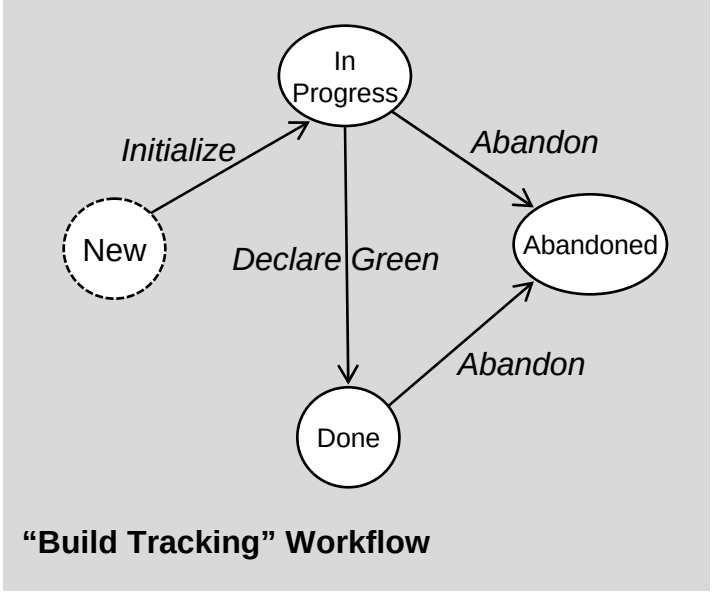
- **Work Item Type** defines:
  - Display name, Icon
  - Built-in attributes: e.g. Summary, Description, Owner, Creator, …
  - Optional custom attributes
  - A state transition matrix (aka workflow)

- Scrum Examples:
  - Build Tracking Item
  - Story
    - Criteria of Acceptance: Large HTML
  - **Adoption Item**
    - Impact: Custom enumeration
    - Affected Teams: Team Area list



Work Item Workflows

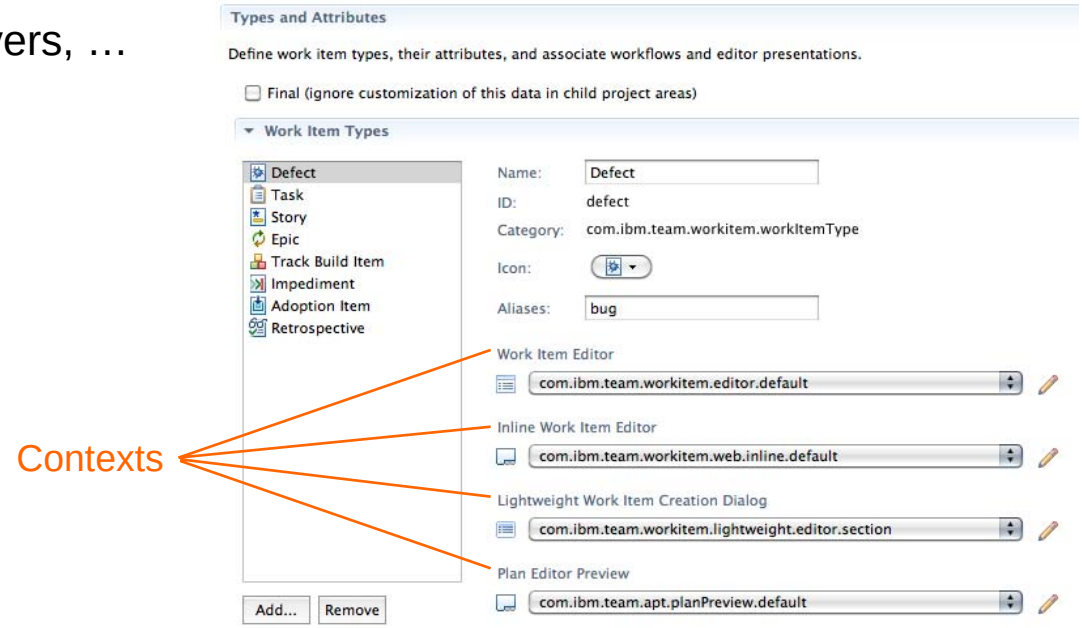
- A **Workflow** consist of:
  - States
  - Actions
  - Resolutions (optional)
  - State Groups
- Scrum Examples:
  - User Story Workflow
  - **Build Tracking Workflow**



“Build Tracking” Workflow

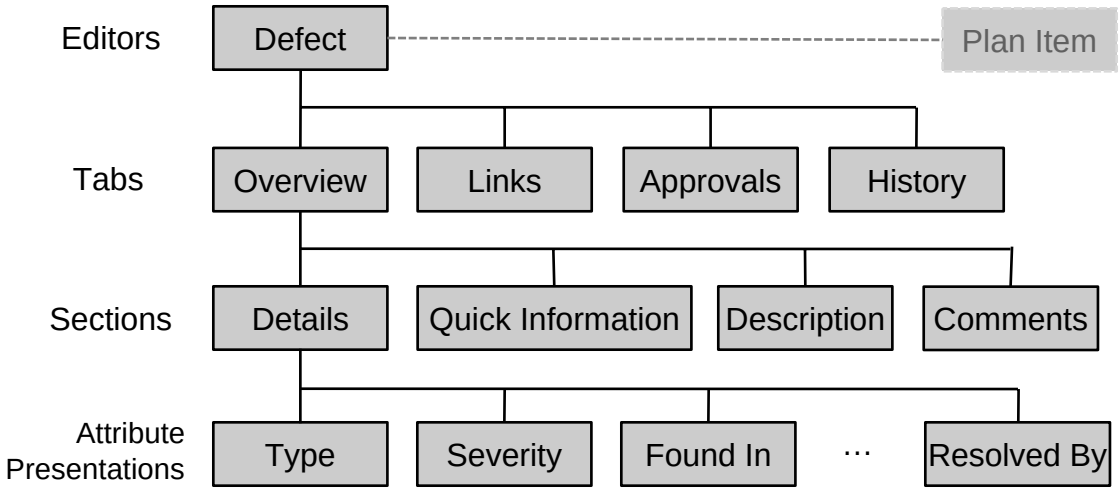
Customizing the UI – Editor & Attribute Presentations

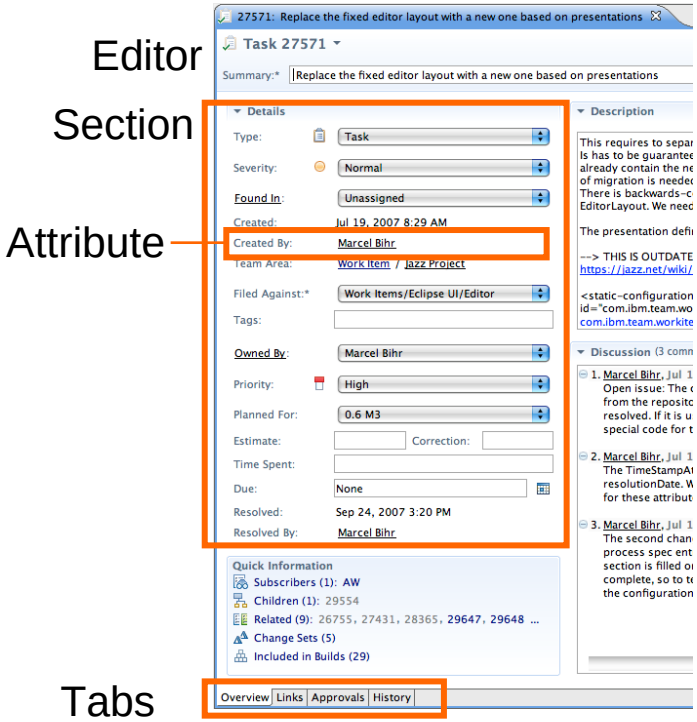
- **Editor Presentations**
  - Define how a work item is presented based on:
    - Tabs, Sections, Attribute presentations
    - Small number of predefined layouts (no free-form layout yet)
  - Can be used in various contexts:
    - Editor, Inline Editors, Dialogs, Hovers, …



Editor Presentation

- Editor's UI element hierarchy:

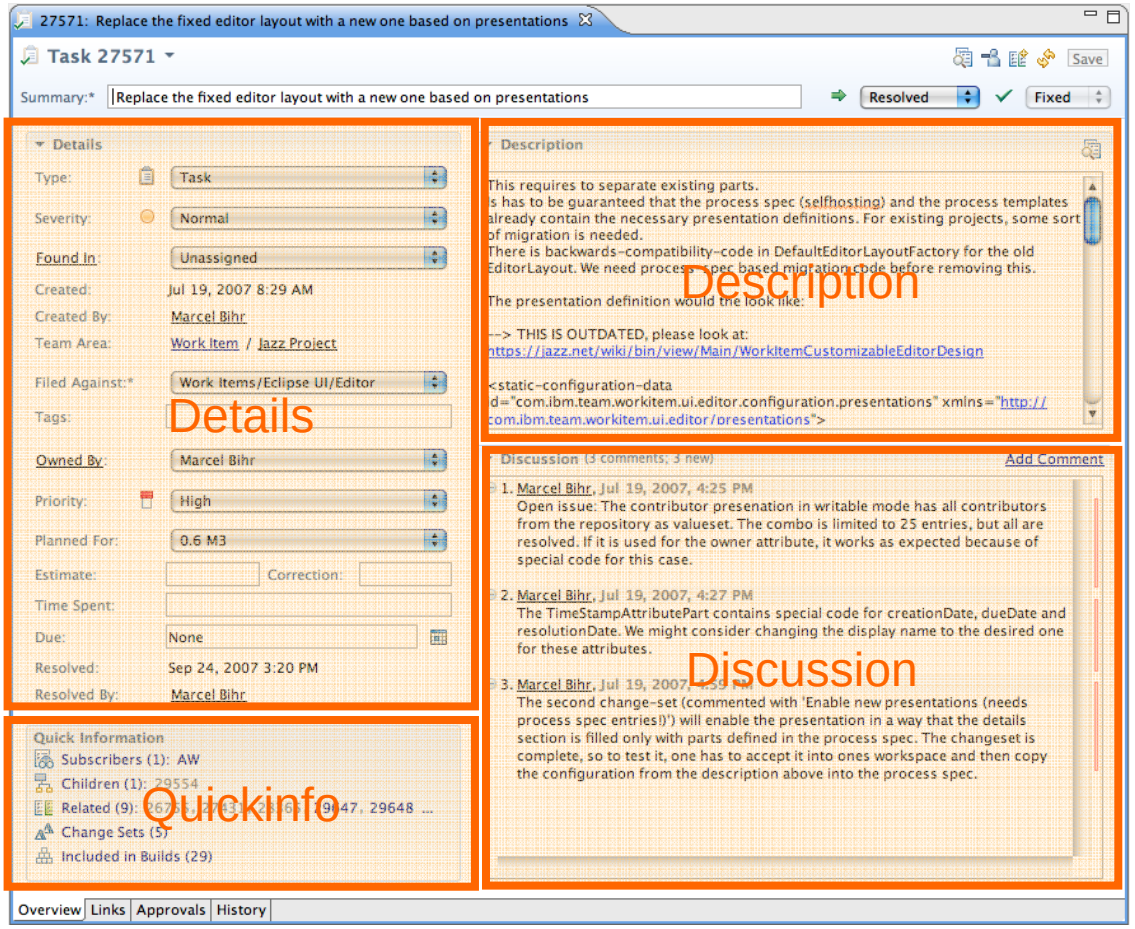




- RTC 3.0.1 supports WYSIWYG editing
  - but only in Web UI

Editor Presentation Layout

- Layout limitations:
  - Tabs have predefined ‘section slots’
  - Sections can be added to slot but not arbitrarily positioned
- Web UI uses same slots
  - but different layout
- Similar limitation for Sections
  - Attributes can be added but not arbitrarily positioned

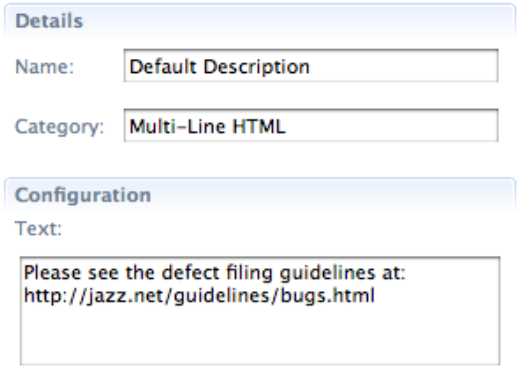


More Customization Features

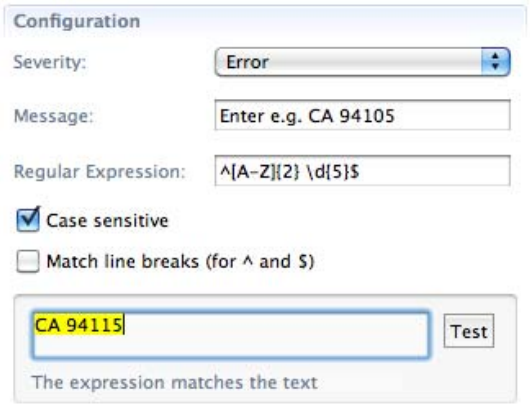
- **Preconditions** for Work Item Save Operation:
  - "Require Approvals"
    - Require an approval for all transitions into a specific state
  - "Strict Group Resolution" (generalization of "All Children Resolved" precondition)
    - Requires all children to be resolved before the parent can be resolved, and requires the parent to be open before a child can be added or re-opened.
  - "Attribute Validation"
    - Verifies that a work item can only be saved if all attribute values are valid
  - "Implied Attributes"
    - Specific attribute changes can trigger sets and clears of dependent attributes
  - "Prevent Editing"
    - Prevents a Work Item from being edited in certain states

More Customization Features

- **Default Values**
  - Freely configurable default value for attributes
    - Supported Types: Text, Wiki text, Users, User Lists, Iterations, Categories, Numbers

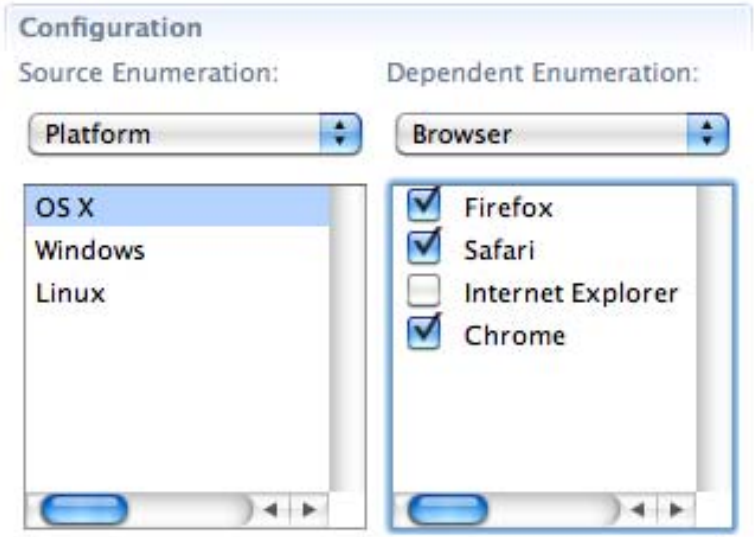


- **Attribute Validation**
  - Two out-of-the-box configurable validators:
    - Number Ranges: Restrict numeric fields to a specific range, e.g. only positive numbers
    - Regular Expressions: Enter a regular expression that validates your string attribute
      - Example: U.S. Zip codes:



More Customization Features

- **Dependent Enumerations**
  - Value set of one enumeration depends on the value of another enumeration
  - Example:
    - "Browser" value set depends on a "Platform" attribute:



Process Customization: Best Practices

- Process can be easily changed at any time
  - But there are some constraints (see below)
- At end of iteration reflect on how to improve process
  - e.g. collect improvements in “Process Improvement” work items
- After enough reflection/discussions customize existing process instance
  - Start with simple customizations
    - e.g. **add** a new work item type, **add** query to find all items of this type
  - Plan how to avoid corruption of existing data
    - e.g. **add only**, do not change IDs,
      do not remove process definitions that are still used
      (use the “check … usages” action links in the Eclipse UI)
- After thorough testing create a process template of your customizations that other projects can use