Implementing Process Change Control

This is the fourth in a series of articles on how Rational Collaborative Lifecycle Management (CLM) solutions support software development compliance.

A highly configurable process provides great flexibility for a team. However, in a regulated environment, you must ensure that the internal controls for IT governance implemented in the process are enforced and cannot be circumvented. Therefore, changes to certain parts of the process specification must be more highly controlled. This article will show how project areas, permissions, and a custom work item type can be leveraged to provide the control required while not overly-restricting the flexibility teams need.

In this article we will demonstrate

- how a shared process configuration is used for controlling parts of the process configuration across an organization
- how to use a custom work item type for capturing and approving process changes
- the process change history recorded by Team Concert

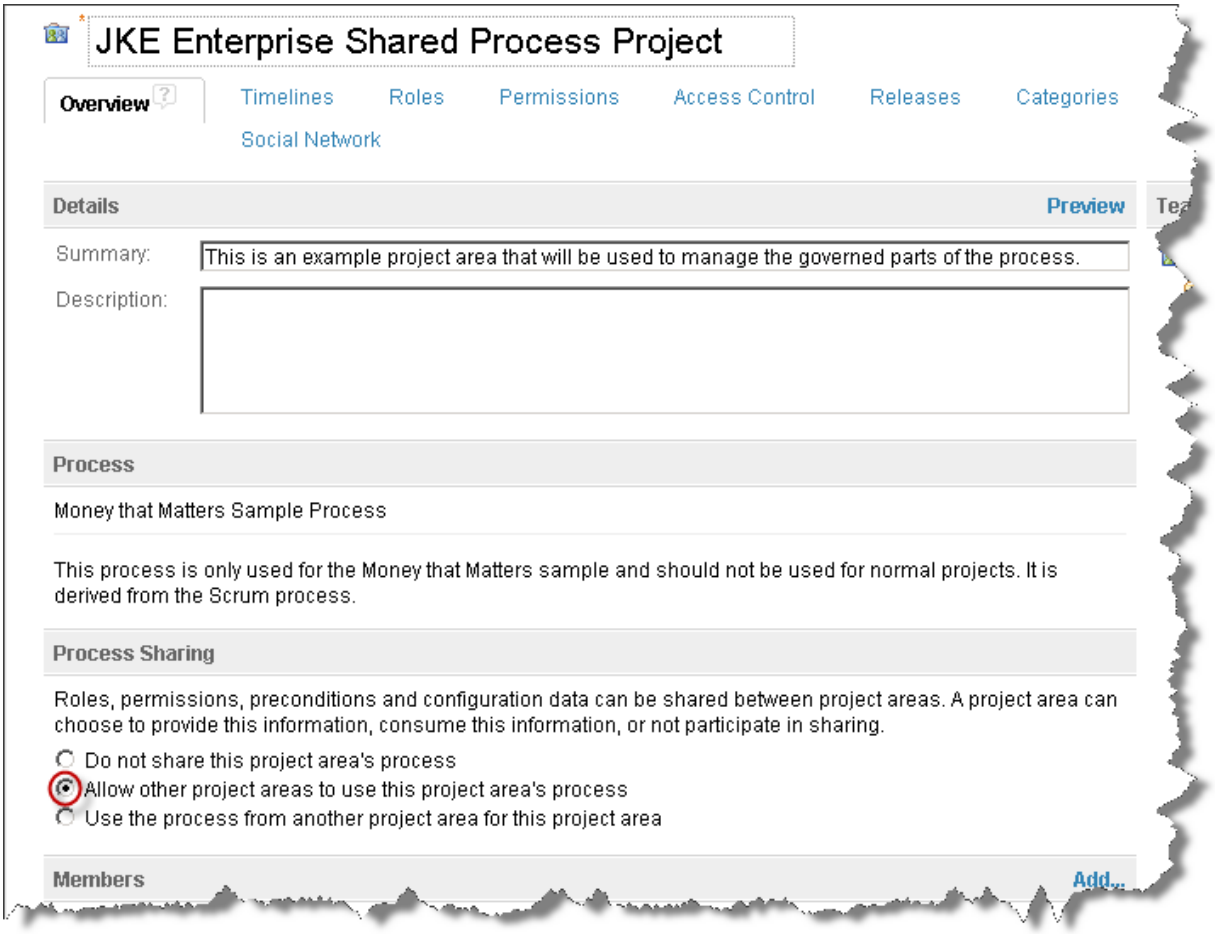
Managing a Controlled Shared Process

Consider this description of a shared process configuration from Team Concert help: “After you create a project area, you can make its process available to other project areas. By sharing a project area process, you ensure that all project areas across your organization use the same process. You also centralize process maintenance.”

In this example, we have created a project area that provides a shared process configuration used by projects across the organization. We’ll look at how the parts of the process configuration are locked down so that it cannot be changed or overridden by the project areas that are using the process. Then we’ll make a change to the shared process and see how it is reflected in the using project area. Lastly, we will try to make a change to a process area that is using the shared process and experience how that change is prevented.

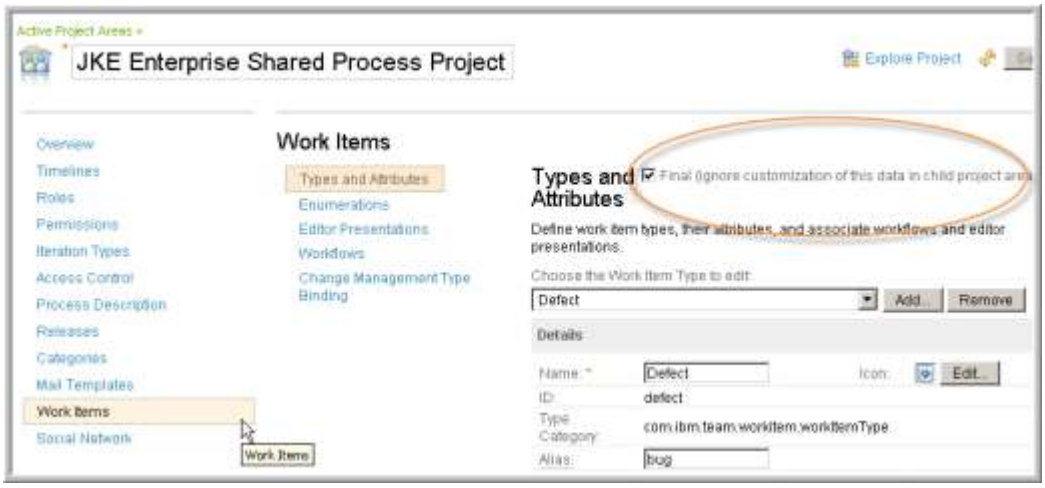
We started with the Money That Matters sample project and created a Project Area called JKE Enterprise Shared Process Project using the Money That Matters process template.

On the **Overview** tab note the option that allows other project areas to use the shared process.



On the **Work Items** tab, we selected the option called "**Final (ignore customization of this data in child project areas)**".

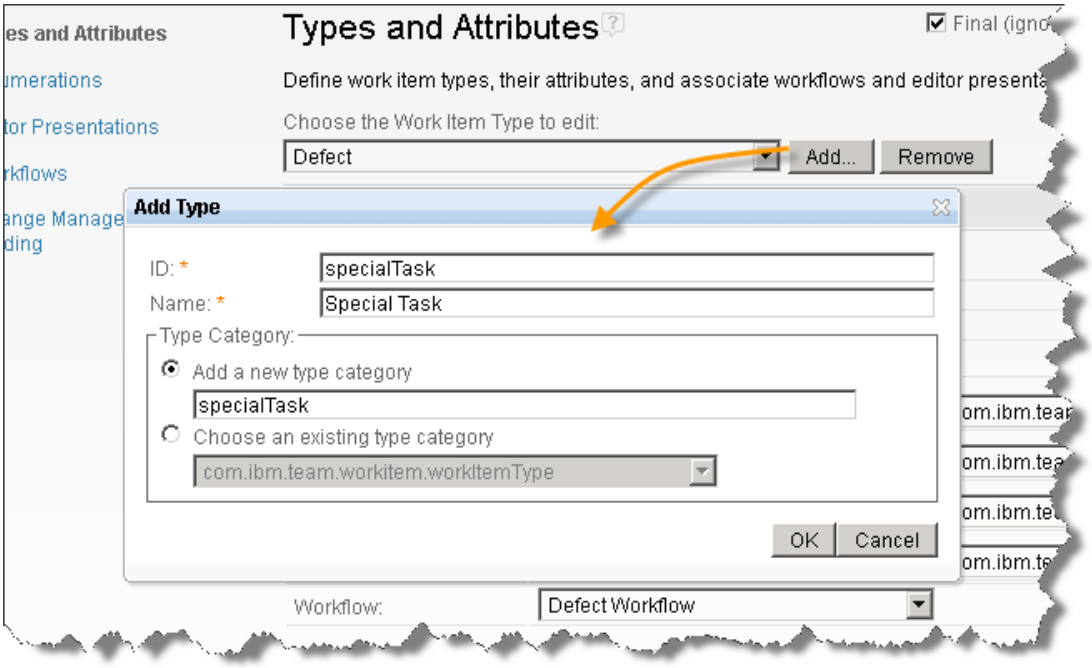
This option prevents addition to or changes to the work item types and attributes in the projects that are using the shared process.



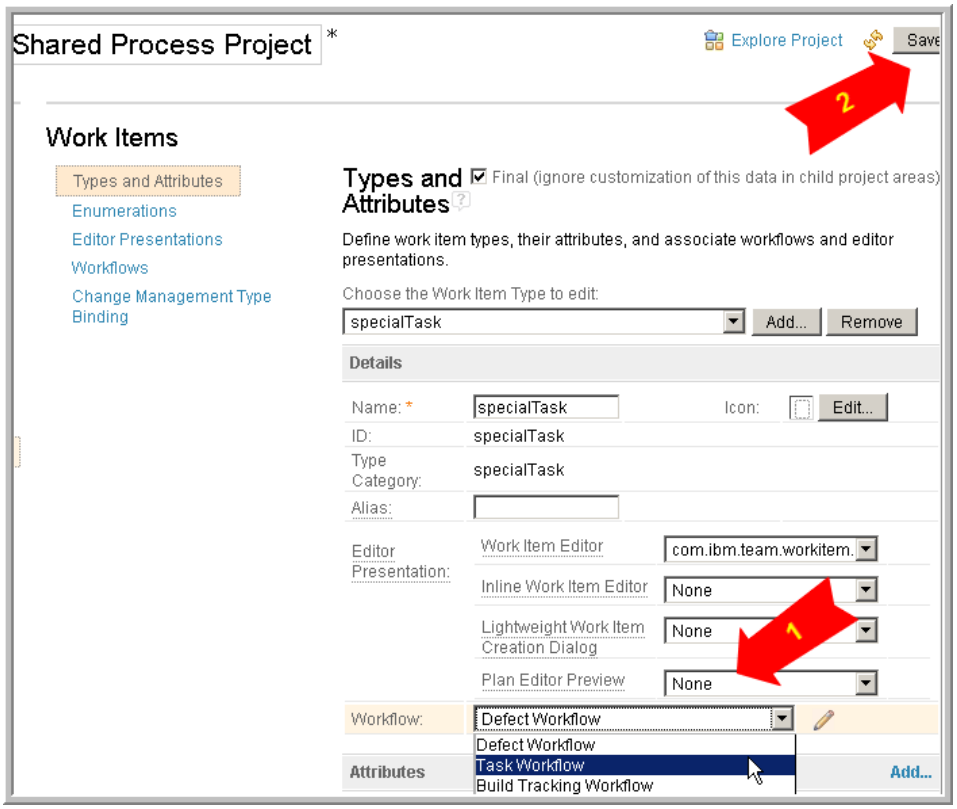
In this example, we have also designated **Editor Presentations** and **Workflows** as **Final**. But **Enumerations** are not “locked down” with this designation. This is a fairly large-grained level at which this option is selected, and therefore not ideal. At the end of the article, we point you to some work items in the Rational Team Concert project that you can review to see the status of relevant enhancements.

Now we will add a simple new work item type to the shared process project to see how it is cascaded to the projects using the shared process.

On the **Types and Attributes** tab, we **Add** a new work item type.

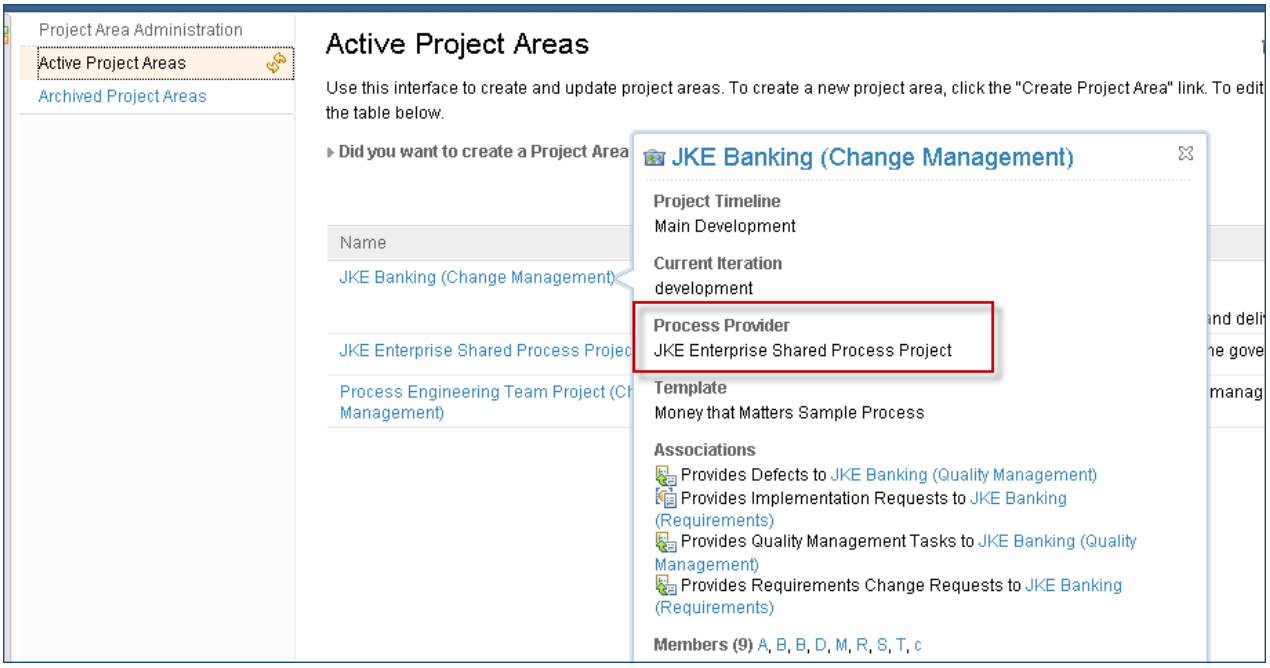


We save the project area and refresh.

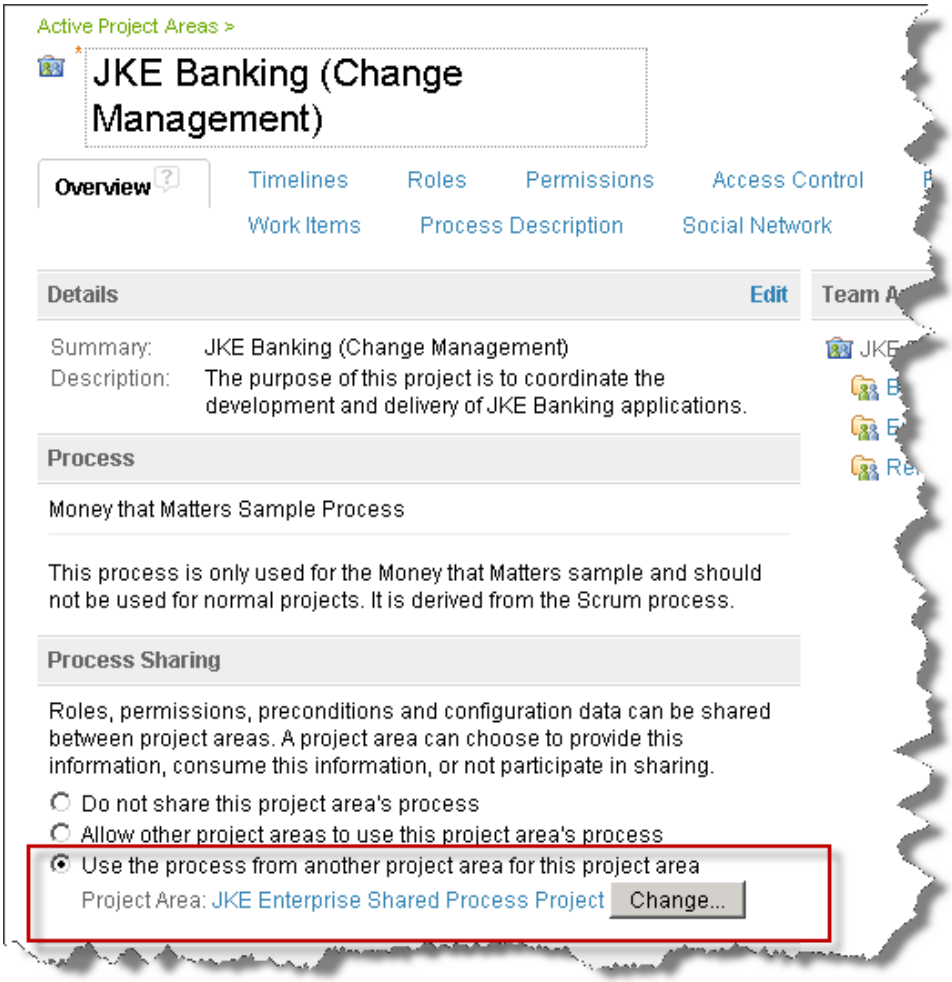


Now we are ready to go look at the project area for a team that is using this process.

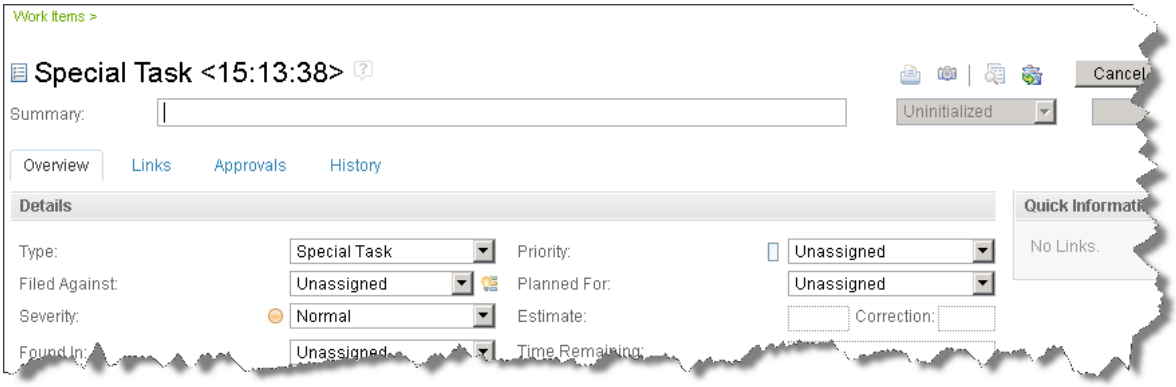
We select **Active Project Areas** on the left side and choose *JKE Banking Change Management*. In this rich hover, we see that this project area uses the shared process.



On the overview tab, we can also see indication of the use of the shared process.



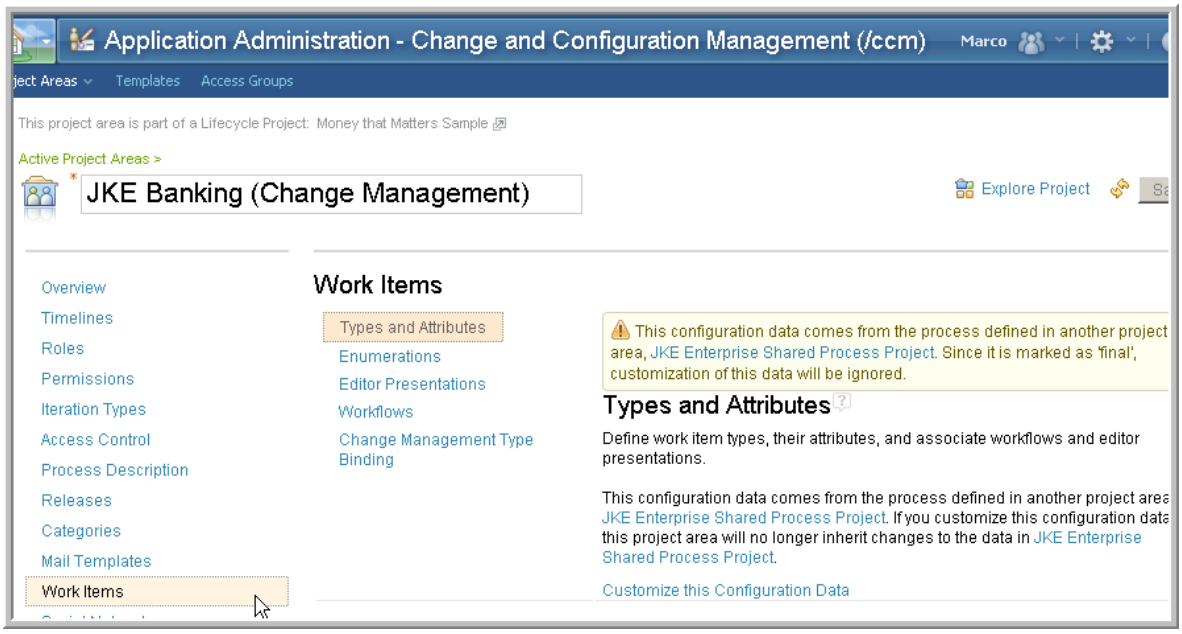
Now when we open the project we can create a new work item of the added type.



Now we will attempt to change the process in a project area where it is not allowed

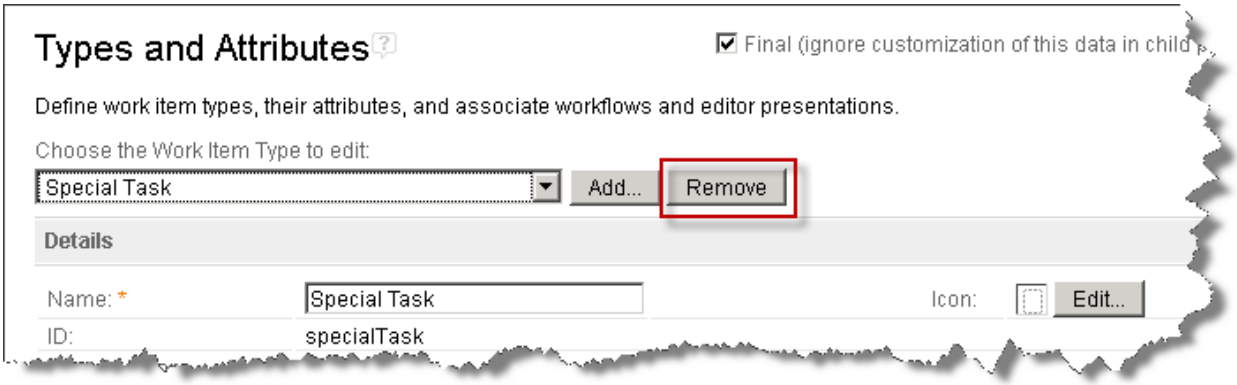
Logged in as a user with privileges to manage this project area's process, we are managing the *JKE Banking (Change Management)* Project Area.

On the Work Items tab, we see a warning about the shared configuration.

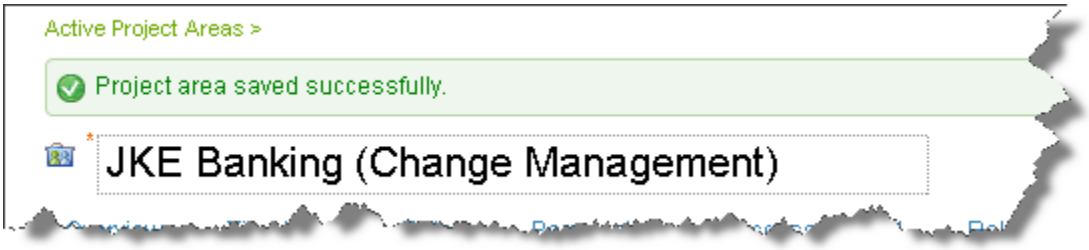


Note that if you are using the Eclipse client and modifying the process for a project area that uses a shared process, you will not see this warning. You will be able to make changes and save the configuration, but the changes will be ignored.

Now, back to managing the project in the web client. We select the link for "Customize this Configuration Data", select the new task type called Special Task that we created earlier, and choose *Remove*.



We save the project configuration and see a message that the project area was saved.



Yet when we back to *Customize this Configuration Data*, the work item type called Special Task is still there. So our changes were ignored.

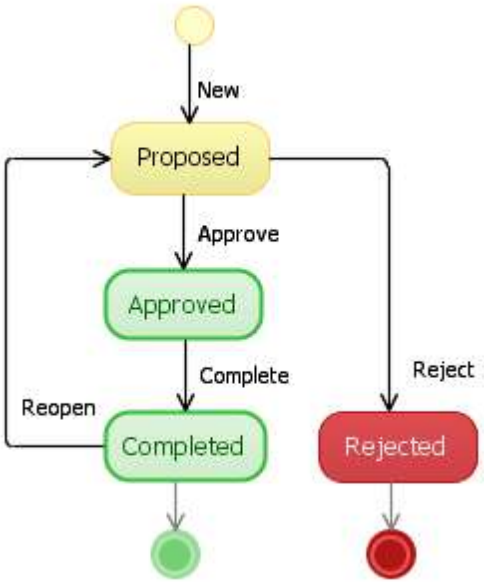
Using a Process Change Authorization Task

In this example we show how you can manage change control over the shared process configuration. We use a custom work item type called a Process Change Authorization to capture a proposed change. Then we show how only a Change Control Board member can approve the process change.

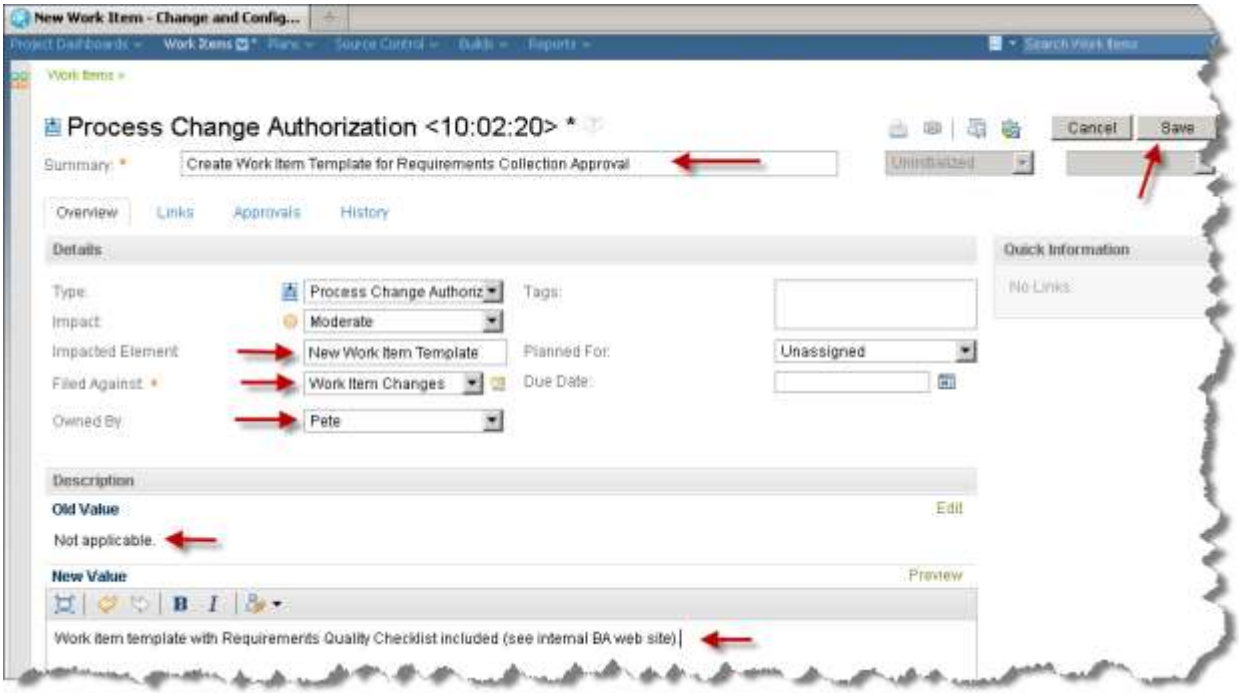
We find that teams responsible for the process configuration and tool configuration in an organization can benefit greatly from the use of the CLM tools for their own work. In fact, this provides a much higher quality of support to people who are using the process and tools because the process & tools team members are also users of the work they are supporting.

Here we see an example Project area established for the Process Engineering Team.

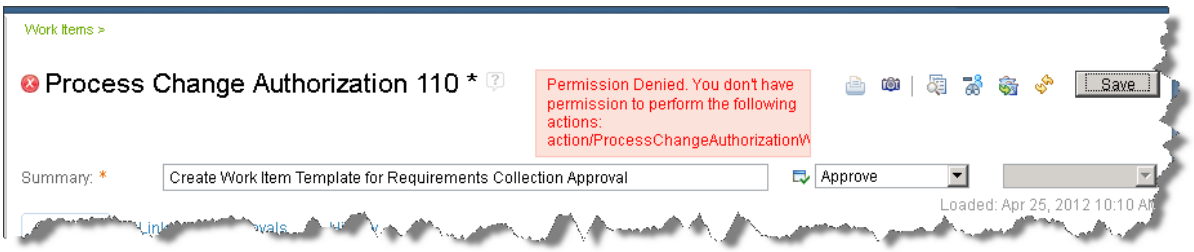
The Process Change Authorization is a custom work item type created from the Adoption Item work item type in the Scrum process template, so it has a simple state model.



We create a new **Process Change Authorization** as shown here:

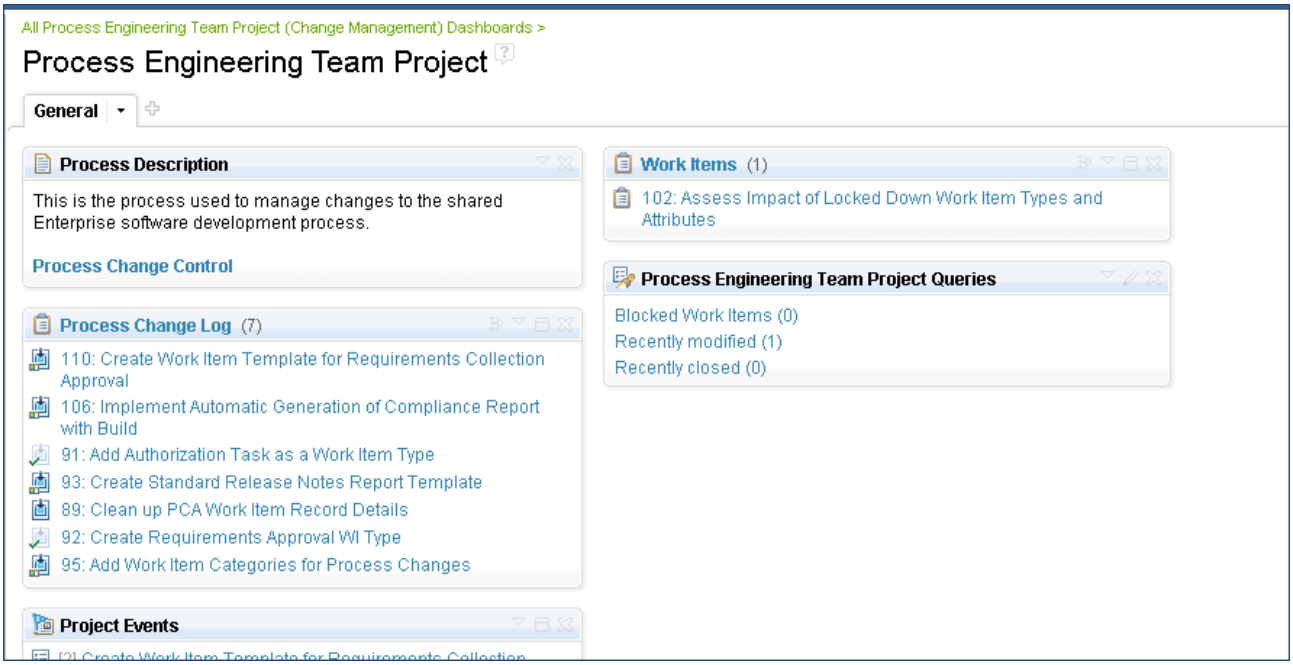


When we save the PCA it is in the **Proposed** state. Because Pete is not assigned the role of Change Control Board, he cannot approve this PCA. When he attempts to do so, the permission is denied.

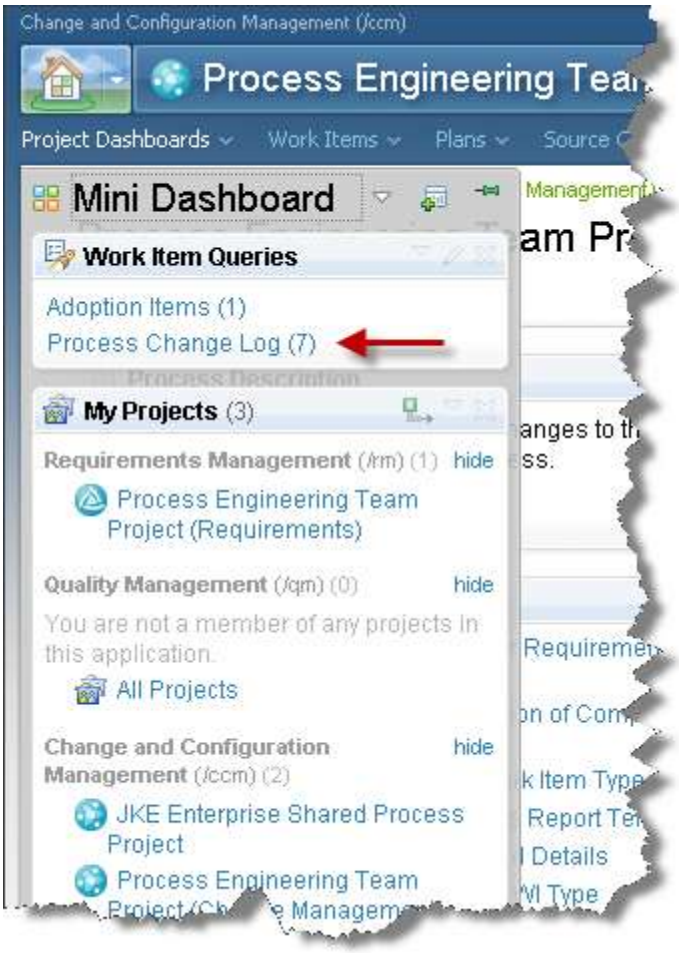


When we log in as a user with the Change Control Board role, that user's privileges allow her to approve the PCA.

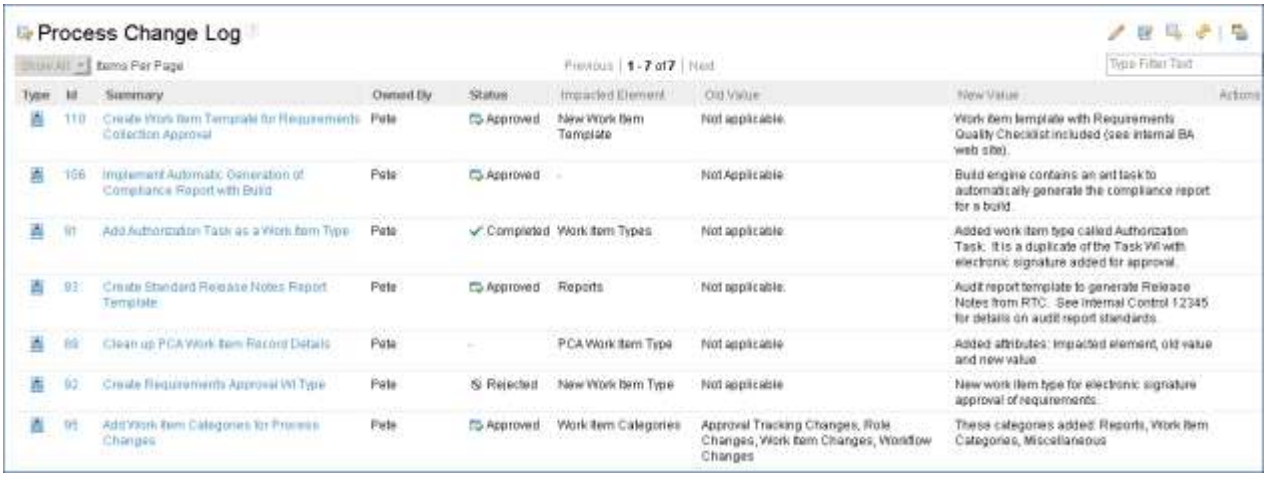
Once we have a set of Process Change Authorization records, we need to see the comprehensive process change log. An easy way to make this visible to all team members it to use a **Process Change Log** widget on the **Dashboard**.



Our process engineer's **Mini Dashboard** is also configured to show the *Process Change Log* query.



When he selects the query from the **Mini Dashboard** he sees the details for the **Process Change Log**.



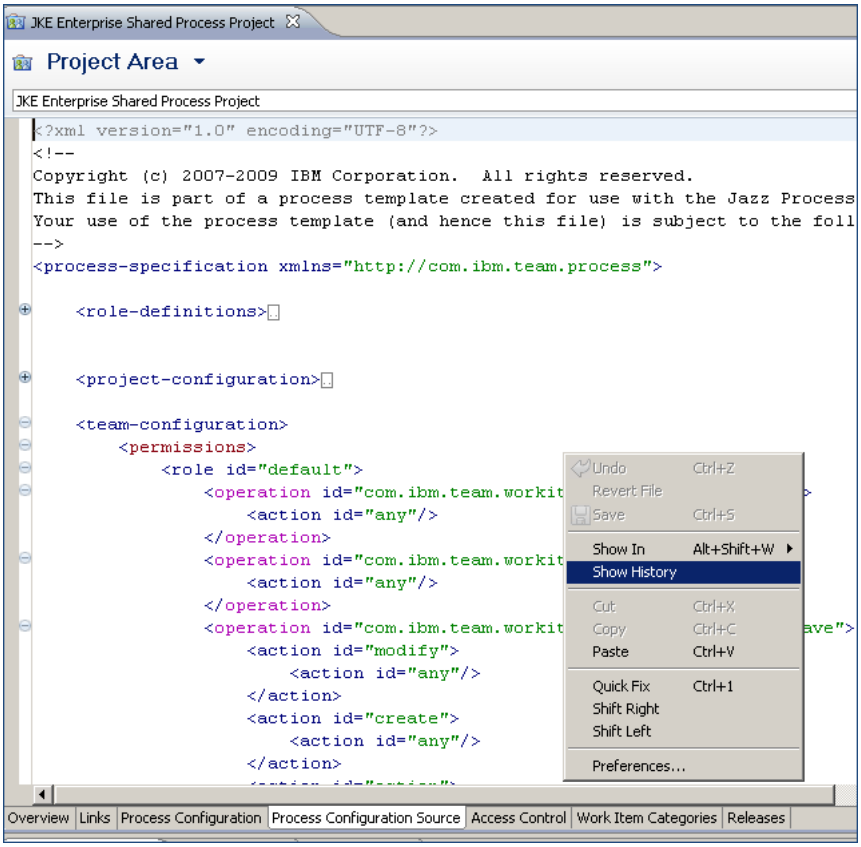
When the work is complete, the process engineer will mark the PCA as complete.

Reviewing Process Change History

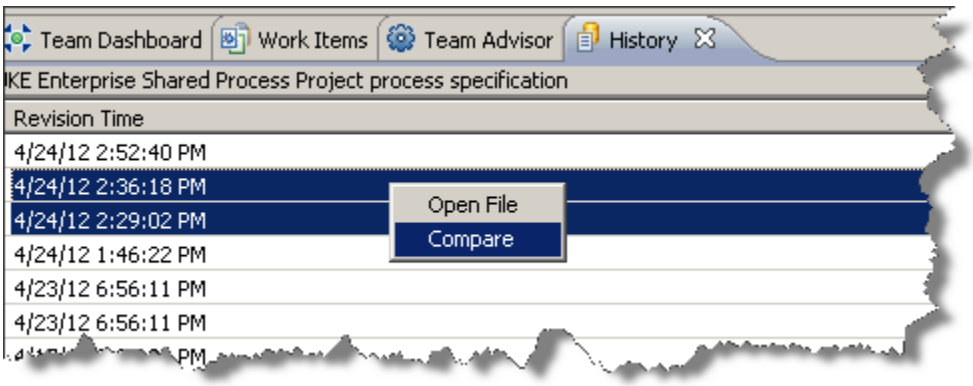
In this example, we review the ways that Team Concert exposes the information about changes to the process configuration.

In the Eclipse client, we open the configuration of the *JKE Enterprise Shared Process Project* and select the **Process Configuration Source** tab.

Right-click in the source file and on the context menu select **Show History**.

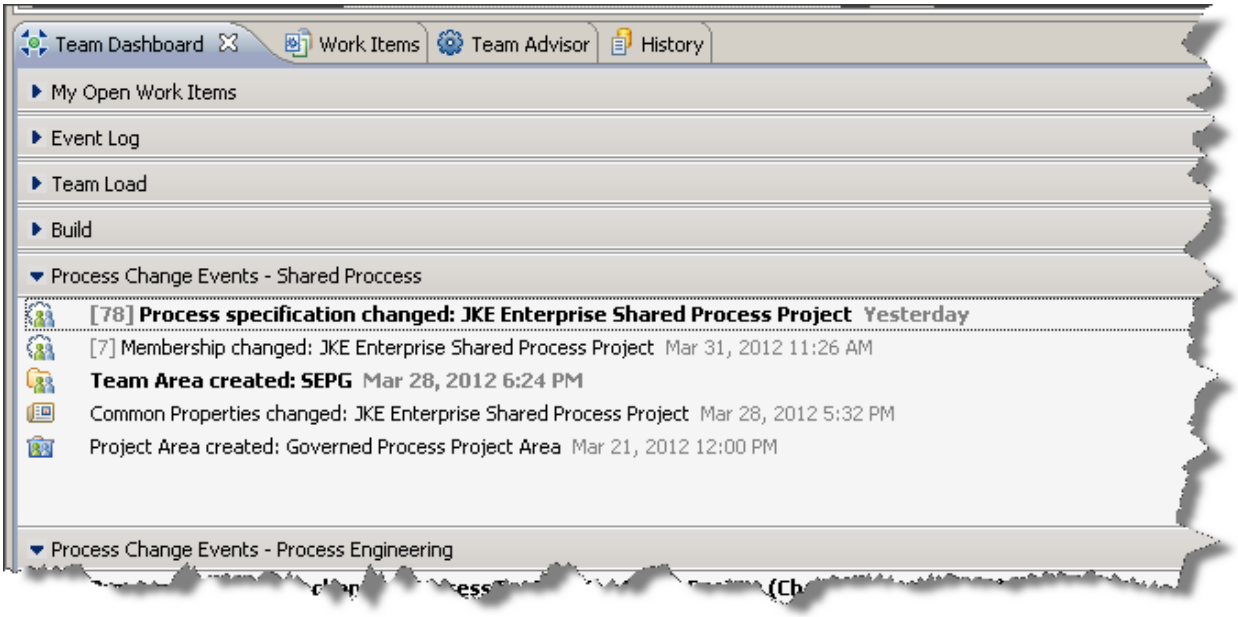


We see a list of the revisions made to the file with a time stamp and indication of who made the change. We select multiple versions from the history, right-click and select **Compare**.



We see the changes from one version to the other indicated in the two versions of the source file. These changes are fairly cryptic because they are in the “vocabulary” of the XML structure of the file. Also note that user and role information is not stored in the configuration source.

We can also view process and membership change events. On the **Team Dashboard** view, we observe the membership and process change events reported from the Team Concert feed service.



The events will tell you that a change has been made, by whom and a date/time stamp. But the details of what was changed are not exposed by the events.

Summary

In this article we examined the use of shared process configuration and a process change authorization task to manage changes to the controlled part of the process configuration. We also looked at how Team Concert exposes information on process changes that were made. While more detail of process changes reported by RTC is highly desired, we are showing you what can be done now.

Here are some requests for enhancement to these capabilities on jazz.net:

Improved Templatization of Project Areas Across CLM

https://jazz.net/jazz/web/projects/Jazz%20Foundation#action=com.ibm.team.workitem.viewWorkItem&id=199691&tab=links

Support "final" on a finer granularity than the configuration-data element for delta configuration

https://jazz.net/jazz/web/projects/Jazz%20Foundation#action=com.ibm.team.workitem.viewWorkItem&itemId=_7sTX0Hh1EeGNGIYqvsTwKg

Allow user to control when changes to a process are "deployed".

https://jazz.net/jazz/web/projects/Jazz%20Foundation#action=com.ibm.team.workitem.viewWorkItem&id=139667

Make it easier to move/copy a process from test environment to real environment

https://jazz.net/jazz/web/projects/Jazz%20Foundation#action=com.ibm.team.workitem.viewWorkItem&id=200389