An IBM Proof of Technology

Collaborative software development using IBM Rational Team Concert

Presentations

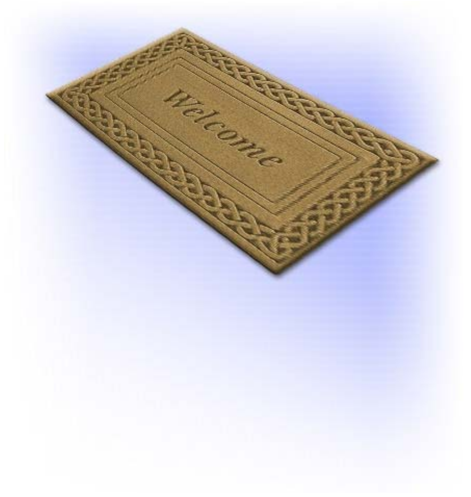
PoT.Rational.07.2.038.02

Collaborative Software Development Using IBM Rational Team Concert

An IBM Proof of Technology

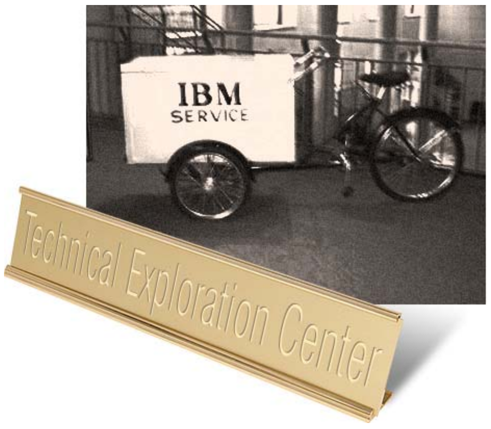
Welcome to the Technical Exploration Center

- Introductions
- Access restrictions
- Restrooms
- Emergency Exits
- Smoking Policy
- Breakfast/Lunch/Snacks – location and times
- Special meal requirements?



Introductions

- Please introduce yourself
- Name and organization
- Current integration technologies/tools in use



What do you want out of this Exploration session?

Agenda

Introduction to Rational® Team Concert™	
Lab Overview	
Module 1	Setting up the Team
Module 2	Planning Your Work
Module 3	Keeping Track of All Our Work
Module 4	Performing and Sharing Your Work
Module 5	Remembering Well Known SCM Configurations
Module 6	User's View of Build
Module 7	Exploring Changes and Traceability
Module 8	Endgame and a Tightened Process
Session Summary	

Optional Modules

Module 9	Taking Control of Your Project
Module 10	Integrating with Other SCM Systems
Module 11	Project Growth and Multi-Stream Development

Objectives

- Explore how Rational Team Concert can
  - Enable development teams to **collaborate in real time in the context** of the work they are doing, especially in globally diverse environments
  - Enable projects to be managed more effectively by providing visibility into **accurate project health information** drawn directly from actual work
  - Automate traceability and auditability by **managing artifacts and their inter-relationships** across the lifecycle empowering teams to deliver more value
  - Provide **customizable process design and enactment** through rule-based process guidance, automation and definable checkpoints
- Provide a hands on experience using Rational Team Concert to automate the software delivery process

Introduction to Rational Team Concert

An IBM Proof of Technology

What if your development tools knew…

… about your teams

… about your artifacts

… who is responsible for what

… about your process

- Code delivery rules, code quality, traceability, test runs, intellectual property

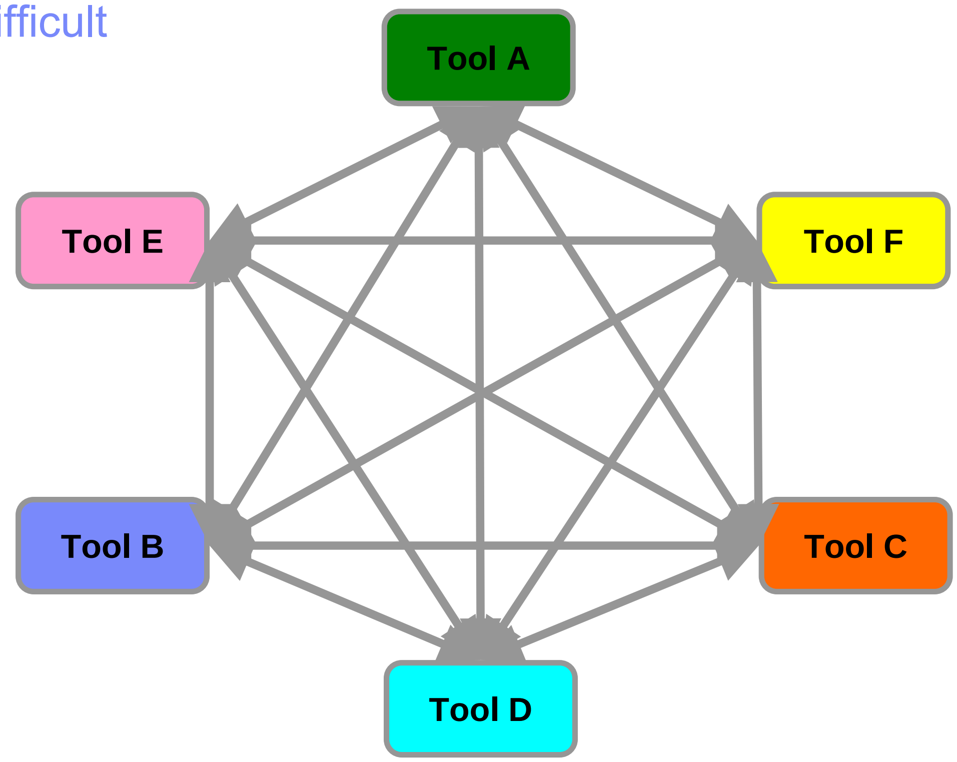
… how to bootstrap a project

… how to help new team members get started

… your favorite work item types and their state transitions

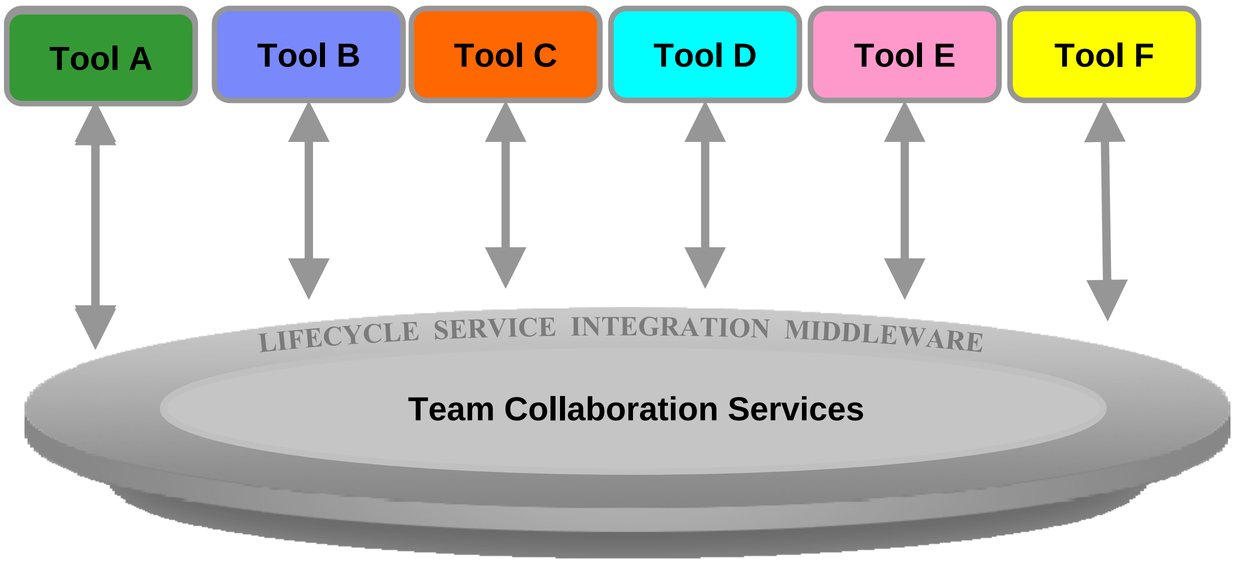
… when the build runs and what to do if it breaks

Enabling Collaborative Application Lifecycle Management is generally difficult



Team collaboration based on middleware services

Built on an extensible platform and common repository


Tool A
Tool B
Tool C
Tool D
Tool E
Tool F
LIFECYCLE SERVICE INTEGRATION MIDDLEWARE
Team Collaboration Services

IBM Rational Team Concert

Software innovation through collaboration

- **Real time, in-context team collaboration**
  - Make software development more automated, transparent and predictive
- **"Think and work in unison"**
  - Integrated planning, source control, work item, build management and project visibility
- **Assess real-time project health**
  - Capture data automatically and unobtrusively
- **Automate best practices**
  - Dynamic processes accelerate team workflow
  - Out-of-the-box choice of agile processes or customize
- **Unify software teams**
  - Integrate a broad array of tools and clients
  - Extend the value of IBM ClearQuest® and IBM ClearCase®
  - Support for IBM System z® and System i® servers (4Q08)
  - Visual Studio Client (1Q09)
  - Integrate document collaboration (1Q09)

IBM Rational Team Concert
transparent integrated presence
wikis OPEN real-time reporting
chat automated hand-offs Web 2.0
custom dashboards automated data gathering
EXTENSIBILITY Eclipse plug-ins services
architecture FREEDOM TO CREATE
JAZZ TEAM SERVER

Open and extensible on jazz

- ✓ Collaborate in context
- ✓ Right-size governance
- ✓ Day one productivity

Pain points before Rational Team Concert

- joining a team
- get my environment configured to be productive
- what is happening in my team
- collecting progress status
- following the team's process
- ad hoc collaboration/sharing of changes
- starting an ad hoc team

Team awareness

- is the fix in the build?
- what will be in the next build?
- tracking a broken build
- Avoid breaking a build/personal build
- why is this change in the build?
- reconstructing a context for a bug/build failure

Build awareness

- creating, tracking iteration plans
- interrupting development due to a high priority bug fix
- working on multiple releases concurrently
- tracking the code review of a fix
- referencing team artifacts in discussions
- how healthy is a component?
- collecting project data/metrics?

Project awareness

Boring and painful

Rational Team Concert: A closer look

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
- Component level baselines
- Server-based sandboxes
- Parallel development
- ClearCase connector

Work Items
- Defects, enhancements and conversations
- View and share query results
- Support for approvals and discussions
- Query editor interface
- ClearQuest connector

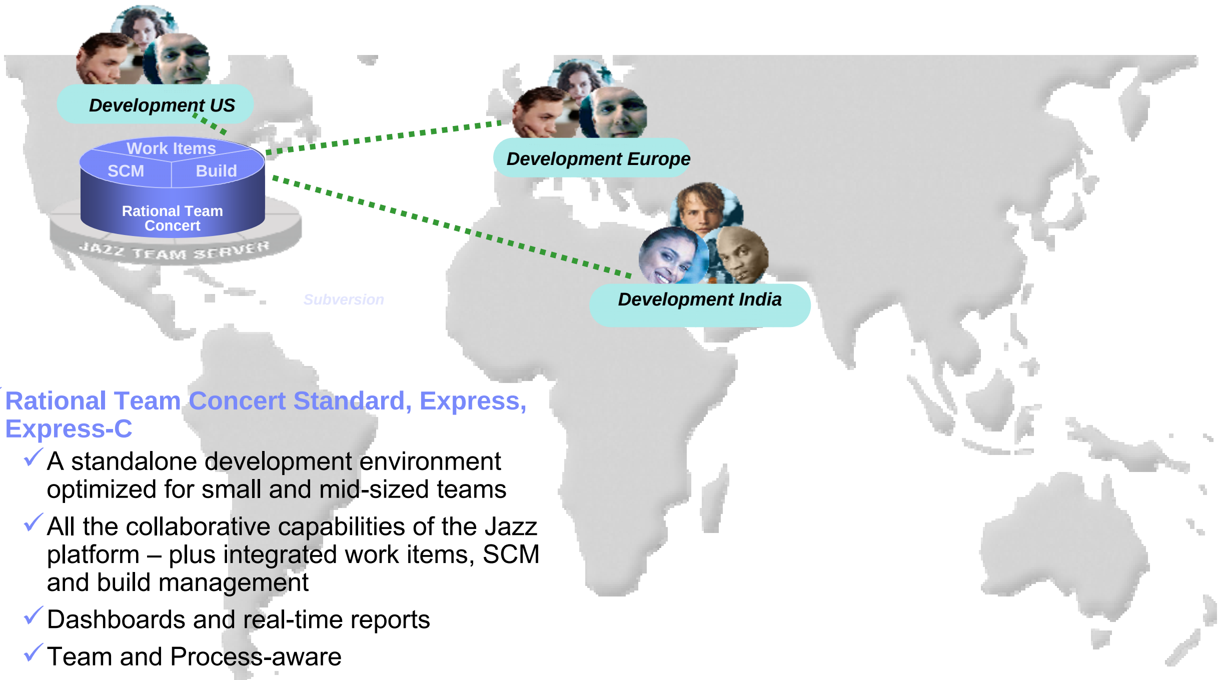
Build
- Work item and change set traceability
- Build definitions for team and private builds
- Local or remote build servers
- Supports Ant and command line tools
- Integration with Build Forge®

IBM Jazz™ Team Server
- Single structure for project related artifacts
- World-class team on-boarding / off-boarding 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 enables team members to communicate in context of their work

Rational Team Concert Key Advantages

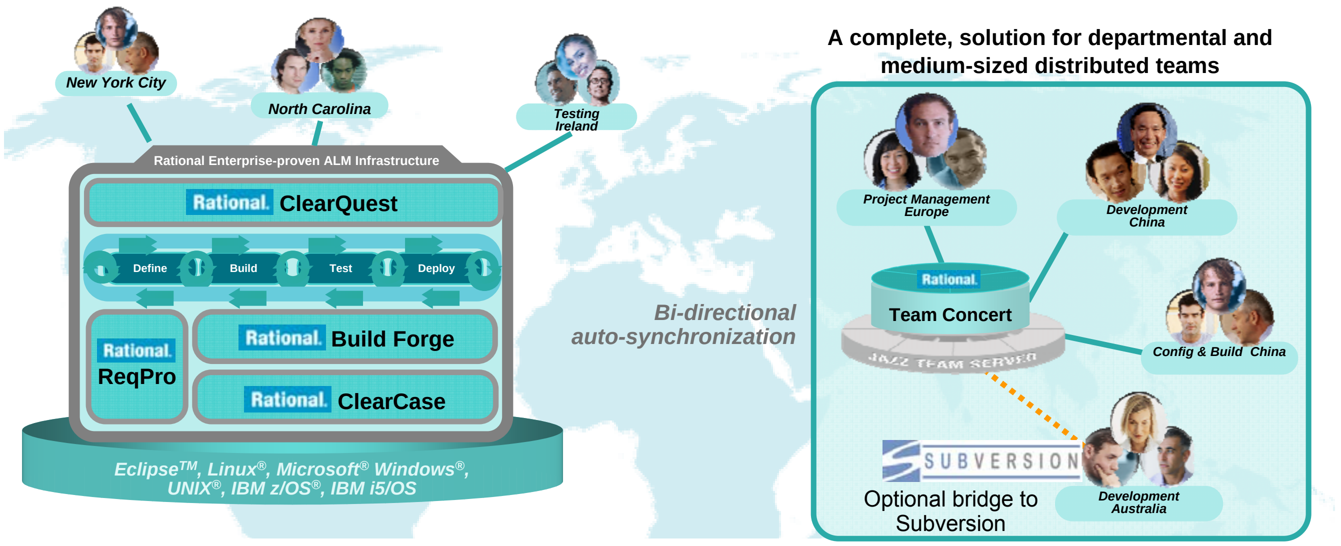
- In-context Collaboration services
- Iteration planning and end to end business visibility
- End to End Support for Integrated Agile Processes
- Powerful source control management
- An Open Platform
- Incremental Adoption
- Open, Community based development model

Leveraging Rational Team Concert independently



✓ **Rational Team Concert Standard, Express, Express-C**
- ✓ A standalone development environment optimized for small and mid-sized teams
- ✓ All the collaborative capabilities of the Jazz platform – plus integrated work items, SCM and build management
- ✓ Dashboards and real-time reports
- ✓ Team and Process-aware

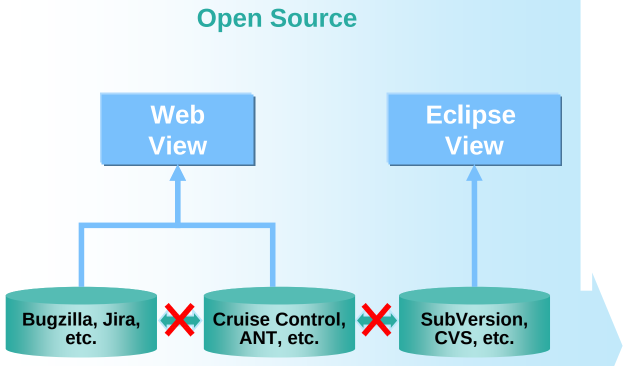
Unifying standalone, departmental and enterprise teams


New York City
North Carolina
Testing Ireland
Rational Enterprise-proven ALM Infrastructure
Rational ClearQuest
Define
Build
Test
Deploy
Rational ReqPro
Rational Build Forge
Rational ClearCase
Eclipse™, Linux®, Microsoft® Windows®, UNIX®, IBM z/OS®, IBM i5/OS
Bi-directional auto-synchronization
A complete, solution for departmental and medium-sized distributed teams
Project Management Europe
Development China
Rational Team Concert
JAZZ TEAM SERVER
Config & Build China
SUBVERSION
Optional bridge to Subversion
Development Australia

- Enables teams to reuse enterprise assets, process and investment in ClearCase/ClearQuest
- Take advantage of new collaborative ALM in an evolutionary way with lower business risk
- Manage status/priorities in ClearQuest and develop with the right ALM solution for your project.
- Use the “right-sized” governance solution for different projects needs.
- Developers can deliver work from “satellite” teams directly into enterprise ClearCase projects

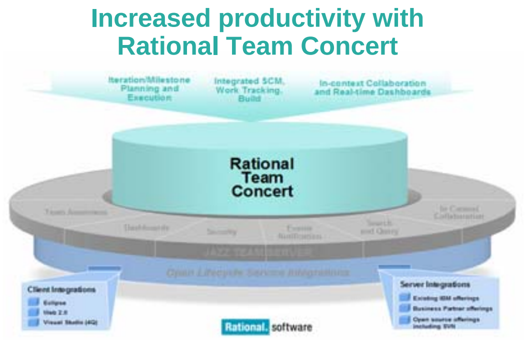
Open Source vs Rational Team Concert

Cut administration costs and increase productivity with Rational Team Concert



Point Solutions for individual productivity

- Significant costs to maintain and administer federated databases
- No concept of project, teams or schedules
- No single, consolidated view into project status and health
- SCM, work items, build functionality unaware of each other providing little to no insight into relationships between artifacts


Increased productivity with Rational Team Concert
Iteration/Milestone Planning and Execution
Integrated SCM, Work Tracking, Build
In-context Collaboration and Real-time Dashboards
Rational Team Concert
JAZZ TEAM SERVER
Open Lifecycle Service Integrations
Client Integrations
Eclipse
Web 2.0
Visual Studio (4Q)
Server Integrations
Existing IBM offerings
Business Partner offerings
Open source offerings including SVN
Rational software

Fully integrated collaborative application lifecycle management solution

- ✓ Designed for agile and distributed teams
- ✓ Fully aware of projects, team process, team composition, who’s doing what and by when, in-context collaboration with instant messaging discussions, discussion threads kept with work items
- ✓ All-in-one solution for the core development team with planning, SCM, work items, workflow, build management, dashboards and reporting fully integrated
- ✓ Single download package, easy to install and get started, minimal administration, easy startup of projects, onboarding and offboarding of team members quickly

Envisioning a platform that can transform software delivery

Jazz is a project and platform for *transforming how people work together* to deliver greater value and performance from their software investments.

- robust, extensible and scaleable
- globally distributed, fluid & dynamic
- community-based & open at Jazz.net

Collaborate in Context

- Enable team transparency of “who, what, when, why”
- Build team cohesion and presence
- Automate hand-offs – so nothing falls through the cracks

Right-size Governance

- Automate team workflow improving productivity
- Automate data collection eliminating administrative overhead
- Real time reporting and alerts reduces project risk

Day One Productivity

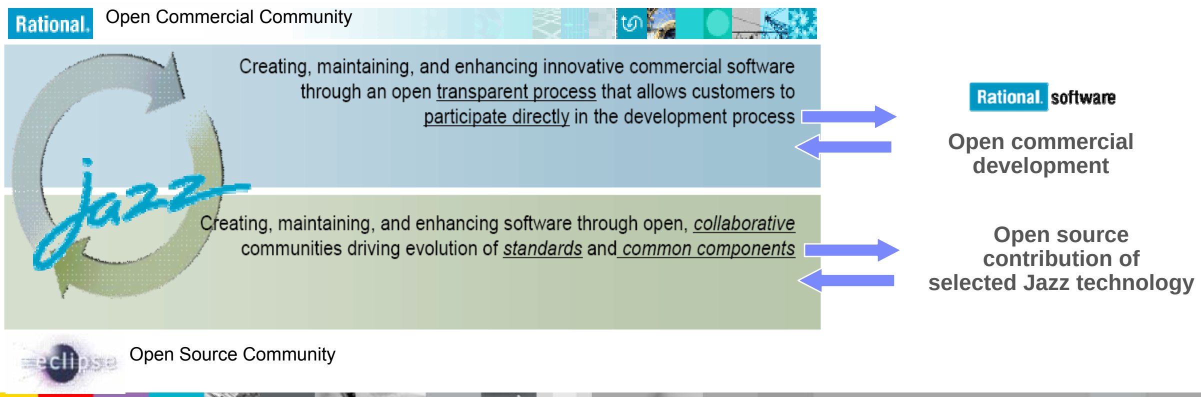
- Dynamic provisioning of projects and teams
- Real-time iteration planning and workload balancing
- Unify teams with tools choice

Dynamic integration of people, process and projects across the lifecycle

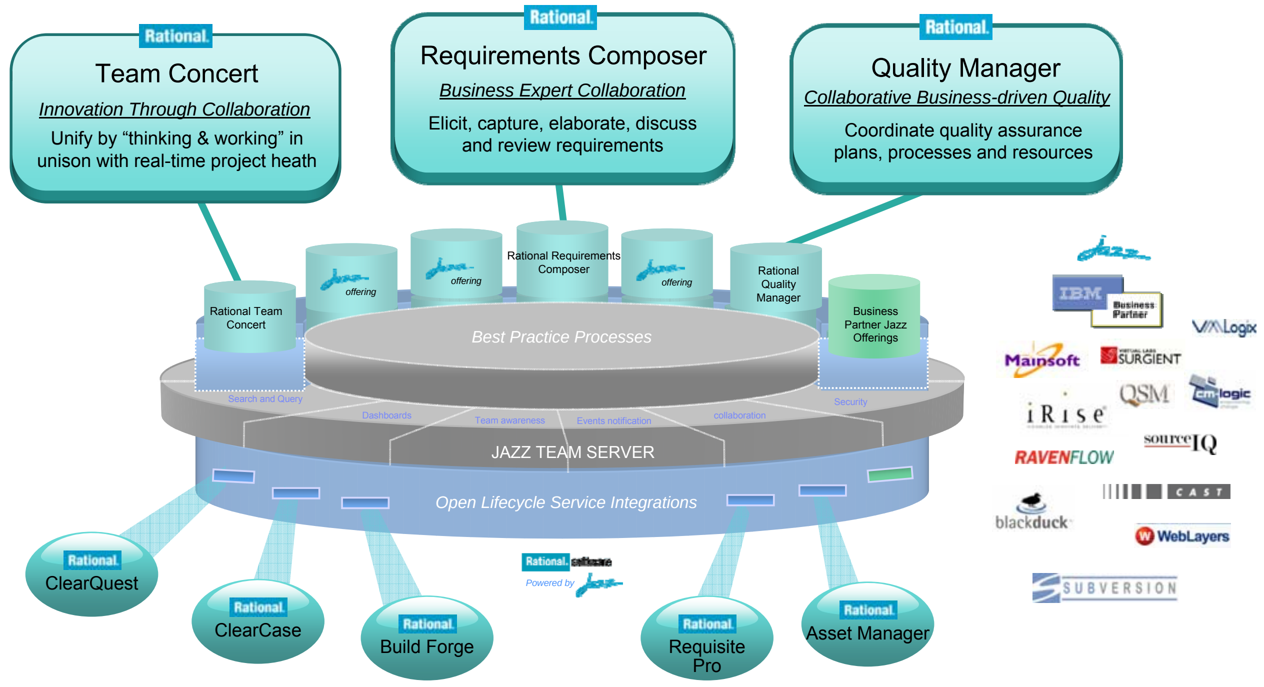
Open Commercial Development at jazz.net

Delivering greater openness and customer participation in the products they depend on for software delivery

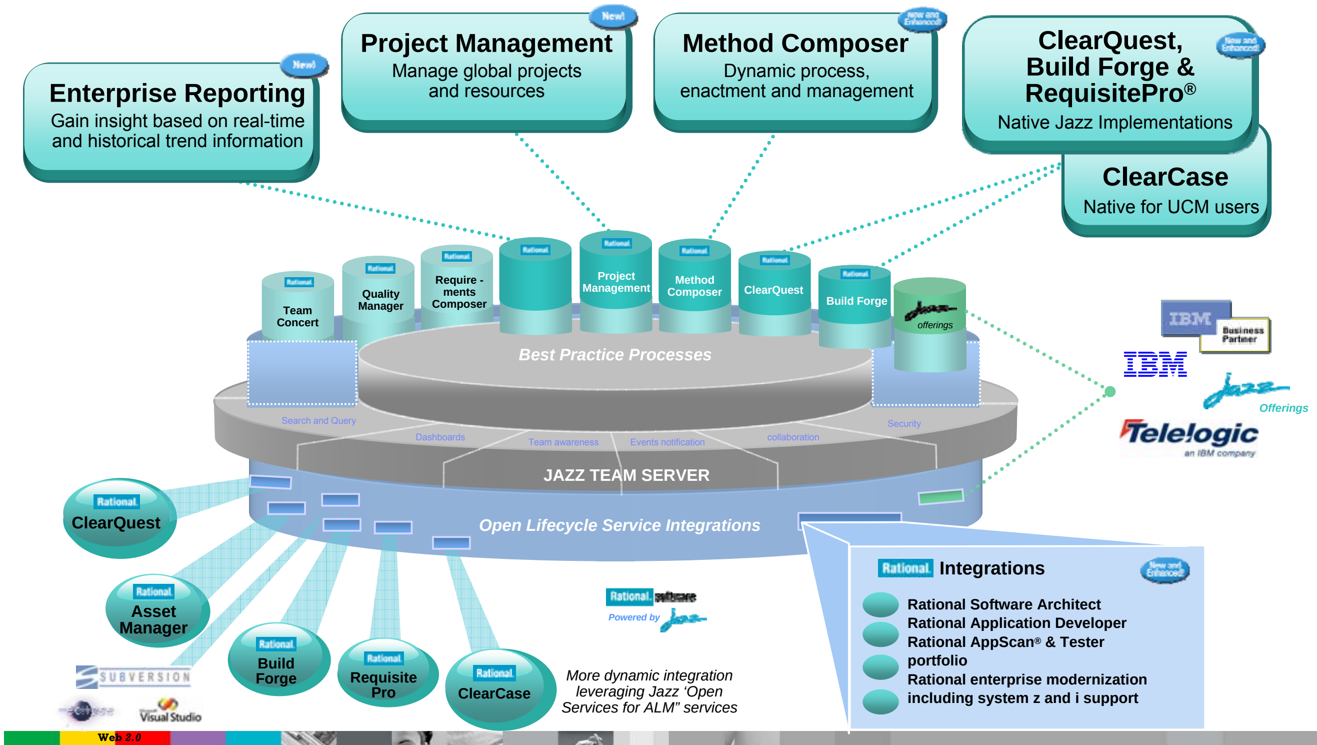
- IBM is opening up the Rational Software Delivery Platform for greater ease of consumption, extensibility and integration to meet the unique usage needs of our customers
- IBM is providing transparent, collaborative customer participation in the development of new Rational technologies through an open commercial community


Rational. Open Commercial Community
Creating, maintaining, and enhancing innovative commercial software through an open transparent process that allows customers to participate directly in the development process
jazz
Creating, maintaining, and enhancing software through open, collaborative communities driving evolution of standards and common components
eclipse Open Source Community
Rational. software
Open commercial development
Open source contribution of selected Jazz technology

Introducing the first wave of new Jazz offerings

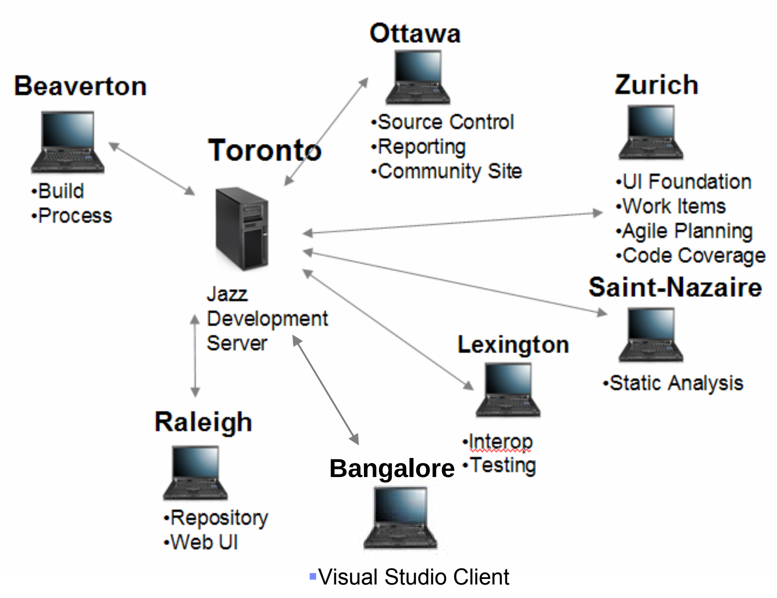


The road ahead: What to expect from Rational **Jazz** in 2009



How we use Rational Team Concert

- 2-way Xeon Server running application server (WAS) and another running IBM DB2®
- Jazz Project – Using bi-weekly iteration builds
  - Jazz and Rational Team Concert self hosting since 4Q06
  - Global team in 7 locations in NA Europe and India
  - ~100 developers, plus jazz.net webclient access
  - Repro > 10G, 66K files, 43K work items
- Established "Rhythm"
- Over 10 internal Rational development teams leveraging Rational Team Concert
- 25+ Other Rational teams using Rational Team Concert



Rational Team Concert customer feedback

"By helping us to make project deliveries more repeatable and predictable, we anticipate that Rational Team Concert will **reduce project overrun costs by 20%.**"

--Matt Pomroy - Executive, Software Engineering, Ascendant Technology

"Its automated project management dashboards are transparent to everyone – not just managers. This immediate and **automated feedback helps keeps teams on track and motivated** to achieve project goals."

--Han Jie - Senior Consultant, Siemens

" Where we previously used separate systems, with Rational Team Concert we now have well integrated functionality. **Our developers are more efficient because they are better able to focus on important issues**. Our project managers greatly value the ability to customize these dashboards and **instantly provide status** on their milestones!"

--Mika Koivuluoma - Production Manager, TietoEnator

"Having a **unified and extensible environment is very compelling for us.** Rational Team Concert provides **the team transparency and visibility needed to keep work progressing** so everyone knows what's going on without finger-pointing."

--Carson Holmes - Unified ALM Services Manager, Noblestar

Questions

IBM
Software
Group

Rational Team Concert Client for Microsoft Visual Studio IDE

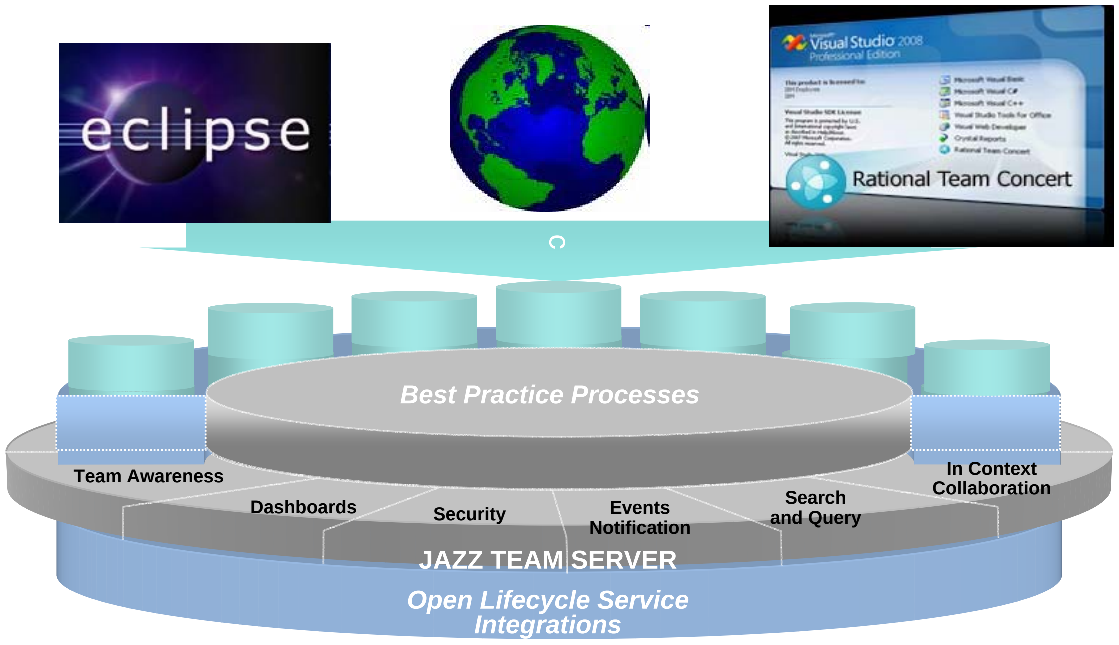
Collaboration, automation and reporting for heterogeneous development teams

An IBM Proof of Technology

Agenda

- Rational Team Concert client for Microsoft® Visual Studio IDE
  - Features and Benefits
- Summary: Supports all your Windows® development needs
  - Windows Platforms
  - SQL Server Database
  - Visual Studio 2005 and 2008 IDE

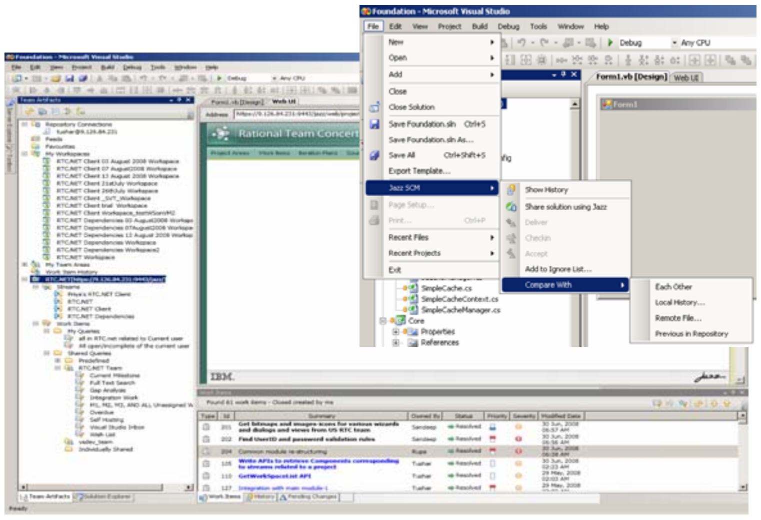
Unify ALL your teams using Eclipse, the web or Visual Studio, with a single collaborative development platform.



Extend team collaboration to Visual Studio developers

Rational Team Concert client for Microsoft Visual Studio IDE

- **Unify Software teams**
  - ✓ Manage Change across development environments
  - ✓ Single repository for both development platforms (.NET and J2EE)
  - ✓ Cross platform team collaboration
    - ✓ Common Work items
    - ✓ Source Code Management



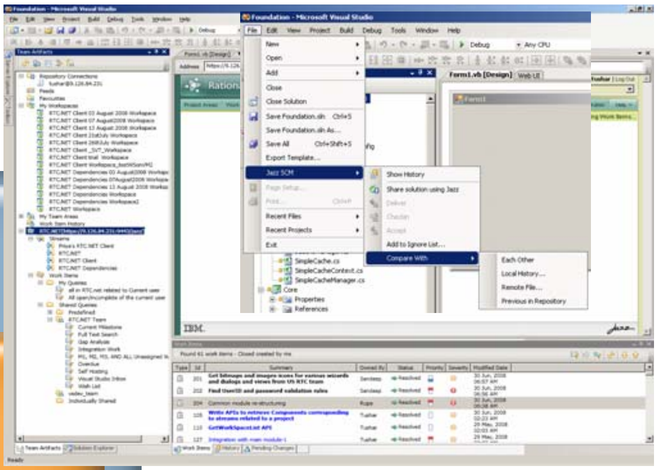
Supports Both Visual Studio 2005 and 2008

Professional and Standard Editions

Microsoft® Visual Studio® 2008

Microsoft® Visual Studio® 2005 Professional Edition

Microsoft® Visual Studio® 2005 Standard Edition



Open and extensible on Jazz

- Collaborate in context
- Right-size governance
- Day one productivity

SCM, Work items directly from VS.NET IDE – Build CLI

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
- Component level baselines
- Server-based sandboxes
- Identifies component in streams and available baselines
- ClearCase connector

Work Items

- Defects, enhancements and conversations
- View and share query results
- Support for approvals and discussions
- Query editor interface
- ClearQuest connector

Build*

*CLI access

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

Jazz Team Server

Microsoft® SQL Server · IBM DB2 · ORACLE®

- 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 enables team members to communicate in context of their work

Developer focused, scm and work item features have direct access from the Microsoft Visual Studio IDE

Inside Visual Studio .NET Shell

Developer

Join Development

- Connect to repository
- Or
- Accept invite to join project

Develop in Visual Studio

- Select Stream
- Create Workspaces
- Load/Unload workspace
- Deliver changes
- Run Queries
- Associate Work items
- Create Work items
- Change Status of Work Items
- View Pending Changes
- Change Delivery Flow
- Merge Changes
- View History

WebUI

- Create Project Area*
- Create Team Area*
- View Dashboards

Developer

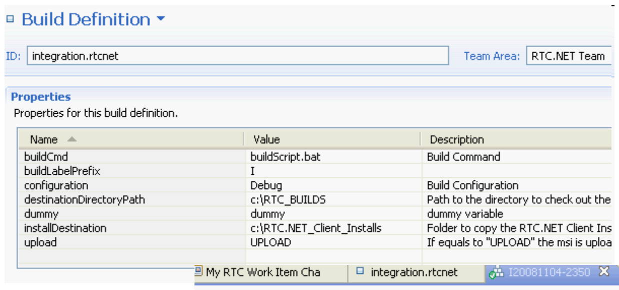
Eclipse UI

- Create Project Area*
- Create Team Area*
- Customize Process*
- Customize Work items*
- Author iteration plans
- Track MS-Build Progress/Result

* Permission provided

Do Visual Studio builds integrate with Team Concert?

- **Yes. They do.**
  - ✓ Just use MS-Build command line builds with Rational Team Concert continuous Integration build engine
  - ✓ Link command line MS-Builds to the Rational Team Concert build toolkit and store results in the repository
  - ✓ Use the Eclipse client to control and view the build results.



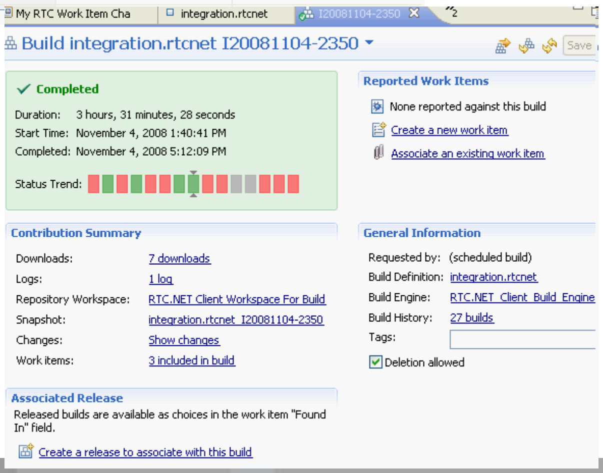
Build Definition

ID: integration.rtcnet Team Area: RTC.NET Team

Properties

Properties for this build definition.

Name	Value	Description
buildCmd	buildScript.bat	Build Command
buildLabelPrefix	I	
configuration	Debug	Build Configuration
destinationDirectoryPath	c:\RTC_BUILDS	Path to the directory to check out the
dummy	dummy	dummy variable
installDestination	c:\RTC.NET_Client_Installs	Folder to copy the RTC.NET Client Ins
upload	UPLOAD	If equals to "UPLOAD" the msi is uploa



Build integration.rtcnet I20081104-2350

✓ Completed

Duration: 3 hours, 31 minutes, 28 seconds
Start Time: November 4, 2008 1:40:41 PM
Completed: November 4, 2008 5:12:09 PM

Status Trend: 🟥🟩🟥🟩🟥🟥🟩🟩🟥🟥⬜⬜🟥🟥🟥

Reported Work Items

None reported against this build
Create a new work item
Associate an existing work item

Contribution Summary

Downloads:	7 downloads
Logs:	1 log
Repository Workspace:	RTC.NET Client Workspace For Build
Snapshot:	integration.rtcnet_I20081104-2350
Changes:	Show changes
Work items:	3 included in build

General Information

Requested by:	(scheduled build)
Build Definition:	integration.rtcnet
Build Engine:	RTC.NET_Client_Build_Engine
Build History:	27 builds
Tags:	

✓ Deletion allowed

Associated Release

Released builds are available as choices in the work item "Found In" field.

Create a release to associate with this build

Details of Team Concert views surfaced in Visual Studio

View (Tool Window)	Feature	Comments
Solution Explorer	Share new or existing Visual Studio solutions with the Jazz repository Control which file types in a solution are versioned Glyphs/icons display files version control state Rename and move files while retaining history Display history of selected files	Jazz is selected as the SCM provider for the Visual Studio solution. All updates to the Solution Explorer are then managed by Jazz and Rational Team Concert,
Team Artifact Navigator	Allows developers to explore multiple Team Concert repositories from here. Join multiple projects View streams View repository workspaces Load and unload workspaces and start work Create new workspace Create new stream Create new component View and run pre-defined queries	Similar to Eclipse Team Artifact Navigator Has components, streams and baselines unlike VS Team System source control Does not copy workspace contents to perform branches (weakness of VS Team System which does not scale for large workspaces)
Pending Changes View	Shows status of all changes made by the developer and provides access to advanced scm features. Deliver change sets View and modify change flow of work to streams Create baseline, rollback to another baseline Merge, rollback a change set	Similar to Eclipse Pending Changes View
Work Item View	Result set from a work item query is displayed here. Can sort the view Quick edit UI to update most work item fields	Similar to Eclipse Work Item View

Details of Team Concert views surfaced in Visual Studio

View (Tool Window)	Feature	Comments
Change Set Explorer	Explore a change set and its contents View before and after state of file in the change set Associate work items Remove work items	Same as Rational Team Concert's Eclipse UI capabilities.
Change Set Search View	Developer can search for change sets using search criteria Useful for looking at past deliveries of work	Same as Rational Team Concert's Eclipse UI capabilities.
History View	Shows history of files and components. Compare with previous version, local version or arbitrary version Can then open in Change Set Explorer view	Similar to Rational Team Concert Eclipse UI
Repository File Browser	Browse repository workspaces, components, files and folders without explicitly downloading them to the users local hard disk Browse and view files selectively.	Same as Rational Team Concert's Eclipse UI capabilities

What do I need to get started?

Rational Team Concert client for Visual Studio

- **Prerequisities**
  - ✓ Rational Team Concert v1.0.1.1
  - ✓ Visual Studio 2005 or 2008
  - ✓ Register at www.jazz.net
  - ✓ Download from www.jazz.net
- **Learn More**
  - Watch the video!
- **Technical Articles**
  - Source Controlling Visual Studio Projects and Solutions in Team Concert
  - Integrating Visual Studio builds with Team Concert
- **Provide Community Feedback**
  - jazz.user forum
  - Submit bugs or enhancement requests

IBM Rational Team Concert
transparent integrated presence
wikis OPEN real-time reporting
chat automated hand-offs Web 2.0
custom dashboards automated data gathering
EXTENSIBILITY Eclipse plug-ins services
architecture FREEDOM TO CREATE
JAZZ TEAM SERVER

IBM
Software
Group

IBM Software Group

Lab Overview

An IBM Proof of Technology

Scenario for PoT Labs

- You are joining a new project called Squawk that has recently been started in your company.
- You will be using Rational Team Concert as the project's collaborative development environment.
- You have joined the project at the start of Milestone 2. You and all your team mates will be contributing new content to the application.

Scenario for PoT Labs

- Squawk is a (simple) program that will print out different sounds depending on who “squawks”. The Dog squawker goes “bark”, the Cat squawker goes “meow”, etc. Your main task is to create a new squawker, along with tests and documentation.
- At the same time as creating new squawkers, you will get to participate in some planning activities, interact with your fellow team members, deliver your work to the project, trigger automated builds and various tasks typical for project teams everywhere.
- The project team structure mimics the four major components:
  - Core Library
  - Documentation
  - User Interface
  - Release Engineering (build)
- You are assigned to the Core Library and Documentation teams with a team leader (one of the instructors). Welcome to the team!

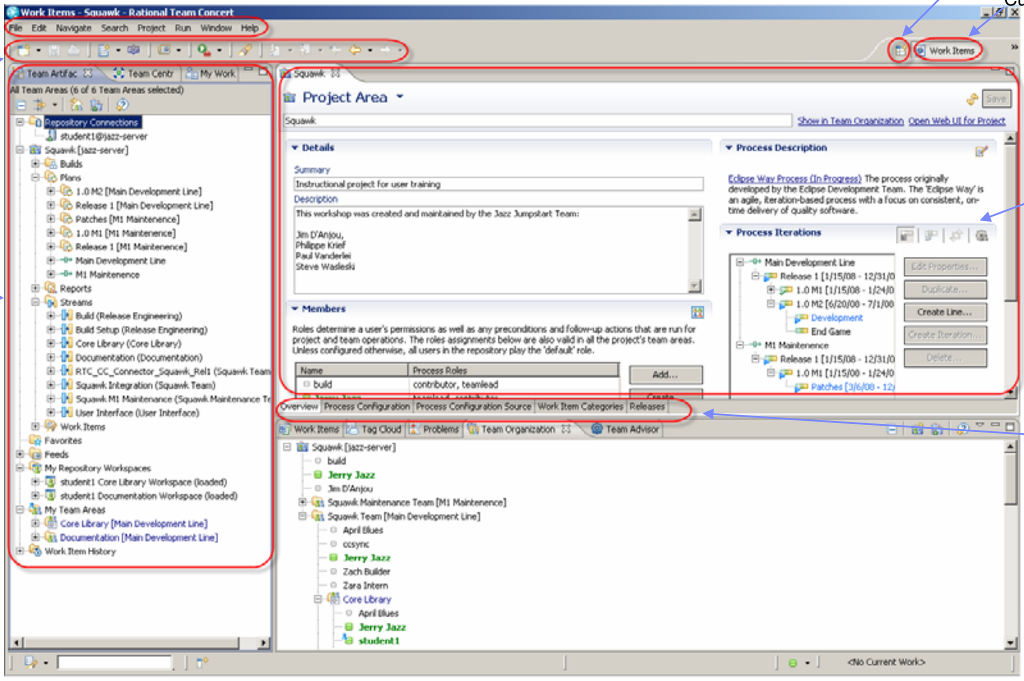
Eclipse Overview

Change Perspective

Current Perspective

Menu

Button bar



All Team Areas (6 of 6 Team Areas selected)

Squawk

Summary: Instructional project for user training

Description: This workshop was created and maintained by the Jazz Jumpstart Team:

Jim D'Anjou,
Philippe Krief
Paul Vanderlei
Steve Wasleski

Eclipse Way Process (In Progress) The process originally developed by the Eclipse Development Team. The 'Eclipse Way' is an agile, iteration-based process with a focus on consistent, on-time delivery of quality software.

Roles determine a user's permissions as well as any preconditions and follow-up actions that are run for project and team operations. The roles assignments below are also valid in all the project's team areas. Unless configured otherwise, all users in the repository play the 'default' role.

Name	Process Roles
build	contributor, teamlead

This View has different areas accessed via the View tabs

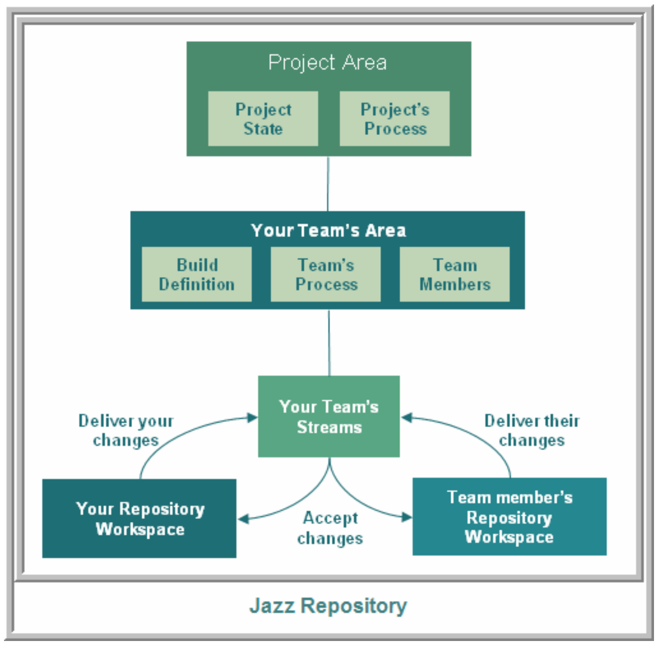
A View

All screen elements below the button bar are called Views

Add new Views using the **Window->Show View** menu

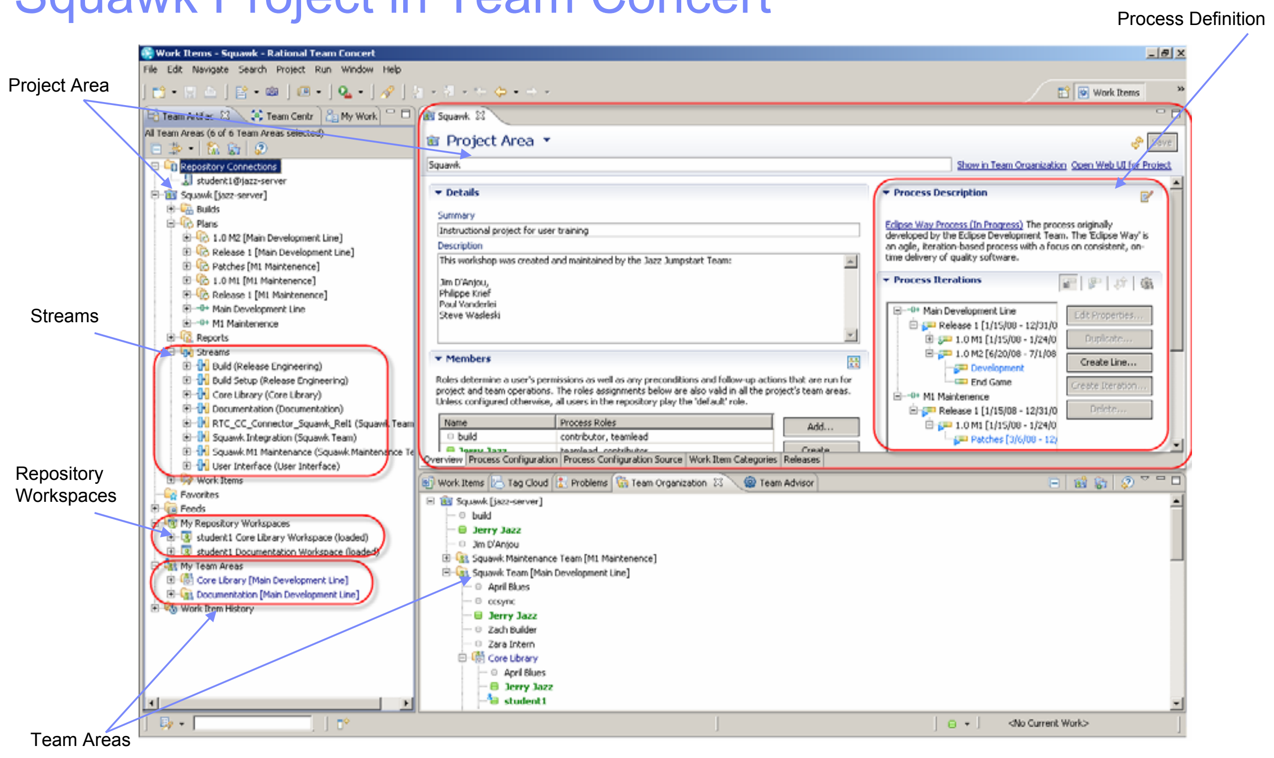
View tabs

Team Concert Terminology



- Jazz artifacts are stored in a repository.
- The repository contains project areas, which are the system's representation of a software projects.
- Each project area has an associated process, which governs how the project is run.
- Project Areas are decomposed into a set of team areas, which describe the teams that work on the project.
- Teams use a stream to store the master copy of project's files.
- Team Members use a personal repository workspace to work on project files.

Squawk Project in Team Concert



All Team Areas (6 of 6 Team Areas selected)

- Repository Connections
  - student1@jazz-server
- Squawk [jazz-server]
  - Builds
  - Plans
    - 1.0 M2 [Main Development Line]
    - Release 1 [Main Development Line]
    - Patches [M1 Maintenence]
    - 1.0 M1 [M1 Maintenence]
    - Release 1 [M1 Maintenence]
    - Main Development Line
    - M1 Maintenence
  - Reports
  - Streams
    - Build (Release Engineering)
    - Build Setup (Release Engineering)
    - Core Library (Core Library)
    - Documentation (Documentation)
    - RTC_CC_Connector_Squawk_Rel1 (Squawk Team
    - Squawk Integration (Squawk Team)
    - Squawk M1 Maintenance (Squawk Maintenance Te
    - User Interface (User Interface)
  - Work Items
- Favorites
- Feeds
- My Repository Workspaces
  - student1 Core Library Workspace (loaded)
  - student1 Documentation Workspace (loaded)
- My Team Areas
  - Core Library [Main Development Line]
  - Documentation [Main Development Line]
- Work Item History

Squawk

Details

Summary: Instructional project for user training

Description: This workshop was created and maintained by the Jazz Jumpstart Team:

Jim D'Anjou,
Philippe Krief
Paul Vanderlei
Steve Wasleski

Members

Roles determine a user's permissions as well as any preconditions and follow-up actions that are run for project and team operations. The roles assignments below are also valid in all the project's team areas. Unless configured otherwise, all users in the repository play the 'default' role.

Name	Process Roles
build	contributor, teamlead
Jerry Jazz	teamlead, contributor

Process Description

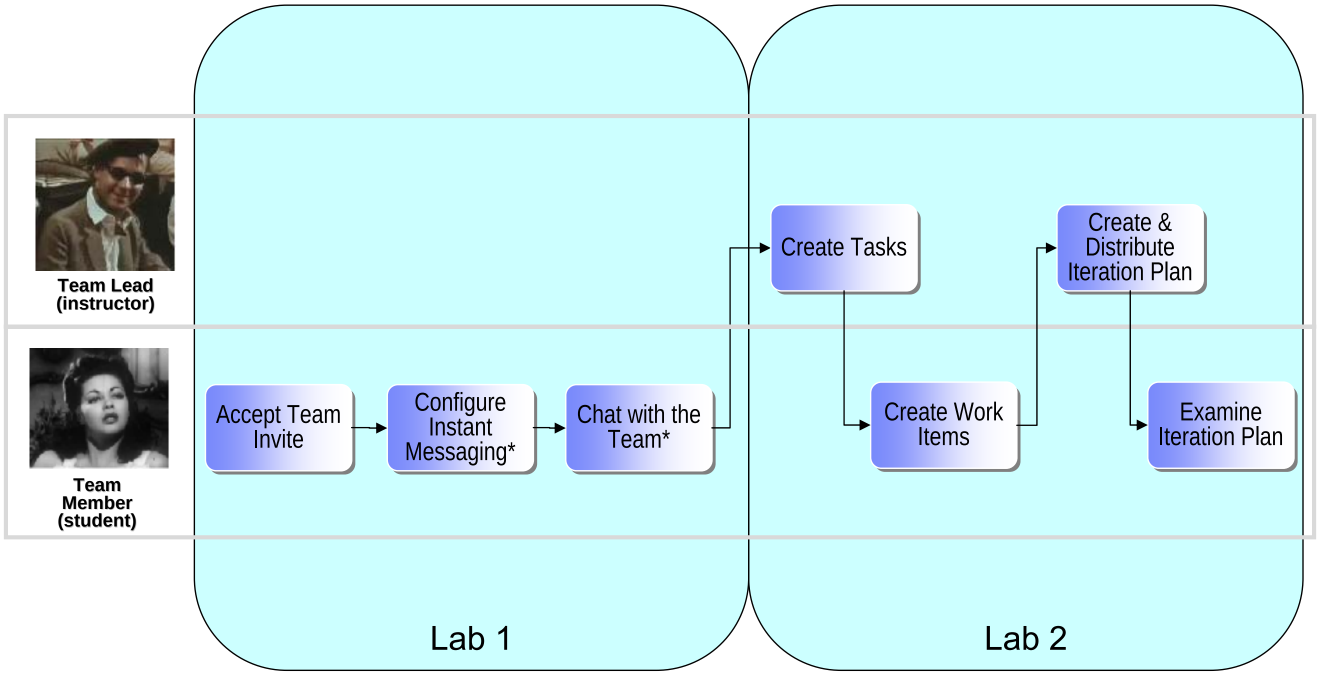
Eclipse Way Process (In Progress) The process originally developed by the Eclipse Development Team. The 'Eclipse Way' is an agile, iteration-based process with a focus on consistent, on-time delivery of quality software.

Process Iterations

- Main Development Line
  - Release 1 [1/15/08 - 12/31/0
    - 1.0 M1 [1/15/08 - 1/24/0
    - 1.0 M2 [6/20/08 - 7/1/08
      - Development
      - End Game
- M1 Maintenence
  - Release 1 [1/15/08 - 12/31/0
    - 1.0 M1 [1/15/08 - 1/24/0
      - Patches [3/6/08 - 12/

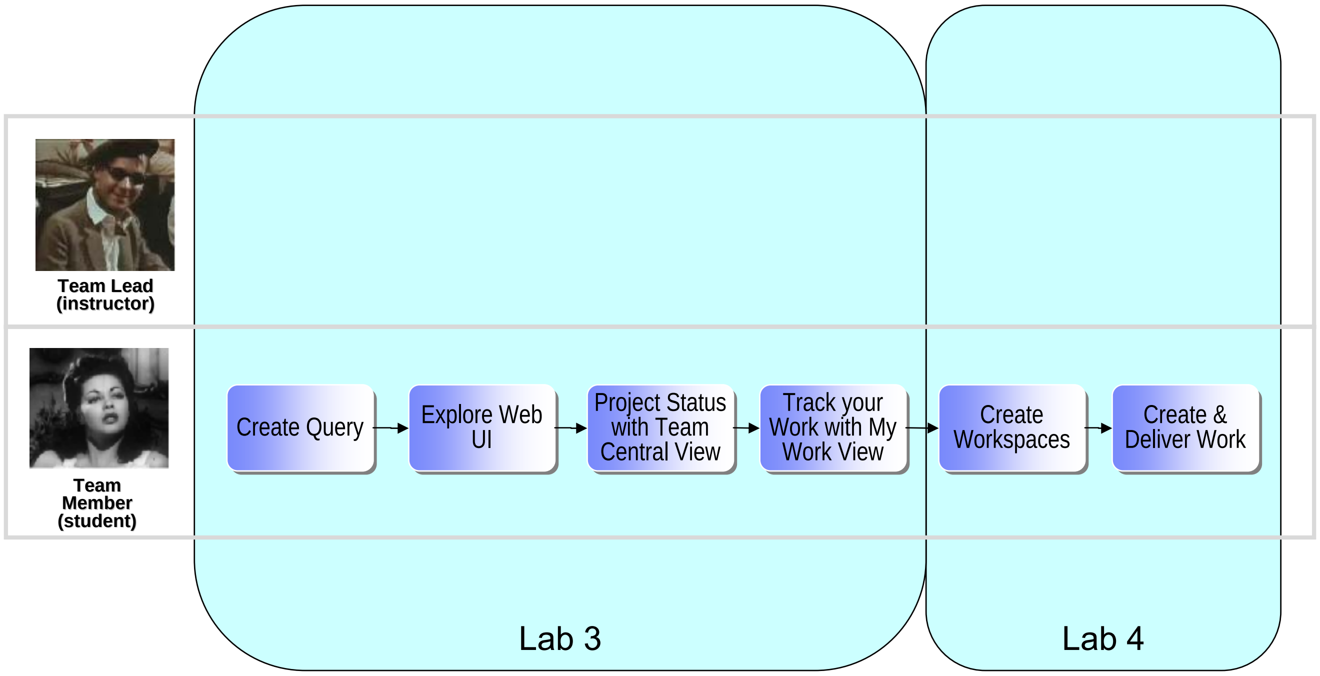
- Squawk [jazz-server]
  - build
  - Jerry Jazz
  - Jim D'Anjou
  - Squawk Maintenance Team [M1 Maintenence]
  - Squawk Team [Main Development Line]
    - April Blues
    - ccsync
    - Jerry Jazz
    - Zach Builder
    - Zara Intern
    - Core Library
      - April Blues
      - Jerry Jazz
      - student1

Sequence of events – Lab 1 & 2

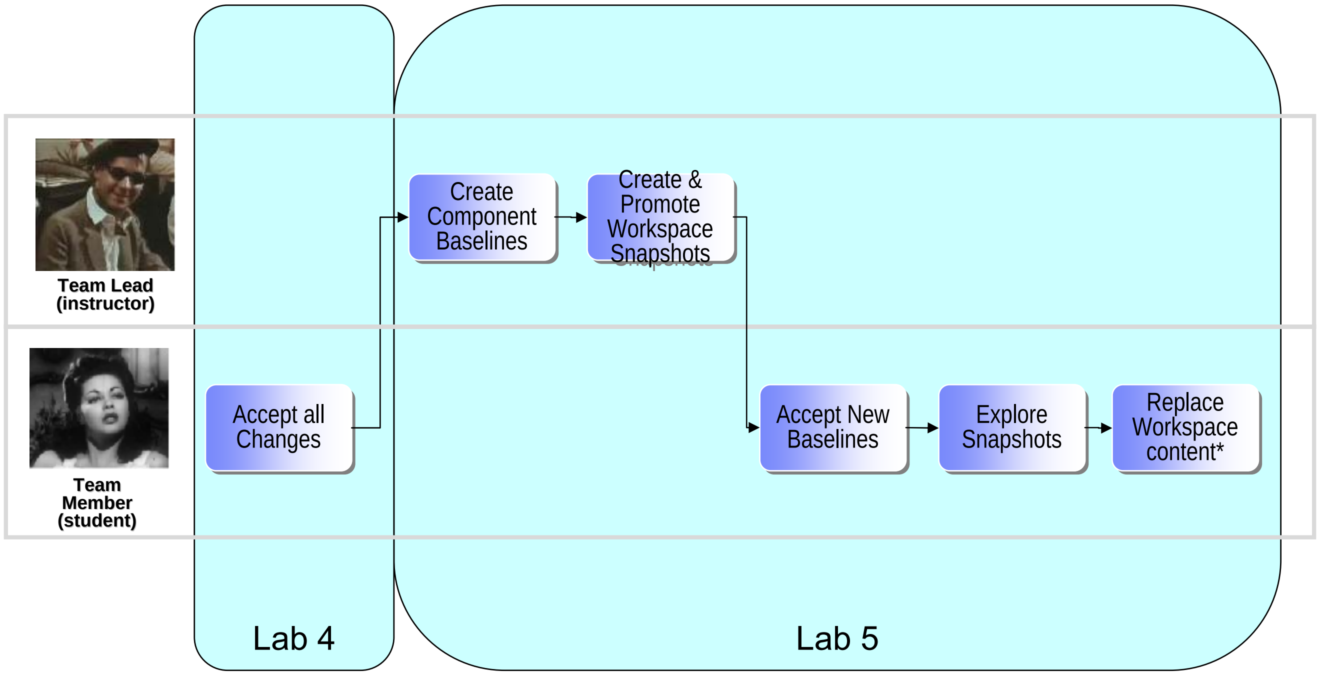


* Not performed in Visual Studio

Sequence of events – Lab 3 & 4

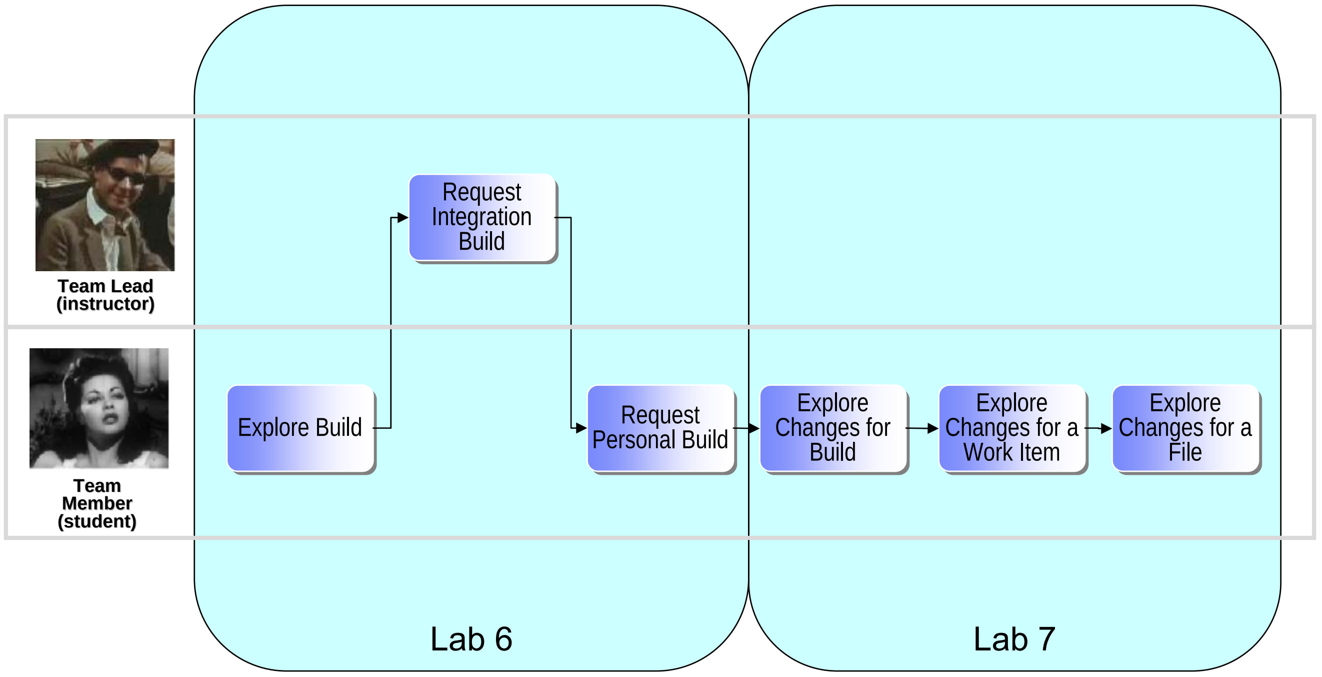


Sequence of events – Lab 4 & 5

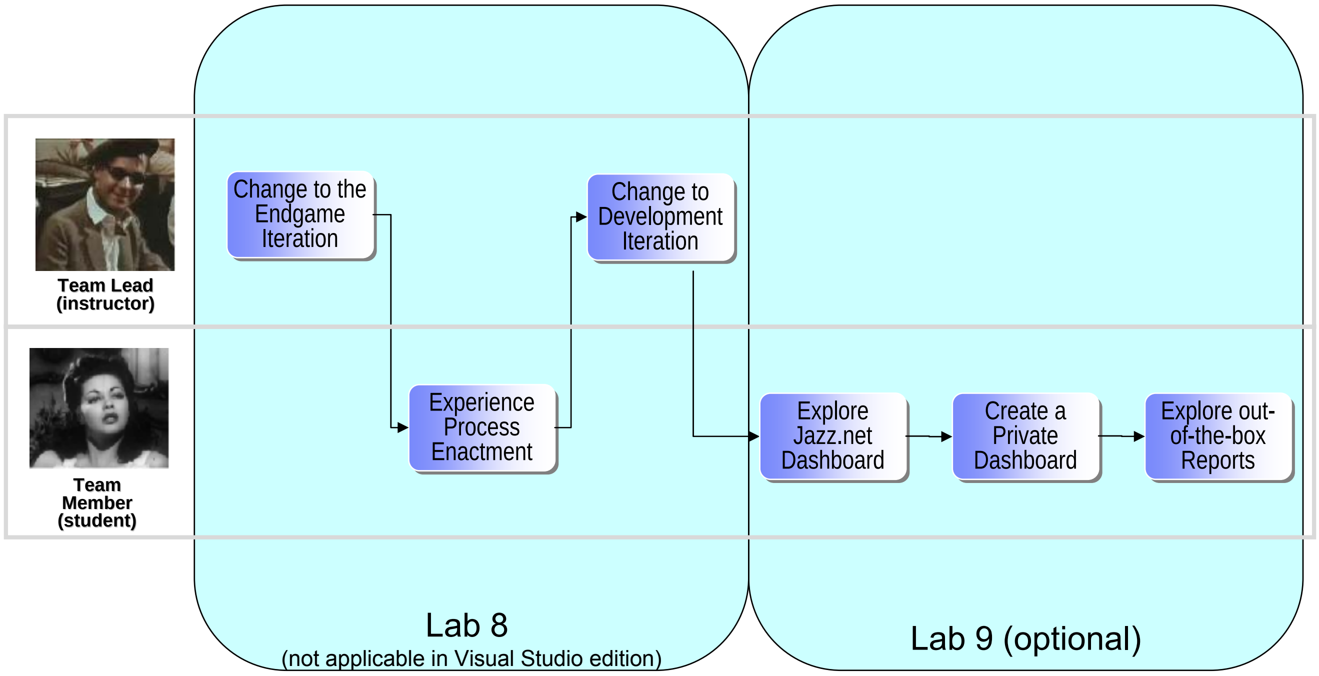


* Not performed in Visual Studio

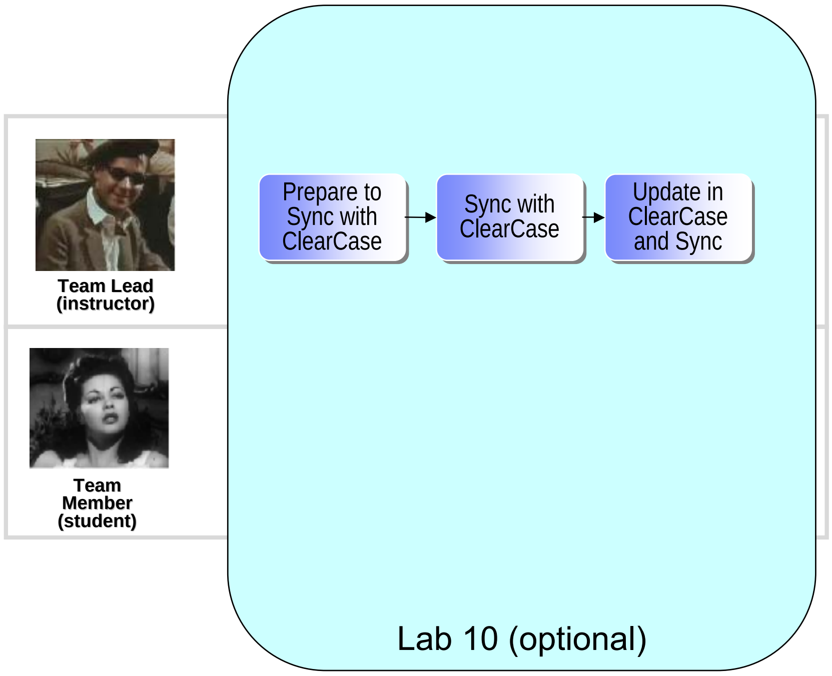
Sequence of events – Lab 6 & 7


Team Lead (instructor)
Team Member (student)
Explore Build
Request Integration Build
Request Personal Build
Explore Changes for Build
Explore Changes for a Work Item
Explore Changes for a File
Lab 6
Lab 7

Sequence of events – Lab 8 & 9


Team Lead (instructor)
Team Member (student)
Change to the Endgame Iteration
Experience Process Enactment
Change to Development Iteration
Explore Jazz.net Dashboard
Create a Private Dashboard
Explore out-of-the-box Reports
Lab 8
(not applicable in Visual Studio edition)
Lab 9 (optional)

Sequence of events – Lab 10

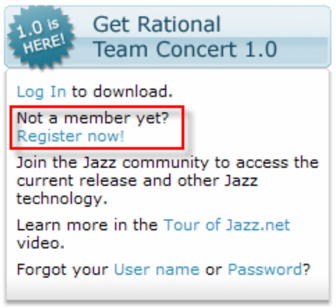


Lab Conventions

- The hostname used to connect to the Jazz Team Server is *jazz-server*
- Each student is assigned a unique user id of the form *student<N>* based on their student number, e.g. *student1*
  - Examples in the lab workbooks use student1, you will need to adjust per your assigned id
- Every student creates their own unique Squawker.
  - Examples in the lab workbooks use *Lion*
- Students can choose any squawker they want but should include your student id in the name
  - *<squawker>_<student id>*, e.g. *Lion_student1*
- Work items created should include the full squawker name in the summary title
  - *<squawker name> Implementation* and *<squawker name> Documentation* e.g. *Lion_student1 Implementation*
- Optionally, adjust the language settings in the VM for international keyboards. Go to **Control Panel -> Regional and Language Options**. Select the **Languages** tab and then click **Details** in the **Text Services and input languages** section. Add your local keyboard and make it the default input language.

Jazz.net Registration

- To run Lab 9, you will need a Jazz.net account.
- Not a Member yet?
  - If you have access web access to your email Server
    - You will receive a confirmation and password resetting instructions
  - Go to www.jazz.net and **register now**.



Get Rational Team Concert 1.0

Log In to download.

Not a member yet?
Register now!

Join the Jazz community to access the current release and other Jazz technology.

Learn more in the Tour of Jazz.net video.

Forgot your User name or Password?

- Creating a Jazz.net account allows you:
  - to take part in the Jazz community.
  - download product trials, betas, and other previews of Jazz technology.
  - have access to articles, tech notes, tutorials
  - interact directly with the development teams and other members of the Jazz community to ask questions, report bugs, provide feedback and help guide the evolution of Jazz technology.

Questions

IBM
Software
Group

Setting up the Team

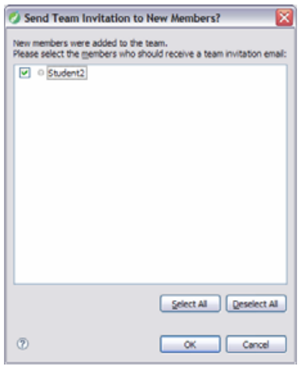
An IBM Proof of Technology

Objectives

- In this lab you will learn how ramp up projects quickly and dramatically improve onboarding and offboarding of team members
- You will perform some initial setup of Rational Team Concert to enable your machine to communicate with the server
- You will enable instant messaging in Rational Team Concert

Joining a project

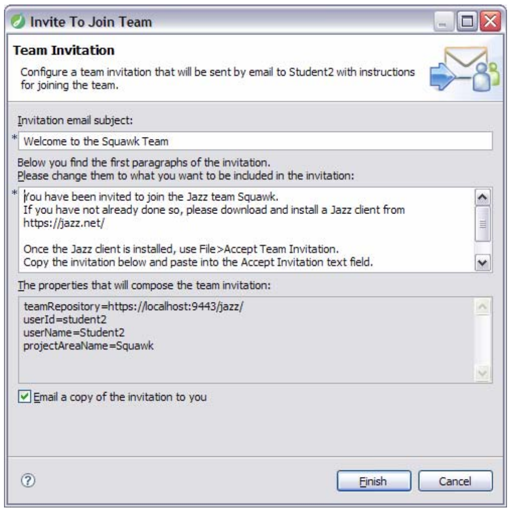
- For most environments, joining a project can be complicated
- Team Concert makes this as easy as possible
- Adding a new team member to a project generates a Team Invitation email
- Contents of the email can be used to set up the new team member's access to the project resources in Team Concert



Send Team Invitation to New Members?

New members were added to the team.
Please select the members who should receive a team invitation email:

✓ Student2



Invite To Join Team

Team Invitation

Configure a team invitation that will be sent by email to Student2 with instructions for joining the team.

Invitation email subject:

Welcome to the Squawk Team

Below you find the first paragraphs of the invitation.
Please change them to what you want to be included in the invitation:

You have been invited to join the Jazz team Squawk.
If you have not already done so, please download and install a Jazz client from https://jazz.net/

Once the Jazz client is installed, use File>Accept Team Invitation.
Copy the invitation below and paste into the Accept Invitation text field.

The properties that will compose the team invitation:

```
teamRepository=https://localhost:9443/jazz/
userId=student2
userName=Student2
projectAreaName=Squawk
```

✓ Email a copy of the invitation to you

Communication

- Users of Team Concert can use a variety of tools to communicate with team members
  - E-mail
  - Instant Messaging/Chat *
  - RSS feeds *
  - Web UI
  - Team Concert client
- Team members can use all the typical communication mechanisms to keep working together as a team, regardless of where they are physically located. This collaboration allows for a single view of project data
  - Integrated Instant Messaging/Chat for immediate feedback
  - RSS feeds to notify you of significant events on the project in real time
  - The Web UI used for anyone on the team, or who has an interest in the project

*** Not currently available in Visual Studio**

Lab #1 Scenario

- You arrive at work on Day 1 and receive an email inviting you to the Squawk project.
- You start Team Concert and get connected to the project right away.
- You use instant messaging to chat to your colleagues on the project

Lab #1 Overview

- Use the team invitation received via email to get connected to the Squawk project
- Configure your Team Concert workspace for Instant Messaging
- Explore the organization of your new team and start up a friendly chat to introduce yourself

Lab #1 Concepts Learned

- Team Invitations make it easier and foolproof to get team member's connected to your project resources managed in Rational Team Concert
- Rational Team Concert has built-in instant messaging support that makes it easy to connect and collaborate with your teammates

IBM
Software
Group

Planning Your Work

An IBM Proof of Technology

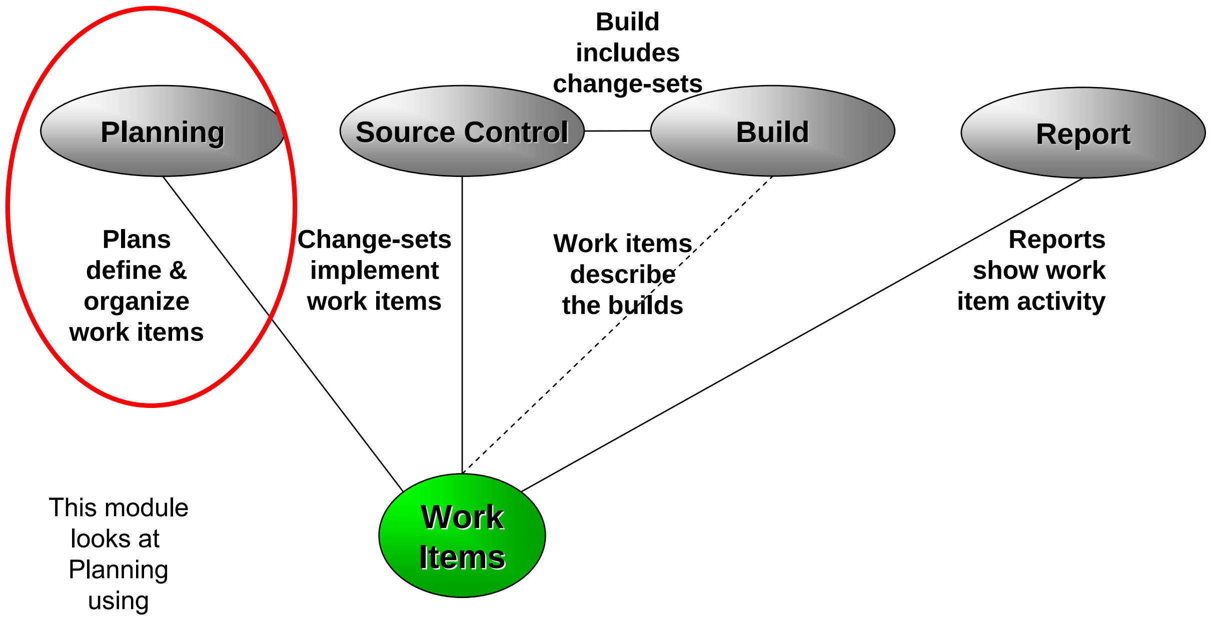
Objectives

- Learn about Work Items and how they are central to Rational Team Concert
- Understand Rational Team Concert’s project planning capabilities
- In the lab you will learn how to create and work with Work Items and Iteration Plans

Units of work

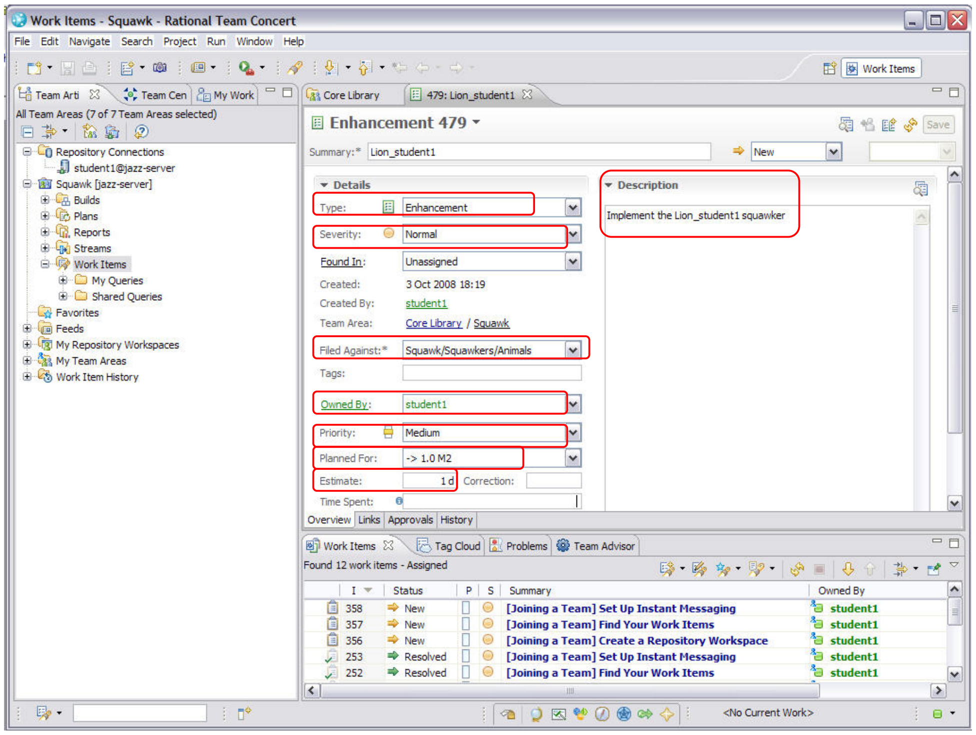
- Rational Team Concert allows you to divide your work into **work items**
- The set of work item types is **open-ended**
- Standard types such as **Task**, **Enhancement**, **Defect**
- The set is **defined by each team**
- The work item **life cycle** is **configurable**
- All work items are **stored** in the **repository**

Work Items are central to Rational Team Concert


Build includes change-sets
Planning
Source Control
Build
Report
Plans define & organize work items
Change-sets implement work items
Work items describe the builds
Reports show work item activity
Work Items

This module looks at Planning using Work items

Work Item details



Work Items - Squawk - Rational Team Concert

File Edit Navigate Search Project Run Window Help

Team Arti | Team Cen | My Work

All Team Areas (7 of 7 Team Areas selected)

- Repository Connections
  - student1@jazz-server
- Squawk [jazz-server]
  - Builds
  - Plans
  - Reports
  - Streams
  - Work Items
    - My Queries
    - Shared Queries
- Favorites
- Feeds
- My Repository Workspaces
- My Team Areas
- Work Item History

Core Library | 479: Lion_student1

Enhancement 479

Summary:* Lion_student1 — New

Details

Type:	Enhancement
Severity:	Normal
Found In:	Unassigned
Created:	3 Oct 2008 18:19
Created By:	student1
Team Area:	Core Library / Squawk
Filed Against:*	Squawk/Squawkers/Animals
Tags:	
Owned By:	student1
Priority:	Medium
Planned For:	-> 1.0 M2
Estimate:	1 d
Correction:	
Time Spent:	

Description

Implement the Lion_student1 squawker

Overview | Links | Approvals | History

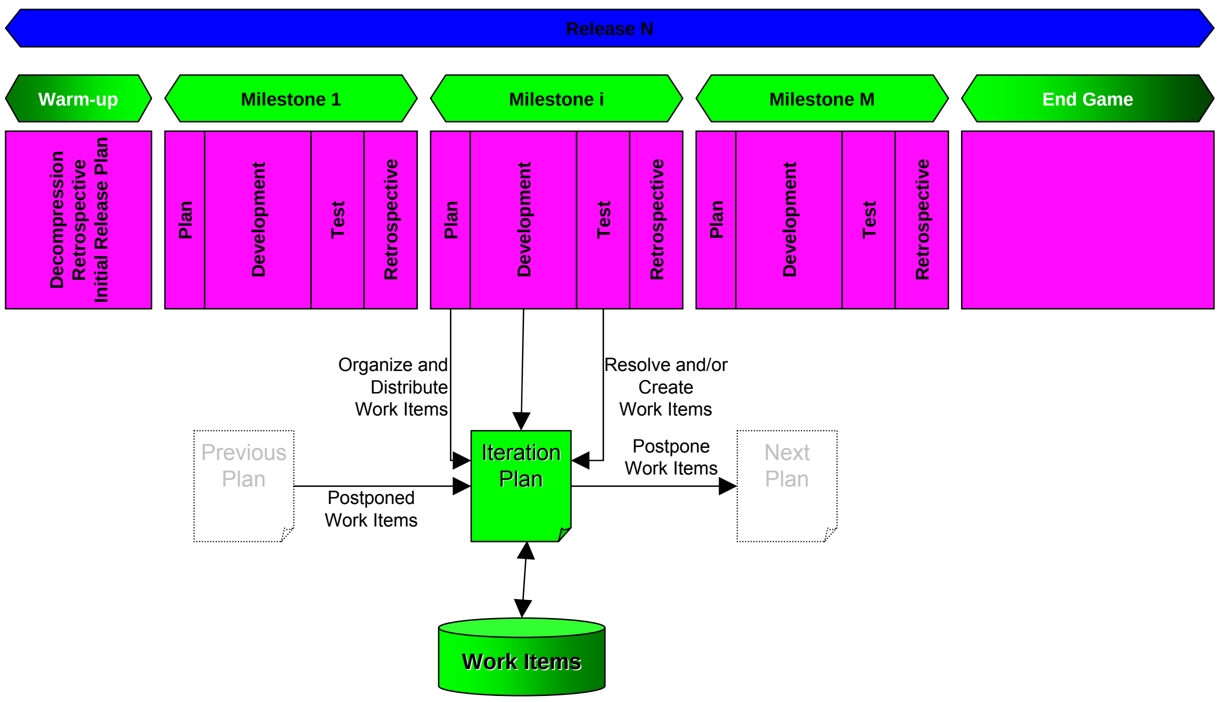
Work Items | Tag Cloud | Problems | Team Advisor

Found 12 work items - Assigned

I	Status	P	S	Summary	Owned By
358	New			[Joining a Team] Set Up Instant Messaging	student1
357	New			[Joining a Team] Find Your Work Items	student1
356	New			[Joining a Team] Create a Repository Workspace	student1
253	Resolved			[Joining a Team] Set Up Instant Messaging	student1
252	Resolved			[Joining a Team] Find Your Work Items	student1

<No Current Work>

Iteration planning in Rational Team Concert



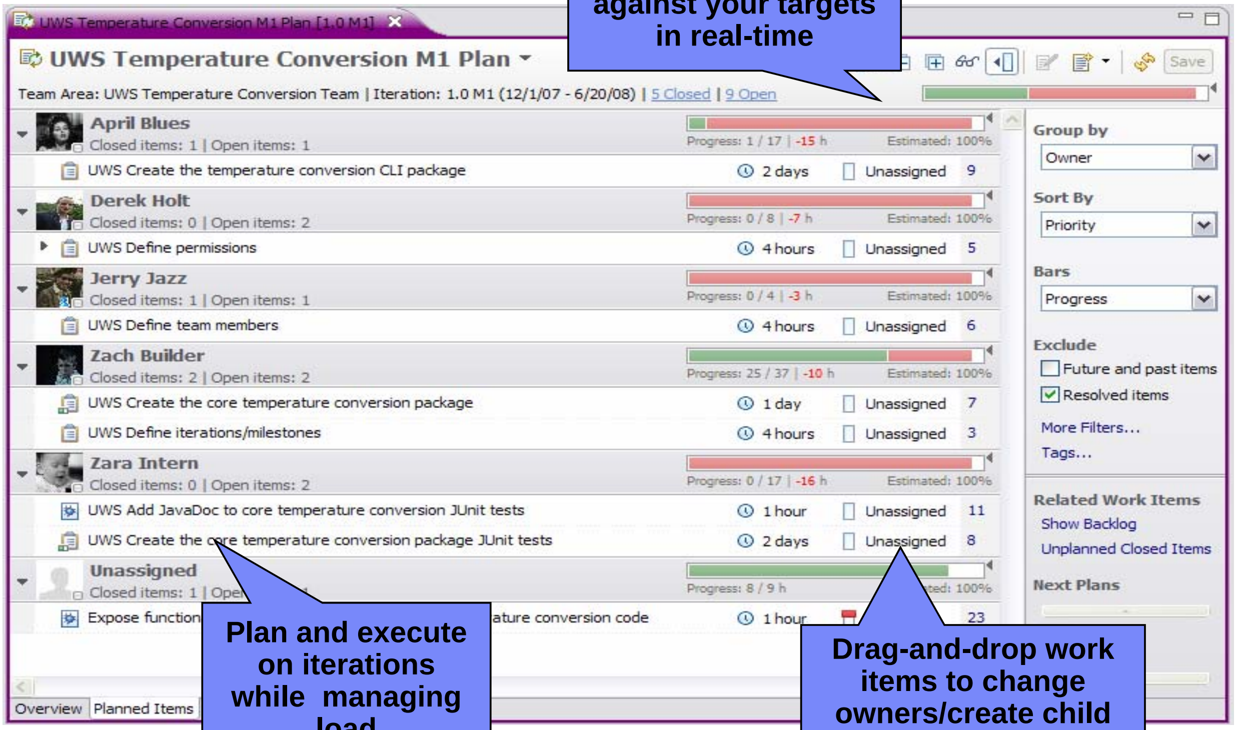
The Iteration Plan

- A collection of work items…
  - Assigned to a given milestone
  - For a given team
- Plans are live
  - Changing work items changes the plan
  - Changing the plan directly changes the work items
  - Create new work items from the plan
- Plan structure is dynamic
  - Easily be grouped by owner, category, duration, priority, etc.
- Plans are visible
  - Available to everybody on the team
  - Observable by interested outsiders

Iteration Planning *

Understand how well you are progressing against your targets in real-time

* Not currently available in Visual Studio



UWS Temperature Conversion M1 Plan [1.0 M1]

UWS Temperature Conversion M1 Plan

Team Area: UWS Temperature Conversion Team | Iteration: 1.0 M1 (12/1/07 - 6/20/08) | 5 Closed | 9 Open

April Blues — Closed items: 1 | Open items: 1 — Progress: 1 / 17 | -15 h — Estimated: 100%

UWS Create the temperature conversion CLI package	2 days	Unassigned	9

Derek Holt — Closed items: 0 | Open items: 2 — Progress: 0 / 8 | -7 h — Estimated: 100%

UWS Define permissions	4 hours	Unassigned	5

Jerry Jazz — Closed items: 1 | Open items: 1 — Progress: 0 / 4 | -3 h — Estimated: 100%

UWS Define team members	4 hours	Unassigned	6

Zach Builder — Closed items: 2 | Open items: 2 — Progress: 25 / 37 | -10 h — Estimated: 100%

UWS Create the core temperature conversion package	1 day	Unassigned	7
UWS Define iterations/milestones	4 hours	Unassigned	3

Zara Intern — Closed items: 0 | Open items: 2 — Progress: 0 / 17 | -16 h — Estimated: 100%

UWS Add JavaDoc to core temperature conversion JUnit tests	1 hour	Unassigned	11
UWS Create the core temperature conversion package JUnit tests	2 days	Unassigned	8

Unassigned — Closed items: 1 | Ope — Progress: 8 / 9 h — ted: 100%

Expose function ature conversion code	1 hour		23

Group by: Owner
Sort By: Priority
Bars: Progress
Exclude: ☐ Future and past items ☑ Resolved items
More Filters…
Tags…
Related Work Items: Show Backlog, Unplanned Closed Items
Next Plans

Plan and execute on iterations while managing load

Drag-and-drop work items to change owners/create child parent relationships

Lab #2 Scenario

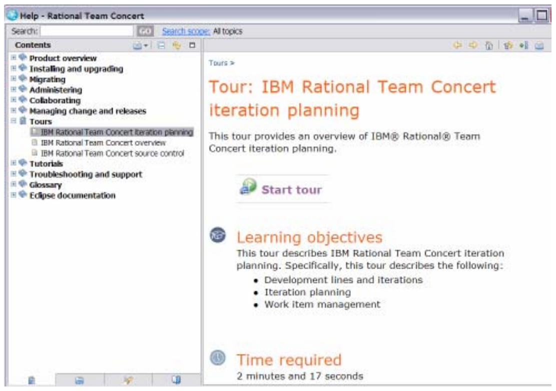
- You want to track all the work on your projects.
- All your work (tasks, defects, enhancements) are based around the concept of Work Items.
- You see how Work Items are fundamental to Rational Team Concert and how you use these work items to track the work you do.
- You prioritize and link your work so that you can do the right things at the right time in the plan.

Lab #2 Overview

- The instructor will create new tasks to create additional squawkers.
- You will create enhancement work items for the Squawk project
- You will assign your work to the right team member (you!)
- You will set the priority for your work items and estimate how long they will take to complete
- You will link your new work items to the ones created by the instructor. The relevant plans will be updated automatically.
- The instructor will create a new Doc for M2 plan and check all your work items are part of the relevant plan.
- The instructor will send you the plans via Chat and you will examine the plan.

Lab #2 Concepts Learned

- Work Items in Rational Team Concert are a central team artifact in the development process
- Everything gets tracked using Work Items so nothing gets lost which provides project transparency and real time data access
- Work Items are used to create the Iteration Plan linking project data to the overall plan
- Iteration Plans are live, dynamic and visible to the entire team helping to create a collaborative project environment
- Video overview available from the online Help under **Tours**


Help - Rational Team Concert
Search:
GO
Search scope: All topics
Contents
Product overview
Installing and upgrading
Migrating
Administering
Collaborating
Managing change and releases
Tours
IBM Rational Team Concert iteration planning
IBM Rational Team Concert overview
IBM Rational Team Concert source control
Tutorials
Troubleshooting and support
Glossary
Eclipse documentation
Tours >
Start tour

Tour: IBM Rational Team Concert iteration planning

This tour provides an overview of IBM® Rational® Team Concert iteration planning.

Learning objectives

This tour describes IBM Rational Team Concert iteration planning. Specifically, this tour describes the following:

- Development lines and iterations
- Iteration planning
- Work item management

Time required

2 minutes and 17 seconds

IBM
Software
Group

Keep Track of All Our Work

An IBM Proof of Technology

Objectives

- Explore Rational Team Concert query capabilities
- Create and run queries
- Use and configure the Team Central * and My Work * views to get a real time view of project, team and individual status

*** Not currently available in Visual Studio**

Real time collaboration

- The modules before showed how the project team plans the work for an iteration.
  - But how does the project keep track of all the planned work items?
  - How do I see who may help me with my actual problem?
  - How do I get the most recent status of the project?

What if your tool knows the actual status of your team's work?

- Rational Team Concert stores all artifacts for the development project in one repository and provides powerful query capabilities to retrieve and display data.

Real time collaboration

- Utilize Rational Team Concert’s extensive collaboration capabilities
  - Define queries on Work Items to find your work and the work of others.
  - See who is online and ready to collaborate with you.
  - See the event log for build or work item events that are interesting to you and follow RSS feeds for News.
  - Generate, display and export reports on the status and health of the project.
- Rational Team Concert displays the information in automatically refreshed views that are configurable, so that you are up to date with the information you need in real time.

Work Item Queries

- Provides real-time project health information and transparency of status through automated data gathering.
- Rational Team Concert provides a query mechanism to find work items in a Project Area allowing for more project transparency.
  - The query scope for work items is the project area.
- The user interface includes
  - an editor for building structured work item queries
  - an end-user configurable work item view to browse the query results.

Lab #3 Scenario

- You recently joined the development staff of the Squawk project
- Your environment was properly set up by accepting the invitation for the Core project team
- Now it is your task to become familiar with the work and the tasks to do.
- You are using the real time collaboration capabilities of Rational Team Concert to be up to date with the
  - latest news feeds,
  - status of the project and
  - work items assigned to you

Lab #3 Objectives

- As a user you will learn how to:
  - Write and run work item queries in the Eclipse client and in the Web UI.
  - Use the capabilities of the Team Central View.
  - Configure the My Work View.

Lab #3 Concepts Learned

- Rational Team Concert provides powerful query capabilities for work items creating real time access to detailed project data
- Create customized queries or use predefined queries to enable unique project views for a wide range of users
- Rational Team Concert helps teams collaborate by creating an environment where real time project status and data are available.
- Easily customized views to fit your needs

Questions

IBM
Software
Group

Performing and Sharing Your Work

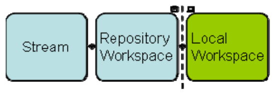
An IBM Proof of Technology

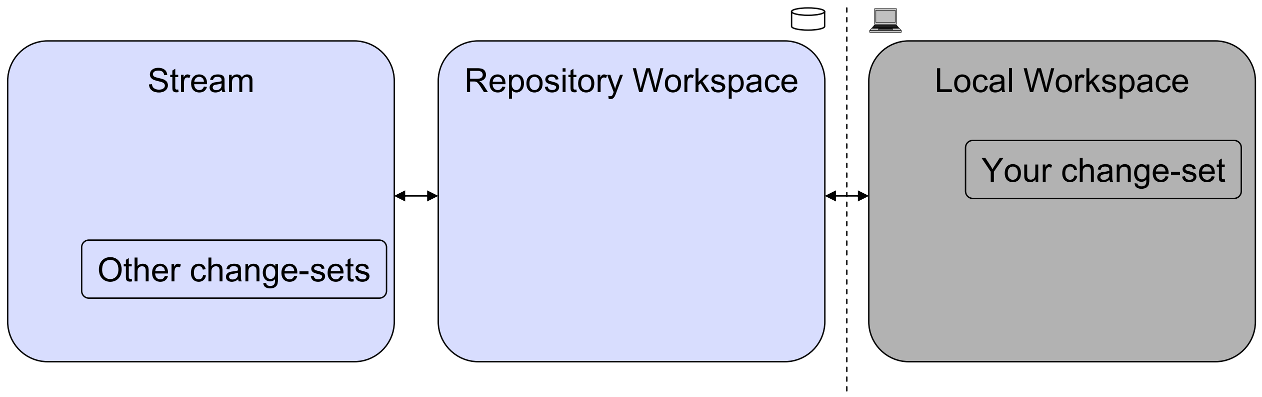
Rational. software

Objectives

- Understand Software Configuration Management (SCM) concepts in Rational Team Concert
- Create and use a Repository Workspace for work assigned to you
- Create or make changes to artifacts under source control
- Associate changes with Work Items
- Deliver changes from Repository Workspaces to Streams
- Accept changes from other members of your team

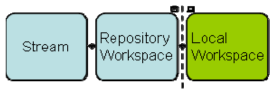
Basic Jazz SCM Anatomy

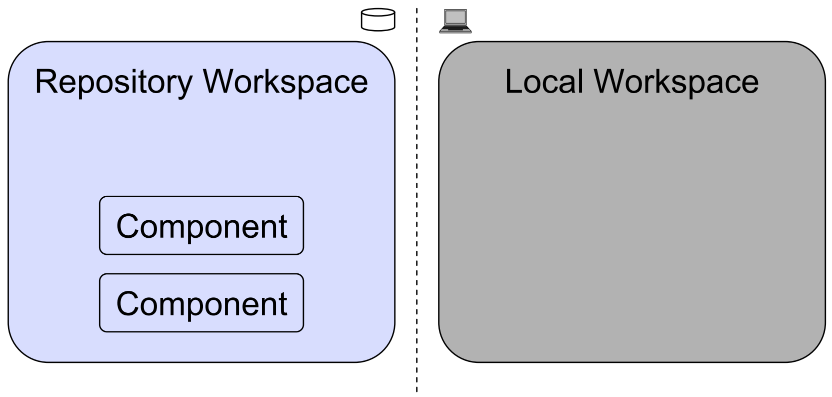




- Streams are for sharing
- Repository workspaces are your personal space
- The local workspace is a folder on your local files system where you develop and test
- Change-sets flow back and forth

Components

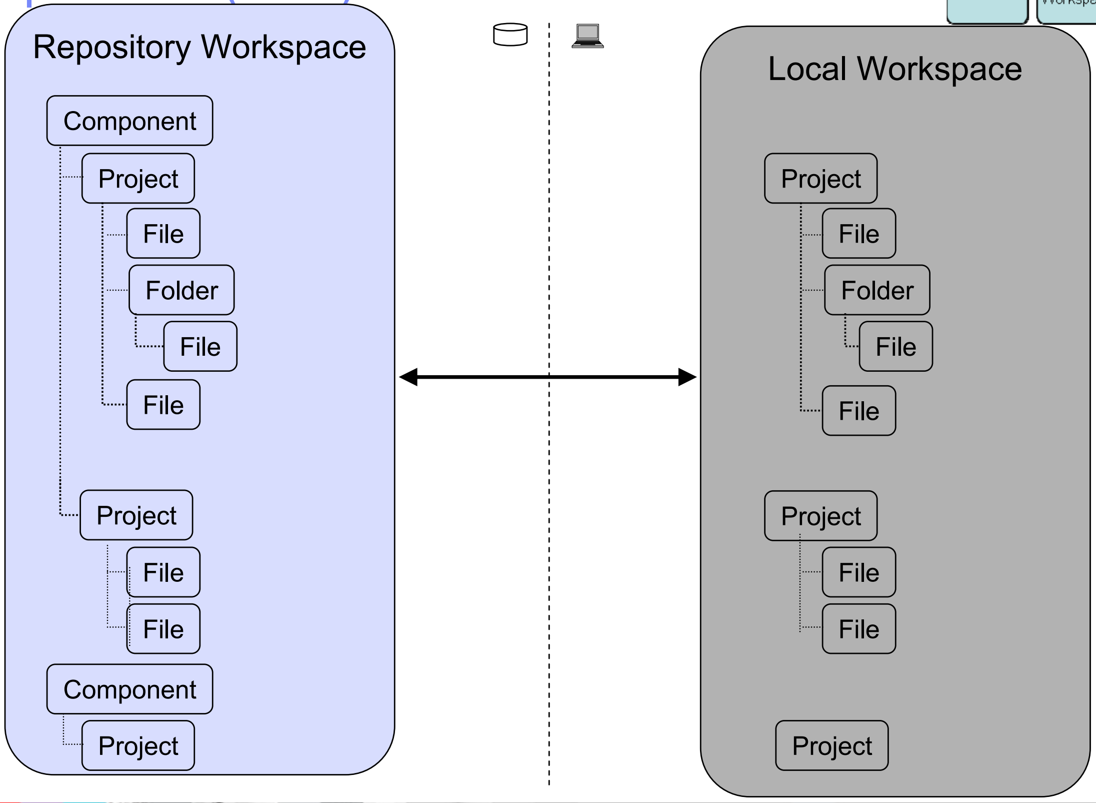




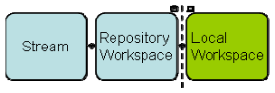
- Repository Workspaces
  - Partitioned into components
  - Jazz understands the structure of your components
  - Jazz directly supports component based development

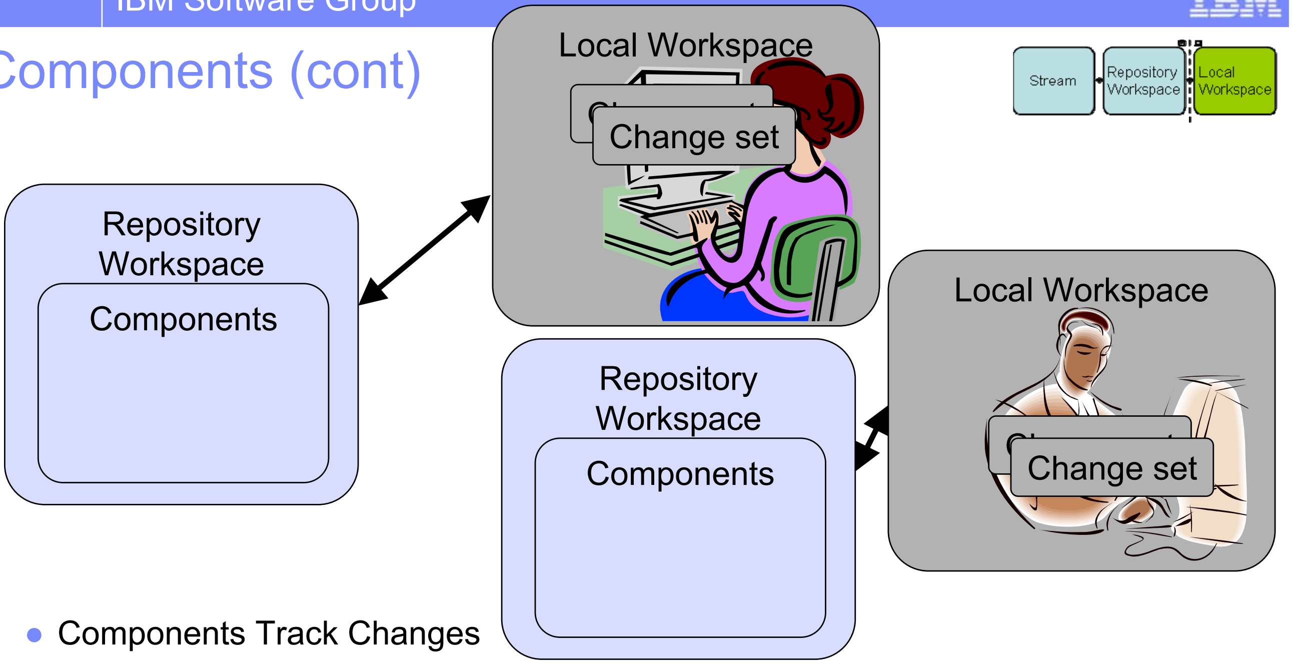
Components (cont)

Stream
Repository Workspace
Local Workspace



Components (cont)



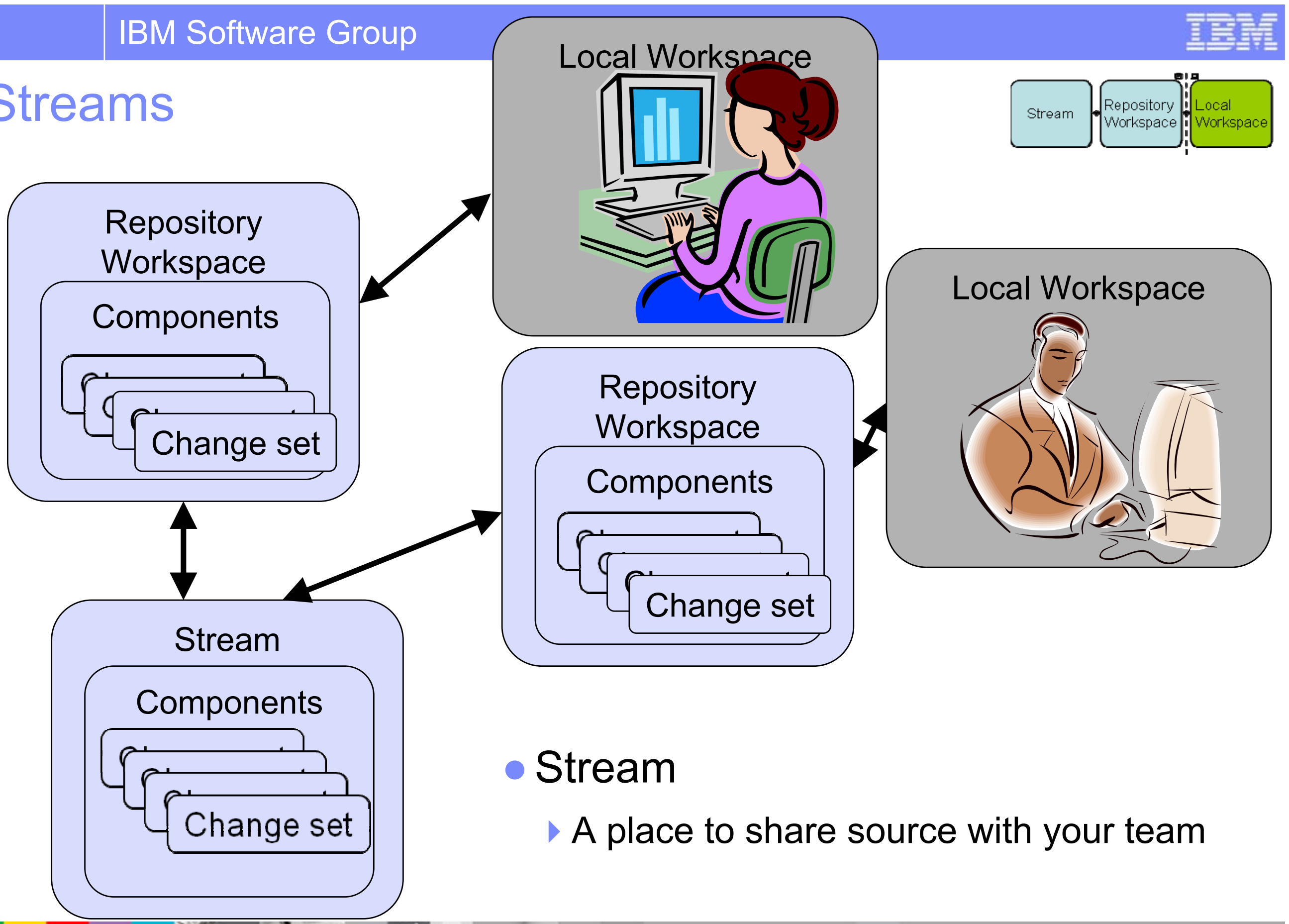


- Components Track Changes
  - Configuration of resources builds from the change set flow
  - Each change set builds on what came before
- Component's Change History
  - A time-ordered sequence of change sets
  - Describes how the component's content was built from nothing

Change set Details

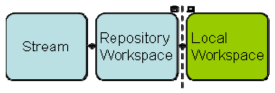
- Composed from a collection of changes to one or more files and folders
  - A change set that affects multiple resources is committed as a single atomic unit
- Indicates the reason for the changes
  - Via a comment, and/or
  - By referencing the relevant work item
- Can be shared with another team member
  - Via a stream, or
  - From your repository workspace via a work item

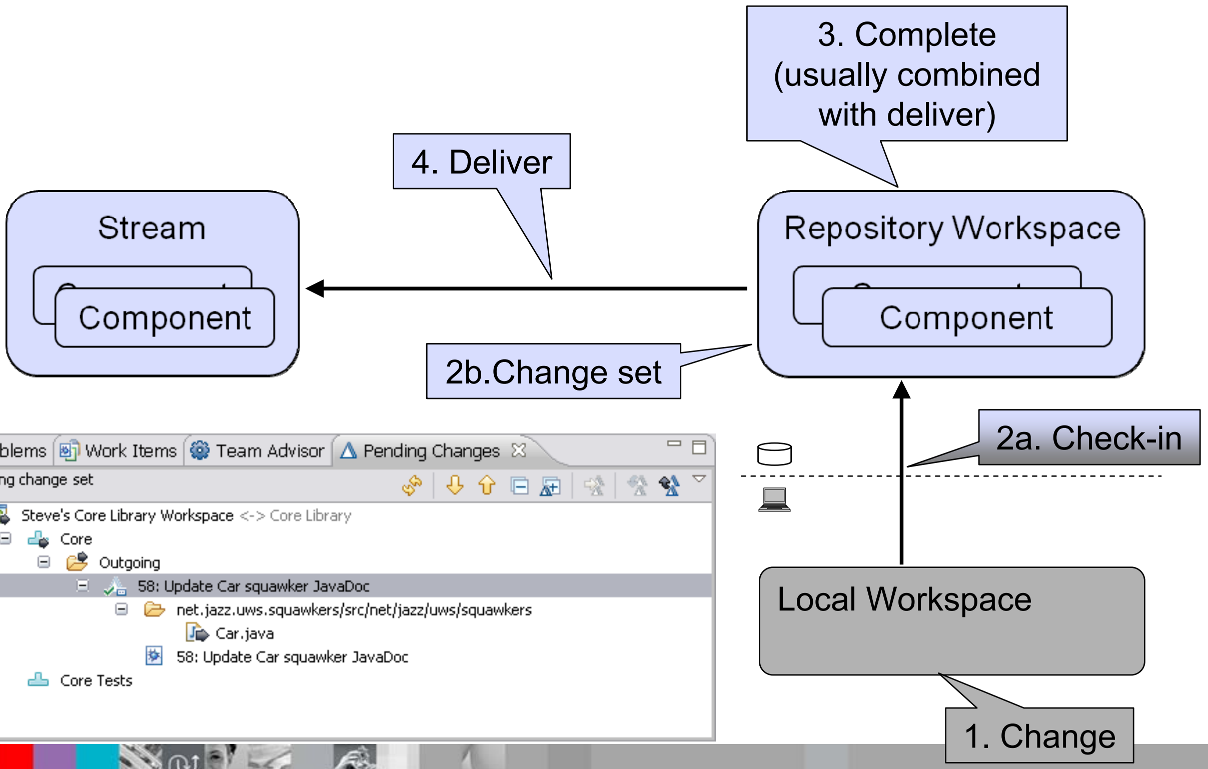
Streams



- Stream
  - A place to share source with your team

Typical Journey For A Change set

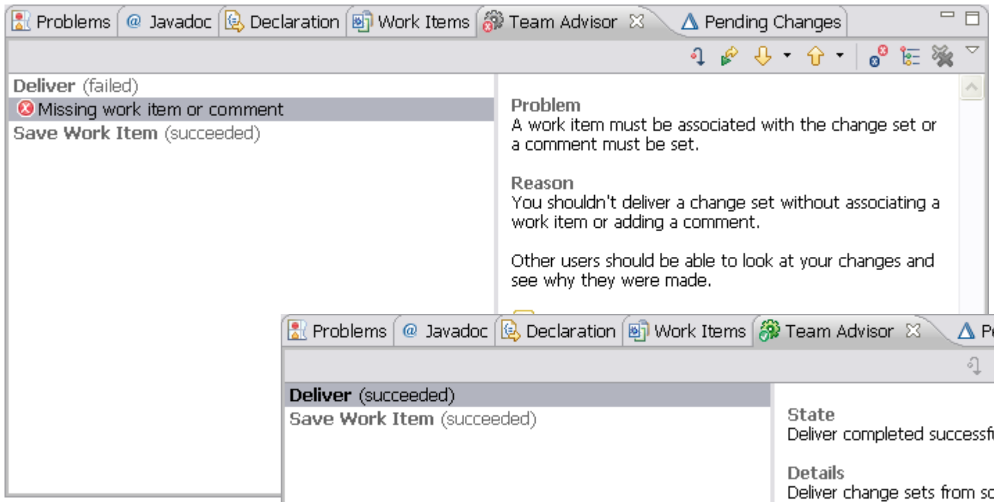




1 outgoing change set
Steve's Core Library Workspace <-> Core Library
Core
Outgoing
58: Update Car squawker JavaDoc
net.jazz.uws.squawkers/src/net/jazz/uws/squawkers
Car.java
58: Update Car squawker JavaDoc
Core Tests

Change-set Delivery is Process Enabled*

- The deliver operation is process-enabled, allowing the team's process to check and enforce delivery rules automatically

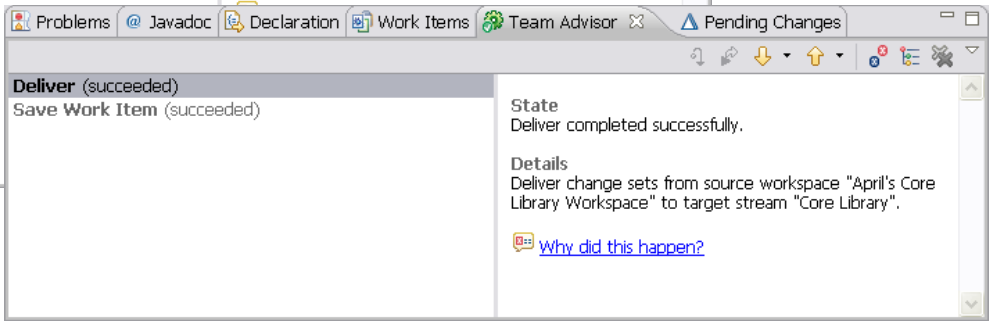


Deliver (failed)
Missing work item or comment
Save Work Item (succeeded)

Problem
A work item must be associated with the change set or a comment must be set.

Reason
You shouldn't deliver a change set without associating a work item or adding a comment.

Other users should be able to look at your changes and see why they were made.



Deliver (succeeded)
Save Work Item (succeeded)

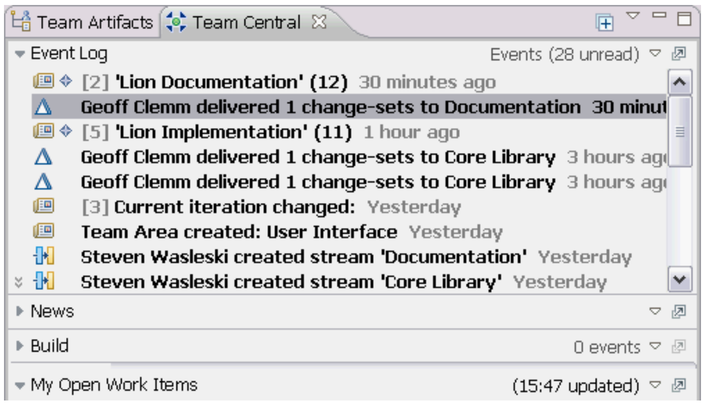
State
Deliver completed successfully.

Details
Deliver change sets from source workspace "April's Core Library Workspace" to target stream "Core Library".

Why did this happen?

*** Not currently available in Visual Studio and command line**

Delivery Notifications



Team Artifacts | Team Central

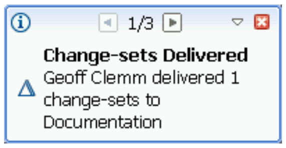
Event Log — Events (28 unread)

- [2] 'Lion Documentation' (12) 30 minutes ago
- Geoff Clemm delivered 1 change-sets to Documentation 30 minut
- [5] 'Lion Implementation' (11) 1 hour ago
- Geoff Clemm delivered 1 change-sets to Core Library 3 hours ag
- Geoff Clemm delivered 1 change-sets to Core Library 3 hours ag
- [3] Current iteration changed: Yesterday
- Team Area created: User Interface Yesterday
- Steven Wasleski created stream 'Documentation' Yesterday
- Steven Wasleski created stream 'Core Library' Yesterday

News

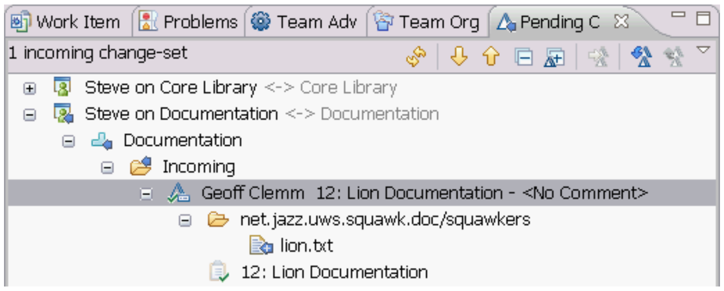
Build — 0 events

My Open Work Items — (15:47 updated)



1/3

Change-sets Delivered
Geoff Clemm delivered 1 change-sets to Documentation

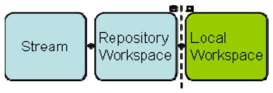


Work Item | Problems | Team Adv | Team Org | Pending C

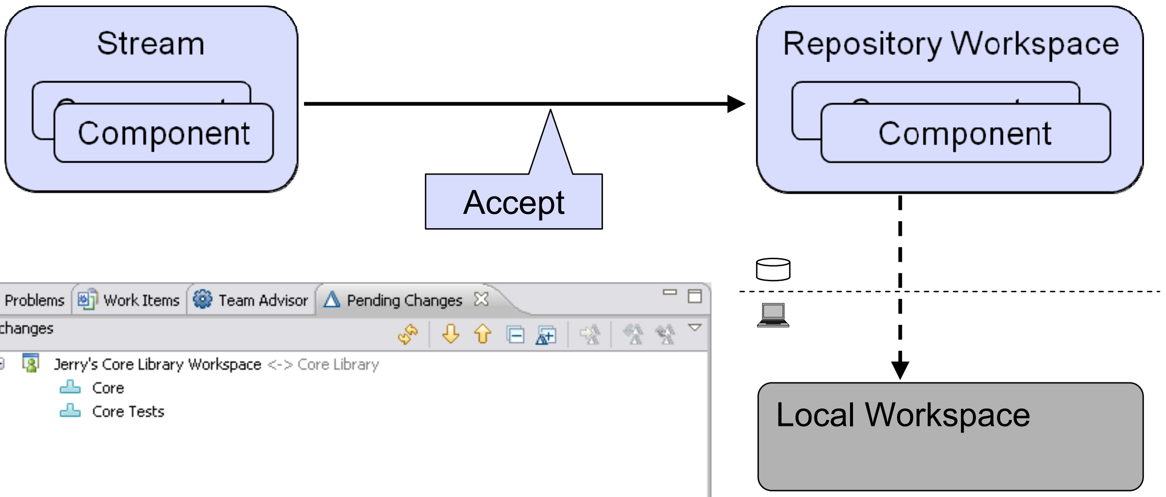
1 incoming change-set

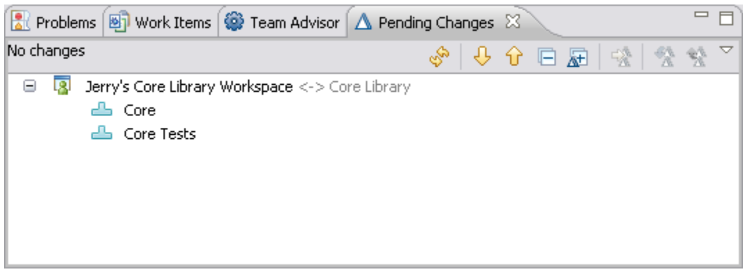
- Steve on Core Library <-> Core Library
- Steve on Documentation <-> Documentation
  - Documentation
    - Incoming
      - Geoff Clemm 12: Lion Documentation - <No Comment>
        - net.jazz.uws.squawk.doc/squawkers
          - lion.txt
        - 12: Lion Documentation

Getting Teammates' Delivered Work



- An incoming change-set is
  - In the change history of the stream, but
  - Not in the change history of your repository workspace
- Accept adds the change-set to your repository workspace's change history





Problems | Work Items | Team Advisor | Pending Changes

No changes

- Jerry's Core Library Workspace <-> Core Library
  - Core
  - Core Tests

Lab #4 Scenario

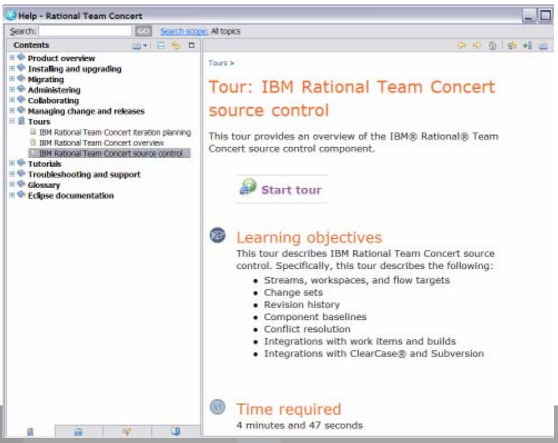
- You have been tasked with contributing your own squawker class along with its documentation and, optionally, its test case

Lab #4 Overview

- You will spend a little time understanding the key concepts of the SCM system in Jazz
- You will create your own squawker, basic documentation and optionally its test case against the work items you created in Module 2 *Planning Your Work*
- You will deliver this work so that other people can use it
- Finally, you will bring in changes from other members of your team so your code is up-to-date with everyone else

Lab #4 Concepts Learned

- Jazz Source Control provides private **repository workspaces** to track and back up your changes before you share them with the team using a **stream** for integration
- A **change set** is the fundamental unit of change and collaboration in your team environment
- A change set can be associated with a **work item**, which can then be **delivered** as a unit and provides traceability and transparency to the development lifecycle
- The **Pending Changes** view is central to these operations by enabling real time updates and efficiency
- Video overview available from the online Help under **Tours**


Help - Rational Team Concert
Search:
GO
Search scope: All topics
Contents
Product overview
Installing and upgrading
Migrating
Administering
Collaborating
Managing change and releases
Tours
IBM Rational Team Concert iteration planning
IBM Rational Team Concert overview
IBM Rational Team Concert source control
Tutorials
Troubleshooting and support
Glossary
Eclipse documentation
Tours >
Start tour

Tour: IBM Rational Team Concert source control

This tour provides an overview of the IBM® Rational® Team Concert source control component.

Learning objectives

This tour describes IBM Rational Team Concert source control. Specifically, this tour describes the following:

- Streams, workspaces, and flow targets
- Change sets
- Revision history
- Component baselines
- Conflict resolution
- Integrations with work items and builds
- Integrations with ClearCase® and Subversion

Time required

4 minutes and 47 seconds

Questions

IBM
Software
Group

Remembering Well Known SCM Configurations

An IBM Proof of Technology

Objectives

- Understand how **Component Baselines** and **Workspace Snapshots** can be used
- Create new repository workspace from a snapshot for maintenance purposes
- Utilize the Pending Changes view to increase productivity

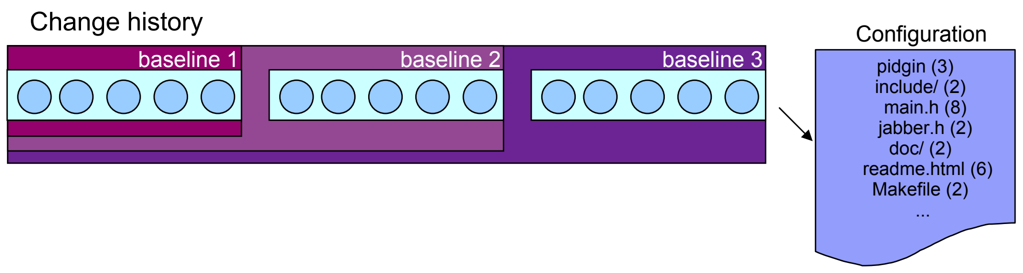
What About These Questions?

- How do I find a known good configuration of a component?
- How about a known good configuration of an entire stream?
- Hey, exactly what was in that milestone build a year ago?
- That is, what about fixed configurations that do not change anymore?

- Use baselines and snapshots…

A Baseline

- Is an immutable copy of a component's configuration
  - At a particular point in time, and
  - There can be multiple baselines of a component
- Serves as a fixed point of reference
  - For initializing streams and repository workspaces
  - For sharing source with people or processes
- Can be easily compared
  - With the current state of a stream or repository workspace
  - With another baseline



A Snapshot

- Is a collection of one baseline per component in a repository workspace or stream
  - Captures an important repository workspace configuration for later recreation
  - There can be multiple snapshots of a repository workspace or stream
  - Provides traceability to historical artifacts
- Like baselines, snapshots are used for sharing and collaborating with team members
  - Create a repository workspace or stream
  - Update the contents of a repository workspace
  - Recreate a prior build via a build created snapshot

Answers to those tough questions

- How do I find a known good configuration of a component?
  - Use a baseline!
- How about a known good configuration of an entire stream?
  - Use a snapshot!
- Hey, exactly what was in that milestone build a year ago?
  - Use a snapshot or baseline!

Lab #5 Scenario

- You have contributed your own squawker class along with documentation and delivered your work
- Your teammates have been creating and delivering their own squawkers and documentation, which you have accepted
- These changes need to be captured so that they can be used for further work or returned to at some point in the future if necessary

Lab #5 Overview

- The instructor will play the role of Team Lead, creating baselines and snapshots to capture all the work completed by the team
- You will then explore the new baselines and snapshots by querying their contents.
- You will revert a component in your workspace to a previous baseline version with the replace operation which provides a convenient way to reconfigure your workspace.

Lab #5 Concepts Learned

- **Baseline** and **snapshot** artifacts increase traceability and enable collaboration among teams and team members
- **Baselines** are an efficient means to mark artifacts within a single component for later reference
- **Snapshots** are an efficient means to mark artifacts across a set of related components for later reference
- It is easy to create a new repository workspace or stream from a snapshot. This is useful for maintenance purposes, fixing builds or forking the code
- The **Pending Changes** view is central to these operations by providing an easy to use interface to review changes and appropriately update your workspace

IBM
Software
Group

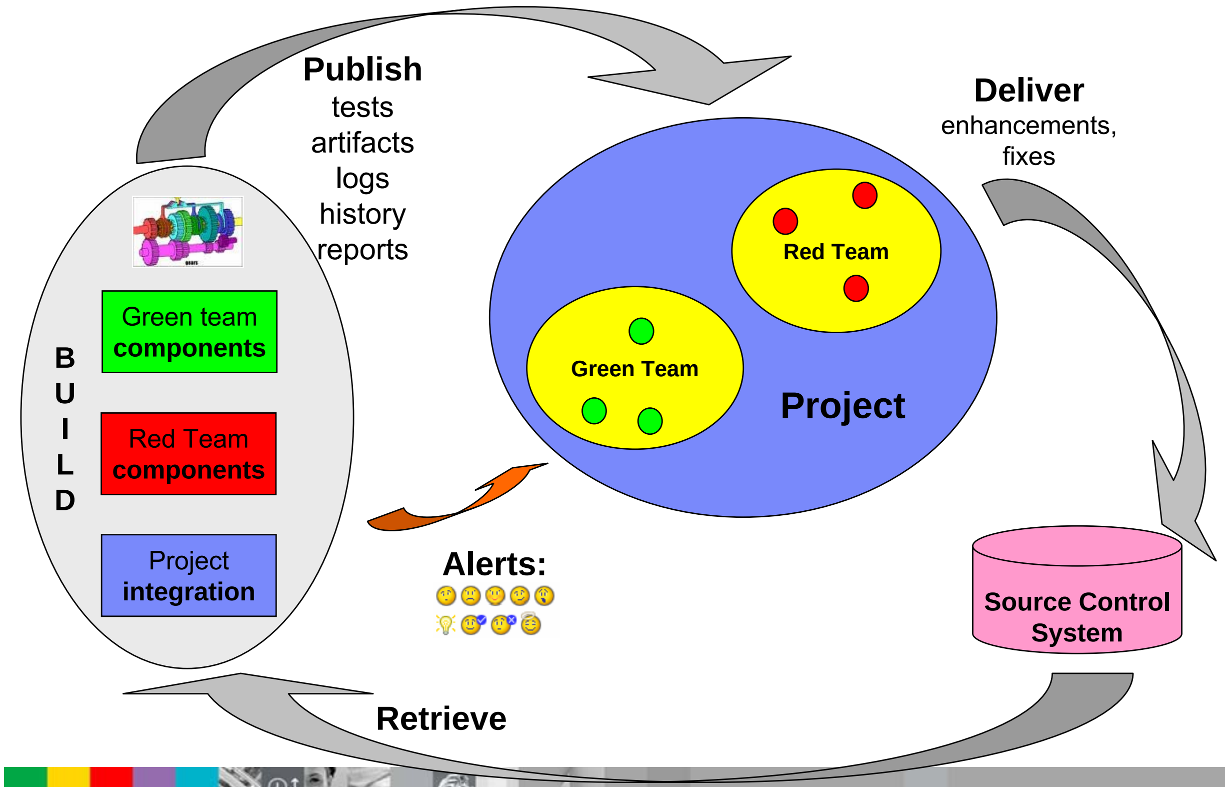
User’s View of Build

An IBM Proof of Technology

Objectives

- Understand the build functionality of Rational Team Concert
- Understand the flexibility of the build process and how it enables collaboration and teaming
- Observe policies and processes that relate to consistency and repeatability
- Explore Build Results and observe traceability to artifacts
- Perform a build or a rebuild

Build in the World of Agile Team Development

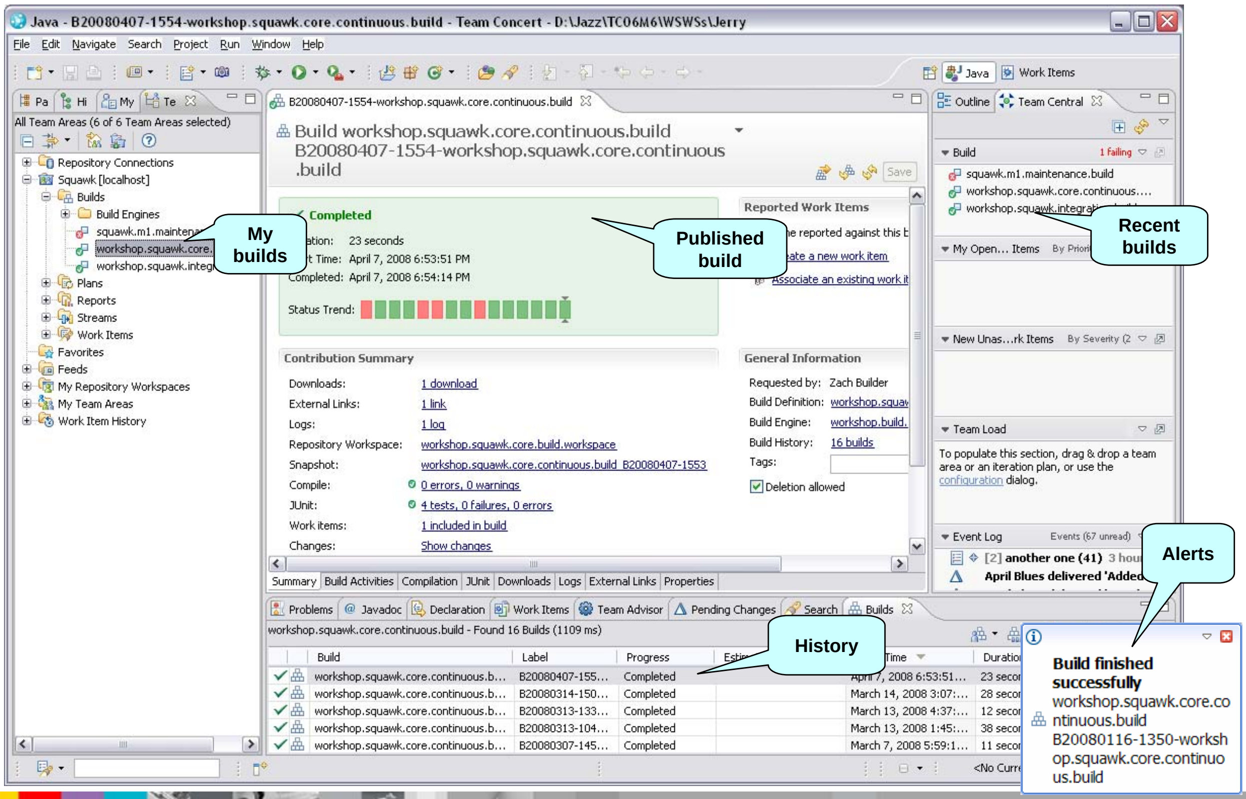


Rational Team Concert Build

- Is an integral part of the project infrastructure
  - Consistent, repeatable process throughout the project
- Brings awareness of build progress and results to developers
  - Easy sharing of information
- Links build results to related Jazz artifacts
  - Integrated experience, traceability and tracking “baked in“
- Allow developers to have a private build area
  - Build and test code before delivering to the main branch
- Accomodates existing build technologies (Ant, CruiseControl , Build Forge, Maven, …)
  - Leverages technology that fits your project best

Build is very visible to the user*

* Not currently available in Visual Studio



Java - B20080407-1554-workshop.squawk.core.continuous.build - Team Concert - D:\Jazz\TC06M6\WSWSs\Jerry

Build workshop.squawk.core.continuous.build B20080407-1554-workshop.squawk.core.continuous.build

Completed
Duration: 23 seconds
Start Time: April 7, 2008 6:53:51 PM
Completed: April 7, 2008 6:54:14 PM
Status Trend:

Contribution Summary

Downloads:	1 download
External Links:	1 link
Logs:	1 log
Repository Workspace:	workshop.squawk.core.build.workspace
Snapshot:	workshop.squawk.core.continuous.build_B20080407-1553
Compile:	0 errors, 0 warnings
JUnit:	4 tests, 0 failures, 0 errors
Work items:	1 included in build
Changes:	Show changes

General Information

Requested by:	Zach Builder
Build Definition:	workshop.squaw
Build Engine:	workshop.build.
Build History:	16 builds
Tags:	

Deletion allowed

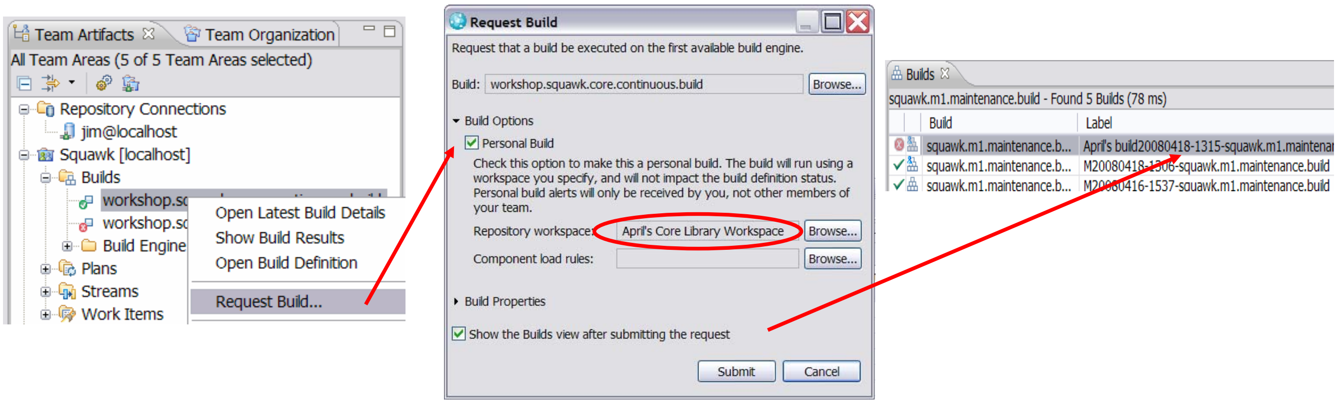
workshop.squawk.core.continuous.build - Found 16 Builds (1109 ms)

Build	Label	Progress	Start Time	Duration
workshop.squawk.core.continuous.b...	B20080407-155...	Completed	April 7, 2008 6:53:51...	23 secon
workshop.squawk.core.continuous.b...	B20080314-150...	Completed	March 14, 2008 3:07:...	28 secon
workshop.squawk.core.continuous.b...	B20080313-133...	Completed	March 13, 2008 4:37:...	12 secon
workshop.squawk.core.continuous.b...	B20080313-104...	Completed	March 13, 2008 1:45:...	38 secon
workshop.squawk.core.continuous.b...	B20080307-145...	Completed	March 7, 2008 5:59:1...	11 secon

Build finished successfully
workshop.squawk.core.continuous.build
B20080116-1350-workshop.squawk.core.continuous.build

Personal builds

- Builds normally run from a dedicated repository workspace.
- Personal Builds
  - run from your repository workspace.
  - allow you to build your changes before delivering them to the stream.
  - provide you with some assurance that your changes will not disrupt the team builds when you deliver them.



Request Build

Request that a build be executed on the first available build engine.

Build: workshop.squawk.core.continuous.build

Build Options

Personal Build [✓]

Check this option to make this a personal build. The build will run using a workspace you specify, and will not impact the build definition status. Personal build alerts will only be received by you, not other members of your team.

Repository workspace: April's Core Library Workspace

Component load rules:

Build Properties

Show the Builds view after submitting the request [✓]

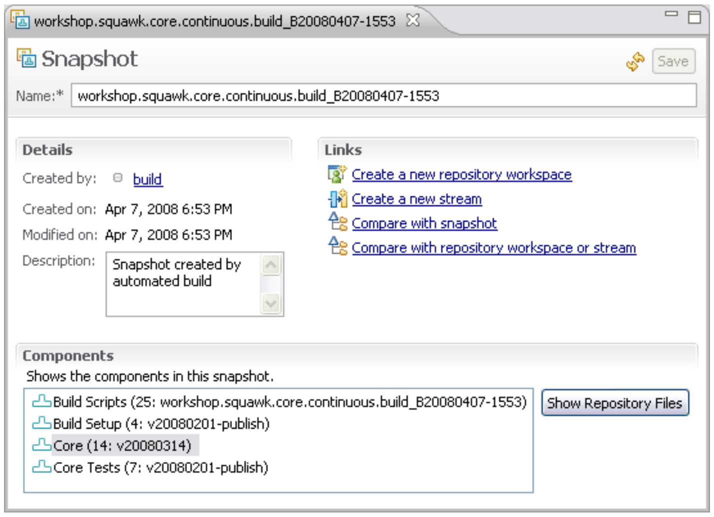
Submit Cancel

squawk.m1.maintenance.build - Found 5 Builds (78 ms)

Build	Label
squawk.m1.maintenance.b...	April's build20080418-1315-squawk.m1.maintenan
squawk.m1.maintenance.b...	M20080418-1306-squawk.m1.maintenance.build
squawk.m1.maintenance.b...	M20080416-1537-squawk.m1.maintenance.build

Builds and Snapshots

- A build can request a snapshot
  - If there are any changes in a component since the last build
    - A new baseline is created with the same name as the snapshot name
  - Convenient for reproducing build problems


workshop.squawk.core.continuous.build_B20080407-1553

Snapshot

Name:* workshop.squawk.core.continuous.build_B20080407-1553

Details

Created by: build

Created on: Apr 7, 2008 6:53 PM

Modified on: Apr 7, 2008 6:53 PM

Description: Snapshot created by automated build

Links

Create a new repository workspace

Create a new stream

Compare with snapshot

Compare with repository workspace or stream

Components

Shows the components in this snapshot.

Build Scripts (25: workshop.squawk.core.continuous.build_B20080407-1553)

Build Setup (4: v20080201-publish)

Core (14: v20080314)

Core Tests (7: v20080201-publish)

Lab #6 Scenario

- You have recently joined your company's exciting new project called Squawk.
- By now you have
  - planned and tracked your work,
  - developed a new squawker,
  - and created baselines and snapshots.
- You are now ready to build your application with help of the Team Concert Build Engine.

Lab #6 Overview

- The instructor will then demonstrate how a build engineer, team lead or other appropriate role, can request a build for use by the project team
- You will explore the results of existing builds
- You will request a private build to ensure that your changes won't break the build

Lab #6 Concepts Learned

- In this module you explored the build capabilities of Rational Team Concert. You have explored existing builds and learned how to request new builds or rebuilds.
- Treating the build as an integral part of the project infrastructure makes it easy to keep processes and policies consistent and repeatable.
- Every team member has access to build data which promotes communication and collaboration among the contributors – on local or remote sites.
- Linking build results directly to Jazz artifacts provides a high level of traceability.
- Using existing build technologies (Ant, CruiseControl , Build Forge, Maven, …) makes it easy to adapt to needs of different projects.

Questions

IBM
Software
Group

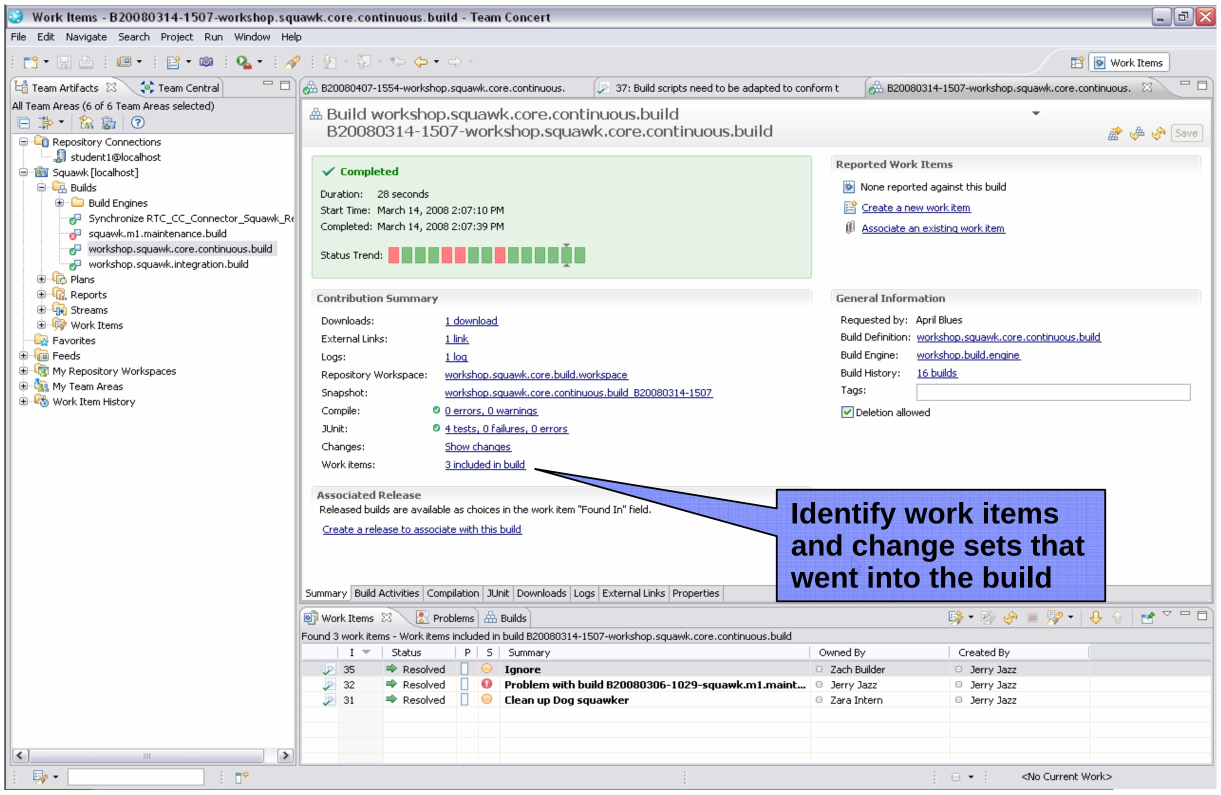
Exploring Changes and Traceability

An IBM Proof of Technology

Objectives

- This lab will demonstrate how information is linked within Rational Team Concert to establish traceability.
- Determine what work items and files are included in a build
- Determine change sets that are included in a build
- Determine who changes files, when and why
- Compare versions of a file
- Observe specific changes to files

Builds



Work Items - B20080314-1507-workshop.squawk.core.continuous.build - Team Concert

Team Artifacts | Team Central

All Team Areas (6 of 6 Team Areas selected)

- Repository Connections
  - student1@localhost
- Squawk [localhost]
  - Builds
    - Build Engines
    - Synchronize RTC_CC_Connector_Squawk_Re
    - squawk.m1.maintenance.build
    - workshop.squawk.core.continuous.build
    - workshop.squawk.integration.build
  - Plans
  - Reports
  - Streams
  - Work Items
  - Favorites
- Feeds
- My Repository Workspaces
- My Team Areas
- Work Item History

Build workshop.squawk.core.continuous.build
B20080314-1507-workshop.squawk.core.continuous.build

Completed

Duration: 28 seconds
Start Time: March 14, 2008 2:07:10 PM
Completed: March 14, 2008 2:07:39 PM

Status Trend: 🟥🟩🟩🟩🟥🟥🟩🟩🟥🟩🟩🟩🟩🟩🟩

Reported Work Items

None reported against this build
Create a new work item
Associate an existing work item

Contribution Summary

Downloads:	1 download
External Links:	1 link
Logs:	1 log
Repository Workspace:	workshop.squawk.core.build.workspace
Snapshot:	workshop.squawk.core.continuous.build_B20080314-1507
Compile:	0 errors, 0 warnings
JUnit:	4 tests, 0 failures, 0 errors
Changes:	Show changes
Work items:	3 included in build

General Information

Requested by:	April Blues
Build Definition:	workshop.squawk.core.continuous.build
Build Engine:	workshop.build.engine
Build History:	16 builds
Tags:	

☑ Deletion allowed

Associated Release

Released builds are available as choices in the work item "Found In" field.
Create a release to associate with this build

Summary | Build Activities | Compilation | JUnit | Downloads | Logs | External Links | Properties

Work Items | Problems | Builds

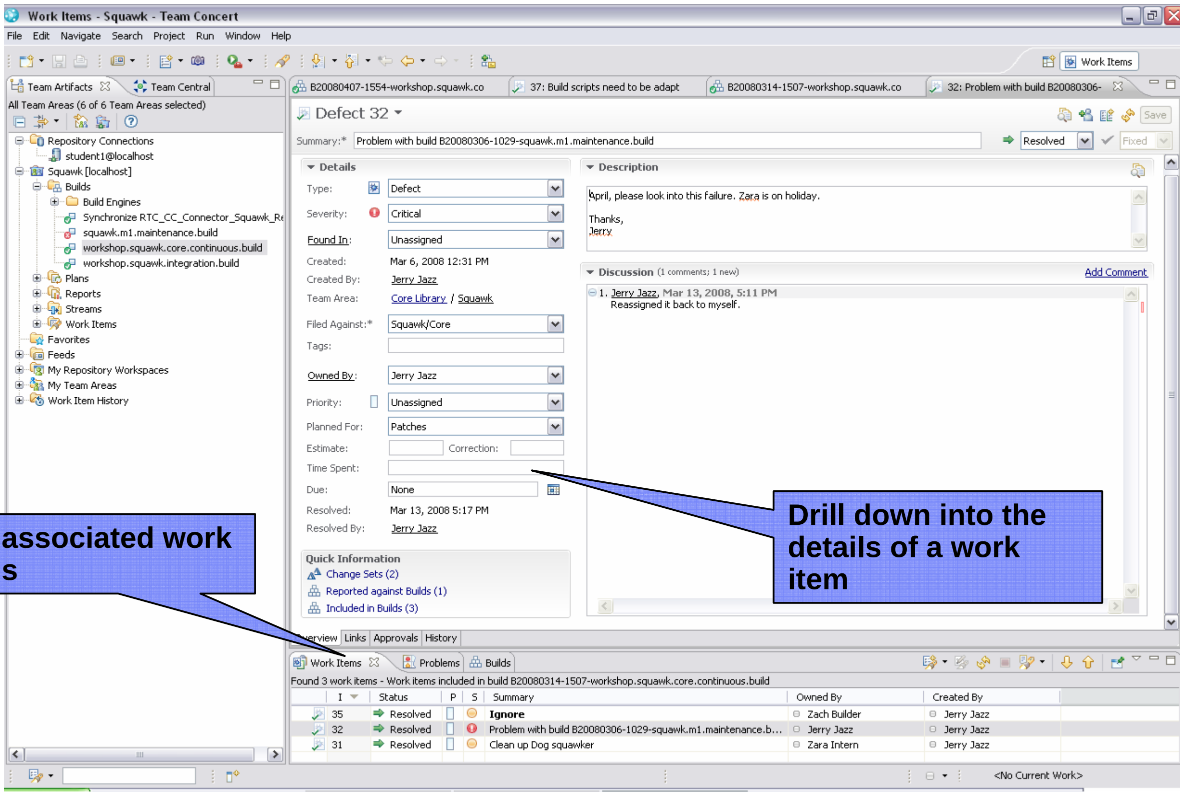
Found 3 work items - Work items included in build B20080314-1507-workshop.squawk.core.continuous.build

I	Status	P	S	Summary	Owned By	Created By
35	Resolved			Ignore	Zach Builder	Jerry Jazz
32	Resolved			Problem with build B20080306-1029-squawk.m1.maint...	Jerry Jazz	Jerry Jazz
31	Resolved			Clean up Dog squawker	Zara Intern	Jerry Jazz

<No Current Work>

Identify work items and change sets that went into the build

Work Items



Work Items - Squawk - Team Concert

Defect 32

Summary:* Problem with build B20080306-1029-squawk.m1.maintenance.build — Resolved — Fixed

Details

Type:	Defect
Severity:	Critical
Found In:	Unassigned
Created:	Mar 6, 2008 12:31 PM
Created By:	Jerry Jazz
Team Area:	Core Library / Squawk
Filed Against:*	Squawk/Core
Tags:	
Owned By:	Jerry Jazz
Priority:	Unassigned
Planned For:	Patches
Estimate:	Correction:
Time Spent:	
Due:	None
Resolved:	Mar 13, 2008 5:17 PM
Resolved By:	Jerry Jazz

Quick Information

Change Sets (2)
Reported against Builds (1)
Included in Builds (3)

Description

April, please look into this failure. Zara is on holiday.

Thanks,
Jerry

Discussion (1 comments; 1 new) — Add Comment

1. Jerry Jazz, Mar 13, 2008, 5:11 PM
Reassigned it back to myself.

Overview | Links | Approvals | History

Work Items | Problems | Builds

Found 3 work items - Work items included in build B20080314-1507-workshop.squawk.core.continuous.build

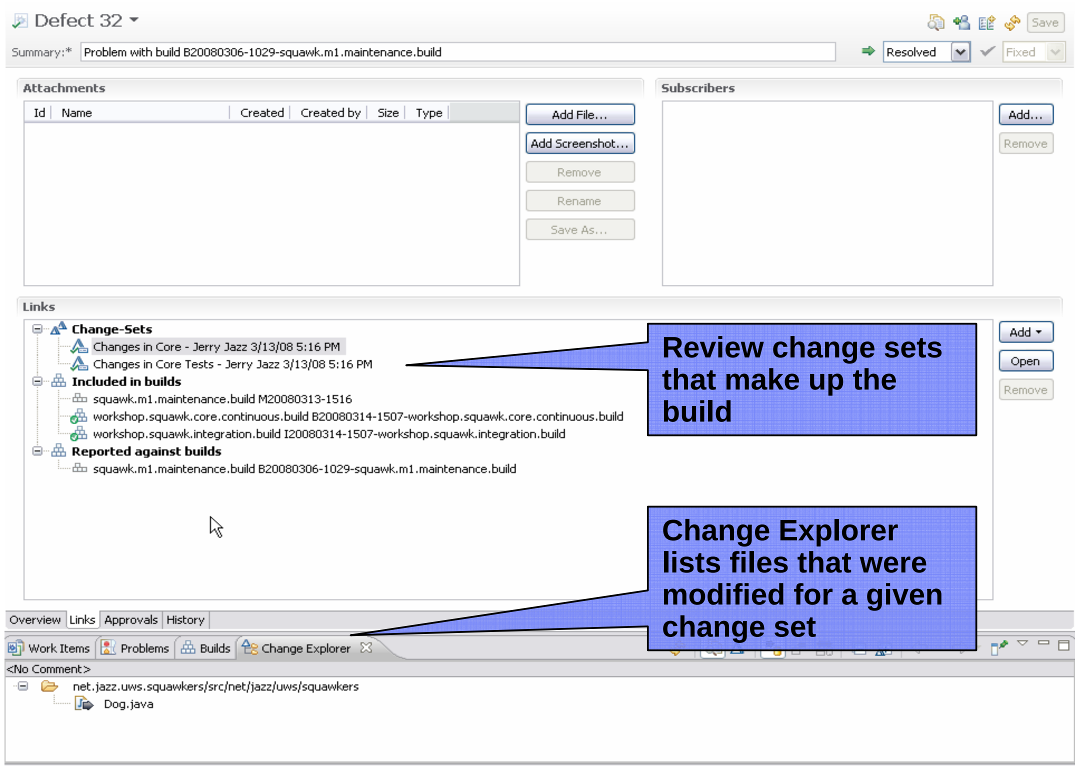
I	Status	P	S	Summary	Owned By	Created By
35	Resolved			Ignore	Zach Builder	Jerry Jazz
32	Resolved			Problem with build B20080306-1029-squawk.m1.maintenance.b...	Jerry Jazz	Jerry Jazz
31	Resolved			Clean up Dog squawker	Zara Intern	Jerry Jazz

<No Current Work>

List associated work items

Drill down into the details of a work item

Change Sets



Defect 32

Summary:* Problem with build B20080306-1029-squawk.m1.maintenance.build

Resolved · Fixed · Save

Attachments

Id	Name	Created	Created by	Size	Type

Add File... · Add Screenshot... · Remove · Rename · Save As...

Subscribers

Add... · Remove

Links

- Change-Sets
  - Changes in Core - Jerry Jazz 3/13/08 5:16 PM
  - Changes in Core Tests - Jerry Jazz 3/13/08 5:16 PM
- Included in builds
  - squawk.m1.maintenance.build M20080313-1516
  - workshop.squawk.core.continuous.build B20080314-1507-workshop.squawk.core.continuous.build
  - workshop.squawk.integration.build I20080314-1507-workshop.squawk.integration.build
- Reported against builds
  - squawk.m1.maintenance.build B20080306-1029-squawk.m1.maintenance.build

Add · Open · Remove

Overview | Links | Approvals | History

Work Items | Problems | Builds | Change Explorer

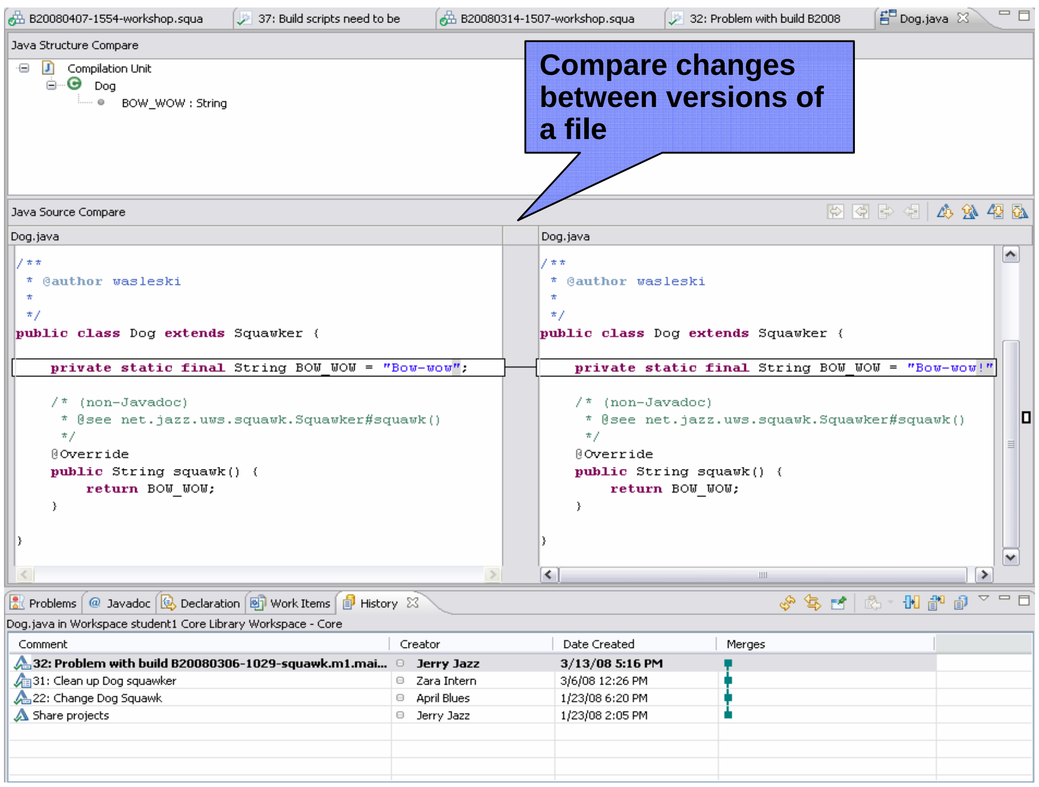
<No Comment>

- net.jazz.uws.squawkers/src/net/jazz/uws/squawkers
  - Dog.java

Review change sets that make up the build

Change Explorer lists files that were modified for a given change set

Compare Changes



B20080407-1554-workshop.squa | 37: Build scripts need to be | B20080314-1507-workshop.squa | 32: Problem with build B2008 | Dog.java

Java Structure Compare

- Compilation Unit
  - Dog
    - BOW_WOW : String

Java Source Compare

Dog.java

```
/**
 * @author wasleski
 *
 */
public class Dog extends Squawker {

    private static final String BOW_WOW = "Bow-wow";

    /* (non-Javadoc)
     * @see net.jazz.uws.squawk.Squawker#squawk()
     */
    @Override
    public String squawk() {
        return BOW_WOW;
    }

}
```

Dog.java

```
/**
 * @author wasleski
 *
 */
public class Dog extends Squawker {

    private static final String BOW_WOW = "Bow-wow!"

    /* (non-Javadoc)
     * @see net.jazz.uws.squawk.Squawker#squawk()
     */
    @Override
    public String squawk() {
        return BOW_WOW;
    }

}
```

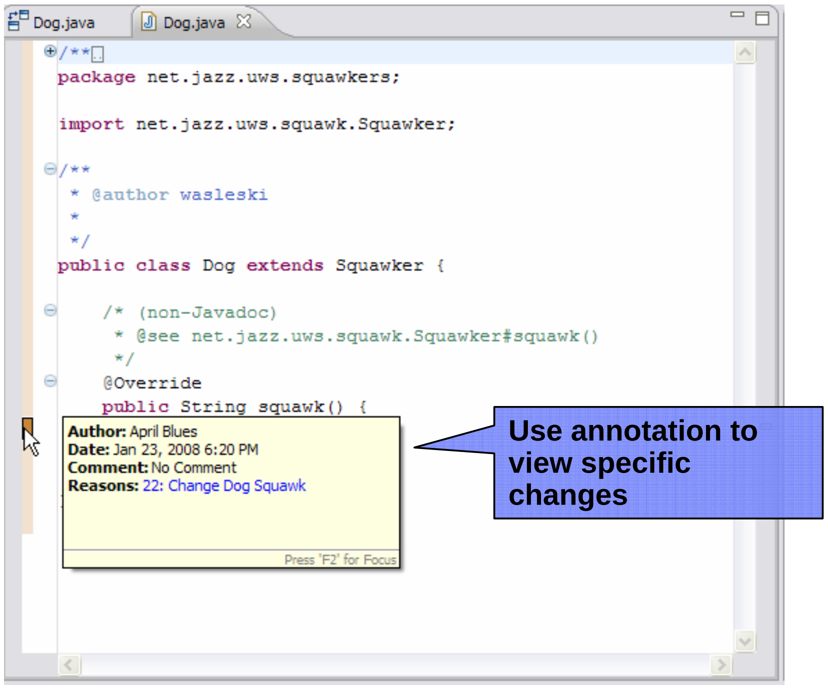
Problems | Javadoc | Declaration | Work Items | History

Dog.java in Workspace student1 Core Library Workspace - Core

Comment	Creator	Date Created	Merges
32: Problem with build B20080306-1029-squawk.m1.mai...	**Jerry Jazz**	**3/13/08 5:16 PM**	
31: Clean up Dog squawker	Zara Intern	3/6/08 12:26 PM	
22: Change Dog Squawk	April Blues	1/23/08 6:20 PM	
Share projects	Jerry Jazz	1/23/08 2:05 PM	

Compare changes between versions of a file

Visualize Change History



```
/**
package net.jazz.uws.squawkers;

import net.jazz.uws.squawk.Squawker;

/**
 * @author wasleski
 *
 */
public class Dog extends Squawker {

    /* (non-Javadoc)
     * @see net.jazz.uws.squawk.Squawker#squawk()
     */
    @Override
    public String squawk() {
```

Author: April Blues
Date: Jan 23, 2008 6:20 PM
Comment: No Comment
Reasons: 22: Change Dog Squawk

*** Not currently available in Visual Studio**

Lab #7 Scenario

- You have completed some builds for the Squawk project and are now ready to look at how Rational Team Concert links the software artifacts that make up the builds.
- You will investigate the build artifacts to see how Rational Team Concert automatically manages traceability.
- You will review the change sets (work items and associated changes under source control) that make up the build and explore the change history.

Lab #7 Objectives

- You will experience how information is linked within Rational Team Concert.
- As a team member you will learn how traceability helps answer questions such as
  - What work items went into a build?
  - What changes were made for a work item?
  - What build did a work item get delivered in?
  - Who changed a file, and why?
  - What are the specific changes made on a resource?
  - How to visualize the change history for a resource?

Lab #7 Concepts Learned

- Rational Team Concert maintains full traceability for changes contained in a build
- Work items maintain a record of the changes made to resources maintaining consistency and transparency in the project
- Changes are collected and managed as Change Sets and available for reporting purposes and analysis
- Users can drill down into the detailed change history of every artifact, enhancing collaboration and quality

Questions

IBM
Software
Group

Endgame and a Tightened Process

An IBM Proof of Technology

Objectives

- Understand how process is defined in Jazz and implemented by Rational Team Concert
- Understand how roles can be used to control process workflow

Motivation for the Team Process Component

- Generally all software teams have some sort of process
  - May be formal, informal…
- Successful teams…
  - Believe their software process helps produce quality software
  - Own their process **and accept accountability for it**
  - Continually adapt their process to changing needs
- However, success depends on…
  - Common understanding by all team members
  - Consistent execution
- Many times…
  - Process relies on *documents (or word of mouth) for understanding and human memory for execution* **and is otherwise very manual**
  - Leads to inconsistent or erroneous execution

What if your tools understood how your team works?

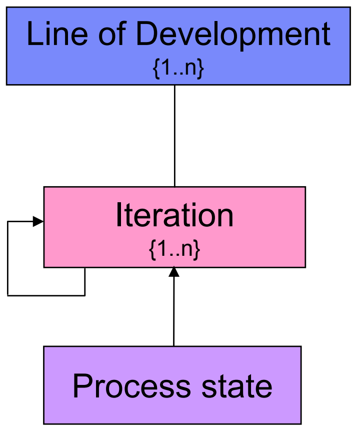
In a Basic Process Model…

- Teams work on projects
- Each project follows a process
- Each team is unique and thus can work differently
- Work inside the scope of a team follows the team's process
- Cross-team work follows the process of the broader team
- Team members play roles defined by the process
- Process manifests itself through artifacts types, operations manipulating the artifacts, and artifact change events.

Jazz Process Support

- Support different degrees of flexibility and formalism
- Allows for **predefined** processes
- Allows for **emerging** processes
- Allows for **variations**
- Allows for **exceptions**
- Allows for process **consolidation**
- Allows for process **evolution** in general
- Allows for **extensions**
- Put knowledgeable human in the center
- Comprises runtime, authoring, and inspection support

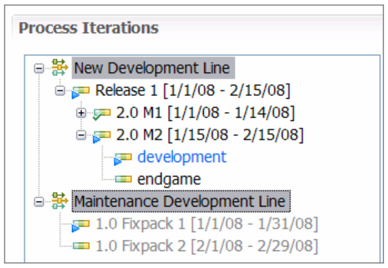
Project Area Iteration Structure and Terminology



Line of Development is an element of a project area that owns a set of deliverables and its production schedule (**maintenance, new release development**).
- Often represents parallel development
- A team area is associated with a development line

Iteration represents some project work interval
- Any depth of nested iterations
- Process specification in any iteration
- May contain start and end dates

Process state is defined as the current iteration in a development line
- Indicated by the blue arrow


Process Iterations

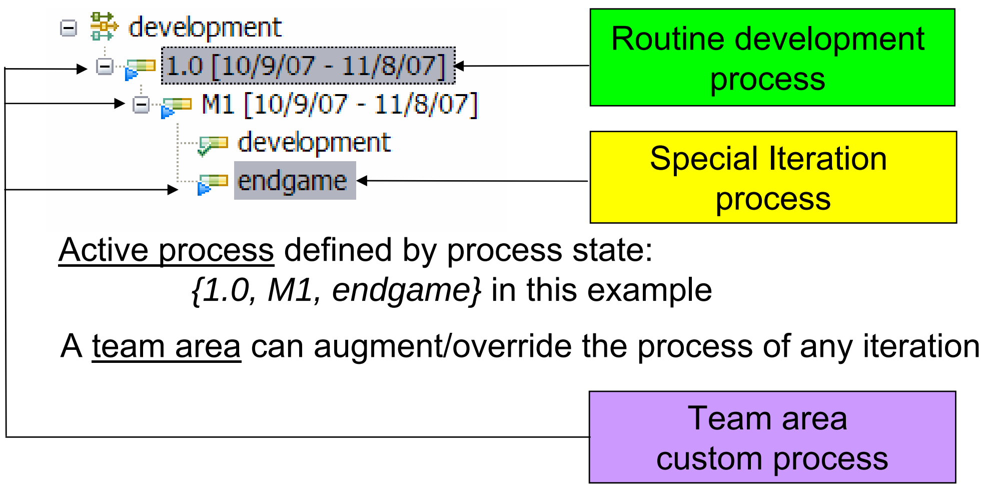
New Development Line
Release 1 [1/1/08 - 2/15/08]
2.0 M1 [1/1/08 - 1/14/08]
2.0 M2 [1/15/08 - 2/15/08]
development
endgame
Maintenance Development Line
1.0 Fixpack 1 [1/1/08 - 1/31/08]
1.0 Fixpack 2 [2/1/08 - 2/29/08]

Example:
Main Development Line process state:
2.0 M2

Maintenance Development Line process state:
1.0 Fix pack 1

Process is Defined in One or More Iterations

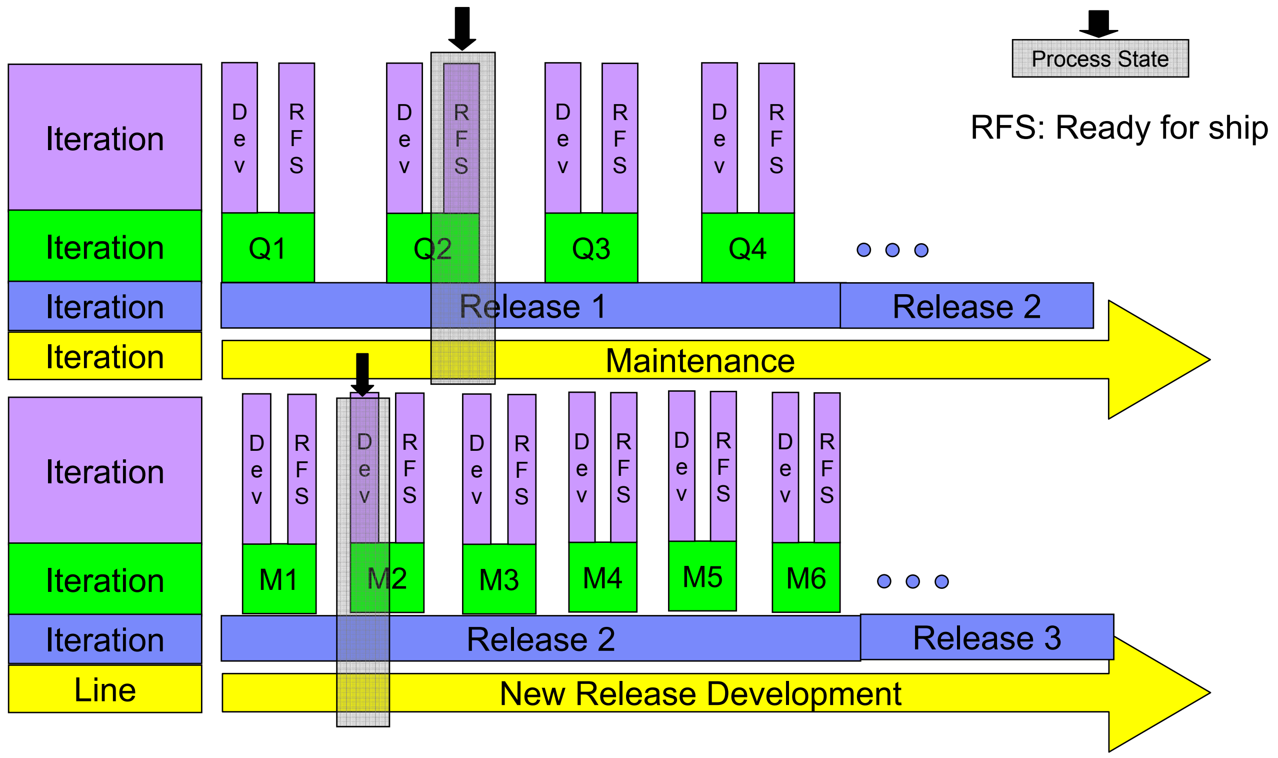
- Specified as a set of component operation rules
- Rules are assigned to user roles (default, contributor, team lead…)
- You can have the general process defined for the project
- Override/augment the general process in planned iterations



Active process defined by process state:
{1.0, M1, endgame} in this example

A team area can augment/override the process of any iteration

Project Delivery Plan in Jazz Terms



Lab #8 Scenario

- As you approach your final milestone, you have the chance to alter the process for the iteration so that your rules get stricter.
- For example, you might insist that all tests run to completion and without error before you are allowed to deliver any changes.

Lab #8 Overview

- In this lab, the team will move to the *Endgame* iteration of milestone *1.0 M2* and will experience a change in the process
  - In the *M2 Endgame* iteration the *Core Library* team has customized the process such that changes can be delivered only if the team lead has approved the work item associated with the delivery
- The instructor will move the project to the *1.0 M2 Endgame* iteration
- As a user with *contributor* role on the *Core Library* teams, you will make a change to the squawker class
- During the delivery, user will notice the change to the squawker class for the *Core Library* will not complete because the Work Item associated with the delivery does not have the approval of the team lead
- The instructor (as *team lead)* will approve the Work Item
- The user will now be able to deliver the Work Item
- The instructor will move the project back to the *1.0 M2 Development* iteration

Lab #8 Concepts Learned

- Jazz processes capture the idea and the notion of choreographies of collaboration
- With Jazz collaboration rules are your friend not something you have to fight. Keep your processes as concrete as possible and as strict as necessary
- Process sandboxes allow ‘good things’ to happen on all levels
- Process support in Jazz is an ongoing endeavor

IBM
Software
Group

Taking Control of Your Project

An IBM Proof of Technology

Objectives

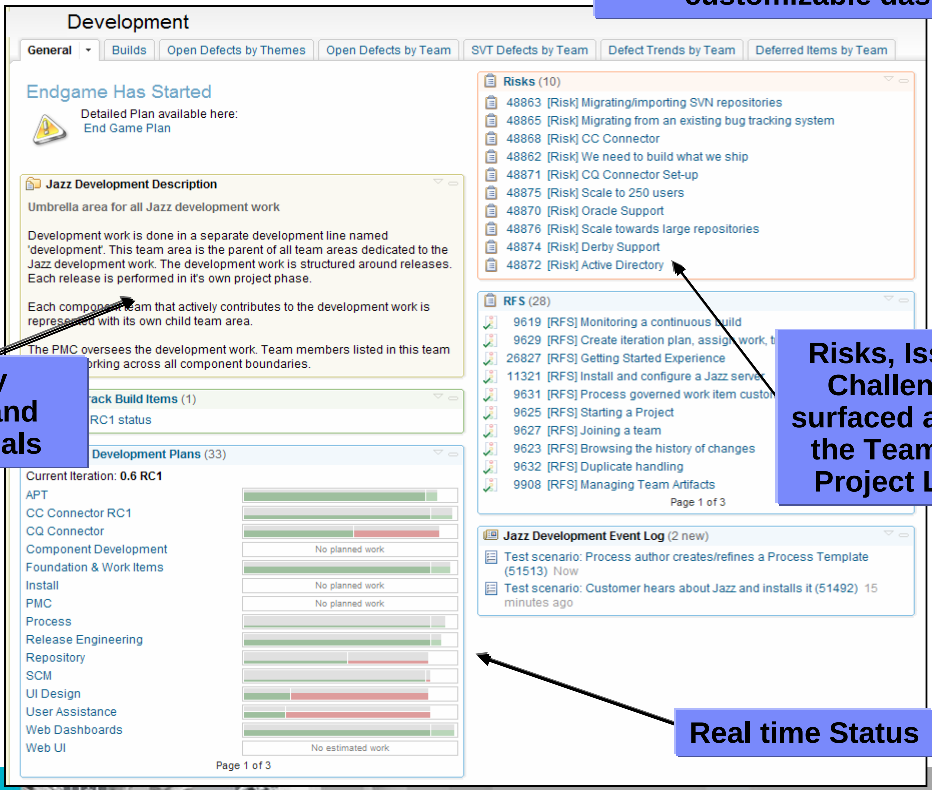
- Learn about the Jazz Dashboards and Reports and how these powerful capabilities can assist your team to track project status and make informed-based decisions to keep it on track.

Dashboards

- Dashboards are a Web UI component intended to provide information about the project status at a glance.
- It provides for easy drill down to get more complete information.
- Dashboards are only available in the Rational Team Concert Standard edition.

Dashboards

Transparency and control via customizable dashboards


Development

General | Builds | Open Defects by Themes | Open Defects by Team | SVT Defects by Team | Defect Trends by Team | Deferred Items by Team

Endgame Has Started

Detailed Plan available here:
End Game Plan

Jazz Development Description

Umbrella area for all Jazz development work

Development work is done in a separate development line named 'development'. This team area is the parent of all team areas dedicated to the Jazz development work. The development work is structured around releases. Each release is performed in it's own project phase.

Each component team that actively contributes to the development work is represented with its own child team area.

The PMC oversees the development work. Team members listed in this team
orking across all component boundaries.

rack Build Items (1)
RC1 status

Development Plans (33)
Current Iteration: **0.6 RC1**

APT	
CC Connector RC1	
CQ Connector	
Component Development	No planned work
Foundation & Work Items	
Install	No planned work
PMC	No planned work
Process	
Release Engineering	
Repository	
SCM	
UI Design	
User Assistance	
Web Dashboards	
Web UI	No estimated work

Page 1 of 3

Risks (10)
48863 [Risk] Migrating/importing SVN repositories
48865 [Risk] Migrating from an existing bug tracking system
48868 [Risk] CC Connector
48862 [Risk] We need to build what we ship
48871 [Risk] CQ Connector Set-up
48875 [Risk] Scale to 250 users
48870 [Risk] Oracle Support
48876 [Risk] Scale towards large repositories
48874 [Risk] Derby Support
48872 [Risk] Active Directory

RFS (28)
9619 [RFS] Monitoring a continuous build
9629 [RFS] Create iteration plan, assign work, t
26827 [RFS] Getting Started Experience
11321 [RFS] Install and configure a Jazz server
9631 [RFS] Process governed work item custo
9625 [RFS] Starting a Project
9627 [RFS] Joining a team
9623 [RFS] Browsing the history of changes
9632 [RFS] Duplicate handling
9908 [RFS] Managing Team Artifacts
Page 1 of 3

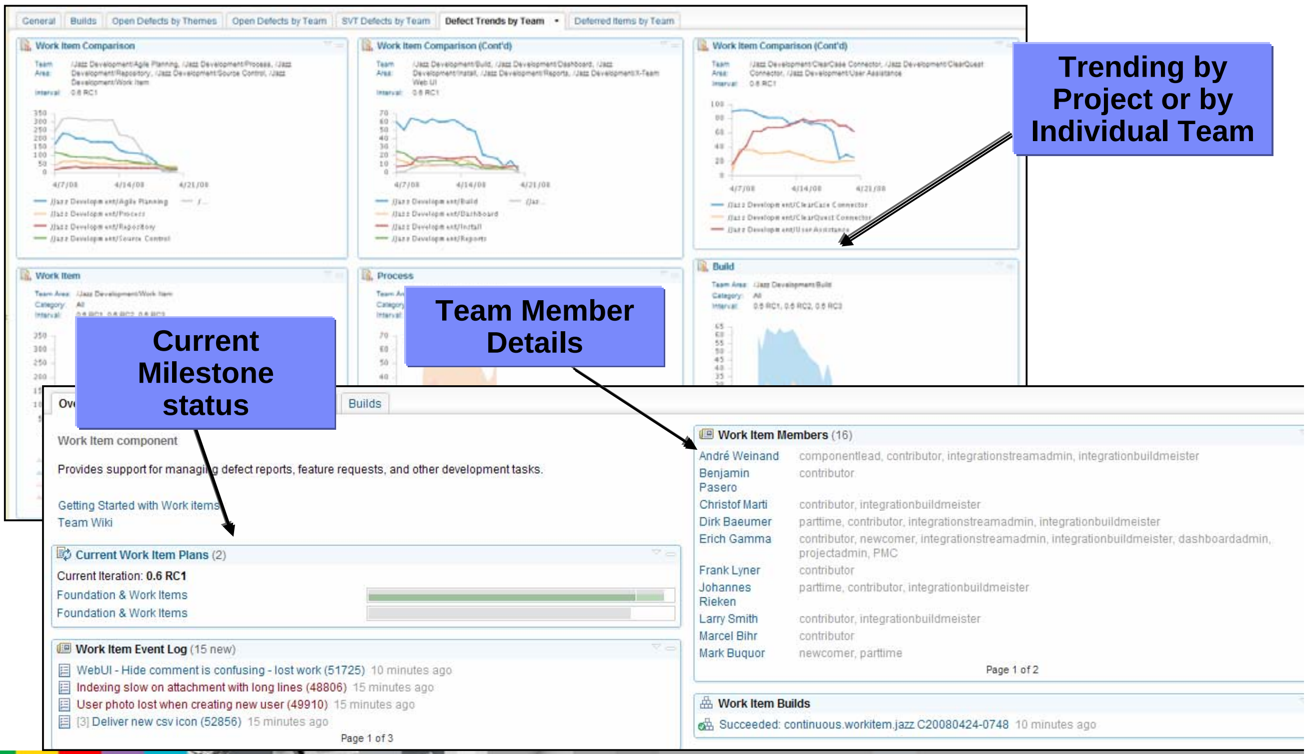
Jazz Development Event Log (2 new)
Test scenario: Process author creates/refines a Process Template (51513) Now
Test scenario: Customer hears about Jazz and installs it (51492) 15 minutes ago

Clearly understand Team Goals

Risks, Issues, Challenges surfaced at both the Team and Project Level

Real time Status

Dashboards and Reporting



General | Builds | Open Defects by Themes | Open Defects by Team | SVT Defects by Team | Defect Trends by Team | Deferred Items by Team

Work Item Comparison | Work Item Comparison (Cont'd) | Work Item Comparison (Cont'd) | Work Item | Process | Build

Builds

Work Item component

Provides support for managing defect reports, feature requests, and other development tasks.

Getting Started with Work Items
Team Wiki

Current Work Item Plans (2)
Current Iteration: **0.6 RC1**
Foundation & Work Items
Foundation & Work Items

Work Item Event Log (15 new)
WebUI - Hide comment is confusing - lost work (51725) 10 minutes ago
Indexing slow on attachment with long lines (48806) 15 minutes ago
User photo lost when creating new user (49910) 15 minutes ago
[3] Deliver new csv icon (52856) 15 minutes ago
Page 1 of 3

Work Item Members (16)

André Weinand	componentlead, contributor, integrationstreamadmin, integrationbuildmeister
Benjamin Pasero	contributor
Christof Marti	contributor, integrationbuildmeister
Dirk Baeumer	parttime, contributor, integrationstreamadmin, integrationbuildmeister
Erich Gamma	contributor, newcomer, integrationstreamadmin, integrationbuildmeister, dashboardadmin, projectadmin, PMC
Frank Lyner	contributor
Johannes Rieken	parttime, contributor, integrationbuildmeister
Larry Smith	contributor, integrationbuildmeister
Marcel Bihr	contributor
Mark Buquor	newcomer, parttime

Page 1 of 2

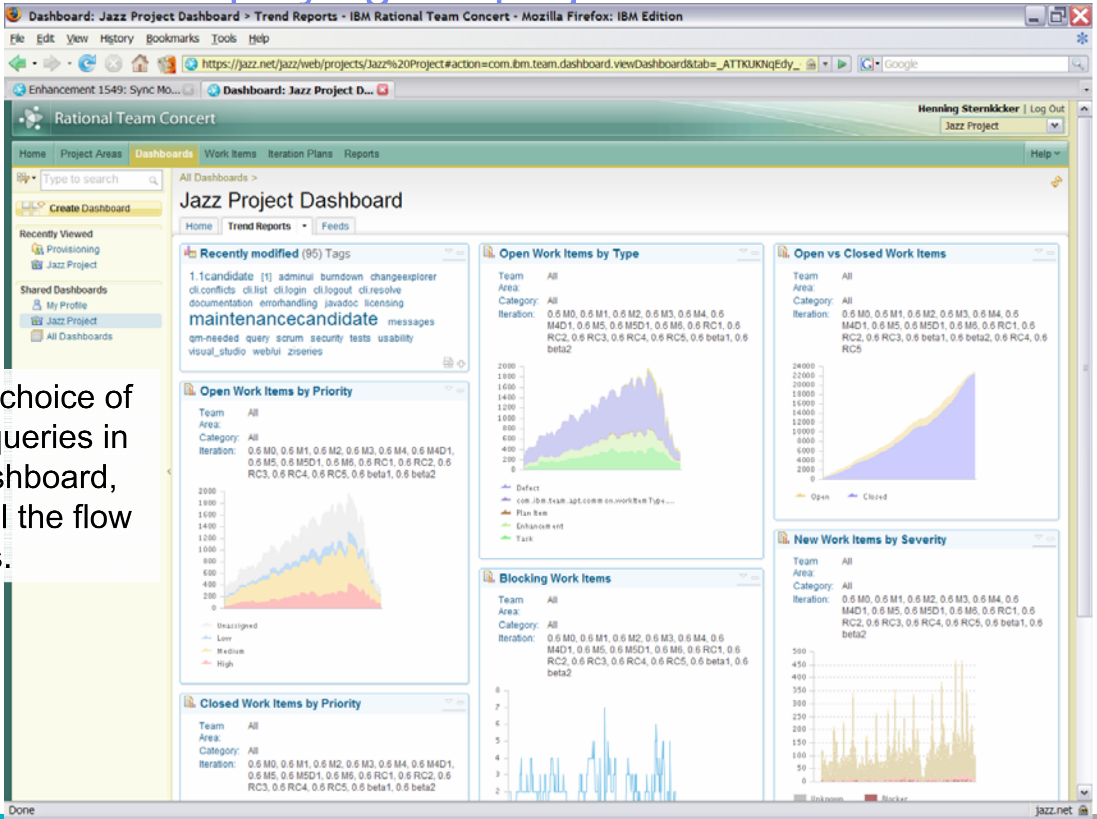
Work Item Builds
Succeeded: continuous.workitem.jazz C20080424-0748 10 minutes ago

Trending by Project or by Individual Team

Current Milestone status

Team Member Details

Dashboards – Displaying the project status



Dashboard: Jazz Project Dashboard > Trend Reports - IBM Rational Team Concert - Mozilla Firefox: IBM Edition

File Edit View History Bookmarks Tools Help

https://jazz.net/jazz/web/projects/Jazz%20Project#action=com.ibm.team.dashboard.viewDashboard&tab=_ATTKUKNqEdy_

Enhancement 1549: Sync Mo... | Dashboard: Jazz Project D...

Rational Team Concert — Henning Sternkicker | Log Out — Jazz Project

Home | Project Areas | Dashboards | Work Items | Iteration Plans | Reports | Help

Create Dashboard

Recently Viewed: Provisioning, Jazz Project

Shared Dashboards: My Profile, Jazz Project, All Dashboards

All Dashboards >

Jazz Project Dashboard

Home | Trend Reports | Feeds

Recently modified (95) Tags
1.1candidate [1] adminui burndown changeexplorer cli.conflicts cli.list cli.login cli.logout cli.resolve documentation errorhandling javadoc licensing maintenancecandidate messages qm-needed query scrum security tests usability visual_studio webui ziseries

Open Work Items by Priority
Team Area: All
Category: All
Iteration: 0.6 M0, 0.6 M1, 0.6 M2, 0.6 M3, 0.6 M4, 0.6 M4D1, 0.6 M5, 0.6 M5D1, 0.6 M6, 0.6 RC1, 0.6 RC2, 0.6 RC3, 0.6 RC4, 0.6 RC5, 0.6 beta1, 0.6 beta2

Closed Work Items by Priority
Team Area: All
Category: All
Iteration: 0.6 M0, 0.6 M1, 0.6 M2, 0.6 M3, 0.6 M4, 0.6 M4D1, 0.6 M5, 0.6 M5D1, 0.6 M6, 0.6 RC1, 0.6 RC2, 0.6 RC3, 0.6 RC4, 0.6 RC5, 0.6 beta1, 0.6 beta2

Open Work Items by Type
Team Area: All
Category: All
Iteration: 0.6 M0, 0.6 M1, 0.6 M2, 0.6 M3, 0.6 M4, 0.6 M4D1, 0.6 M5, 0.6 M5D1, 0.6 M6, 0.6 RC1, 0.6 RC2, 0.6 RC3, 0.6 RC4, 0.6 RC5, 0.6 beta1, 0.6 beta2

Blocking Work Items
Team Area: All
Category: All
Iteration: 0.6 M0, 0.6 M1, 0.6 M2, 0.6 M3, 0.6 M4, 0.6 M4D1, 0.6 M5, 0.6 M5D1, 0.6 M6, 0.6 RC1, 0.6 RC2, 0.6 RC3, 0.6 RC4, 0.6 RC5, 0.6 beta1, 0.6 beta2

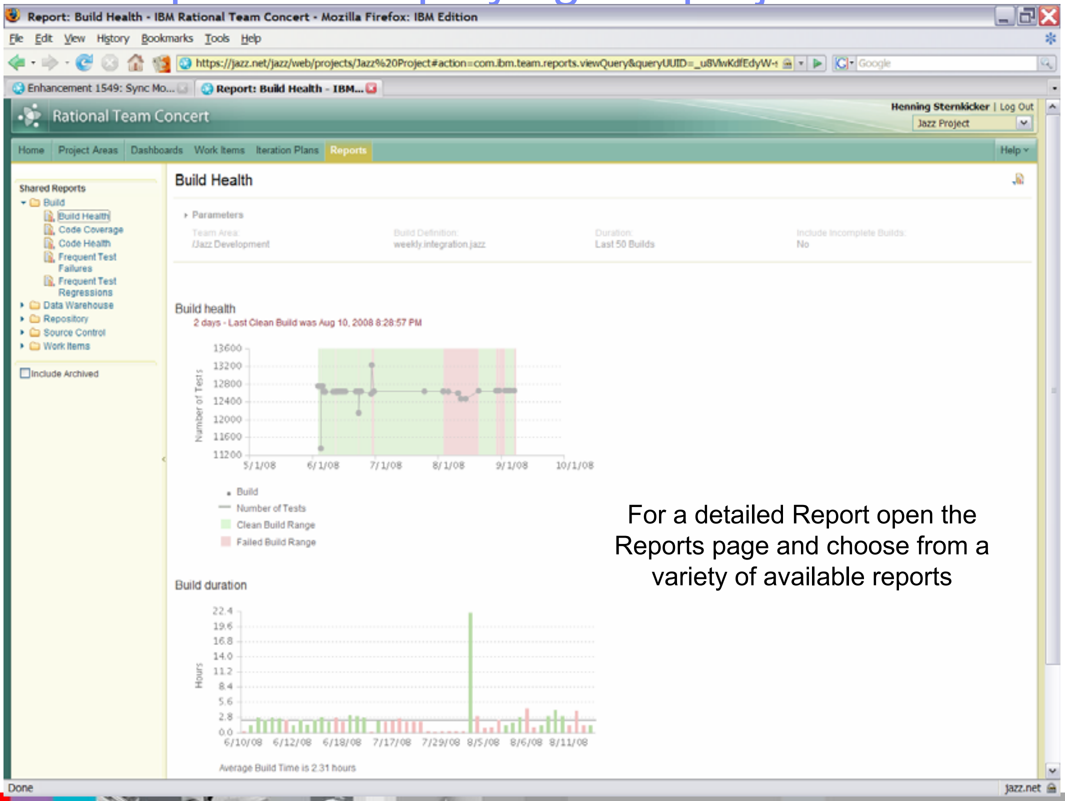
Open vs Closed Work Items
Team Area: All
Category: All
Iteration: 0.6 M0, 0.6 M1, 0.6 M2, 0.6 M3, 0.6 M4, 0.6 M4D1, 0.6 M5, 0.6 M5D1, 0.6 M6, 0.6 RC1, 0.6 RC2, 0.6 RC3, 0.6 beta1, 0.6 beta2, 0.6 RC4, 0.6 RC5

New Work Items by Severity
Team Area: All
Category: All
Iteration: 0.6 M0, 0.6 M1, 0.6 M2, 0.6 M3, 0.6 M4, 0.6 M4D1, 0.6 M5, 0.6 M5D1, 0.6 M6, 0.6 RC1, 0.6 RC2, 0.6 RC3, 0.6 RC4, 0.6 RC5, 0.6 beta1, 0.6 beta2

Done

Display your choice of reports and queries in your own dashboard, e.g. to control the flow of work items.

Reports – Displaying the project status



Report: Build Health - IBM Rational Team Concert - Mozilla Firefox: IBM Edition

Build Health

Parameters

Team Area: /Jazz Development
Build Definition: weekly.integration.jazz
Duration: Last 50 Builds
Include Incomplete Builds: No

Build health

2 days - Last Clean Build was Aug 10, 2008 8:28:57 PM

Build duration

Average Build Time is 2.31 hours

For a detailed Report open the Reports page and choose from a variety of available reports

Reports – Displaying the project status

- Reports are only available in the Rational Team Concert Standard edition.
- Rational Team Concert uses the BIRT* reporting engine
- A huge variety of report formats are designed and available to display an actual overview of your projects:
  - Reports for the health of your builds
  - Reports for viewing the team load and the distribution of work items
  - Reports for your code
  - Etc.
- Reports can be arranged in the Web UI to Dashboards
- Reports can be exported to: .pdf, .xls, .doc, .ppt formats

*BIRT is an open source Eclipse-based reporting system that integrates with your Java™/J2EE application to produce compelling reports.

Lab #9 Concepts Learned

- Rational Team Concert provides **transparency and control** via customizable dashboards
- Rational Team Concert automated project management dashboards are transparent to everyone – not just managers. This immediate and **automated feedback helps keeps teams on track and motivated** to achieve project goals
- Rational Team Concert comes with a variety of report formats to display and export the actual real time, in context project status.

IBM
Software
Group

Integrating with Other SCM Systems

Rational Team Concert ClearCase Connector

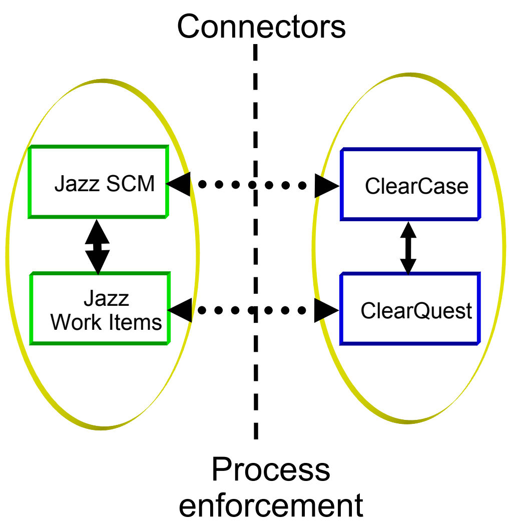
An IBM Proof of Technology

Objectives

- Explore how Rational Team Concert integrates with other SCM systems
- See how Rational Team Concert synchronizes its repository with Rational ClearCase

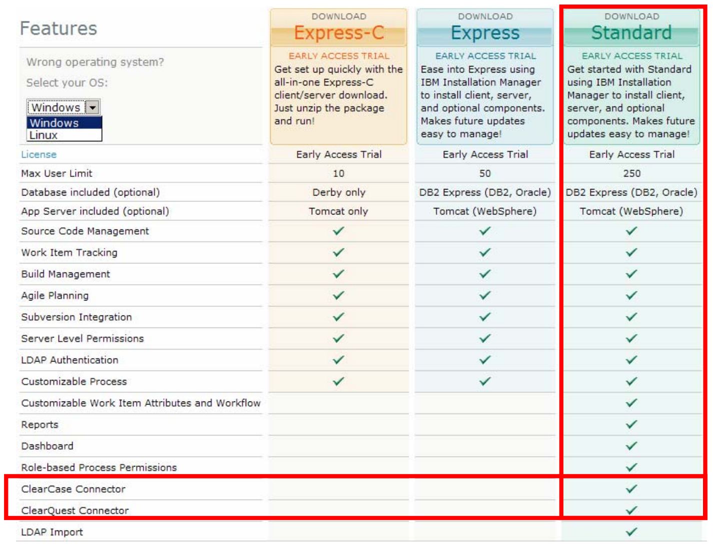
Integrating Other Repositories with Team Concert

- Import – migrate to Rational Team Concert
  - Jazz SCM – CVS, Subversion
  - Jazz Work Items – JIRA, Bugzilla
- Co-Existence (Bridge) – Lifecycle integration
  - Jazz Work Items – Subversion (SCM)
- Connectors – interoperation between repositories
  - Jazz SCM – ClearCase
  - Jazz Work Items - ClearQuest



Team Concert Editions and Connectors

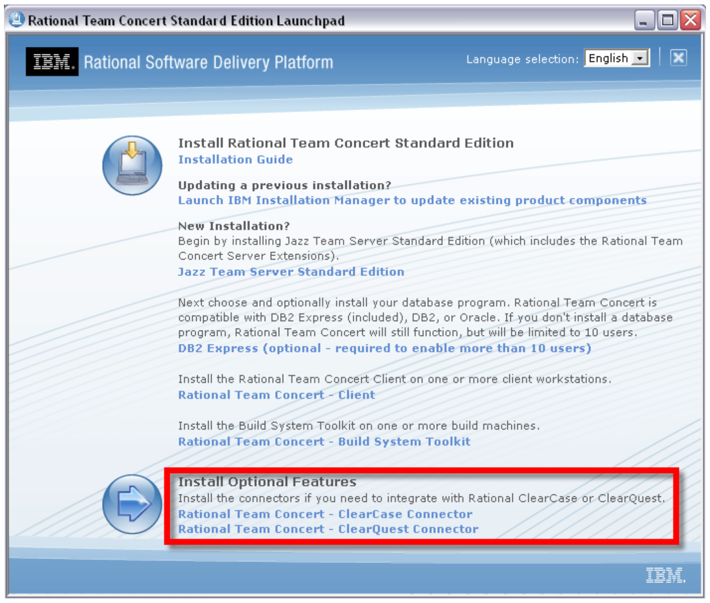
- **ClearCase Connector is only available in the** Team Concert Standard Edition**.**



Features	DOWNLOAD Express-C	DOWNLOAD Express	DOWNLOAD Standard
Wrong operating system? Select your OS: Windows / Windows / Linux	EARLY ACCESS TRIAL Get set up quickly with the all-in-one Express-C client/server download. Just unzip the package and run!	EARLY ACCESS TRIAL Ease into Express using IBM Installation Manager to install client, server, and optional components. Makes future updates easy to manage!	EARLY ACCESS TRIAL Get started with Standard using IBM Installation Manager to install client, server, and optional components. Makes future updates easy to manage!
License	Early Access Trial	Early Access Trial	Early Access Trial
Max User Limit	10	50	250
Database included (optional)	Derby only	DB2 Express (DB2, Oracle)	DB2 Express (DB2, Oracle)
App Server included (optional)	Tomcat only	Tomcat (WebSphere)	Tomcat (WebSphere)
Source Code Management	✓	✓	✓
Work Item Tracking	✓	✓	✓
Build Management	✓	✓	✓
Agile Planning	✓	✓	✓
Subversion Integration	✓	✓	✓
Server Level Permissions	✓	✓	✓
LDAP Authentication	✓	✓	✓
Customizable Process	✓	✓	✓
Customizable Work Item Attributes and Workflow			✓
Reports			✓
Dashboard			✓
Role-based Process Permissions			✓
ClearCase Connector			✓
ClearQuest Connector			✓
LDAP Import			✓

Team Concert Standard Edition Installation

- **ClearCase Connectors installation is optional and performed only in the designated Synchronization Host.**



Rational Team Concert Standard Edition Launchpad

IBM. Rational Software Delivery Platform

Language selection: English

Install Rational Team Concert Standard Edition
Installation Guide

Updating a previous installation?
Launch IBM Installation Manager to update existing product components

New Installation?
Begin by installing Jazz Team Server Standard Edition (which includes the Rational Team Concert Server Extensions).
Jazz Team Server Standard Edition

Next choose and optionally install your database program. Rational Team Concert is compatible with DB2 Express (included), DB2, or Oracle. If you don't install a database program, Rational Team Concert will still function, but will be limited to 10 users.
DB2 Express (optional - required to enable more than 10 users)

Install the Rational Team Concert Client on one or more client workstations.
Rational Team Concert - Client

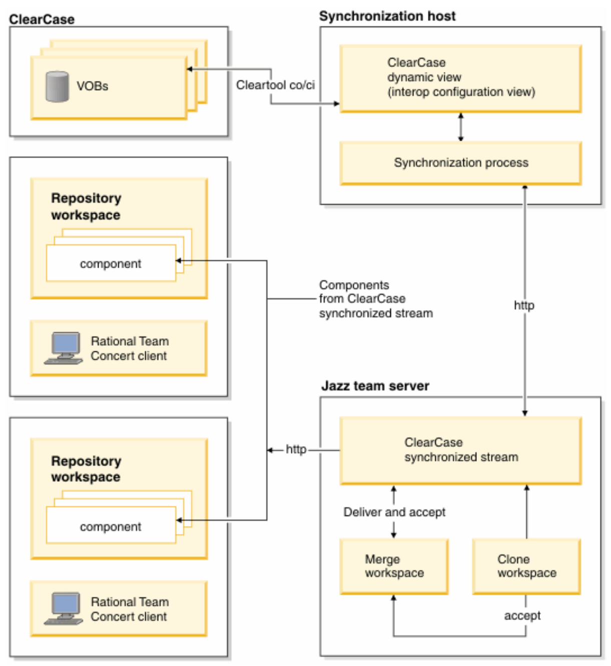
Install the Build System Toolkit on one or more build machines.
Rational Team Concert - Build System Toolkit

Install Optional Features
Install the connectors if you need to integrate with Rational ClearCase or ClearQuest.
Rational Team Concert - ClearCase Connector
Rational Team Concert - ClearQuest Connector

Interoperation Overview

- ClearCase is the corporate software configuration management solution
- Product integration and release are performed in ClearCase repositories
- Development teams opting for Team Concert can collaborate with other teams by deploying the ClearCase Connector
- The ClearCase Connector is not designed to import every version of an artifact from Rational ClearCase to Jazz source control. Instead, it enables synchronization of selected files and folders between both environments, so that development can continue in each.

Interoperation Architecture Overview

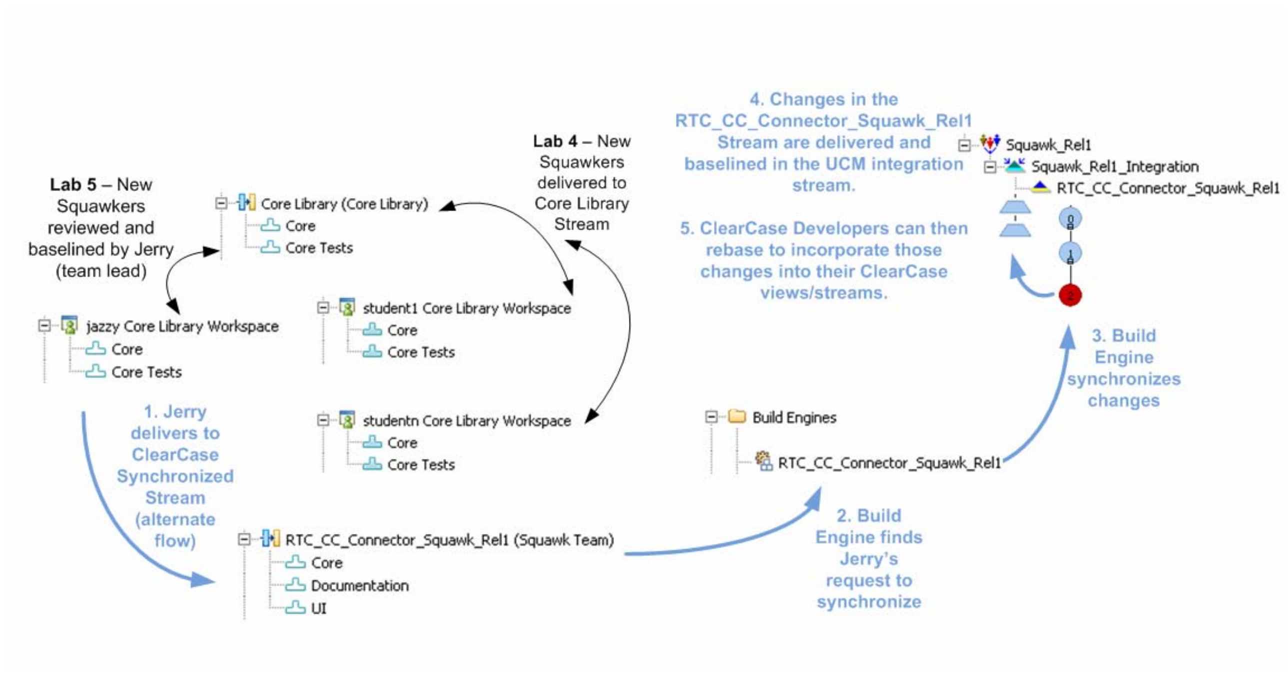


- Versions selected by the ClearCase interoperation configuration view are committed to the ClearCase Synchronized Stream in Team Concert by the synchronization process.
- If any item in the ClearCase Synchronized Stream have been modified concurrently by Jazz source control users and ClearCase, they appear in incoming change sets for the merge workspace.
- The merge workspace owner resolves the conflicts, and then delivers the merged results to the ClearCase Synchronized Stream.
- The change sets delivered to the ClearCase Synchronized stream are propagated back to ClearCase the next time the synchronization process runs.

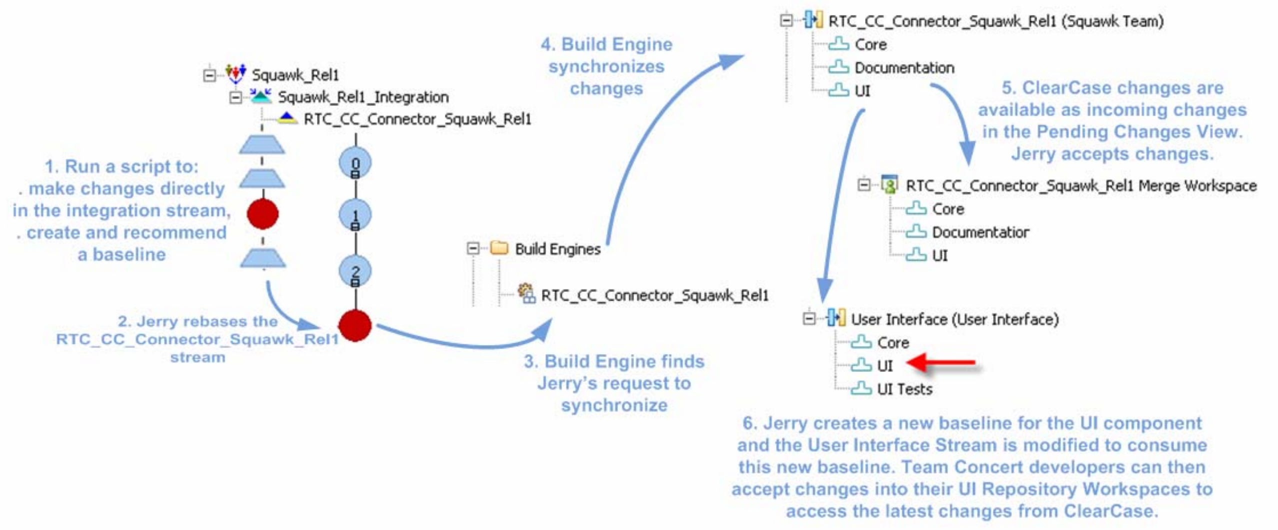
Lab #9 Overview

- In this lab the instructor will demonstrate how to synchronize code changes between Rational Team Concert and Rational ClearCase

Synchronize (export) Team Concert code changes to ClearCase


Lab 5 – New Squawkers reviewed and baselined by Jerry (team lead)
Core Library (Core Library)
Core
Core Tests
Lab 4 – New Squawkers delivered to Core Library Stream
jazzy Core Library Workspace
Core
Core Tests
student1 Core Library Workspace
Core
Core Tests
studentn Core Library Workspace
Core
Core Tests
1. Jerry delivers to ClearCase Synchronized Stream (alternate flow)
RTC_CC_Connector_Squawk_Rel1 (Squawk Team)
Core
Documentation
UI
2. Build Engine finds Jerry's request to synchronize
Build Engines
RTC_CC_Connector_Squawk_Rel1
3. Build Engine synchronizes changes
4. Changes in the RTC_CC_Connector_Squawk_Rel1 Stream are delivered and baselined in the UCM integration stream.
5. ClearCase Developers can then rebase to incorporate those changes into their ClearCase views/streams.
Squawk_Rel1
Squawk_Rel1_Integration
RTC_CC_Connector_Squawk_Rel1
0
1
2

Synchronize (import) ClearCase code changes to Team Concert


Squawk_Rel1
Squawk_Rel1_Integration
RTC_CC_Connector_Squawk_Rel1
1. Run a script to: . make changes directly in the integration stream, . create and recommend a baseline
0
1
2
2. Jerry rebases the RTC_CC_Connector_Squawk_Rel1 stream
Build Engines
RTC_CC_Connector_Squawk_Rel1
3. Build Engine finds Jerry's request to synchronize
4. Build Engine synchronizes changes
RTC_CC_Connector_Squawk_Rel1 (Squawk Team)
Core
Documentation
UI
5. ClearCase changes are available as incoming changes in the Pending Changes View. Jerry accepts changes.
RTC_CC_Connector_Squawk_Rel1 Merge Workspace
Core
Documentatior
UI
User Interface (User Interface)
Core
UI
UI Tests
6. Jerry creates a new baseline for the UI component and the User Interface Stream is modified to consume this new baseline. Team Concert developers can then accept changes into their UI Repository Workspaces to access the latest changes from ClearCase.

Questions

IBM
Software
Group

Project Growth and Multi-stream Development

An IBM Proof of Technology

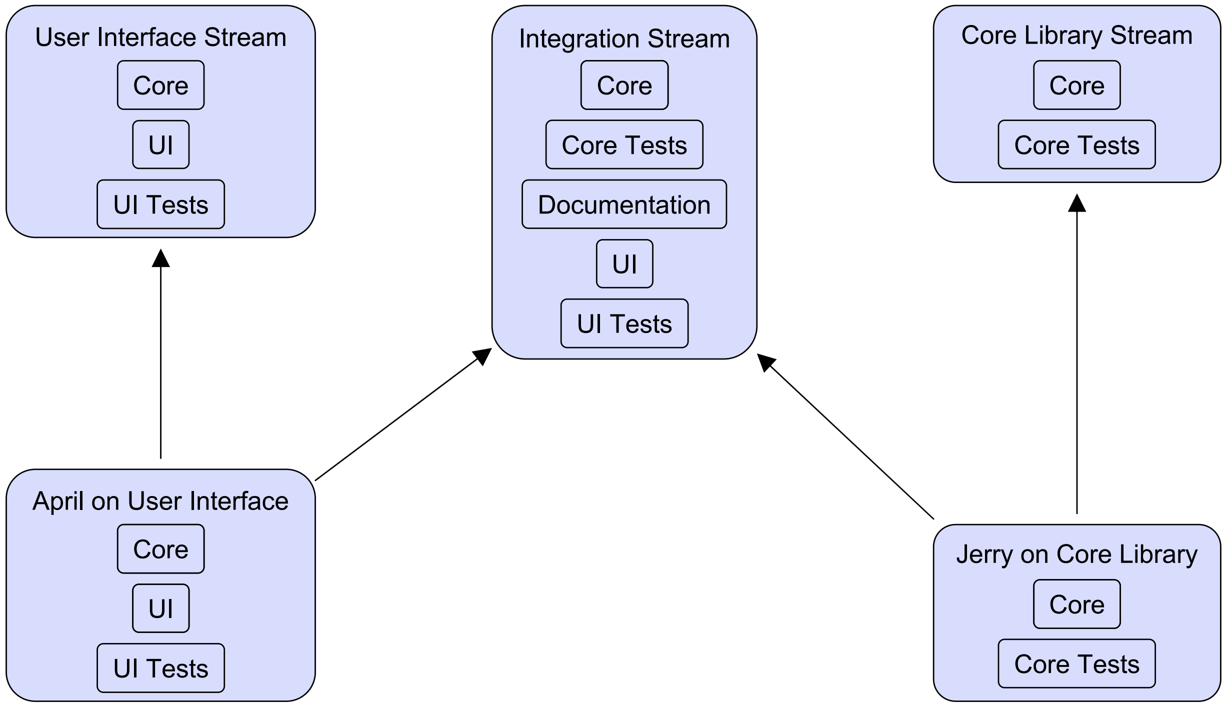
Objectives

- Understand the support for parallel development in Rational Team Concert

Growth and Adding Teams

- Project growth leads to multiple inter-dependent teams
- Each team needs its own stream
  - Quickly collaborate and share changes with each other
  - Run builds on a scheduled bases, as well as ad hoc
- Enhance ability for cross team collaboration and communication
  - Manage cross team dependencies
  - Project build stability and transparency
- Need a stream for cross team sharing and project builds
  - Well known change adoption schedule and process
  - Consistent and repeatable successful full project builds

Typical Component Baseline Flow



Other Scenarios

- Maintenance
  - New stream for maintenance
  - Created from final release snapshot
  - Isolated from daily development
  - Easy to move changes between streams
- Community exploration
  - Single person exploration can use a repository workspace
  - Community exploration will require sharing and collaboration
  - New stream created from a development stream snapshot

Concepts Learned

- In this module you explored the support for parallel development offered by Rational Team Concert.
- You have explored handling multiple streams and the sharing of component dependencies between them.
- Rational Team Concert enables
  - Increased team productivity by allowing parallel development
  - Enhances the delivery policies and process while improving baseline consistency
  - Supports seamless interaction for globally distributed teams
  - Establishes traceability and transparency among project artifacts

Questions

IBM
Software
Group

Session summary

An IBM Proof of Technology

Session summary

- We have described current collaboration challenges with distributed teams
- We have explored how Rational Team Concert can
  - Enable development teams to **collaborate in real time in the context** of the work they are doing, especially in globally diverse environments
  - Enable projects to be managed more effectively by providing visibility into **accurate project health information** drawn directly from actual work
  - Automate traceability and auditability by **managing artifacts and their inter-relationships** across the lifecycle empowering teams to deliver more value
  - Provide **customizable process design and enactment** through rule-based process guidance, automation and definable checkpoints
- We have provided a hands on experience using Rational Team Concert to automate the software delivery process

Next steps

- Engage your local Rational team
  - Provide a customized demo for your team
  - Conduct a targeted proof of concept
- Register on jazz.net and explore learning tutorials and videos
  - http://www.ibm.com/developerworks/spaces/jazz

Additional resources

- Learn more about and download free trials of Rational Team Concert at

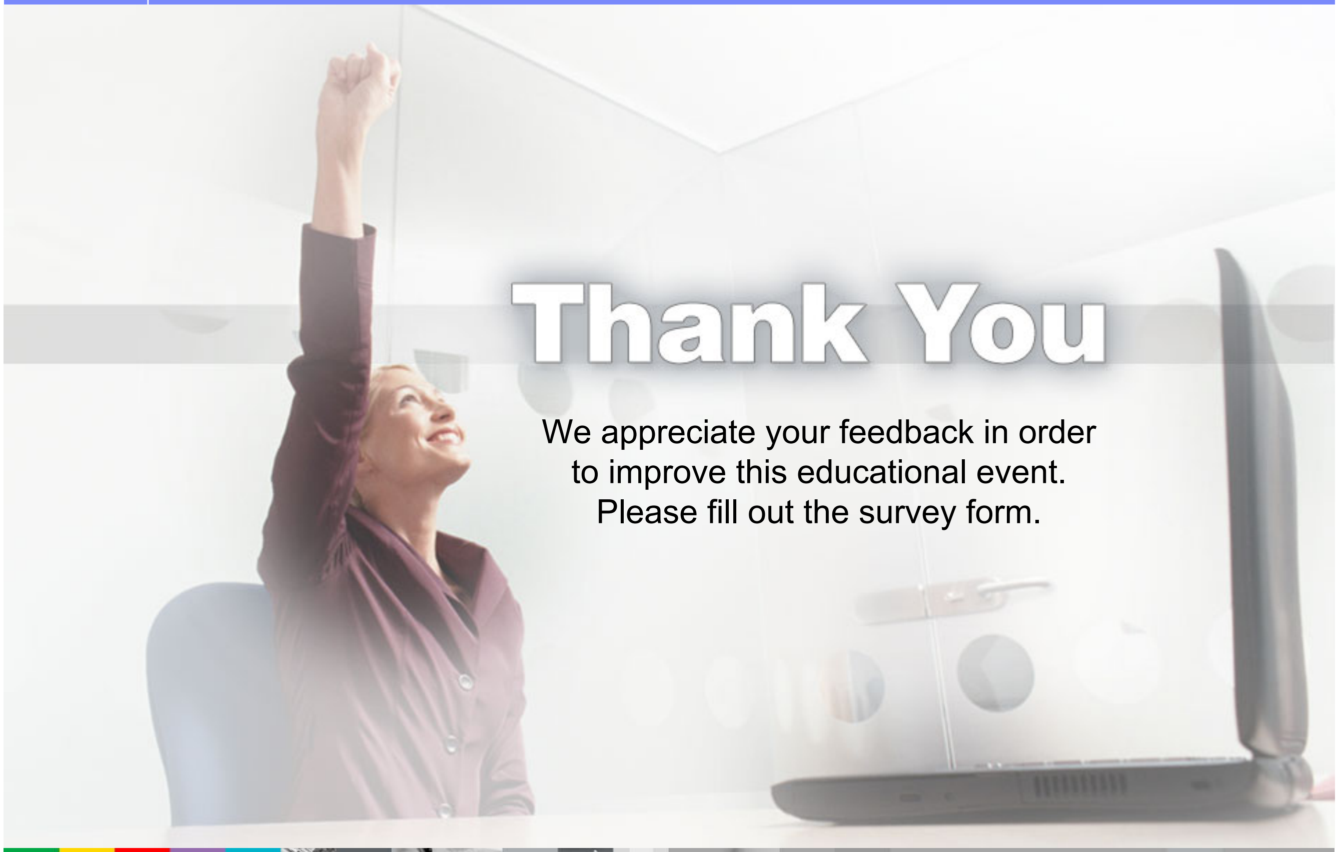
  http://ibm.com/rational/rtc
- Explore Rational Team Concert tutorials, demos and other developer learning resources

  http://ibm.com/developerworks/spaces/jazz
- Participate in the open commercial development of Jazz by joining the community

  http://jazz.net
- Learn more about the Jazz technology and the future IBM Rational product roadmap

  http://ibm.com/rational/jazz/roadmap

Questions



Thank You

We appreciate your feedback in order to improve this educational event. Please fill out the survey form.

NOTES

NOTES

NOTES