



IBM Software Group

Rational Team Concert



Rational software

Agenda

- Jazz platform introduction.
- Rational Team Concert introduction.
- Rational Team Concert in-depth.
 - ▶ Jazz Team Server features
 - ▶ Change management with work items and planning
 - ▶ Version control
 - ▶ Build Management
- Extensions to Rational Team Concert.
 - ▶ RTCp
 - ▶ RTCz
- Live Demo.



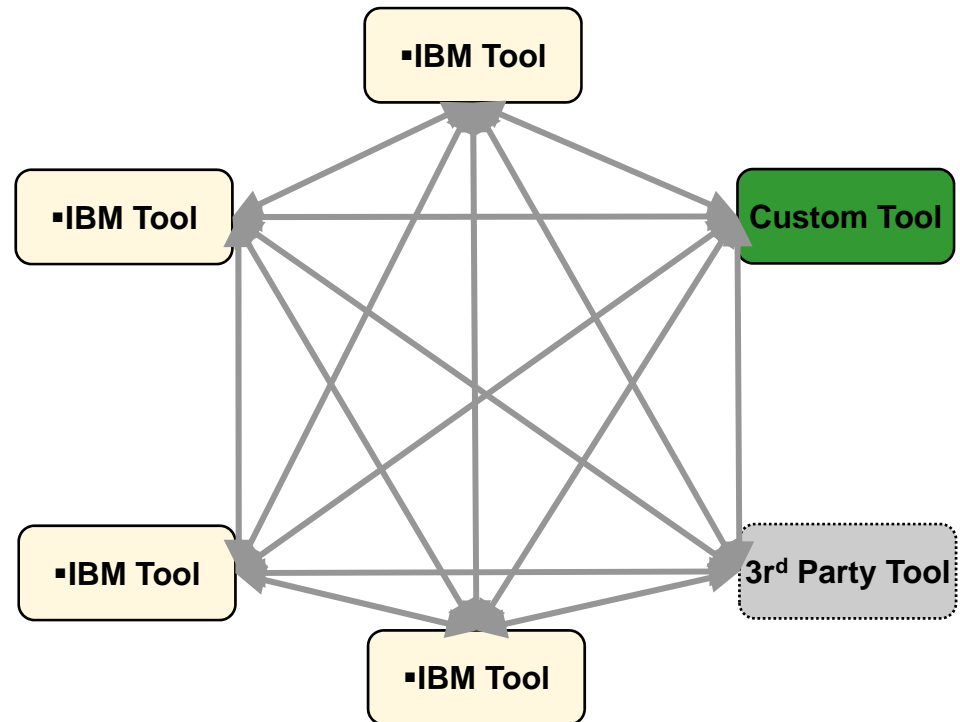
Agenda

- Jazz platform introduction.
- Rational Team Concert introduction.
- Rational Team Concert in-depth.
 - ▶ Jazz Team Server features
 - ▶ Change management with work items and planning
 - ▶ Version control
 - ▶ Build Management
- Extensions to Rational Team Concert.
 - ▶ RTCp
 - ▶ RTCz
- Live Demo.



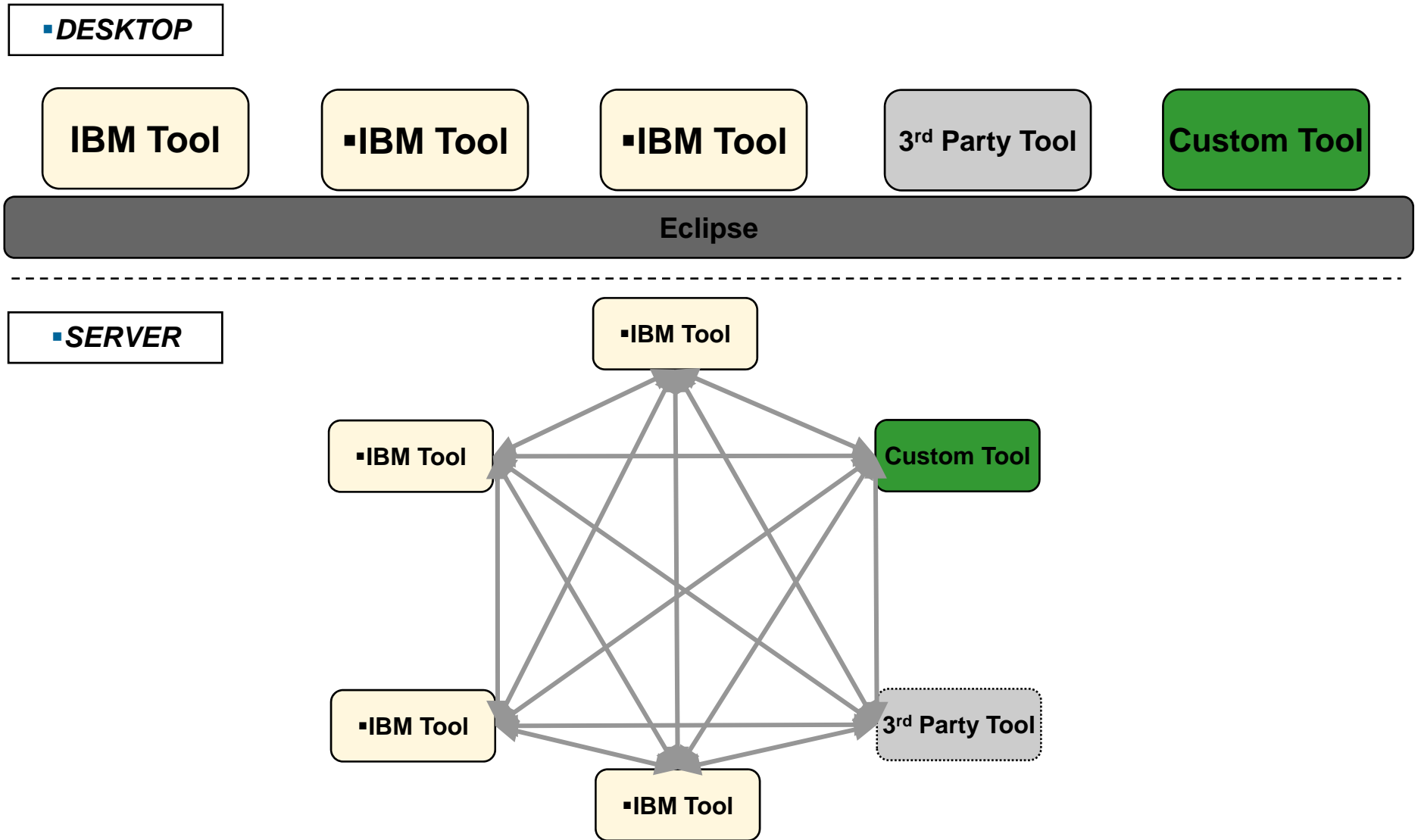
Our customers have invested in a diverse set of tools...

- Traditionally, each tool came with its own
 - ▶ UI - Web and desktop presentations of views and tasks
 - ▶ Logic – Workflow, process, search, query, scale, security and collaboration
 - ▶ Storage – Availability, traceability
 - ▶ Privacy, backup/archive
- Resulting in...
 - ▶ Brittle integrations
 - ▶ Silos everywhere
 - ▶ High cost to maintain and administer
 - ▶ Proprietary API's

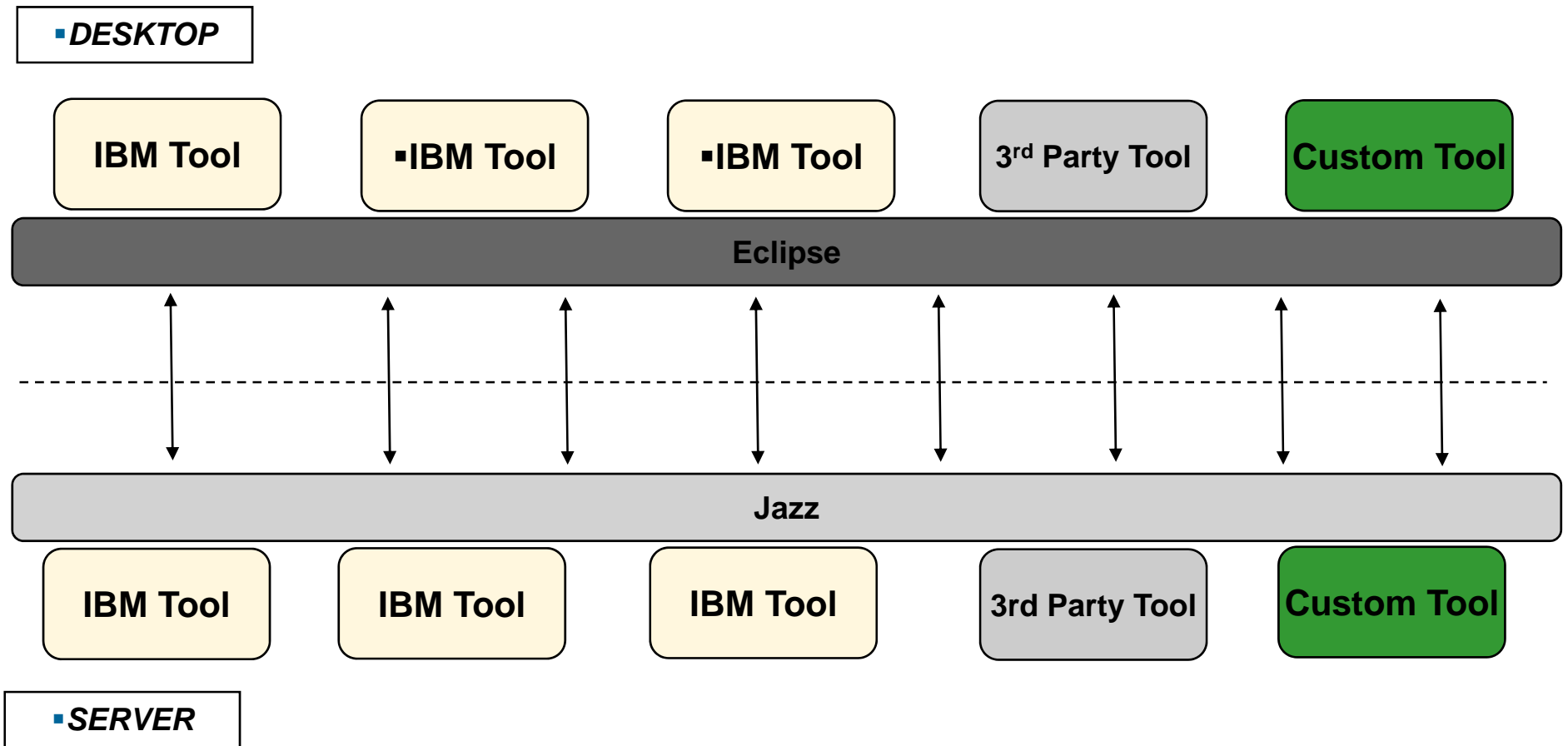


How do you solve this?

First step with Eclipse some years ago...

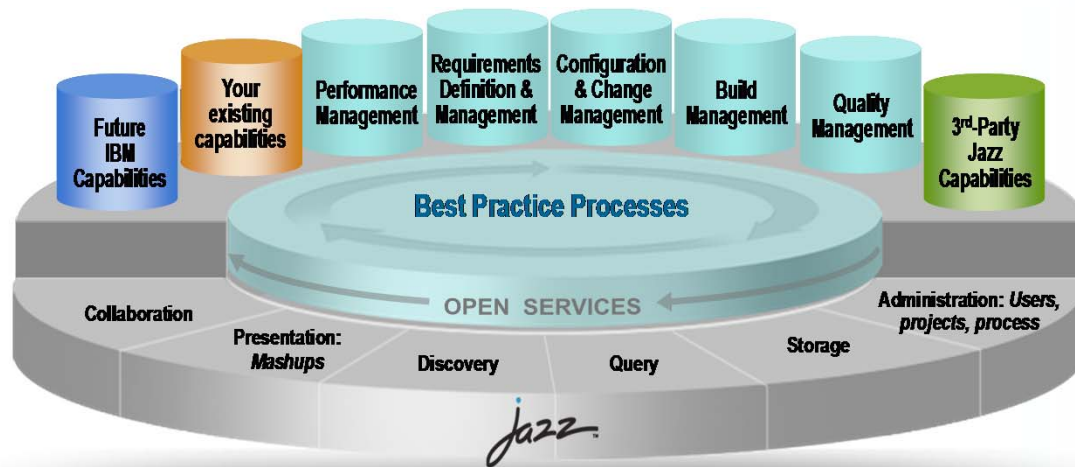


Next step with Jazz today...



The Jazz technology

Collaborative hub for Systems and Software delivery solutions

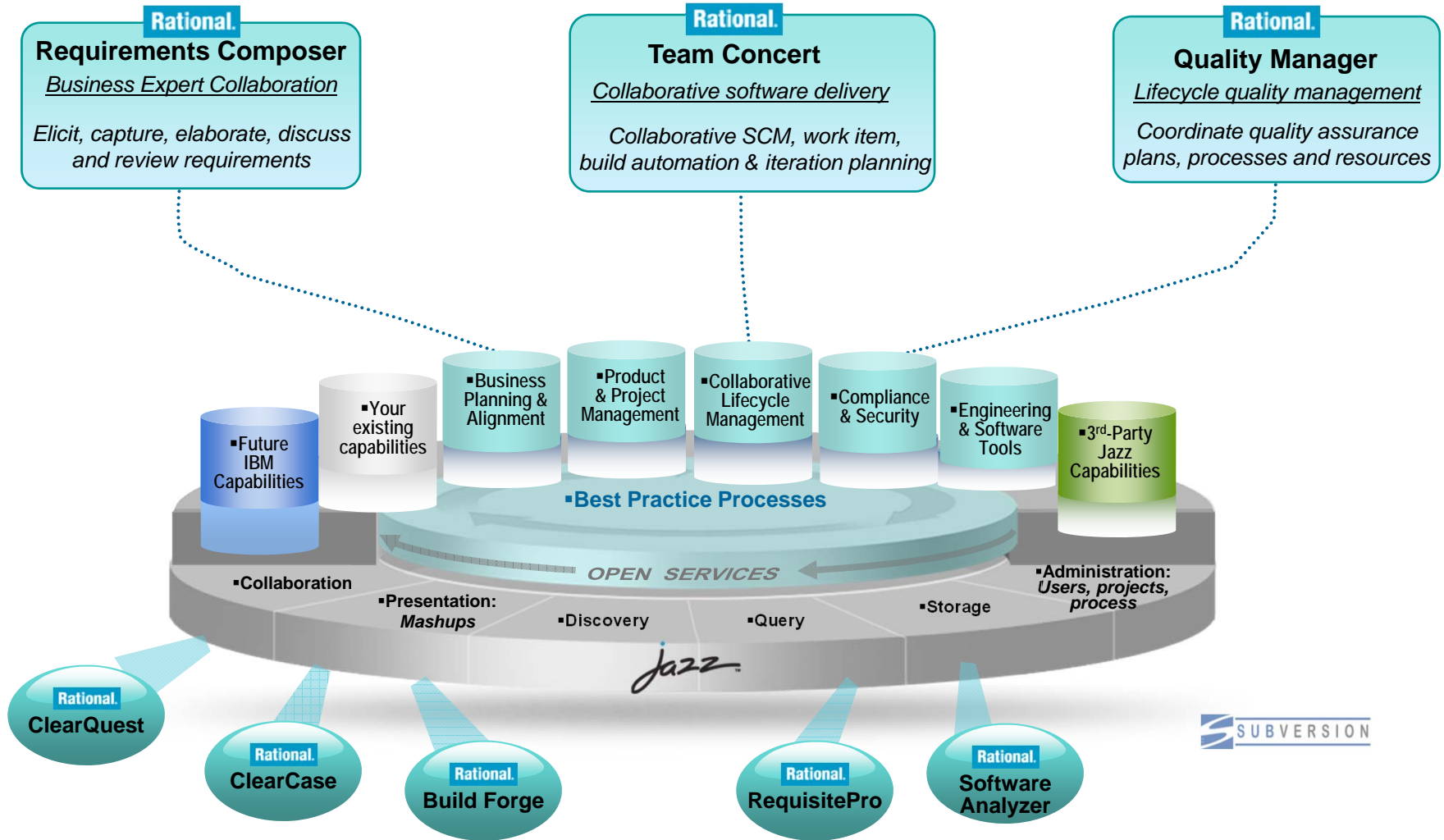


▪ **Jazz is a project and platform for *transforming how people work together* to deliver greater value and performance from their software investments.**

▪ ***Jazz is...***

- A scalable, extensible team collaboration platform
- A community at Jazz.net where you can see Jazz-based products being built
- Our vision of the future of systems and software delivery, supporting globally distributed teams
- An integration architecture enabling mashups and non-Jazz based products to participate
- An evolution of our portfolio which will evolve to leverage Jazz technology over time

First wave of products built on Jazz technology



Agenda

- Jazz platform introduction.
- Rational Team Concert introduction.
- Rational Team Concert in-depth.
 - ▶ Jazz Team Server features
 - ▶ Change management with work items and planning
 - ▶ Version control
 - ▶ Build Management
- Extensions to Rational Team Concert.
 - ▶ RTCp
 - ▶ RTCz
- Live Demo.



Agenda

- Jazz platform introduction.
- Rational Team Concert introduction.
- Rational Team Concert in-depth.
 - ▶ Jazz Team Server features
 - ▶ Change management with work items and planning
 - ▶ Version control
 - ▶ Build Management
- Extensions to Rational Team Concert.
 - ▶ RTCp
 - ▶ RTCz
- Live Demo.



Rational Team Concert Overview

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 bridge and connector

Work Items

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

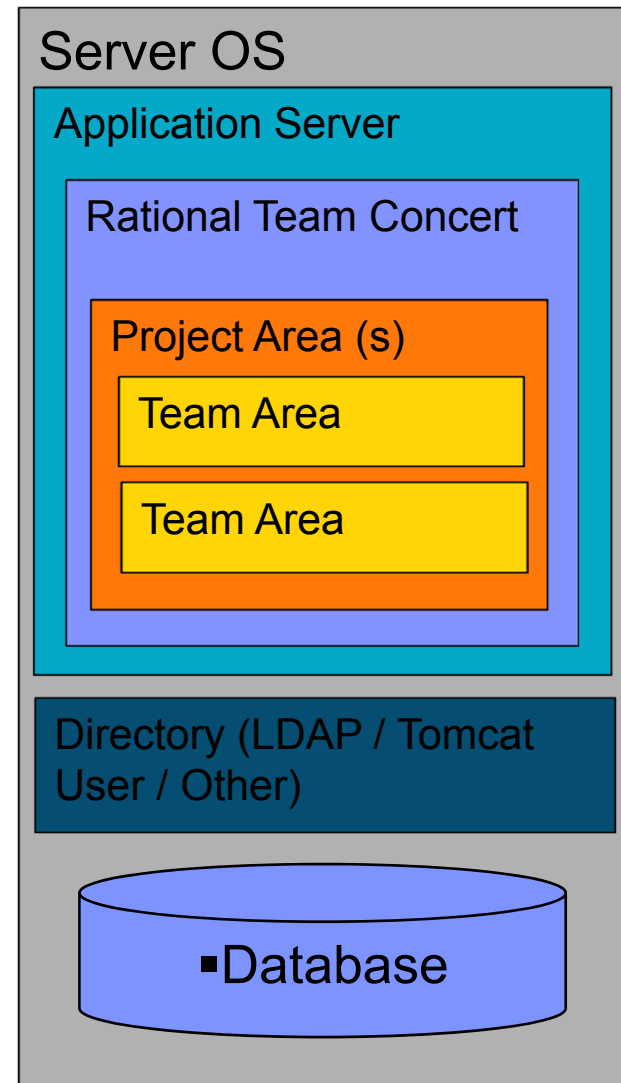
Jazz Team Server

- 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

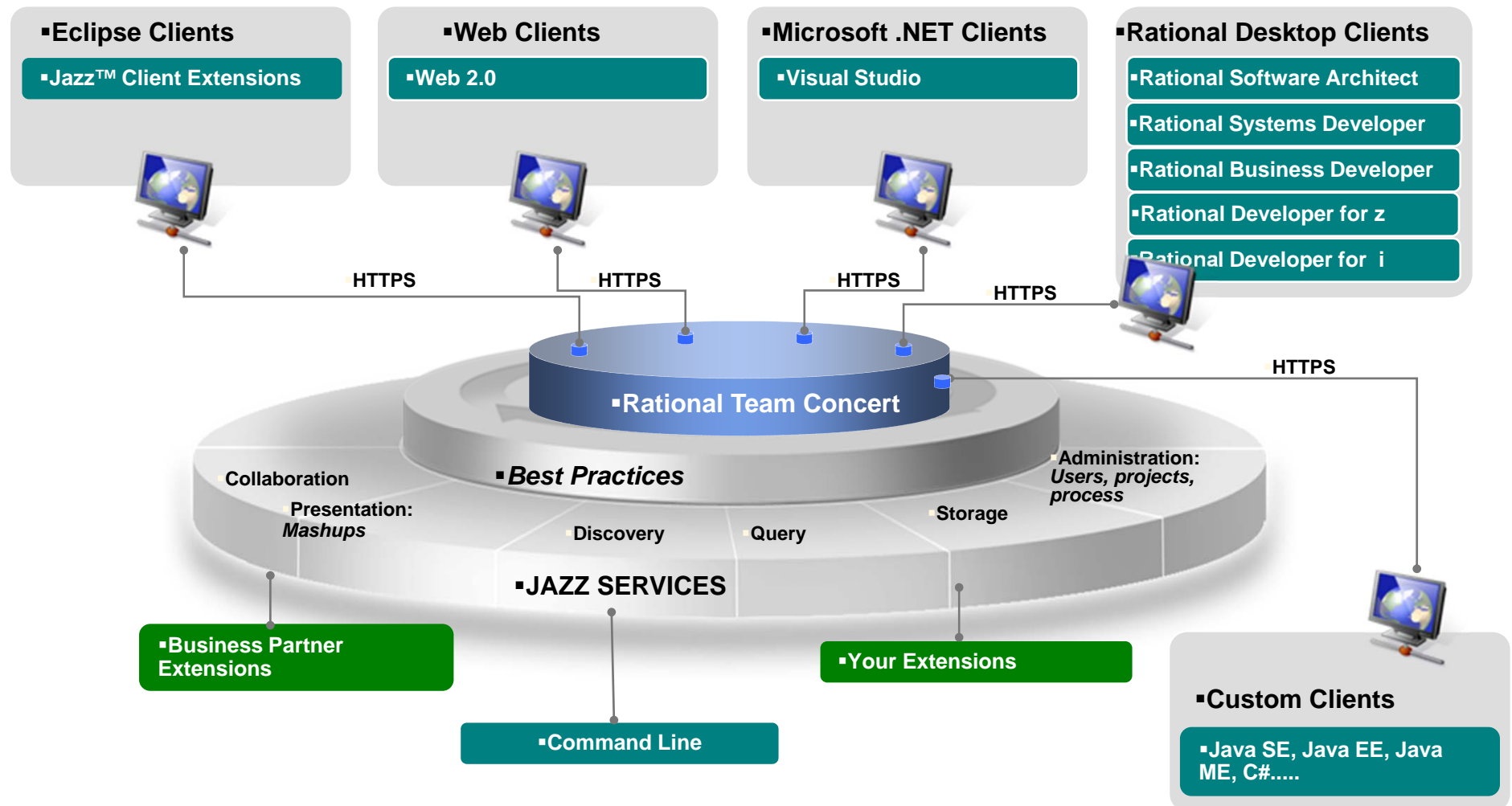
RTC Server Logical Architecture

- Server Operating System
 - ▶ Windows
 - ▶ Linux
 - ▶ AIX
 - ▶ Solaris
 - ▶ OS/400
 - ▶ z/OS
- Application Server
 - ▶ WebSphere Application Server (WAS)
 - ▶ Apache Tomcat Application Server
- Database
 - ▶ Derby
 - ▶ DB2
 - ▶ Microsoft SQL
 - ▶ Oracle
- Directory Services
- Rational Team Concert Server



Rational Team Concert clients

Supporting a broad range of desktop clients, IDE's and languages



Agenda

- Jazz platform introduction.
- Rational Team Concert introduction.
- Rational Team Concert in-depth.
 - ▶ Jazz Team Server features
 - ▶ Change management with work items and planning
 - ▶ Version control
 - ▶ Build Management
- Extensions to Rational Team Concert.
 - ▶ RTCp
 - ▶ RTCz
- Live Demo.



Rational Team Concert Overview

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 bridge and connector

Work Items

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

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 Overview

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

Organization levels in Jazz

The screenshot displays the IBM Jazz Project Area interface. On the left, a tree view shows 'All Team Areas (5 of 5 Team Areas selected)' with a hierarchy including Repository Connections, Squawk [localhost], Builds, Build Engines, Build Queue, Plans, Reports, Streams, Work Items, Favorites, Feeds, My Repository Workspaces, My Team Areas, and Work Item History. The main area is titled 'Project Area' and shows details for 'Squawk'. The 'Details' section includes a Summary ('Instructional project for user training') and a Description ('This workshop was created and maintained by the Jazz Jumpstart Team: Jim D'Anjou, Philippe Krief, Paul Vanderlei, Steve Wasleski'). The 'Members' section lists roles for 'build' and 'Jeny Jazz'. The 'Administrators' section is currently empty. The 'Process Description' section shows the 'Eclipse Way Process (In Progress)'. The 'Process Iterations' section displays a tree view of development lines: 'Main Development Line', 'Release 1 [1/15/08 - 12/31/08]', '1.0 M1 [1/15/08 - 1/25/08]', '1.0 M2 [1/25/08 - 12/31/08]', 'Development', and 'End Game'. Buttons for 'Edit Properties...', 'Duplicate...', 'Create Line...', 'Create Iteration...', and 'Delete...' are visible. The 'Attachments' section is at the bottom.

Access control

- Two levels of access control :
 - Access control controlled by application server (LDAP)
 - Access control in Jazz platform (roles with specific permissions for SCM related functions)
- All jazz products share the same access control menus (each tool adds its own permissions)

Control operation behavior based on Roles

Configure which actions are permitted for each role.

Show by role Show all roles

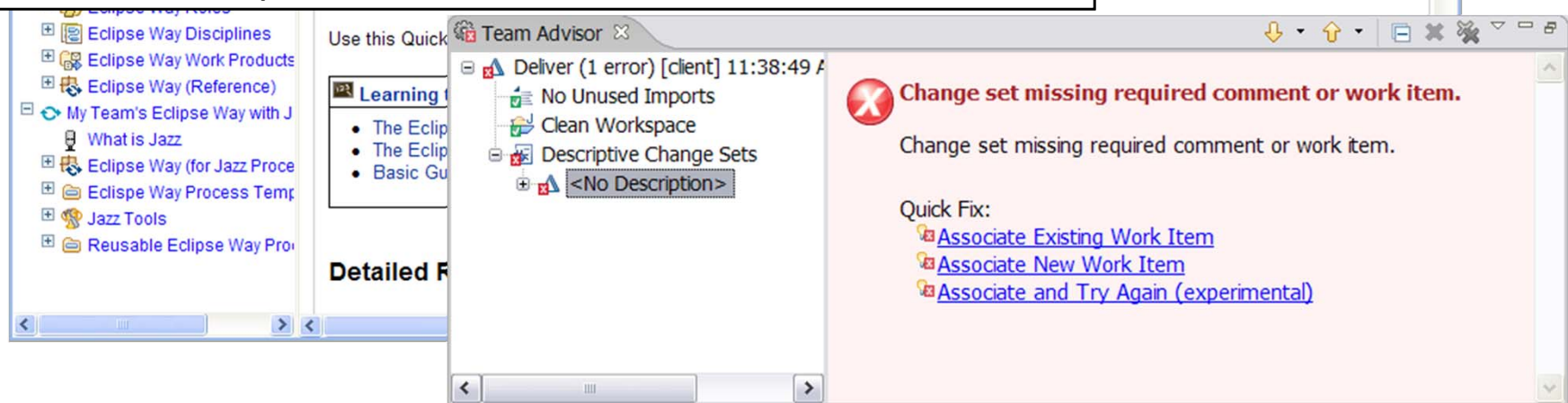
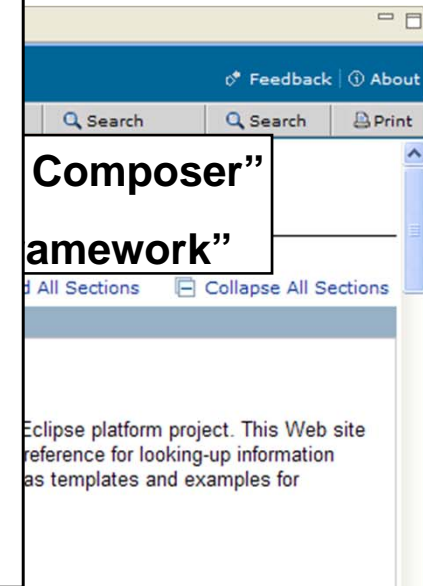
Actions	contributor	teamlead	default
Process			
Generate Team Invitation (server)	✓	✗	✓
Save Team Area (server)	✗	✓	✗
<Uncategorized>			
Abandon Build (server)	✓	✓	✗
Cancel Pending Build Request (server)	✓	✗	✗
Control The Build Lifecycle (server)	✗	✓	✗
Create Private Component (server)	✗	✗	✓
Create Repository Workspace (server)	✗	✗	✓
Delete Build Definition (server)	✗	✓	✗
Delete Build Engine (server)	✗	✓	✗
Delete Build Result (server)	✗	✓	✗
Delete Iteration Plan (server)	✓	✓	✗
Delete Query (server)	✓	✗	✗
Deliver (server)	✓	✗	✗
Deploy Report (server)	✓	✗	✗
Deploy Report Template (server)	✗	✗	✗
Display Report (server)	✗	✗	✗
Manage Report Folder (server)	✓	✗	✗
Request Build (server)	✓	✗	✗
Save Build Definition (server)	✗	✓	✗
Save Build Engine (server)	✗	✓	✗
Save Build Result (server)	✓	✗	✗
Save Component (server)	✗	✓	✗
Save Iteration Plan (server)	✓	✓	✗
Save Query (server)	✓	✗	✗

Provide flexible process at each line of development

Enables Process Automation and Guidance

Team Advisor

- Rules can be run when delivering changes to enforce team or organizational standards
- Helps ensure higher quality results through enforcement of agreed-upon standards
- Rules are configurable
- “Quick Fixes” can be specified to simplify corrective action
- Process rules can be defined, refined “on the fly”, enabling continual improvements



Process templates

- Process templates provide a starting point and a guide for a project area process configuration and iteration structure
- A process template defines the:
 - ▶ Iteration structure
 - ▶ Roles
 - ▶ Permissions
 - ▶ Work item types
 - ▶ Queries
- You apply a process template when you create a project area

Predefined process templates

- Rational Team Concert provides the following predefined process templates:
 - ▶ **Cloudburst**
Supports the Cloudburst reference project available with the Jazz technology platform
 - ▶ **OpenUp**
Preserves the essential characteristics of the Rational Unified Process
 - ▶ **Simple team**
Enables teams to get started quickly
 - ▶ **Scrum**
Supports the agile approach to managing projects
 - ▶ **Formal Project Management**
Supports traditional projects

In Context Collaboration

Team Awareness

- Shows Team members and their online status
- Shows what they are working on

The screenshot displays the Eclipse IDE interface for the 'Jazz Project - Team Concert'. The left sidebar shows a 'Team Load' view with team members and their work items. The main workspace shows a 'Story 45274' with a progress bar (11% to 89%), a discussion thread, and a chat window. A context menu is open over the 'Chat...' option, listing actions like 'Send File...', 'Send Clipboard...', 'Send Mail...', 'Call...', 'Subscribe to Events Generated by User', 'Add to Favorites...', 'Show Recent Events', 'Show Recent Work', 'Show Current Work', and 'Invite to Join Team...'. The chat window shows a conversation between Kevin (K.G.) Haaland and kartik@ca... regarding the story.

Rational Team Concert Overview

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 bridge and connector

Work Items

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

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 Overview

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

Project Plans

- A plan revolves around the following elements
 - ▶ Teams
 - ▶ Time
 - ▶ Work
- Planning levels
 - ▶ Release
 - ▶ Iteration/Sprint
 - ▶ Your day-to-day work

 1.0 M3 3 Aug 2009 - 31 Dec 2009

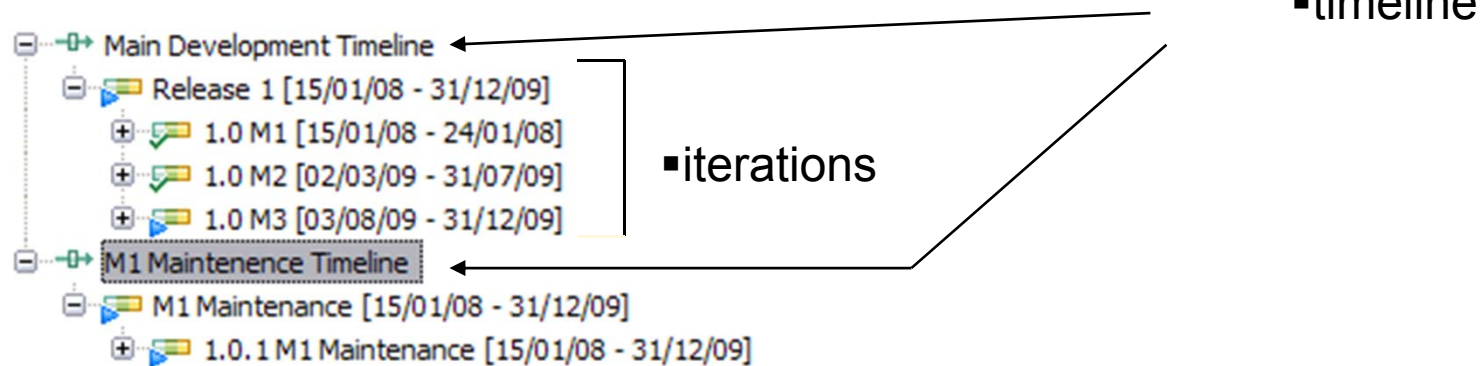
 **Squawk Teams (7)**

- ▼ Squawk Maintenance Team
 - Core Library M1 Maintenance
- ▼ Squawk Team
 - Core Library
 - Documentation
 - Release Engineering
 - User Interface

	493	Create futuristic squawkers
	470	Support theme based squawk parties
	471	Support other types of technologies like audio and video squawkers
	484	Predefine template squawker classes
	469	Allow new squawkers to be defined from a user interface instead of code

Time

- A project has one or more schedules or **timelines**.
- Each timeline is broken down into a series of iterations
- Each iteration can be broken into smaller iterations as required
- A milestone marks the end of an **iteration**
- At the end of any iteration, you may produce a **release**



Work

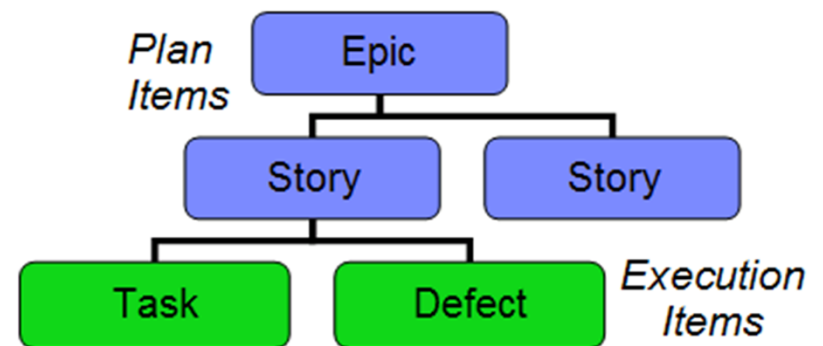
- All work in your project is tracked as one or more Work Items
- Different kinds of work items are available including

- ▶ Plan work item types, for example:

- Plan Item
- Epic
- Story

- ▶ Execution work item types, for example:

- Task
- Defect



- ▶ Plan work item types are used to capture high-level plan elements

- ▶ Execution work item types are used to capture the lower-level details and the work should be completed in a single iteration

- Each kind of work item has its own lifecycle and data

Work planned is captured as work items

The screenshot displays a JIRA board for the project 'UWS Temperature Conversion M1 Plan'. The board is organized into columns for 'Planned Items' and 'Charts'. It shows a list of work items assigned to team members: April Blues, Derek Holt, Jerry Jazz, Zach Builder, and Zara Intern, along with an 'Unassigned' group. Each item includes a progress bar, estimated completion time, and priority. Callouts highlight the following features:

- Understand how well you are progressing against your targets in real-time**: A callout points to the progress bars and status indicators for each work item.
- Plan and execute on iterations while managing load**: A callout points to the 'Unassigned' group and its associated work items.
- Drag-and-drop work items to change owners/create child parent relationships**: A callout points to the 'Unassigned' group, indicating the ability to reassign or create relationships between items.

Additional interface elements visible include a 'Group by' dropdown set to 'Owner', a 'Sort By' dropdown set to 'Priority', and a 'Bars' dropdown set to 'Progress'. The 'Exclude' section is checked for 'Resolved items'. The 'Related Work Items' section includes links for 'Show Backlog' and 'Unplanned Closed Items'. The 'Next Plans' section is also visible.

Customizable web dashboard

The dashboard is titled "Development" and features a navigation bar with tabs: "General", "Builds", "Open Defects by Themes", "Open Defects by Team", "SVT Defects by Team", "Defect Trends by Team", and "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 its own project phase.
Each component team that actively contributes to the development work is represented with its own child team area.

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, tracking progress
26827 [RFS] Getting Started Experience
11321 [RFS] Install and configure a Jazz server
9631 [RFS] Process governed work item customisation
9625 [RFS] Starting a Project
9627 [RFS] Joining a team
9623 [RFS] Browsing the history of work items
9632 [RFS] Duplicate handling
9908 [RFS] Managing Team Artifacts
Page 1 of 3

Current Development Plans (33)
Current Iteration: 0.6 RC1
APT
CC Connector RC1
CQ Connector
Component Development
Foundation & Work Items
Install
PMC
Process
Release Engineering
Repository
SCM
UI Design
User Assistance
Web Dashboards
Web UI
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

Configurable Project or teams dashboard

Access to development info (SCM, RQM..)

Real time event notifications

Real time Status of planning

Rational Team Concert Overview

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 bridge and connector

Work Items

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

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 Overview

Work Items

Defects, enhancements
and conversations

View and share query results

Support for approvals and
discussions

Query editor interface

ClearQuest bridge, connector

Work items

- Work planned for a project is captured in work items
- There are different types of work items
 - ▶ **Defect**—identifies a bug
 - ▶ **Enhancement**—describes a requested new feature
 - ▶ **Task**—describes a specific piece of work
 - ▶ **Plan Item**—provides a high-level description of work targeted for a specific iteration
 - ▶ **Story**—describes part of a use case
- Typical work item attributes
 - ▶ Workflows and Status
 - ▶ Properties and Values
 - ▶ Approvals
 - ▶ Attachments

Work Items

The screenshot displays the Squawk Work Items application interface. A central window titled "Work Item Query" is open, showing a query editor with the following configuration:

- Operator: AND All must match
- Severity is: Blocker, Critical, Major, Normal, Minor, Unclassified (1 value selected)
- Priority is: High, Medium, Low, Unassigned (0 values selected)

Other interface elements include:

- Attachments:** A table with columns "Id" and "Name".
- Links:** A section for managing links.
- Task 7:** Summary: Core Code.
- Bottom Panel:** Shows "Found 8 Work Items (782 n)", a table with columns "I...", "Status", and "Res", and a list of work items with status "Resolved".
- Callouts:** Several blue callout boxes highlight features: "Attach files and Screen shots", "Support for discussions around", "RSS Feed support", "Query editor interface", "Support Approv", and "team men".

Visual configurator for states, actions and fields

Área de proyecto

Mostrar en organización de equipo Abrir interfaz de usuario Web de proyecto

Configuración

Roles

- Configuración de proyecto
 - Inicialización de área de proyecto
 - Permisos
 - Comportamiento de operación (sin configurar)
 - Manejo de sucesos (sin configurar)
 - Datos de configuración
 - Elementos de trabajo
 - Consultas predefinidas
 - Enlaces de flujo de trabajo
 - Enlaces de presentación de trabajo
 - Enumeraciones
 - Flujos de trabajo
 - Presentaciones de editor
 - Presentaciones de editor de configuración
 - Presentaciones de Informes
 - Seguimientos de aprobación
 - Tipos y atributos
 - Paneles de instrumentos
 - Planes de iteración
 - Configuración de equipo

Flujos de trabajo

Defina transiciones de estado de elemento de trabajo (flujos de trabajo). Los flujos de trabajo se componen de estados, acciones y resoluciones. Para añadir una acción, defina dos estados como mínimo en la tabla Estados. A continuación seleccione "Añadir acción..." en la combinación deseada de la tabla Transiciones. Especifique uno de los grupos (abierto, en proceso, cerrado) para los estados para proporcionar el contexto necesario para el sistema.

Elija el flujo de trabajo que desea editar:

Peticion sistemas Añadir... Duplicar... Eliminar...

Nombre: Peticion sistemas

Descripción: Flujo que representa una peticion a sistemas

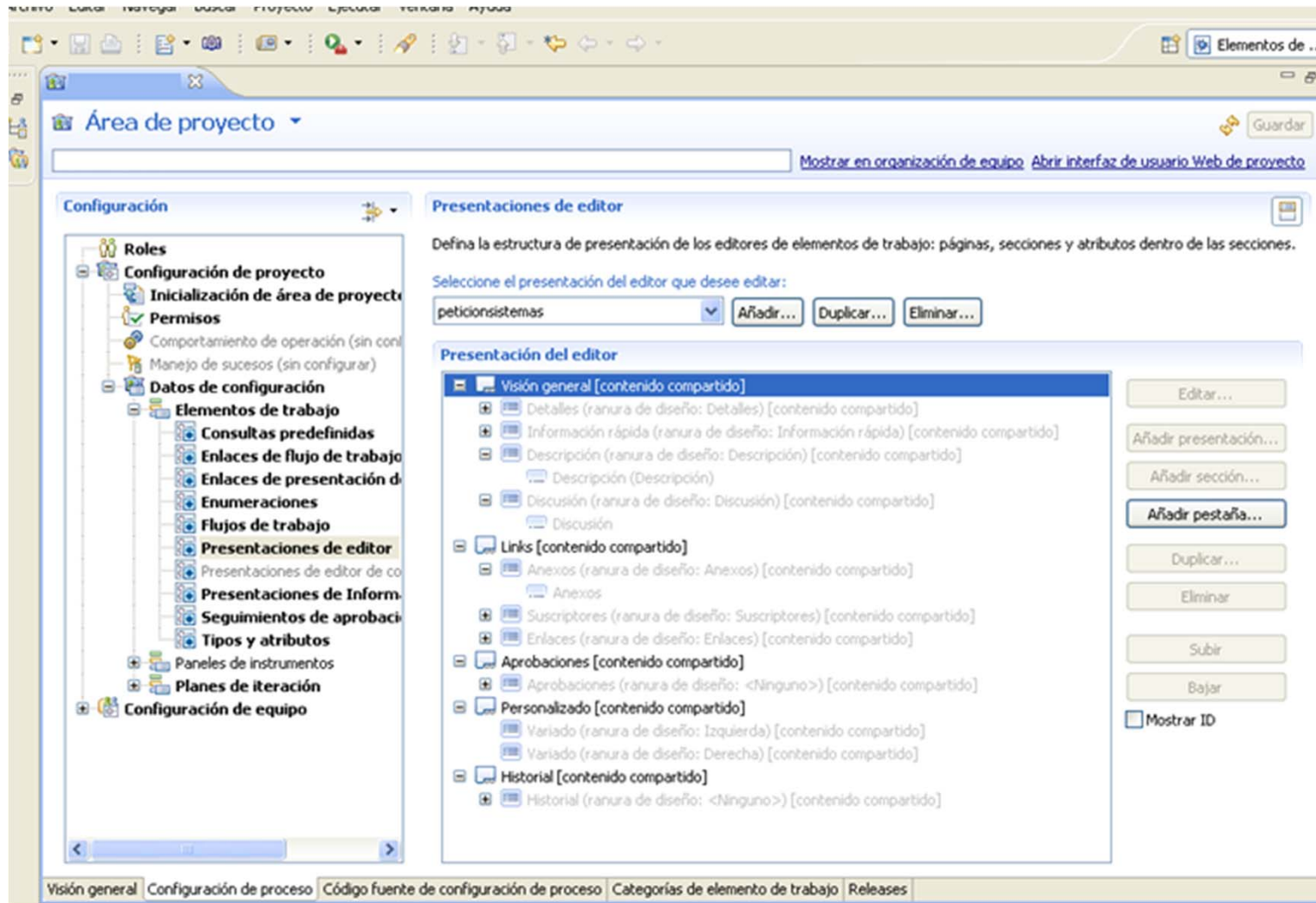
Acción de iniciar: Crear peticion sistemas Acción de resolver: Ninguno

Transiciones

De	A	Nueva petición	Petición autorizada	Petición cerrada
Nueva petición		<Ninguno>	Autorizar petición	Ninguno
Petición autorizada		Ninguno	<Ninguno>	Ninguno
Petición cerrada		Ninguno	Ninguno	<Ninguno>
Petición en proceso		Ninguno	Ninguno	Cerrar petición
Petición rechazada		Ninguno	Ninguno	Ninguno
Se requiere más info		Ninguno	Autorizar petición	Ninguno

Visión general Configuración de proceso Código fuente de configuración de proceso Categorías de elemento de trabajo Releases

Visual configurator to select the format of displayed info



Rational Team Concert Overview

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 bridge and connector

Work Items

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

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 Overview

SCM

Integrated stream management

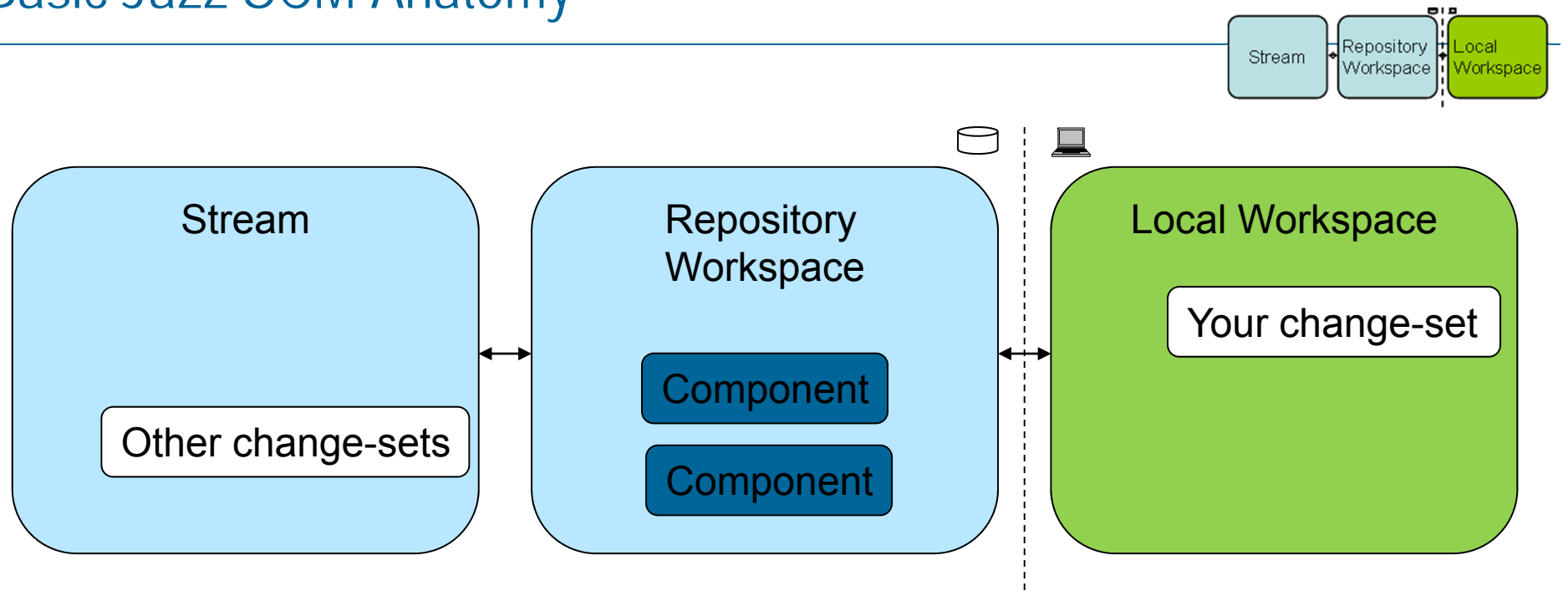
Component level baselines

Server-based sandboxes

Parallel development

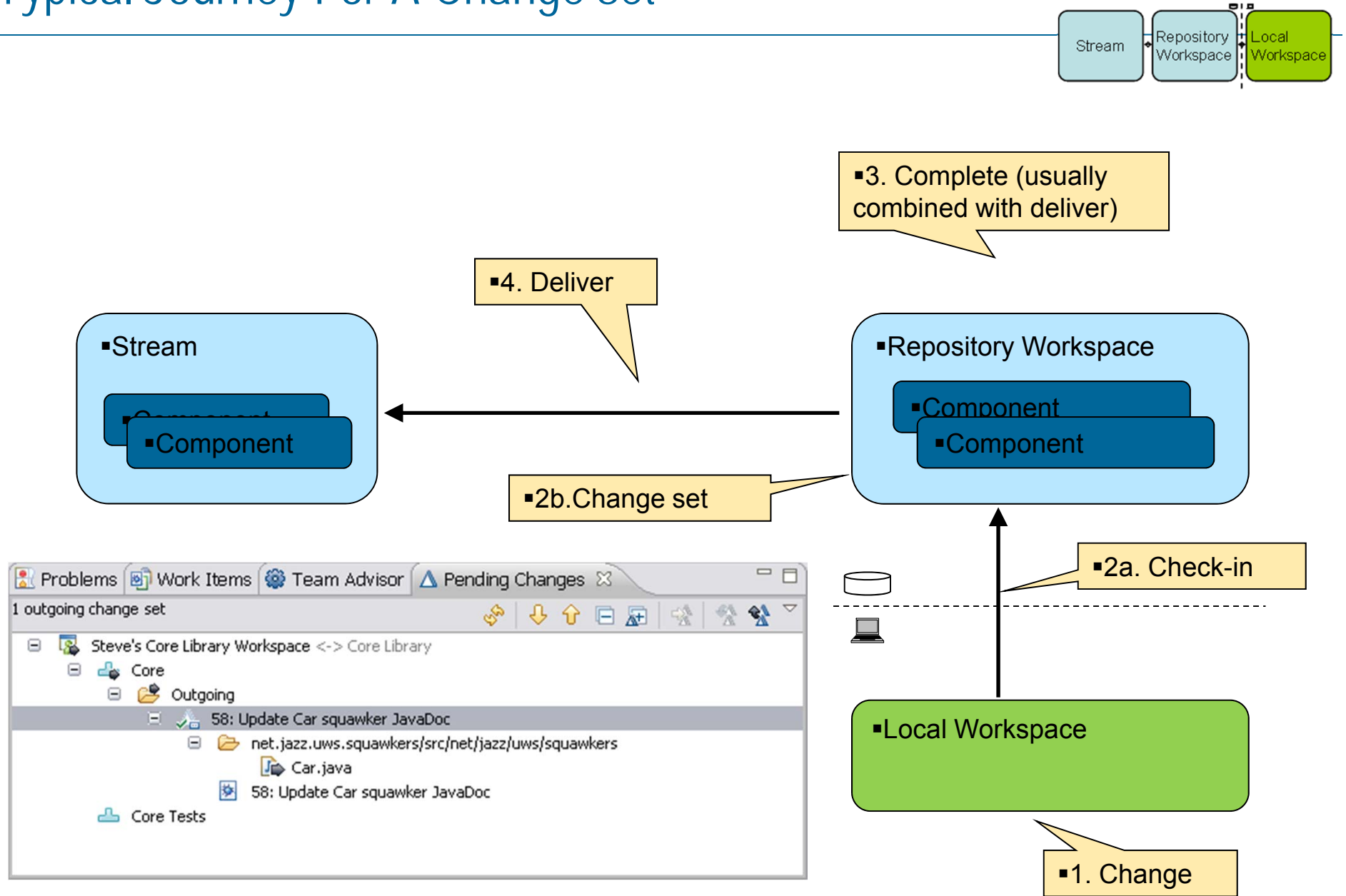
ClearCase bridge and connector

Basic Jazz SCM Anatomy



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

Typical Journey For A Change set



Change-set Delivery is Process Enabled

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

The screenshot displays the Eclipse IDE's 'Pending Changes' view. The top window shows a failed 'Deliver' operation with the error 'Missing work item or comment'. The 'Problem' section states: 'A work item must be associated with the change set or a comment must be set.' The 'Reason' section explains: '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.'

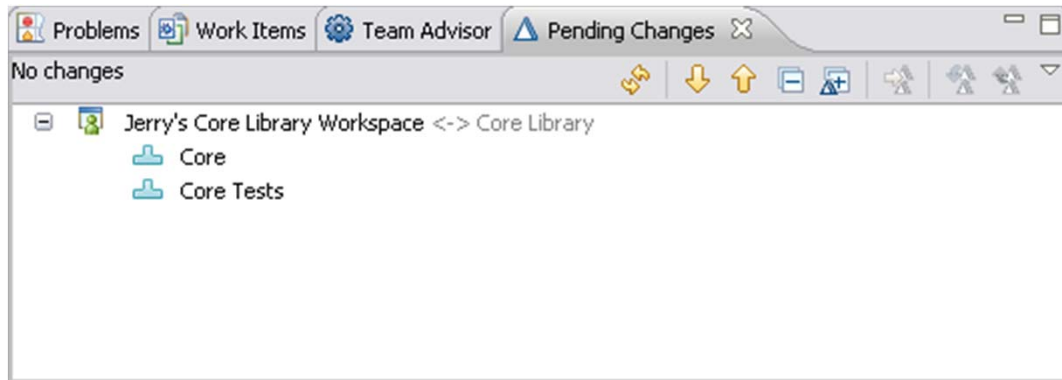
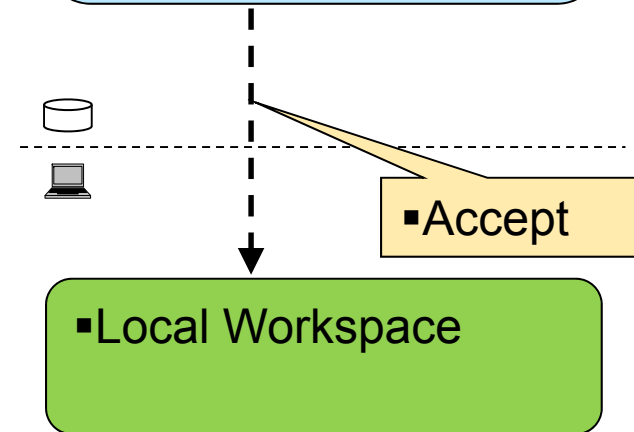
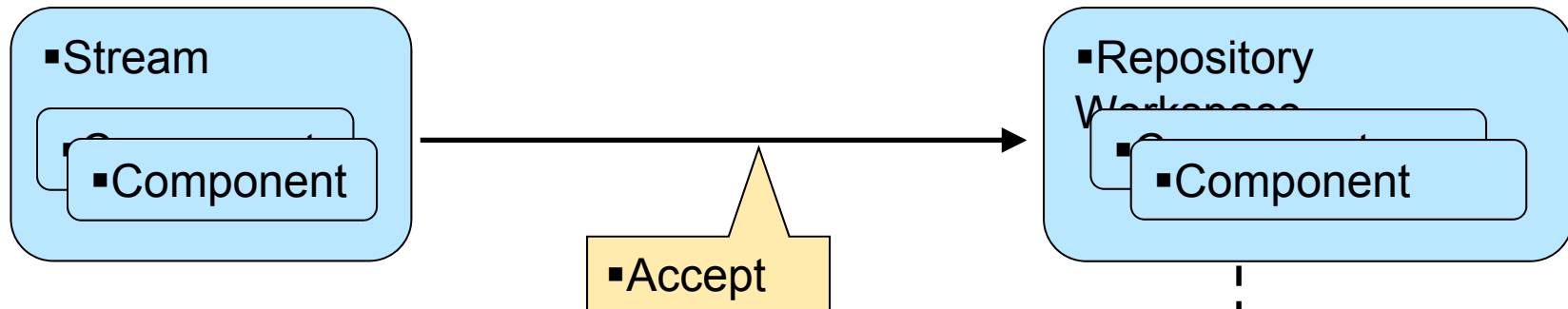
The bottom window shows a successful 'Deliver' operation. The 'State' section indicates: 'Deliver completed successfully.' The 'Details' section provides further information: 'Deliver change sets from source workspace "April's Core Library Workspace" to target stream "Core Library".' A link labeled 'why did this happen?' is visible.

A dialog box titled 'Deliver Changes' is open in the foreground, displaying a red 'X' icon and the message: 'Deliver failed. Preconditions have not been met: The associated work item (502) does not meet the specified approval requirements.' A 'Show details...' link and an 'OK' button are also present.

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



Rational Team Concert Overview

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 bridge and connector

Work Items

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

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 Overview

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

Build Management

The screenshot displays the IBM Team Explorer interface. The left sidebar shows a tree view of 'Team Artifacts' with 'Builds' expanded. The main pane shows details for a completed build: 'Build workshop.squawk.core.continuous.build'. The build status is 'Completed' with a duration of 27 seconds, starting at 1:41:40 PM and ending at 1:42:07 PM on March 3, 2008. A status trend bar shows a sequence of green and red bars. Below this, a 'Contribution Summary' lists changes, logs, repository workspace, snapshot, and work items. To the right, 'General Information' includes the requester (Zach Builder), build definition, engine, and history. At the bottom, a 'Builds' table lists 12 builds with columns for Build, Label, Progress, Estimated Completion, Start Time, Duration, and Tags.

Build definitions: steps to follow to complete the build

Support to Ant, Maven and command line build tools

Public or private builds

Automatic association with work items included in the build and change sets

View with a history of all compilations

Build	Label	Progress	Estimated Completion	Start Time	Duration	Tags
workshop.squawk.core.c...	B20080310-2231...	Completed		March 10, 2008 10:30...	34 seconds	
workshop.squawk.core.c...	B20080305-2325...	Completed		March 5, 2008 11:25...	19 seconds	
workshop.squawk.core.c...	B20080305-2324...	Completed		March 5, 2008 11:24...	33 seconds	
workshop.squawk.core.c...	B20080303-1057...	Completed		March 3, 2008 1:57:0...	24 seconds	
workshop.squawk.core.c...	B20080303-1048...	Completed		March 3, 2008 1:48:1...	28 seconds	
workshop.squawk.core.c...	B20080303-1042...	Completed		March 3, 2008 1:41:4...	27 seconds	
workshop.squawk.core.c...	20080303-1017...	Completed		March 3, 2008 1:17:5...	26 seconds	
workshop.squawk.core.c...	B20080201-1707...	Completed		February 1, 2008 8:06...	30 seconds	v20...
workshop.squawk.core.c...	B20080128-1359...	Completed		January 28, 2008 4:59...	39 seconds	wor...
workshop.squawk.core.c...	B20080123-1623...	Completed		January 23, 2008 7:23...	10 seconds	
workshop.squawk.core.c...	B20080123-1621...	Completed		January 23, 2008 7:21...	18 seconds	

Jazz database build objects

- **Build definition**

- ▶ Defines build script and other properties that specify how the build is invoked
- ▶ Defines a build schedule for automated builds
- ▶ Specifies which workspace to build from

- **Build engine**

- ▶ Represents a process on a build machine that polls the repository for build requests

- **Build request**

- ▶ Represents a request to run a build (as specified by the selected build definition) on the next available build engine

- **Build result**

- ▶ Contains the output from a build, such as execution summaries, logs, and properties

History and traceability build-work items-file changes

- **Objective:** for each build know which work items have been included and which files have been modified for each one.

Build junit I20090902-1757

✓ Completed

Duration: 48 seconds
Start Time: September 2, 2009 5:57:16 PM
Completed: September 2, 2009 5:58:04 PM

Status Trend:

Contribution Summary

Compile: 0 errors, 15 warnings
Downloads: [1 download](#)
External Links: [1 link](#)
JUnit: 31 tests, 0 failures, 0 errors
Logs: [1 log](#)
Repository Workspace: [Bill Private Workspace](#)
Snapshot: [junit I20090902-1757](#)
Work items: None included
Changes: [Show changes](#)

Bill Private Workspace (some components loaded)

- ComponentA (7: Baseline Protocolo de Seguridad)
- ComponentB (2: Baseline 1)
- Docs (3: Baseline Protocolo de Seguridad)
- J2EEArchitecture (4: Baseline Protocolo de Seguridad)
- JUnit (8: Baseline 1)

Change Explorer

Compare junit_I20

- ComponentA
 - userSWFactoryA - 70: Update component A for france subsidiary
 - ComponentA/src/componentA/com
 - FranceSubsidiary.java
- J2EEArchitecture
 - Jason Mitchell - 69: update J2EE architecture
 - StrutsFramework/src/architecture/struts/com
 - JavaStruts.java

Agenda

- Jazz platform introduction.
- Rational Team Concert introduction.
- Rational Team Concert in-depth.
 - ▶ Jazz Team Server features
 - ▶ Change management with work items and planning
 - ▶ Version control
 - ▶ Build Management
- Extensions to Rational Team Concert.
 - ▶ RTCp
 - ▶ RTCz
- Live Demo.



Rational Team Concert for System p

Rational Team Concert For Power Systems Software

Rational Team Concert

The Jazz Platform:

An Open, Services Oriented Architecture that

- Enables teams to **collaborate in real time** in the context of their work with integrated software delivery lifecycle tools
- Provides visibility into accurate **current project health information** drawn directly from actual work across the team
- Automates traceability and auditability by **managing artifacts and their inter-relationships** across the lifecycle
- Enables custom **process enactment** via process automation & definable checkpoints
- Provides an **extensible technology platform** for building products & adapters

A New Family of Products based on Jazz Technology

- Consolidates the team's assets and activities on a Team Server
- Optimized for Agile development teams
- In-place collaboration between team members
- Support and enforcement for development processes
- Transparency of status and trends through automated data-gathering and reporting
- Low administrative footprint
- ClearCase & ClearQuest Connectors

New Family of Products based on Jazz Technology for IBM i

- Jazz Team Server which runs natively on the IBM i OS and using DB2 for i
- Integration with RDP to enable team collaboration support
- Capabilities for native IBM i application developers writing in RPG, COBOL, CL, DDS, etc.
- Build system toolkit which runs natively on IBM i (Run IBM i commands and call native programs)
- Support for IBM i file systems: Libraries and IFS

Rational Team Concert for System z

Rational Team Concert For System z

Rational Team Concert

The Jazz Platform:

An Open, Services Oriented Architecture that

- Enables teams to **collaborate in real time** in the context of their work with integrated software delivery lifecycle tools
- Provides visibility into accurate **current project health information** drawn directly from actual work across the team
- Automates traceability and auditability by **managing artifacts and their inter-relationships** across the lifecycle
- Enables custom **process enactment** via process automation & definable checkpoints
- Provides an **extensible technology platform** for building products & adapters

A New Family of Products based on Jazz Technology

- Consolidates the team's assets and activities on a Team Server
- Optimized for Agile development teams
- In-place collaboration between team members
- Support and enforcement for development processes
- Transparency of status and trends through automated data-gathering and reporting
- Low administrative footprint
- ClearCase & ClearQuest Connectors

Foundation for a New Family of Products based on Rational Team Concert for System z

- Provides a Team Server that is hosted natively on z/OS or zLinux; primarily supporting multi platform **Java and C++** development
- SCM and Build support for COBOL, PL/I, HLASM.

Agenda

- Jazz platform introduction.
- Rational Team Concert introduction.
- Rational Team Concert in-depth.
 - ▶ Jazz Team Server features
 - ▶ Change management with work items and planning
 - ▶ Version control
 - ▶ Build Management
- Extensions to Rational Team Concert.
 - ▶ RTCp
 - ▶ RTCz
- Live Demo.



DEMO

Thank
YOU