

Using IBM Rational Team Concert with other Software Configuration Management Systems

Khurram Nizami
Worldwide Enablement, IBM
knizami@us.ibm.com

IBM Software

Innovate2011

The Premier Event for Software and Systems Innovation



More | **August 11**, Delhi



Agenda

- **RTC SCM integration overview**
- SCM integrations with RTC
 - ▶ Types of integrations and deployment considerations
 - Subversion, Git
 - ClearCase



Integrating to other SCM's with Rational Team Concert

- RTC provides a modern change set / stream based SCM system that will meet the needs of nearly every software development organization
- However, most organizations already have existing SCM's in place and may be using multiple SCM's across groups in their organization
- It is very easy to get siloed from other development teams in separate repositories or different SCM tools
- RTC can bring heterogeneous SCM environments together increasing collaboration, productivity, and control



Differences between Jazz SCM and other SCM's

- Change set based from its roots
- Doesn't use branches – instead, streams and flows
- Compare “snapshots” between builds
- Suspend and resume change sets
- Cherry-pick change sets
- Resolve structural conflicts
- Distributed SCM support



Benefits Achieved by integrating your existing SCM

- Establish an open integration and lifecycle management platform driven by Jazz
- Realtime planning and rebalancing – to your needs: agile and waterfall approaches
- Process Awareness, Enactment, and Enforcement – Scrum, Waterfall, Hybrid
- Multi-level dashboards with realtime metrics / transparency for individual, team, project, and programs
- Team collaboration features – feeds on everything, rich ALM queries, discussion threads, tags, project events



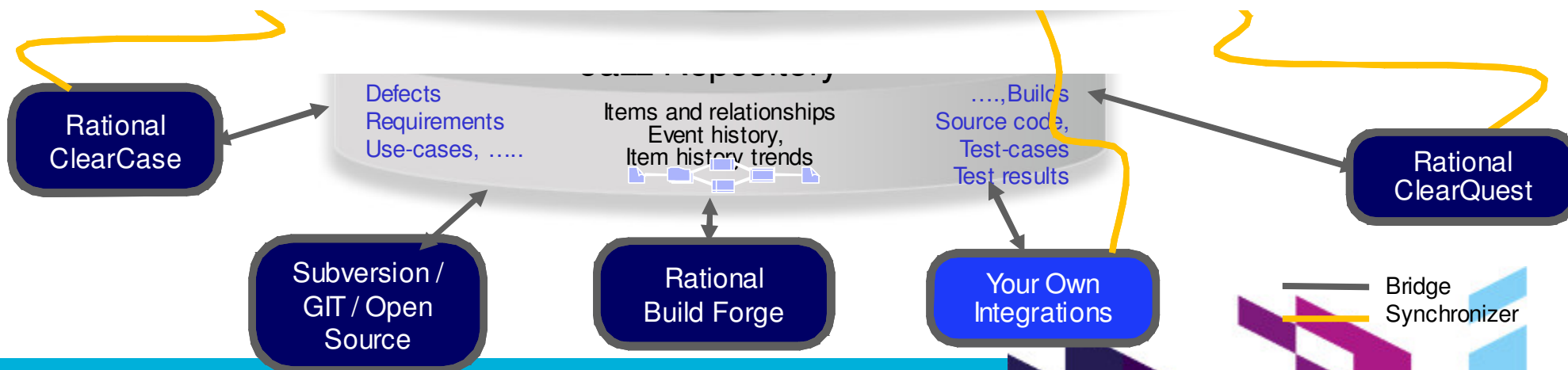
Beyond SCM with Jazz – Enable collaboration and teaming

- It's no longer simply about the “code” or the SCM system – teams need to effectively collaborate in order to deliver good software
- Leverage linking as much as possible between deliverables: builds, defects, plans...
- A platform for rich integration across *your* application life cycle.

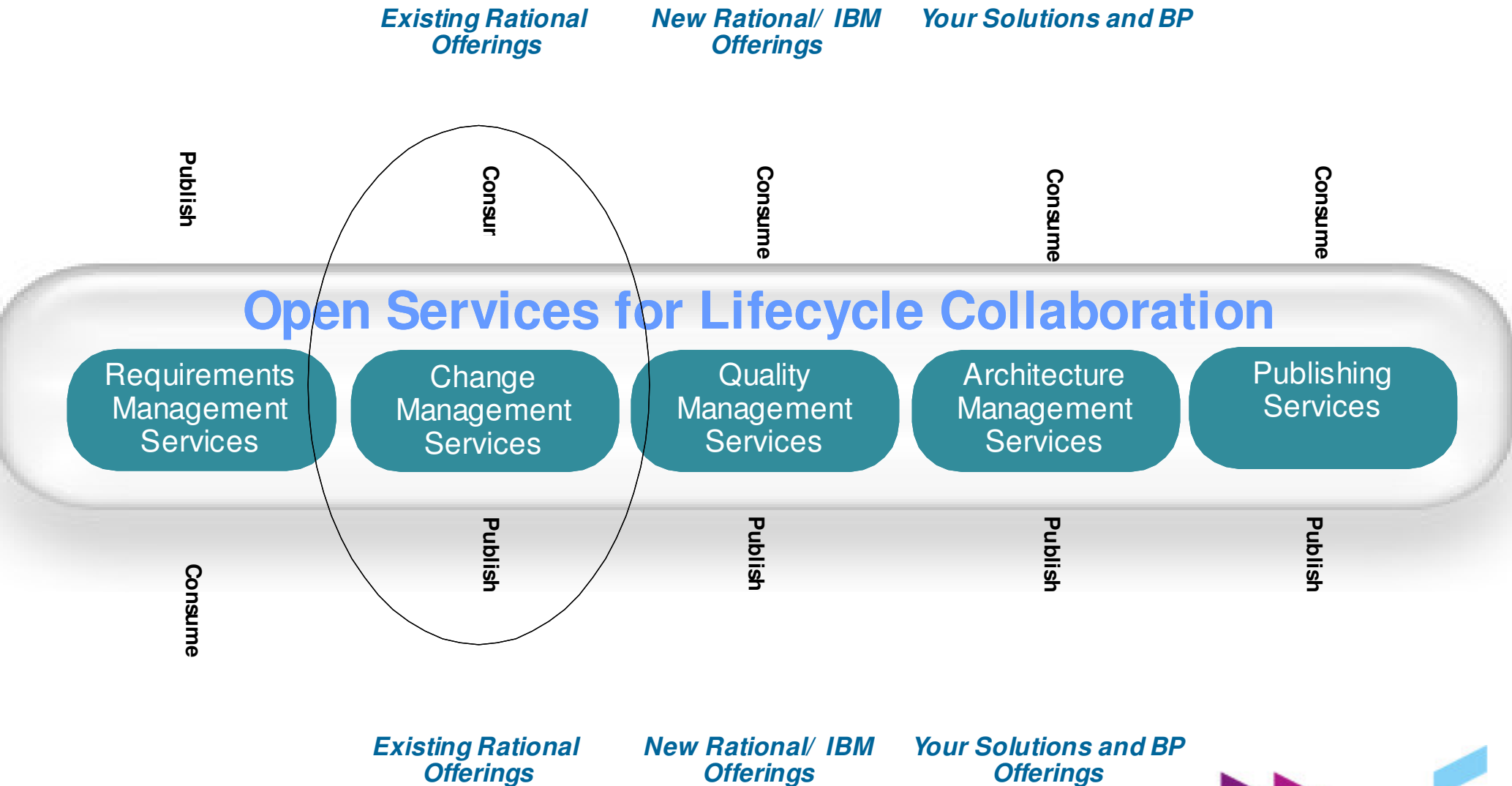
The screenshot displays the IBM Jazz IDE interface. The left pane shows a 'Defect 1' with the summary 'AssertEquals' and status 'Resolved'. The 'Links' section is expanded to show 'Change-sets', with a red underline under 'Changes in new import - [smr99] Changes'. The right pane shows a 'Build CppUnit Team build' with ID 'I20080314-2213'. The 'Contribution Summary' lists various metrics: Changes (with a red underline and 'Show changes' link), Downloads (1 download), External Links (1 link), Logs (1 log), Repository Workspace (build), Snapshot (CppUnit Team build_20080314-2213, with a red underline), and Work items (None included). The bottom pane shows a table of builds:

Build	Label	Progress	Estimated Completion	Start Time
✓ CppUnit Team build	I20080314-2213	Completed		March 14, 2008 10

Jazz enables RTC to be an open, flexible, pluggable platform



Our solution provides an integration framework for ALM protecting existing and future customer investments





Existing SCM users can adopt and/or transition to RTC based on their needs

Iteration Planning

- Integrated iteration planning and execution
- Task estimation linked to key milestones
- Out of the box agile process templates

Project Transparency

- Customizable web based dashboards
- Real time metrics and reports
- Project milestone tracking and status

SCM

- Integrated stream management with flow relationships
- Component level baselines
- Server-based sandboxes
- Identifies component in streams and available baselines
- ClearCase connector

Subversion / ClearCase

Change Management

- Defects, enhancements and conversations
- Query results view and share queries with team or member
- Support for approvals and discussions
- ClearQuest connector
- Query editor interface

Build

- Work item and change set traceability
- Local or remote build servers
- Supports ant and command line tools
- Integration with build forge
- Build definitions for team and private builds

Jazz Team Server

- Single structure for project related artifacts
- World-class team on-boarding / offboarding including team membership, sub-teams and project inheritance
- Role-based operational control for flexible definition of process and capabilities
- Team advisor for defining / refining "rules" and enabling continuous improvement
- Process enactment and enforcement
- In-context collaboration shows team members and status of their work



Establish an ALM solution powered by Jazz using the SCM of your choice

Rational solution for Collaborative Lifecycle Management

CREATE SOFTWARE

Real-time Planning, Lifecycle Traceability, Team Collaboration, Development Intelligence, Continuous Improvement



Requirements Management

Planning, Change, Configuration & Build Management

Quality Management

Rational ClearCase

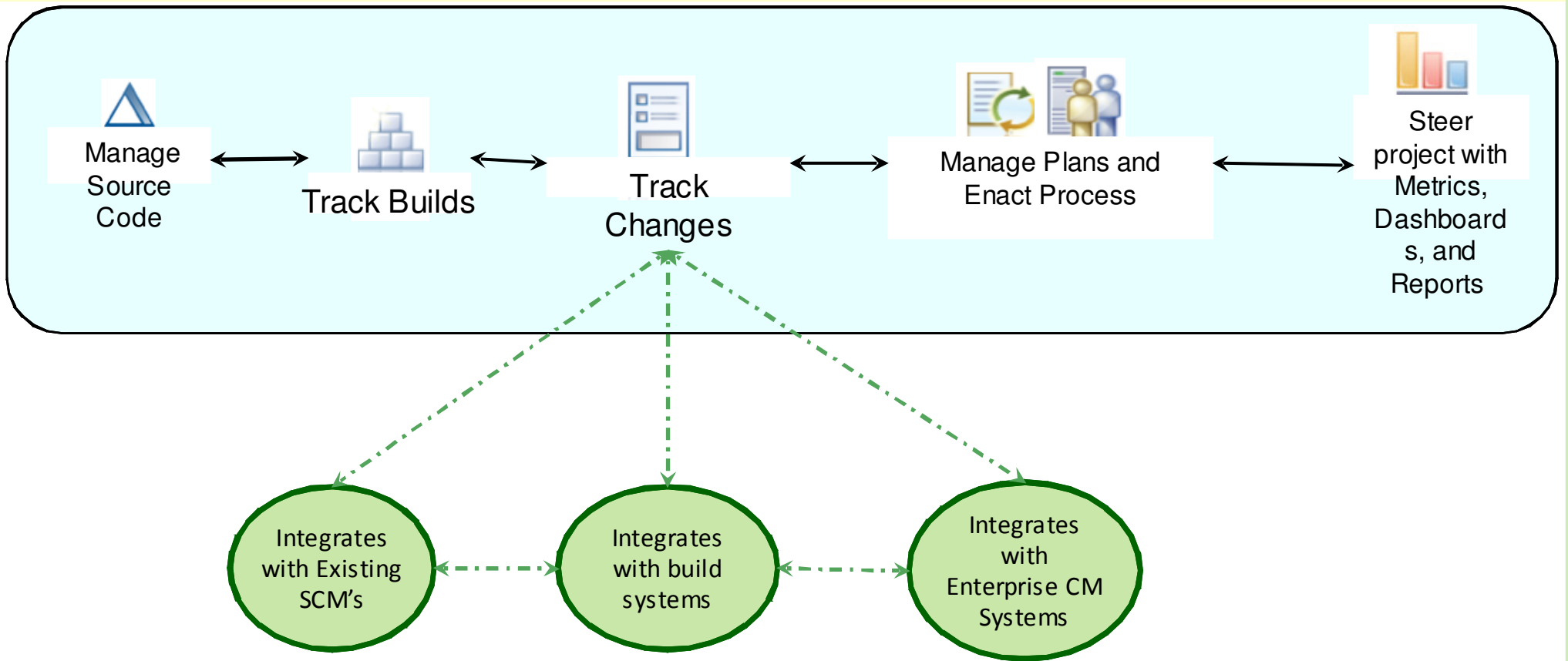
Subversion



There are many points of integration with RTC, but work items are generally the hub

- Quality of releases improves 12%*
- Project governance improvements of 12%

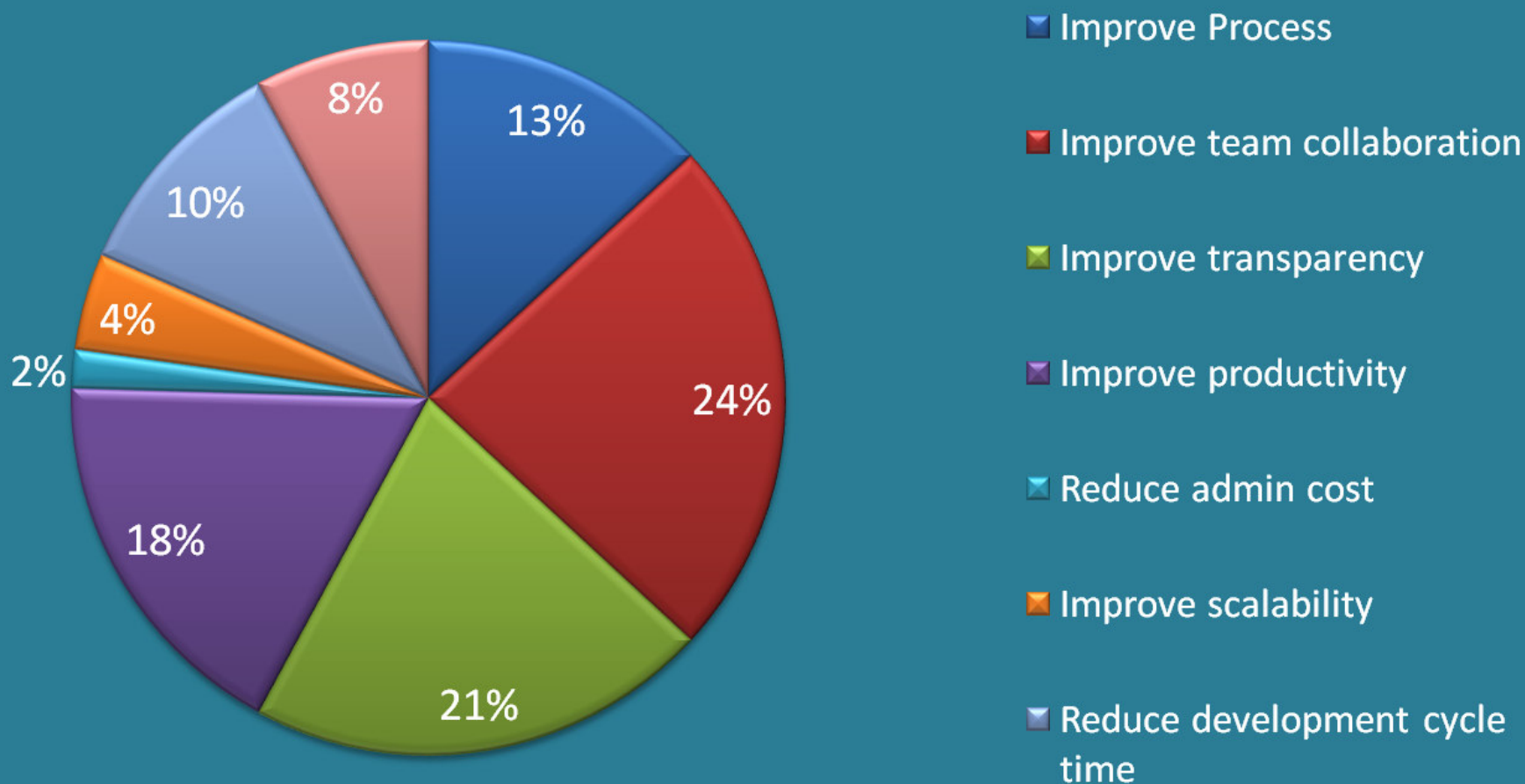
- Team productivity increases of up to 50%
- Team collaboration performance up 25%*



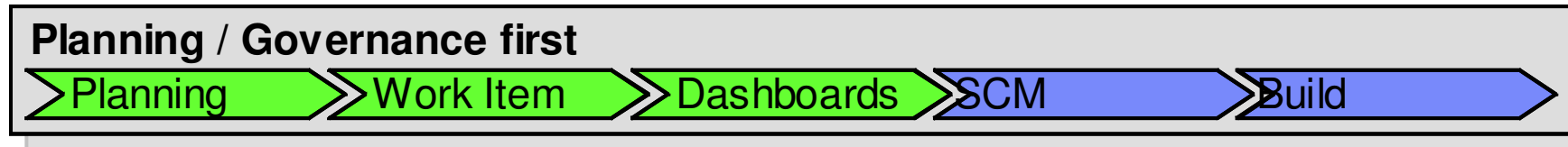
* Based on IBM customer surveys

Our customer stated ROI benefits: Complementary to SCM

Customer Quoted Benefits from RTC



Planning/Governance Adoption Pattern is Very Common



- Many of our customers are using RTC to address task coordination/ planning – integrating to their existing systems
- A Major Insurance company has 500 agile planners who use RTC to plan and collaborate their daily plans, iterations and teams. They don't care what scm or build tools their teams use. But they wanted a platform to encourage collaboration and transparency and consistency (process).
- Major Electronics Co.– Uses Rational Team Concert for 400 globally distributed developers to coordinate (plan) and collaborate (communicate) and encourage re-use of common components (using team structure)



Agenda

- **RTC SCM integration overview**
- SCM integrations with RTC
 - ▶ Types of integrations and deployment considerations
 - Subversion, Git
 - ClearCase



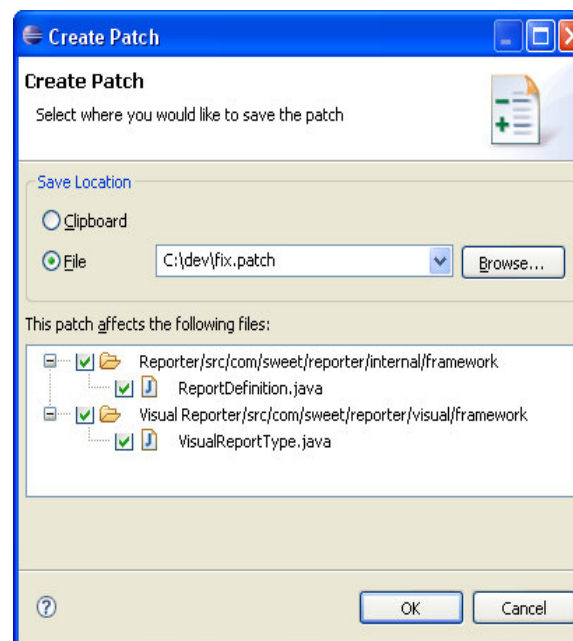
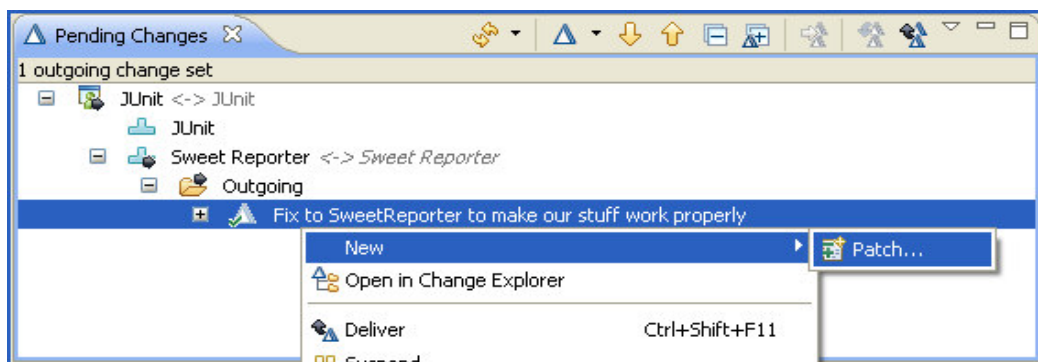
Types of integrations: Jazz connectors

- **Synchronizer – used to synchronize changes between two systems**
 - ▶ Requires synchronization rules to be configured, synch conflicts to be resolved when they surface
 - ▶ More work to write your own since it requires you to handle rules for handling changes between the systems as well as mapping data
 - ▶ Synchronizers are used in co-existence situations when you want to keep two systems around temporarily or indefinitely
 - ▶ Available for ClearCase and ClearQuest – Most common synchronizers are for work items
- **Bridge – most common, used to link changes between a system and work items**
 - ▶ Use change / configuration management tool alongside Jazz-based products
 - ▶ Easily create and navigate relationships between objects in different repositories
 - Multiple teams using separate SCM's as well as different SCM repositories can collaborate and work together on work items / change requests. Requires little administration overhead to setup and use
 - ▶ More generic, allows you to link any system, not just SCM to RTC
 - ▶ If one doesn't exist, this is the simplest to write
 - ▶ Available for ClearCase, Synergy, Subversion, Change, ClearQuest, DOORS, Requirements Composer, Quality Manager



Types of integrations: Client side and Patching

- **Development Client Integration – Eclipse or Visual Studio**
 - ▶ RTC alongside any SCM system that works in Eclipse or Visual Studio
 - ▶ Unified experience for complementary capabilities like planning and work items but no traceability
- **Mylyn – Tasktop integration:** Tasktop provides mylyn integrations for a number of products such as Team System, git, etc.
- **RTC Patch capability - using RTC SCM and interacting with groups using other SCM's**
 - ▶ Allows you to share and take contributions from groups using other SCM's like cvs, subversion, git, etc
 - ▶ Uses the standard “patch” diff format allowing sharing between products like git easy
 - ▶ Provides an easy to use GUI for selecting the files and change sets that a patch will include



Agenda

- **RTC SCM integration overview**
- SCM integrations with RTC
 - ▶ Types of integrations and deployment considerations
 - Subversion, Git
 - ClearCase



Why use RTC with Subversion

Planning, Collaboration, and ALM

- Can complement Subversion without a need for replacement if OSS SCM solution is working well
- Reduced management burden and reduced risk from fully integrated ALM solution that works out of the box with Subversion
- Increased productivity from traceability between scm, build, change management
- Increased transparency and metrics resulting in better planning and management
- Accelerated agile development adoption and execution
- Easy migration from existing tools if desired: Subversion, Bugzilla, Jira, CVS, CSV Import & Update

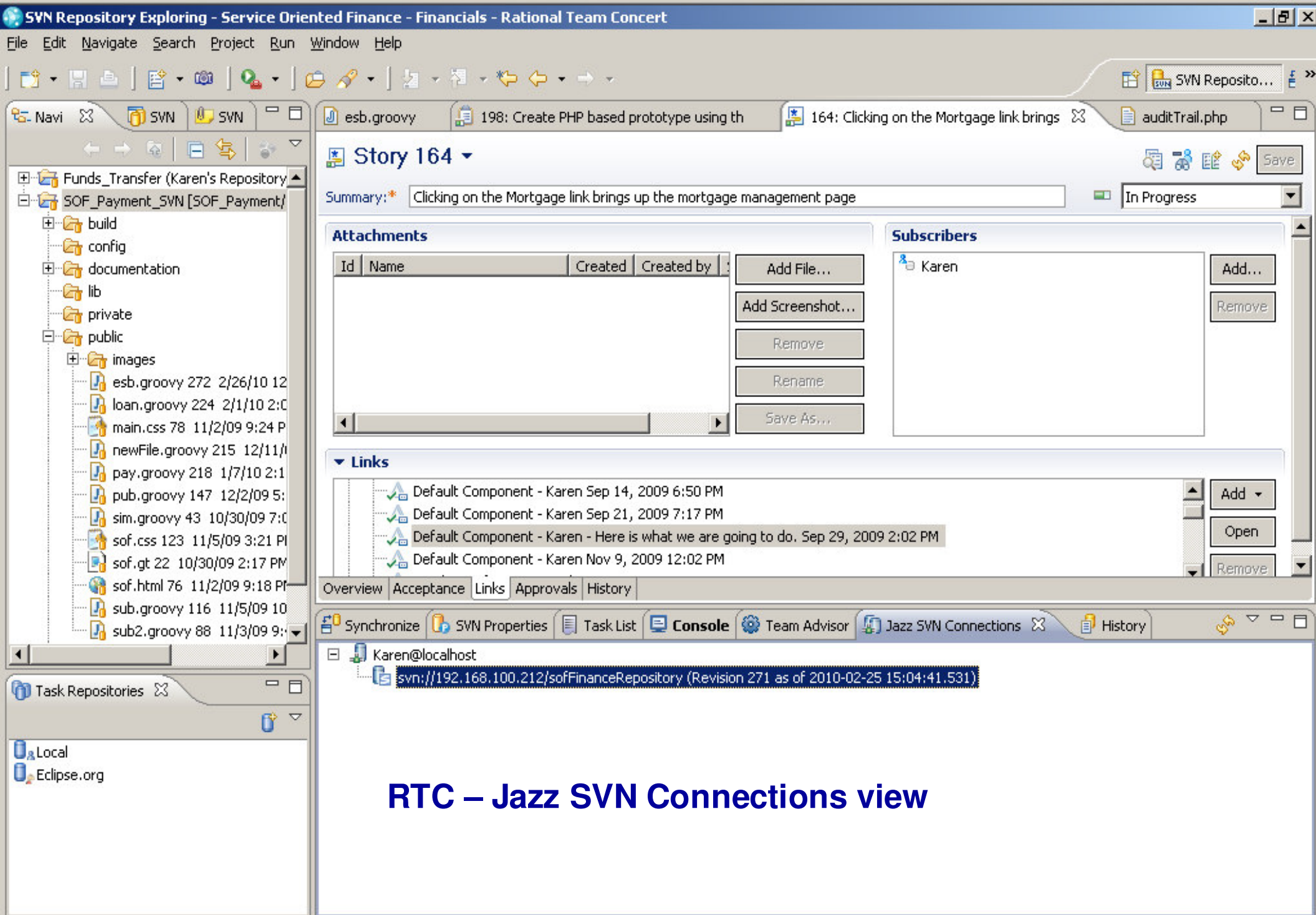


How the Subversion Integration Works

□ RTC with Subversion

- Requires Subclipse or Subversive
- Uses bugtraq properties and alters commit command
- Uses extra text box for work item number
- RTC periodically queries SVN for commits and makes association with work item





RTC – Jazz SVN Connections view

Search for ID or Text

Service Oriented Finan...

svn://192.168.100.212/sofFinanceRepository

164: Clicking on the Mortgage li ...

Svn properties for SOF_Payment_SVN

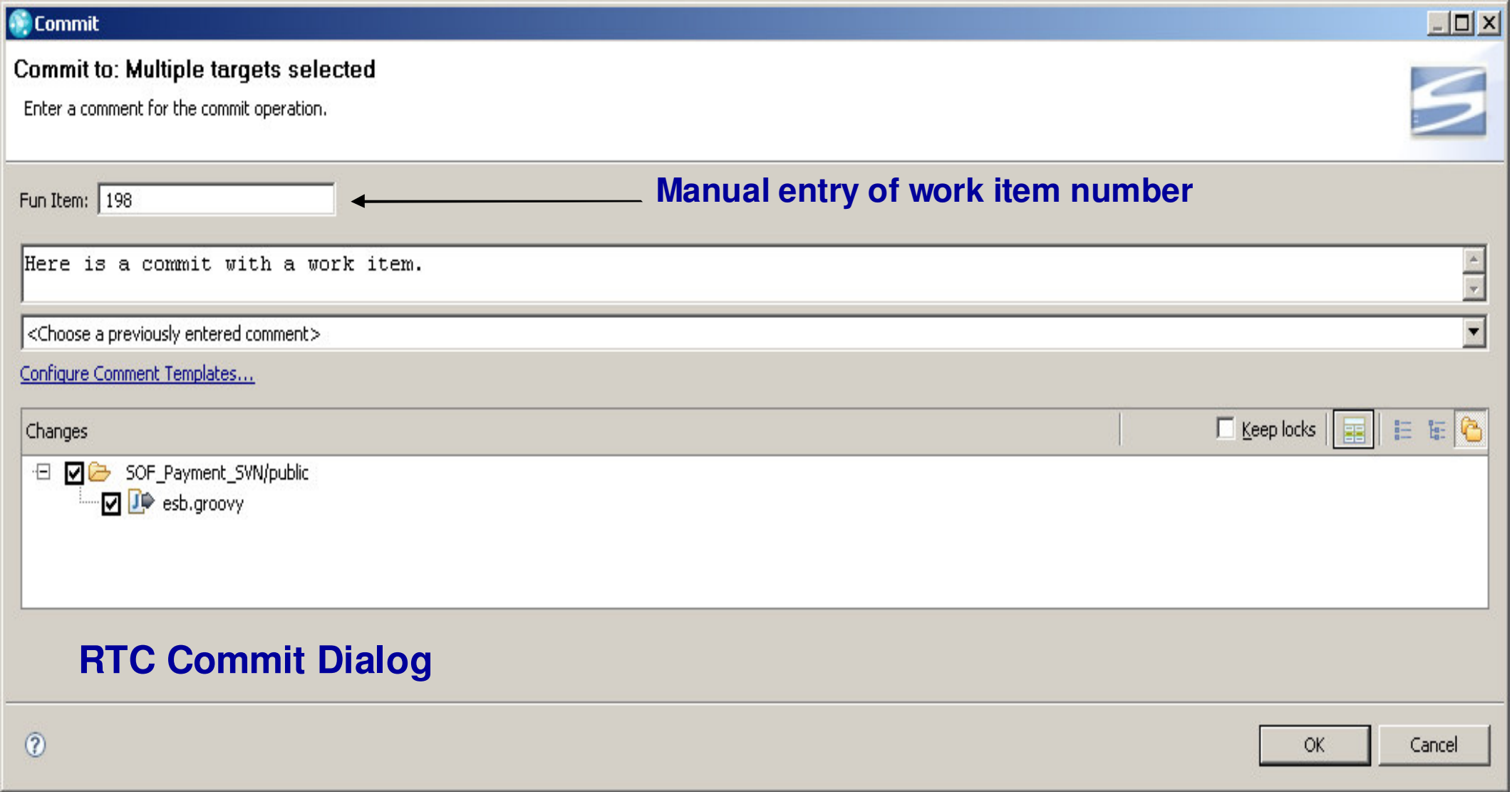
No properties have been modified since last commit

name	value
bugtraq:warnifnoissue	true
subclipse:tags	95,newBranch,/SOF_Payment/branches/SOF_Payment_V2.2/newBranch,branch□243,Rel-V1.4,/SOF...
svn:mergeinfo	/SOF_Payment/branches/SOF_Payment_V2.2/FrontEnd:69-77□/SOF_Payment/branches/SOF_Paym...
bugtraq:url	https://192.168.100.211:9443/jazz/resource/itemName/com.ibm.team.workitem.WorkItem/%BUGID%
bugtraq:label	Fun Item:
bugtraq:append	false
bugtraq:message	Work Item: %BUGID%
bugtraq:number	true

Property example – work item entry must be numeric

false

RTC bugtraq properties



RTC Commit Dialog



Task 198

Summary: Create PHP based prototype using the CommEd RESTful API.

In Progress

Attachments

Id	Name	Created	Created by	Size	Type

Subscribers

- Karen

Links

- Change Sets
 - Fund Transfer - Karen Feb 22, 2010 10:53 AM
- Parent
 - 153: Clicking on Submit Loan Application causes the system to collect user data, format it, and write it to an MQ queue for the loan process to pick up
- SVN Revisions
 - Revision 257: Work Item: 198□Property change for ip address.
 - Revision 258: Work Item: 198□Second update with 198
 - Revision 266: Work Item: 198□Commit with wrong yet existing WI num.
 - Revision 268: Change after property change.□Work Item: 198□
 - Revision 269: Fix to see how often Update Links runs.□Work Item: 198□

RTC – Work item with links to the associated SVN revisions

How the Git Integration Works

• RTC with Git

- Based on scripting in git hook scripts to link to RTC work items – uses the REST API
- REST API follows a Resource Oriented Architecture – CRUD = POST / GET / PUT / DELETE
- cURL / WWW:CURL:EASY - used to facilitate requests with the RTC server
- The integration is fairly configurable – allowing various data to be displayed on work items such as files committed, link to gitweb..
- Primary scripts used to integrate into git are:
 - **commit-msg** & **post-commit** – private repo commits
 - **update** & **post-receive** – public repo / multiple commits



Agenda

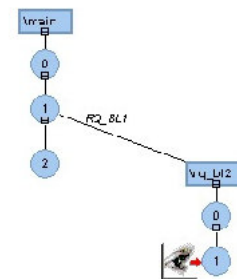
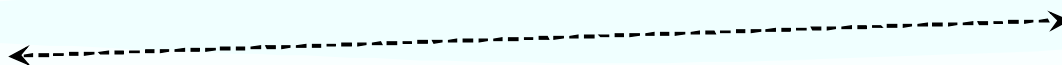
- **RTC SCM integration overview**
- SCM integrations with RTC
 - ▶ Types of integrations and deployment considerations
 - Subversion, Git
 - ClearCase



Who should use the ClearCase Synchronizer?

Scenario:

- ▶ ClearCase is the enterprise software configuration management system
- ▶ Product integration and release are performed in ClearCase repositories
- ▶ Small/medium teams responsible for specific components (“agile islands”) may opt to use Rational Team Concert to do work destined for a ClearCase branch or UCM stream



Synchronizer benefits for this scenario:

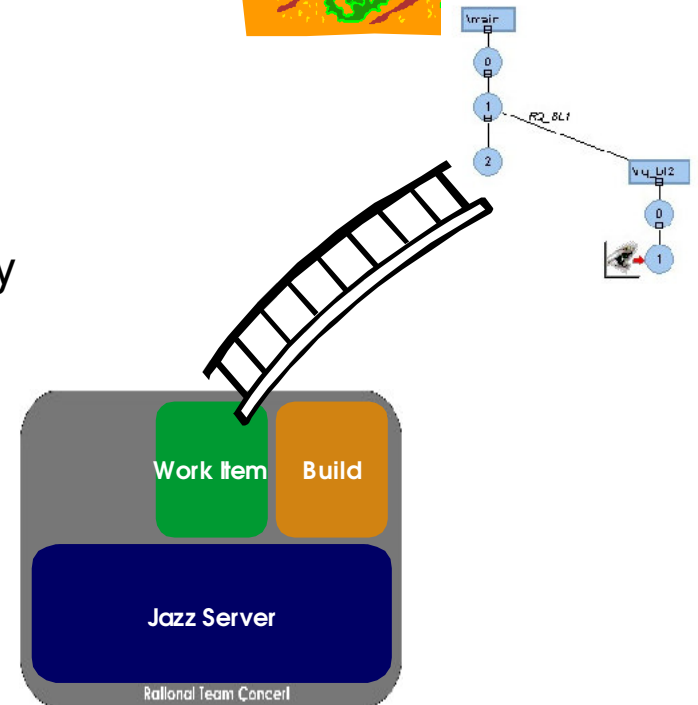
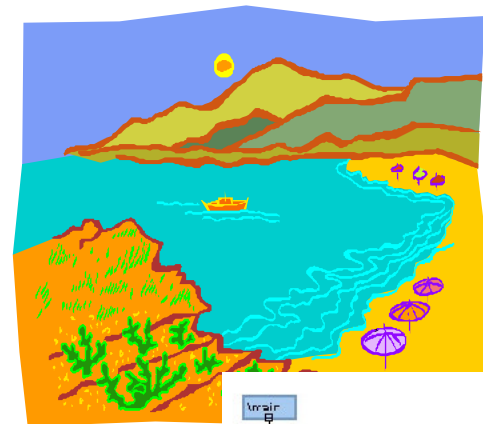
- ▶ Allows ClearCase customers to try Rational Team Concert with its full benefits without isolating trial users from the rest of the organization
- ▶ Allows customers to use either tool/repository
- ▶ Work can occur in parallel in both repositories
- ▶ Customers use standard Jazz/SCM merge tooling to resolve conflicts
- ▶ Supports both base ClearCase branches and UCM streams



Who should use the ClearCase Bridge?

- Scenario:
 - ▶ ClearCase is the enterprise software configuration management system
 - ▶ ClearCase customers don't necessarily want to (or can't) switch SCM systems to get the benefits of the other Jazz features

- Bridge benefits for this scenario:
 - ▶ Allows ClearCase/ClearQuest customers to benefit from many features in Rational Team Concert while continuing to use ClearCase for configuration management
 - ▶ Links ClearCase information with Jazz work items to provide traceability in both directions
 - ▶ No data to synchronize
 - ▶ Loosely coupled
 - ▶ Less restrictive than ClearQuest-enabled UCM
 - ▶ Can be used with ClearQuest Bridge



Who should use the ClearCase Importer?

- Scenario:
 - ▶ Customer wants to move from using ClearCase as the enterprise software configuration management system to Jazz source control

- Importer benefits for this scenario:
 - ▶ Allows ClearCase customers (who want to) to move to Rational Team Concert with its full benefits
 - ▶ Based on synchronizer technology; imports baseline-to-baseline differences
 - ▶ Allows selection of historical data to be brought over
 - ▶ Supports import of both ClearCase UCM streams and base ClearCase branches

Limitation: CCLT is not natively supported; customers wanting to move from CCLT should work with their sales rep to get a temporary full ClearCase license for the purpose of importing



Availability of ClearCase Connectors to Rational Team Concert

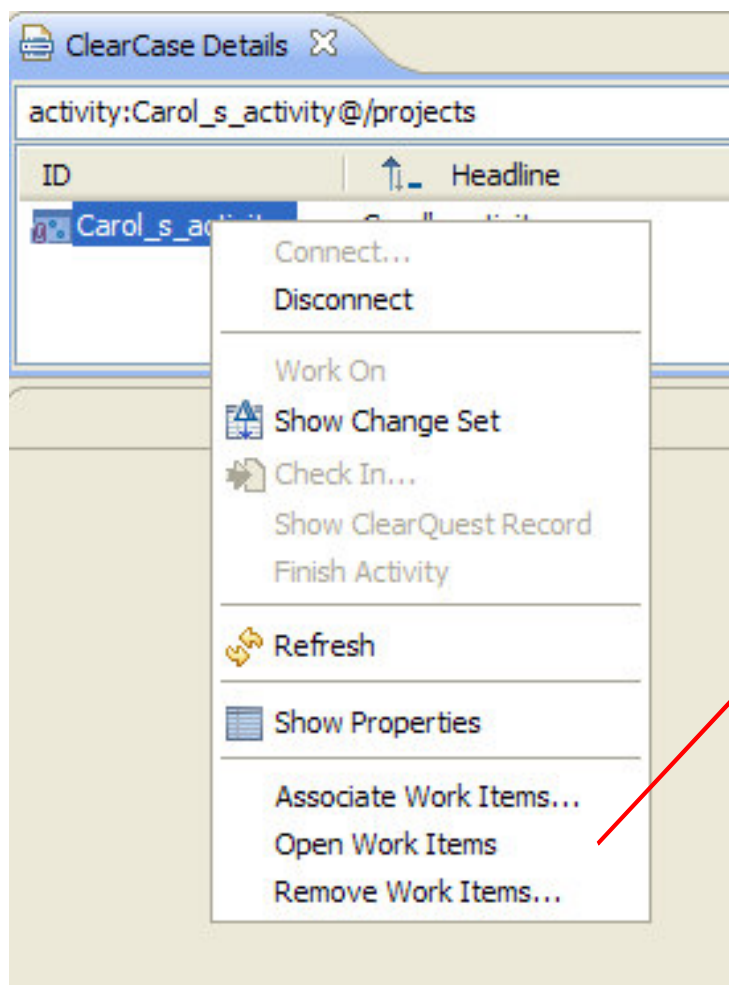
- **Bridge**
 - ▶ Introduced in Rational Team Concert 2.0
 - ▶ Earliest ClearCase release supported: ClearCase 7.1.0.2 (works with 7.1.0.x, 7.1.x)
 - ▶ Base ClearCase support added in ClearCase v7.1.2 and RTC v3.0
 - ▶ “Tolerates” earlier versions

- **Synchronizer**
 - ▶ Introduced in Rational Team Concert 1.0
 - ▶ Works with ClearCase 7.0, 7.0.1, 7.1.x

- **Importer**
 - ▶ Introduced in Rational Team Concert 2.0
 - ▶ Works with ClearCase 7.0, 7.0.1, 7.1.x



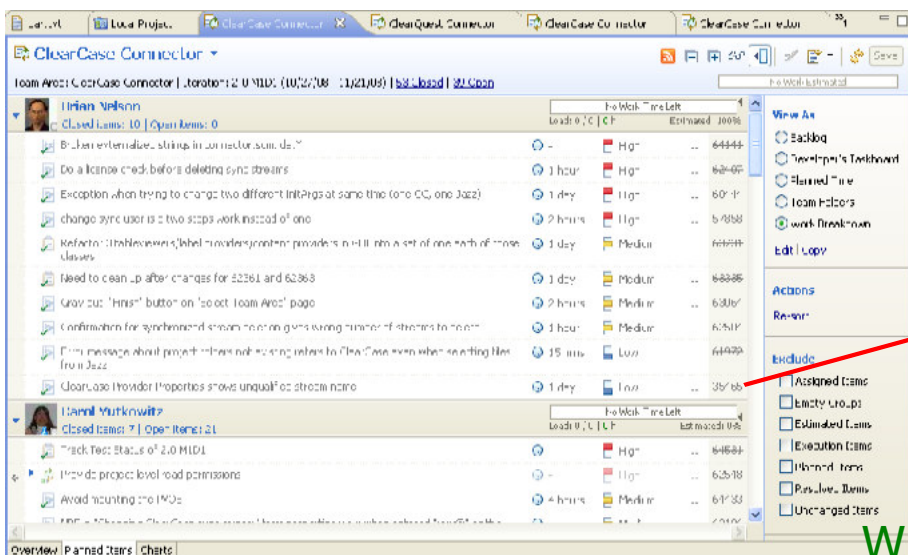
ClearCase Bridge user interface



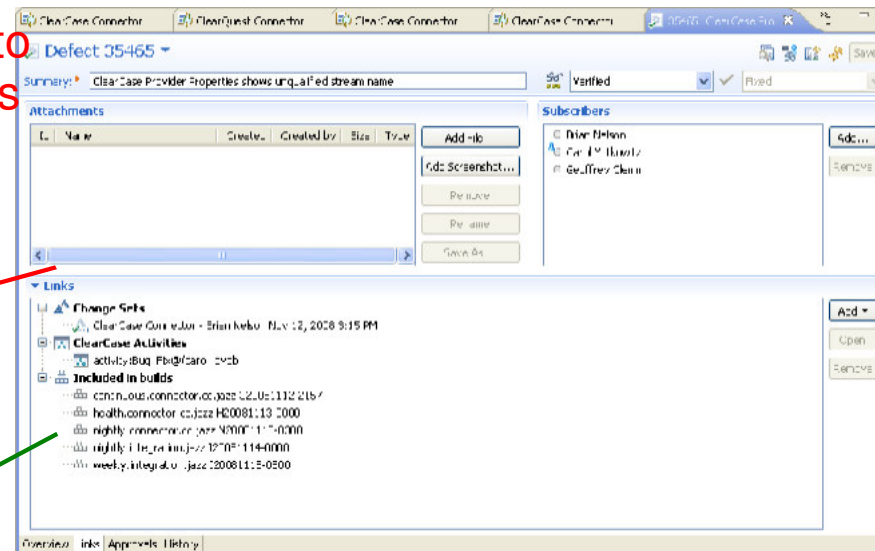
- Context menus on objects in ClearCase Unified Client are enhanced to allow ClearCase users to create and manage associations between ClearCase UCM activities (or base ClearCase versions) and Jazz work items.
- Once a link has been created, it may be used in queries, dashboards, etc., just like any other work item link type.



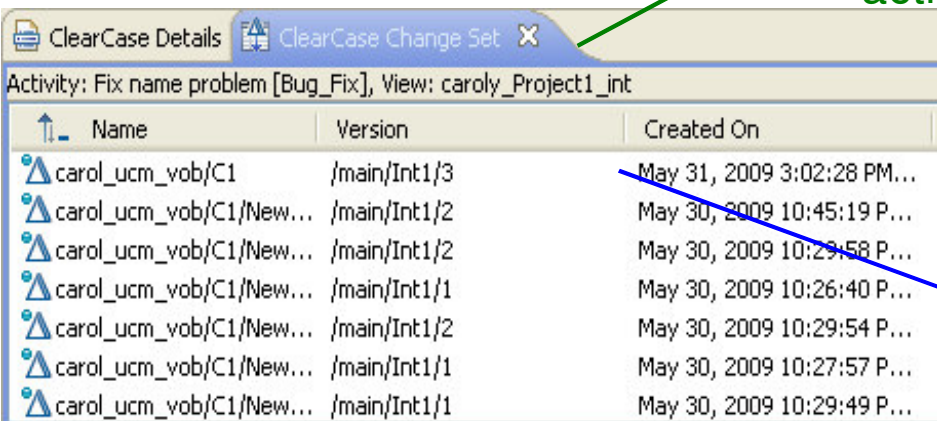
Full traceability from Plans to code with the ClearCase Bridge



Plan links to Work Items



Work Items link to ClearCase activities



ClearCase activities link to change-sets

- References to ClearCase objects show up on the Links tab of their associated work items
- Opening a ClearCase link from a work item editor shows the ClearCase Unified Client activity change set
- Once a link has been created, it may be used in queries, dashboards, etc., just like any other work item link type.



Guidance - First, make sure the synchronizer is right for you

- Only use the synchronizer if some teams will be using the Source Control built in to Rational Team Concert
- Users who work exclusively in Rational Team Concert do not need to synchronize
- Don't choose the synchronizer just because you are using dynamic views
 - ▶ ClearCase and Rational Team Concert can coexist "side by side" (some teams at IBM use the products this way)
 - ▶ Many benefits in Rational Team Concert don't require linkage to ClearCase
 - ▶ Can consider incremental adoption of the bridge
- Summary: don't pay the overhead of synchronizing data if you don't need the data in two places



If deploying the synchronizer makes sense...

- Select the teams that will use Rational Team Concert
 - ▶ Synchronize the result of a component team's work rather than splitting up the team
- Set up the synchronization host/engine (ideally, on the VOB server)
- Test your configuration before adding all of the synchronization roots
 - ▶ Start out with just one or a few files
 - ▶ Then incrementally add roots

- Summary: plan carefully how you will divide the effort between Rational Team Concert and ClearCase and test your setup before diving in



Avoid large and expensive synchronizations that you don't need

- Don't choose as synchronization roots whole VOBs, whole components
 - ▶ Take advantage of the synchronizer's ability to do "partial VOB replication"
 - ▶ Don't synchronize more data than you need to
- Don't duplicate every branch and stream
 - ▶ E.g. if you're synchronizing so you can do release engineering from ClearCase, only synchronize the "release" branch / stream
- Consider whether files will be changing a lot on both sides
 - ▶ Speed of initial import is proportional to the number of files
 - ▶ Speed of synchronization after the initial import is proportional to the number of files with changes to synchronize
- Summary: synchronization of a large amount of data necessarily takes time. Plan to synchronize only what you need to have in two places



Best practices for adopting RTC in existing SCM environments

- An **incremental approach** is best
 - ▶ A **team-by-team approach** is most successful
 - Because teams have different needs, and different constraints on when they can adopt new tooling
 - ▶ Introduce **incremental value first**
 - For example, introduce the work item management and agile planning capabilities (before the RTC SCM capabilities)
- Simplify existing **stream structures**
 - When introducing RTC SCM capabilities, this is a good time to simplify the existing stream structure, since it is easy to incrementally extend RTC stream structures.
- **Lightweight task management system**
 - Introduce RTC work item capabilities using the pre-defined process templates.
- Try to use **existing process mechanisms**
 - When customizing RTC process, experiment with the existing declarative process mechanisms before writing custom process.



QUESTIONS

www.ibm.com/software/rational

