Internal Control Audit Support

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

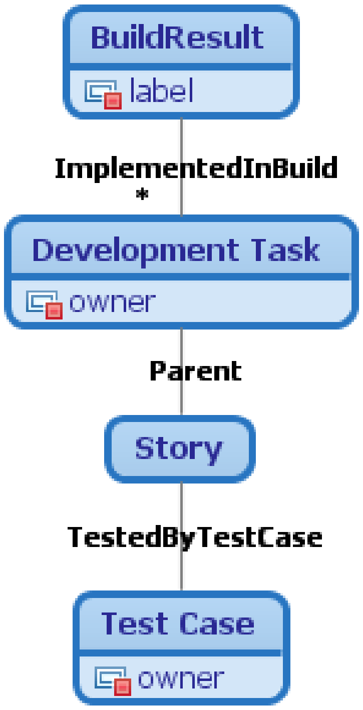
To support internal control audits, you first have to document how you have implemented the controls then prove that your teams are following them. The other articles in this series give examples of how you can automate internal controls related to work authorization, segregation of duties and process change control.

Part 1: Generating Audit Reports

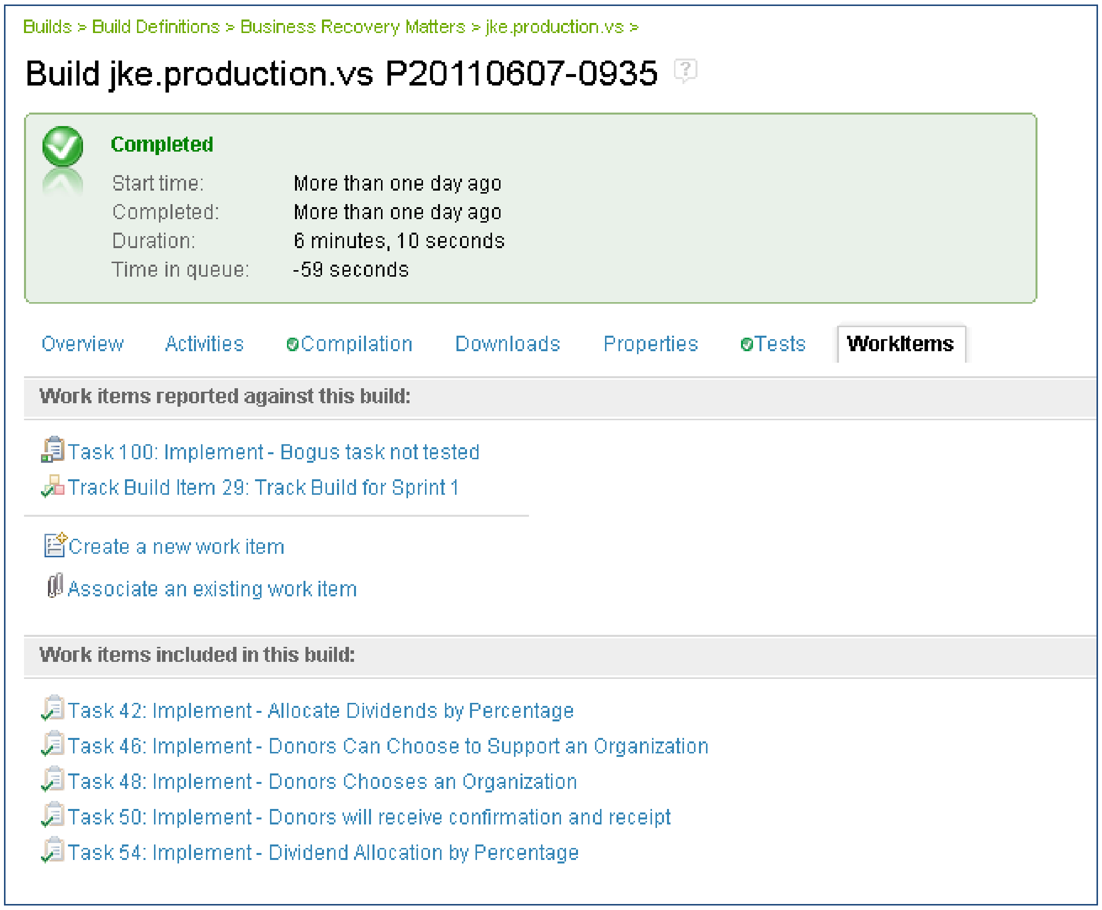
Let's start with proving adherence to controls. A key element of reporting compliance is proving a history of consistent adherence. Here are two examples of audit reports that can be generated for a build.

In the first example, we generate a report on a build that flags any violations of the segregation of duties between development and testing. It will also report a quality violation for any task included in the build for which a corresponding test case is not found. This is a custom report that was built using Rational Publishing Engine and the reportable REST APIs exposed by Rational Team Concert and Rational Quality Manager. This is the same report example used in the Segregation of Duties article

These are the records and their relationships used in the Money That Matters sample application that are relevant for this example:



Here is the Build Result in the example that we will report against:



Builds > Build Definitions > Business Recovery Matters > jke.production.vs >

Build jke.production.vs P20110607-0935

Completed

Start time:	More than one day ago
Completed:	More than one day ago
Duration:	6 minutes, 10 seconds
Time in queue:	-59 seconds

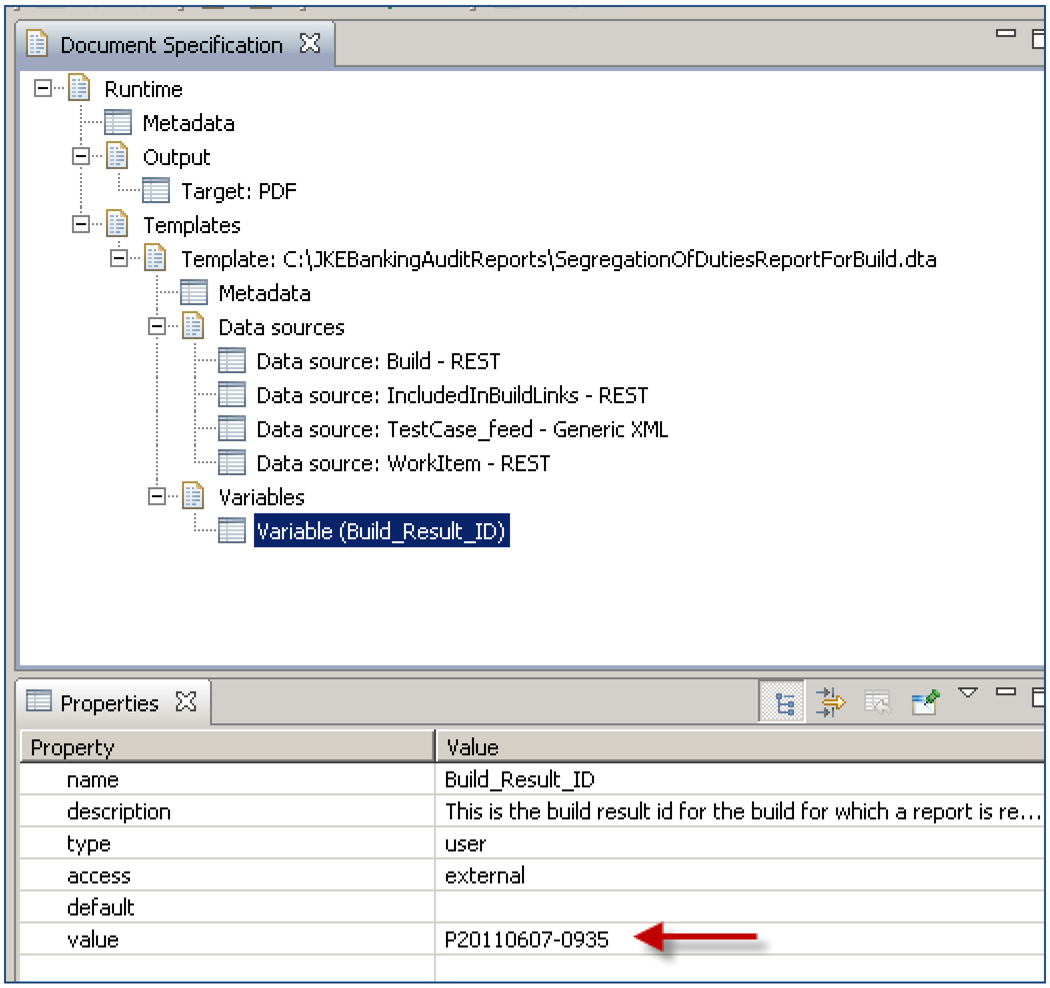
Work items reported against this build:

- Task 100: Implement - Bogus task not tested
- Track Build Item 29: Track Build for Sprint 1

Work items included in this build:

- Task 42: Implement - Allocate Dividends by Percentage
- Task 46: Implement - Donors Can Choose to Support an Organization
- Task 48: Implement - Donors Chooses an Organization
- Task 50: Implement - Donors will receive confirmation and receipt
- Task 54: Implement - Dividend Allocation by Percentage

The document specification for the report uses an input variable that defines the Build Label of the build result on which we want to report. This value can be changed to make the report against any build result desired.

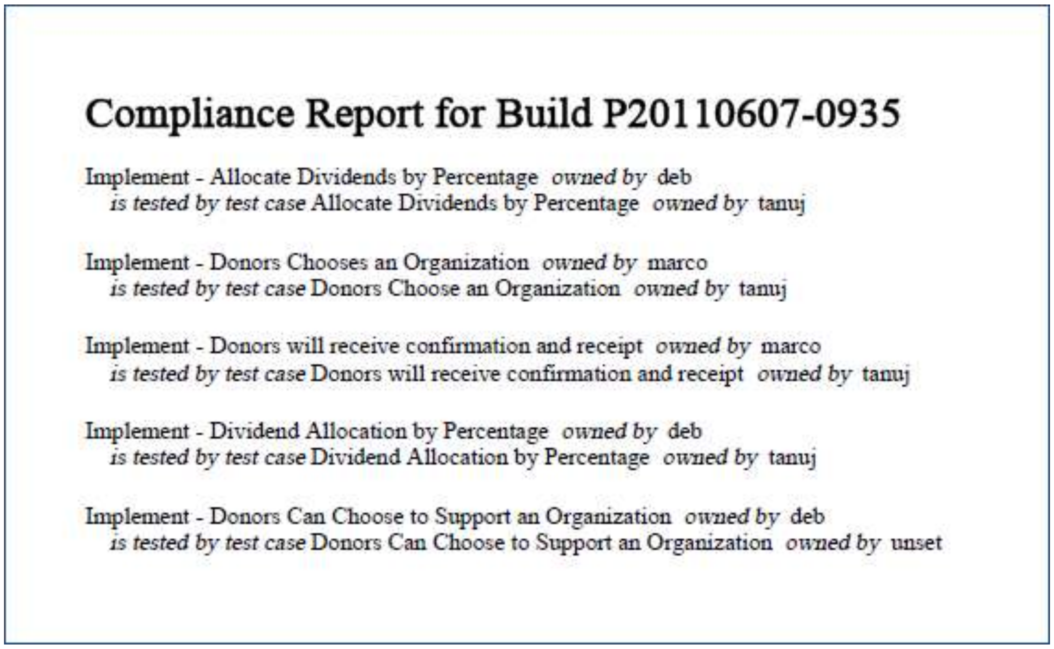

Document Specification

- Runtime
  - Metadata
  - Output
    - Target: PDF
  - Templates
    - Template: C:\JKEBankingAuditReports\SegregationOfDutiesReportForBuild.dta
      - Metadata
      - Data sources
        - Data source: Build - REST
        - Data source: IncludedInBuildLinks - REST
        - Data source: TestCase_feed - Generic XML
        - Data source: WorkItem - REST
      - Variables
        - Variable (Build_Result_ID)

Properties

Property	Value
name	Build_Result_ID
description	This is the build result id for the build for which a report is re…
type	user
access	external
default	
value	P20110607-0935

The published document from the Money That Matters example looks like this:



Compliance Report for Build P20110607-0935

Implement - Allocate Dividends by Percentage *owned by* deb
is tested by test case Allocate Dividends by Percentage *owned by* tanuj

Implement - Donors Chooses an Organization *owned by* marco
is tested by test case Donors Choose an Organization *owned by* tanuj

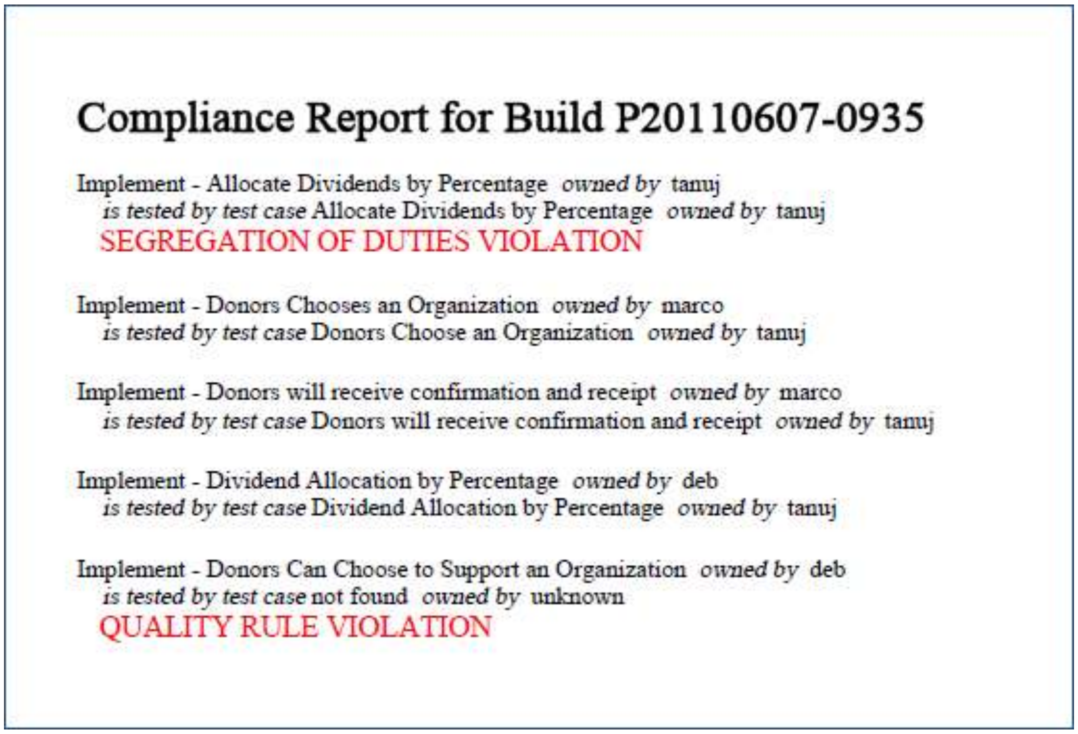
Implement - Donors will receive confirmation and receipt *owned by* marco
is tested by test case Donors will receive confirmation and receipt *owned by* tanuj

Implement - Dividend Allocation by Percentage *owned by* deb
is tested by test case Dividend Allocation by Percentage *owned by* tanuj

Implement - Donors Can Choose to Support an Organization *owned by* deb
is tested by test case Donors Can Choose to Support an Organization *owned by* unset

Although the report looks a bit suspect because the test case name and implementation work item name are almost exactly alike, this is correct. The naming conventions used in the Money That Matters sample application are designed to make the relationships between elements easily understood.

Now if we tweak the example data a bit we can cause some violations, as reported when we run the report again:



Compliance Report for Build P20110607-0935

Implement - Allocate Dividends by Percentage *owned by* tanuj
is tested by test case Allocate Dividends by Percentage *owned by* tanuj
SEGREGATION OF DUTIES VIOLATION

Implement - Donors Chooses an Organization *owned by* marco
is tested by test case Donors Choose an Organization *owned by* tanuj

Implement - Donors will receive confirmation and receipt *owned by* marco
is tested by test case Donors will receive confirmation and receipt *owned by* tanuj

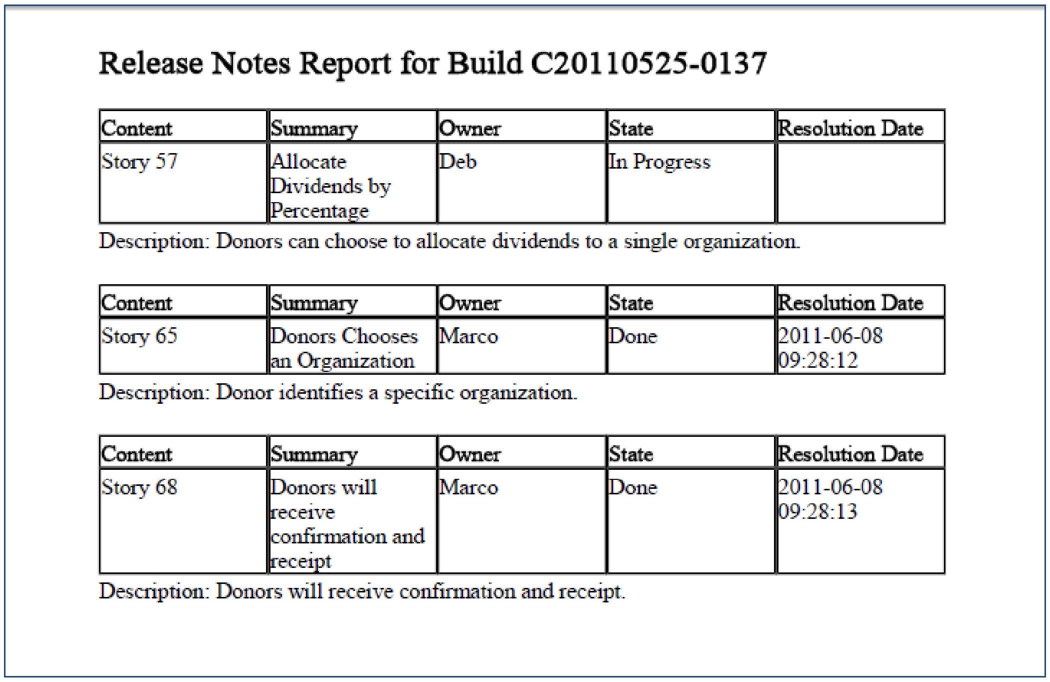
Implement - Dividend Allocation by Percentage *owned by* deb
is tested by test case Dividend Allocation by Percentage *owned by* tanuj

Implement - Donors Can Choose to Support an Organization *owned by* deb
is tested by test case not found *owned by* unknown
QUALITY RULE VIOLATION

In our second example, we use the Rational Publishing Engine launcher to run an audit report on the build contents. Our Money That Matters project has a set of development tasks that are implemented in a build. Each development task has a parent Story that provides the information we want to report as the content for the build. Our report traverses the **ImplementedInBuild** links to the development tasks then the **Parent/Child** links to the Stories.

We are also using a variable to provide the build label, but we are using a different build label for this example.

Here is the result of this report:



Release Notes Report for Build C20110525-0137

Content	Summary	Owner	State	Resolution Date
Story 57	Allocate Dividends by Percentage	Deb	In Progress	

Description: Donors can choose to allocate dividends to a single organization.

Content	Summary	Owner	State	Resolution Date
Story 65	Donors Chooses an Organization	Marco	Done	2011-06-08 09:28:12

Description: Donor identifies a specific organization.

Content	Summary	Owner	State	Resolution Date
Story 68	Donors will receive confirmation and receipt	Marco	Done	2011-06-08 09:28:13

Description: Donors will receive confirmation and receipt.

The example reports we used work against the Money That matters sample in CLM 2011 and are downloadable here. Although not demonstrated here, you can configure an ant script to generate a set of reports automatically with each build.

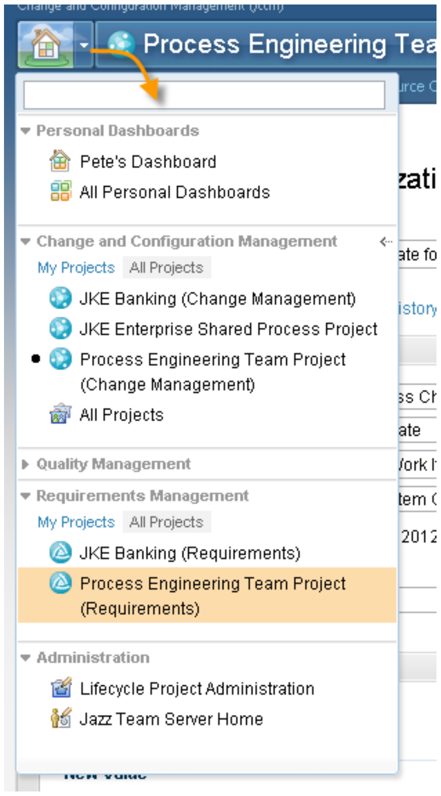
Part 2: Implementing Internal Controls

Now we'll show an example of how a Process Engineering Team leverages the CLM solution to manage Internal Controls and CMMI compliance. This team

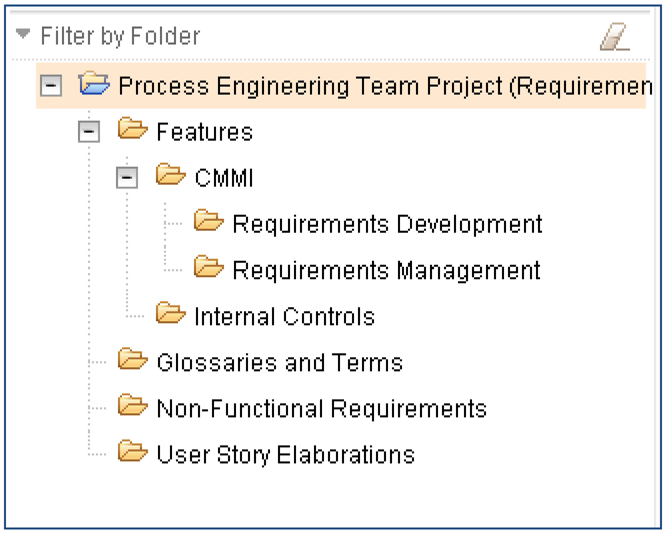
- Uses Rational Requirements Composer to capture features representing Internal Controls and CMMI process capabilities
- Leverages team discussions on how they could automate a specific process capability
- Creates User Story Elaborations to include the agreed-upon solutions for the different internal controls and guidelines to be implemented in the software development process

- Links each process requirement to an analysis task and a Process Change Authorization record for the Process Engineering team to manage the work of implementing the proposed changes. This provides an audit trail and helps implement changes consistently across an organization.

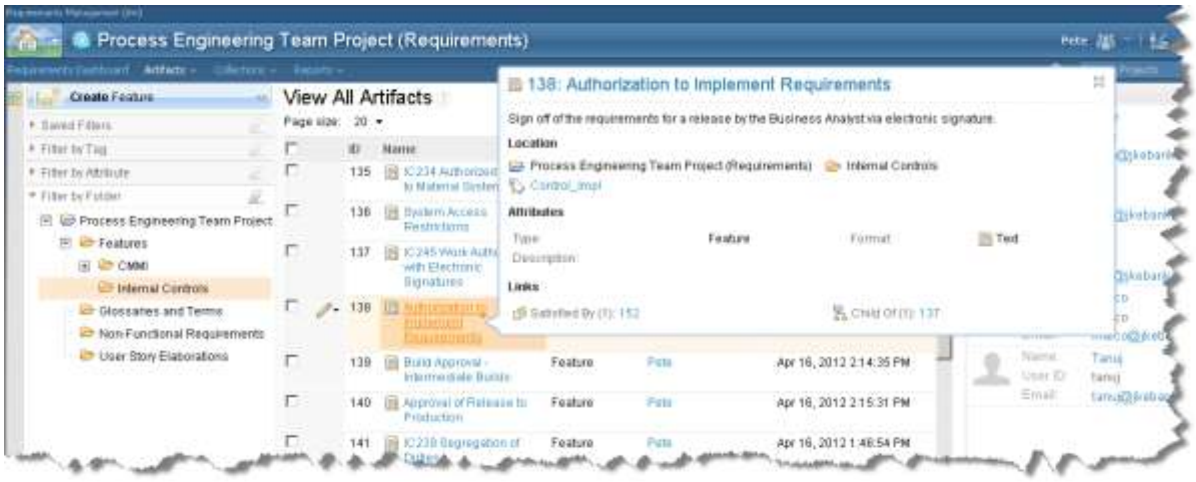
The process engineering team has their own requirements project called *Process Engineering Team Project (Requirements)*


Process Engineering Tea
Personal Dashboards
Pete's Dashboard
All Personal Dashboards
Change and Configuration Management
My Projects
All Projects
JKE Banking (Change Management)
JKE Enterprise Shared Process Project
Process Engineering Team Project (Change Management)
All Projects
Quality Management
Requirements Management
My Projects
All Projects
JKE Banking (Requirements)
Process Engineering Team Project (Requirements)
Administration
Lifecycle Project Administration
Jazz Team Server Home

In the **Features** folder we see two **CMMI** process areas captured and a folder for **Internal Controls**.



In the Internal Controls folder, we hover over *Feature 138: Authorization to Implement Requirements* to see the description and other details.


Process Engineering Team Project (Requirements)

Create Feature

View All Artifacts

Page size: 20

138: Authorization to Implement Requirements

Sign off of the requirements for a release by the Business Analyst via electronic signature.

Location

Process Engineering Team Project (Requirements) Internal Controls

Control_impl

Attributes

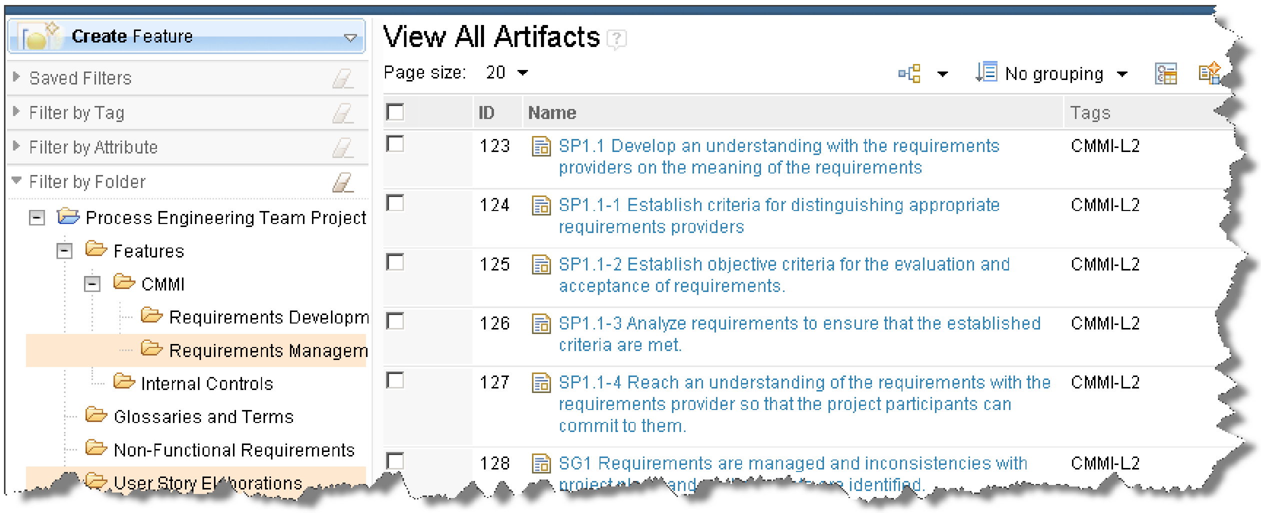
Type	Feature	Format	Text
Description			

Links

Satisfied By (1): 152 Child Of (1): 137

ID	Name			
135	IC234 Authorized to Material System			
136	System Access Restrictions			
137	IC245 Work Auth with Electronic Signatures			
138	Authorization to Implement Requirements			
139	Build Approval - Intermediate Builds	Feature	Pete	Apr 16, 2012 2:14:35 PM
140	Approval of Release to Production	Feature	Pete	Apr 16, 2012 2:15:31 PM
141	IC238 Segregation of Duties	Feature	Pete	Apr 16, 2012 1:48:54 PM

In the CMMI folder and sub-folder to review the **Requirements Management** Specific Goals (SG) and Specific Practices (SP).


Create Feature

Saved Filters

Filter by Tag

Filter by Attribute

Filter by Folder

- Process Engineering Team Project
  - Features
    - CMMI
      - Requirements Developm
      - Requirements Managem
    - Internal Controls
  - Glossaries and Terms
  - Non-Functional Requirements
  - User Story Elaborations

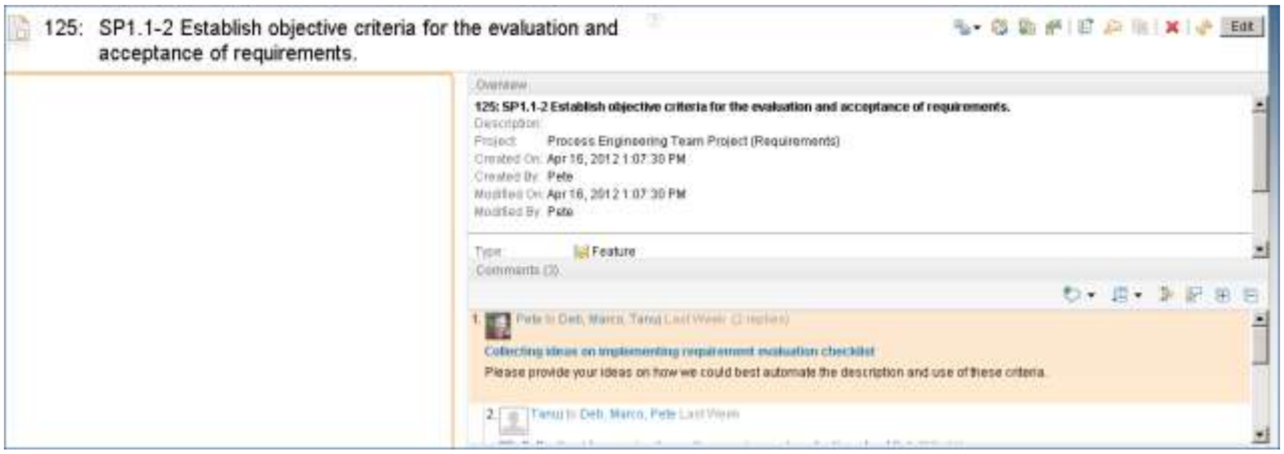
View All Artifacts

Page size: 20 No grouping

ID	Name	Tags
123	SP1.1 Develop an understanding with the requirements providers on the meaning of the requirements	CMMI-L2
124	SP1.1-1 Establish criteria for distinguishing appropriate requirements providers	CMMI-L2
125	SP1.1-2 Establish objective criteria for the evaluation and acceptance of requirements.	CMMI-L2
126	SP1.1-3 Analyze requirements to ensure that the established criteria are met.	CMMI-L2
127	SP1.1-4 Reach an understanding of the requirements with the requirements provider so that the project participants can commit to them.	CMMI-L2
128	SG1 Requirements are managed and inconsistencies with project plans and work products are identified.	CMMI-L2

Here you see a capture of the SGs and SPs as specified by CMMI. We have also leveraged tags to indicate the CMMI level supported.

On *Feature 125: SP1.1-2 Establish objective criteria for the evaluation and acceptance of requirements* we can review the details, including a team discussion on options for automating this practice.


125: SP1.1-2 Establish objective criteria for the evaluation and acceptance of requirements.

Overview

125: SP1.1-2 Establish objective criteria for the evaluation and acceptance of requirements.

Description:

Project: Process Engineering Team Project (Requirements)

Created On: Apr 16, 2012 1:07:30 PM

Created By: Pete

Modified On: Apr 16, 2012 1:07:30 PM

Modified By: Pete

Type: Feature

Comments (3)

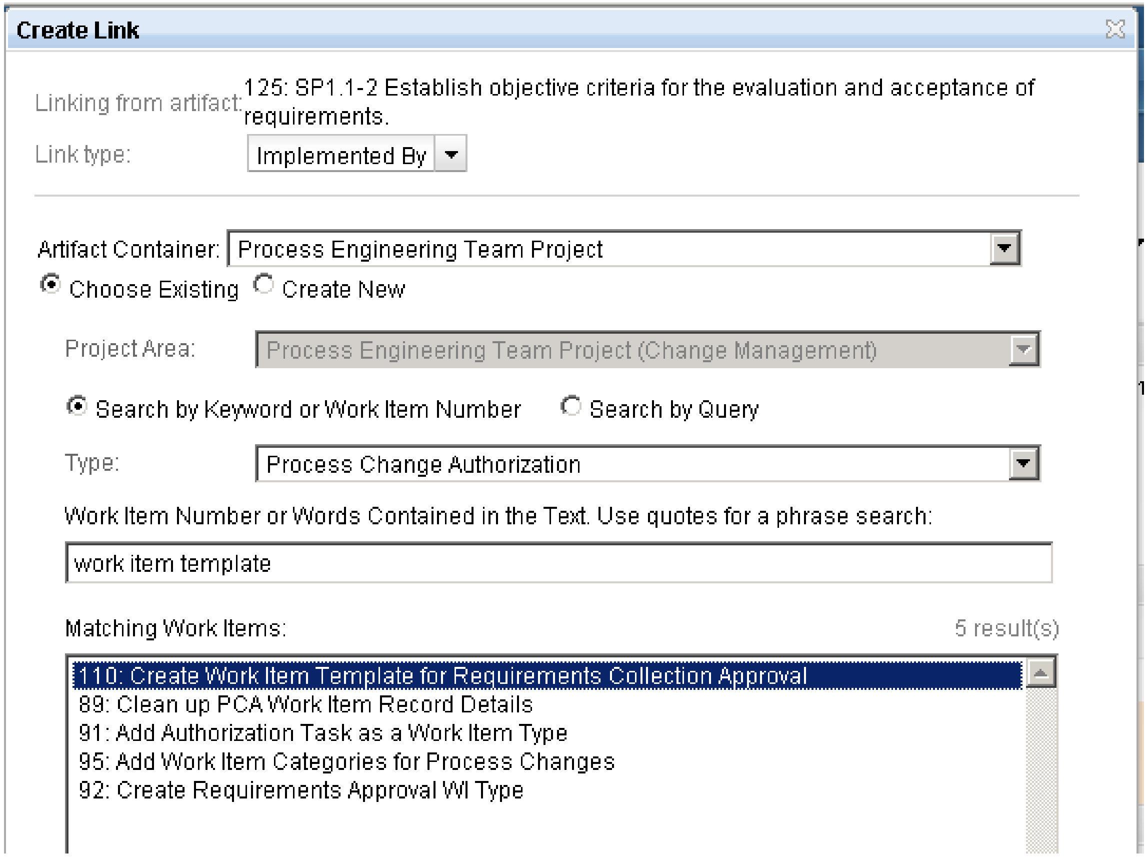
1. Pete to Deb, Marco, Tanuj Last Week (2 replies)

Collecting ideas on implementing requirement evaluation checklist

Please provide your ideas on how we could best automate the description and use of these criteria.

2. Tanuj to Deb, Marco, Pete Last Week

Based on the discussion, the process engineer links this Feature to a Process Change Authorization already in the Process Engineering Change Management project.


Create Link

Linking from artifact: 125: SP1.1-2 Establish objective criteria for the evaluation and acceptance of requirements.

Link type: Implemented By

Artifact Container: Process Engineering Team Project

[✓ Choose Existing] Create New

Project Area: Process Engineering Team Project (Change Management)

[✓ Search by Keyword or Work Item Number] Search by Query

Type: Process Change Authorization

Work Item Number or Words Contained in the Text. Use quotes for a phrase search:

work item template

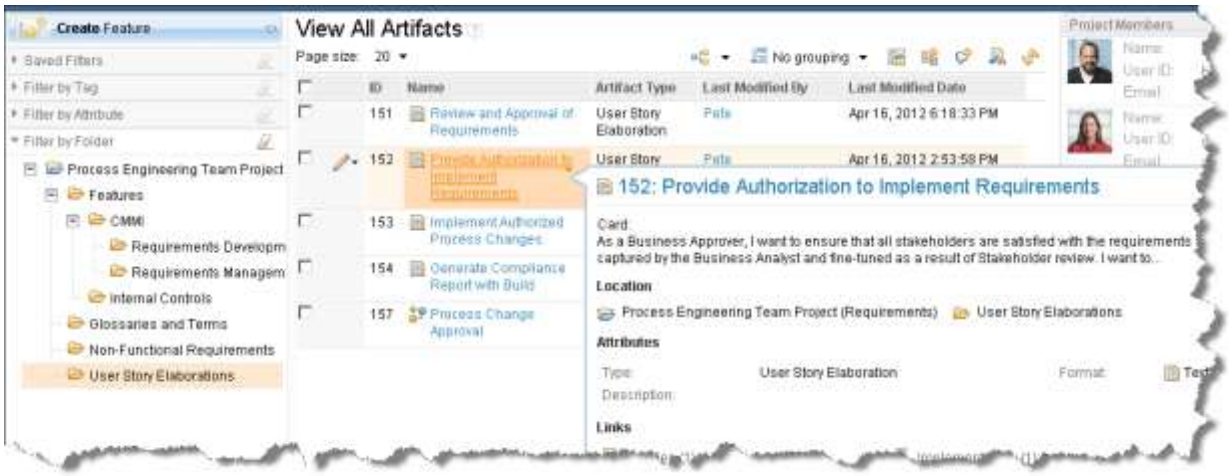
Matching Work Items: 5 result(s)

110: Create Work Item Template for Requirements Collection Approval
89: Clean up PCA Work Item Record Details
91: Add Authorization Task as a Work Item Type
95: Add Work Item Categories for Process Changes
92: Create Requirements Approval WI Type

Often, the way a process is tailored for a project or organization is influenced by many factors including regulations, internal controls and guidelines. In these cases it is extremely useful to maintain the set of User Story Elaborations that represent how the process is used. This helps tie test cases to the requirements and helps ensure that a change to one process requirement does not break the implementation for another.

An alternative for a more robust and comprehensive representation of the process is to use Rational Method Composer.

Let’s look at an example of how this team uses User Story Elaborations. Here you see where the User Story Elaborations are managed in the requirements project.



ID	Name	Artifact Type	Last Modified By	Last Modified Date
151	Review and Approval of Requirements	User Story Elaboration	Pete	Apr 16, 2012 6:18:33 PM
152	Provide Authorization to Implement Requirements	User Story	Pete	Apr 16, 2012 2:53:58 PM
153	Implement Authorized Process Changes			
154	Generate Compliance Report with Build			
157	Process Change Approval			

152: Provide Authorization to Implement Requirements

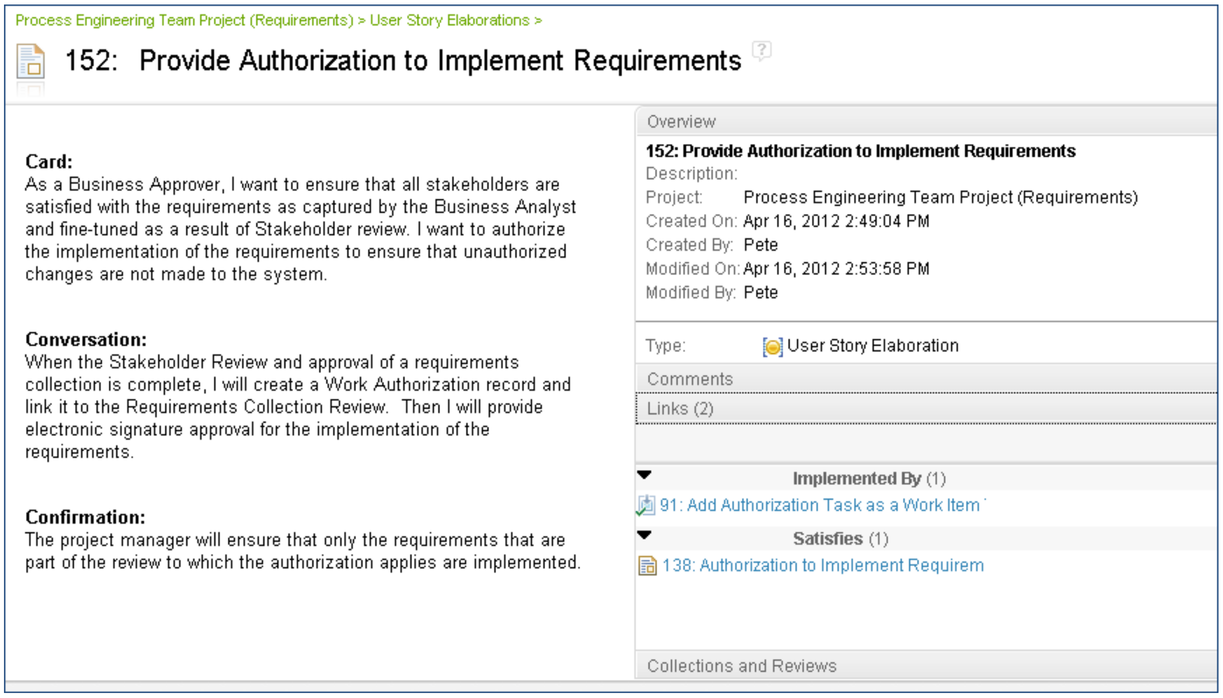
Card:
As a Business Approver, I want to ensure that all stakeholders are satisfied with the requirements captured by the Business Analyst and fine-tuned as a result of Stakeholder review. I want to...

Location
Process Engineering Team Project (Requirements) User Story Elaborations

Attributes
Type: User Story Elaboration Format:
Description:

Links

And here are the details for *User Story Elaboration 152: Provide Authorization to Implement Requirements*.


Process Engineering Team Project (Requirements) > User Story Elaborations >

152: Provide Authorization to Implement Requirements

Card:
As a Business Approver, I want to ensure that all stakeholders are satisfied with the requirements as captured by the Business Analyst and fine-tuned as a result of Stakeholder review. I want to authorize the implementation of the requirements to ensure that unauthorized changes are not made to the system.

Conversation:
When the Stakeholder Review and approval of a requirements collection is complete, I will create a Work Authorization record and link it to the Requirements Collection Review. Then I will provide electronic signature approval for the implementation of the requirements.

Confirmation:
The project manager will ensure that only the requirements that are part of the review to which the authorization applies are implemented.

Overview

152: Provide Authorization to Implement Requirements
Description:
Project: Process Engineering Team Project (Requirements)
Created On: Apr 16, 2012 2:49:04 PM
Created By: Pete
Modified On: Apr 16, 2012 2:53:58 PM
Modified By: Pete

Type: User Story Elaboration

Comments

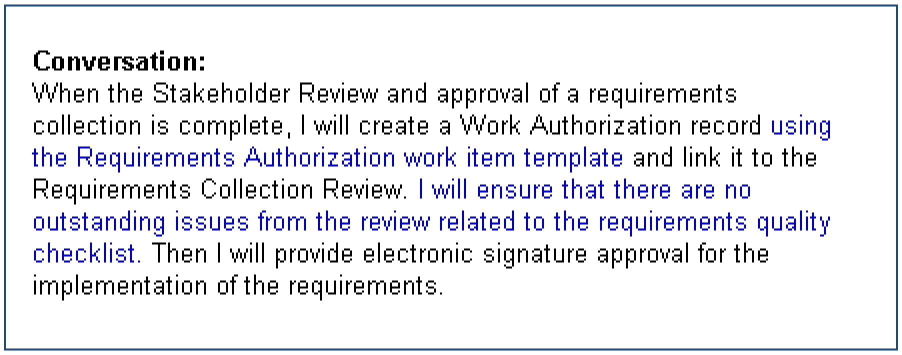
Links (2)

Implemented By (1)
91: Add Authorization Task as a Work Item

Satisfies (1)
138: Authorization to Implement Requirem

Collections and Reviews

Our process engineer edits the User Story Elaboration to add the agreed-upon solution (work item template and review checklist) as part of the conversation:


Conversation:
When the Stakeholder Review and approval of a requirements collection is complete, I will create a Work Authorization record using the Requirements Authorization work item template and link it to the Requirements Collection Review. I will ensure that there are no outstanding issues from the review related to the requirements quality checklist. Then I will provide electronic signature approval for the implementation of the requirements.

Then he links the User Story Elaboration to the Process Change Authorization work item that represents its approval and implementation.

We have not detailed the steps for implementation and testing of the solution. Rather, we provide the observation that management of the work into iterations or releases is accomplished in Team Concert, while capture of test cases and test results is accomplished in Quality Manager. Lifecycle traces help ensure that all requirements are implemented and tested, thus supporting an audit trail for proving CMMI compliance and/or adherence to internal controls. For rollout of consistent process solutions incrementally throughout an organization or Enterprise, the ability to manage the complexity of requirements, implementation projects and testing repeated across parts of the organization is challenging. The foundational capabilities in the Collaborative Lifecycle Management solution provide a means of accomplishing the work most efficiently and effectively.

In this article we showed an example of using Collaborative Lifecycle Management to help manage process compliance to internal controls and industry standards.