



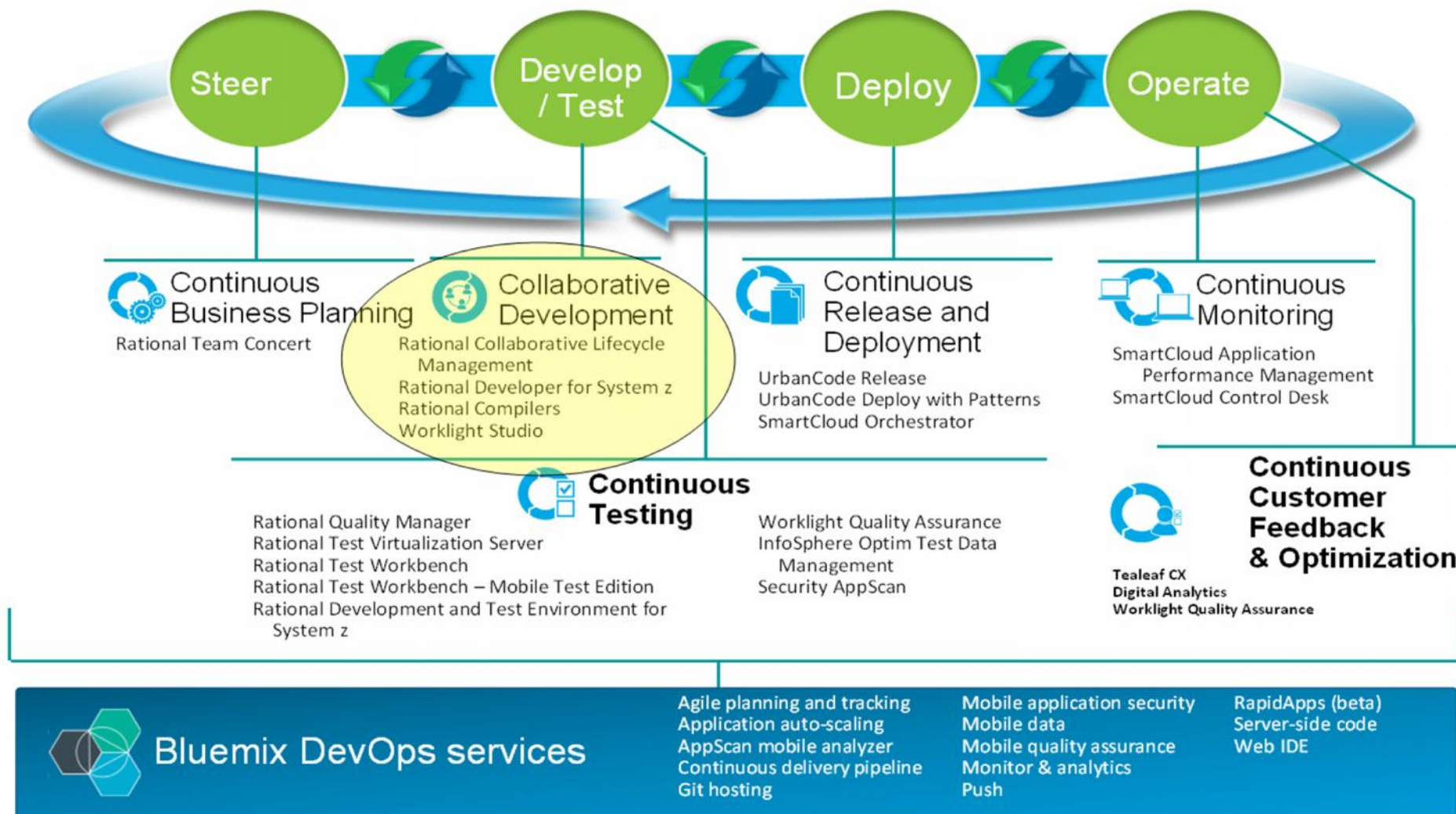
Managing Source Code and Unifying Development Teams

Rational Team Concert
for System z Users



IBM DevOps – Broad set of DevOps capabilities

Address bottlenecks across the application delivery lifecycle



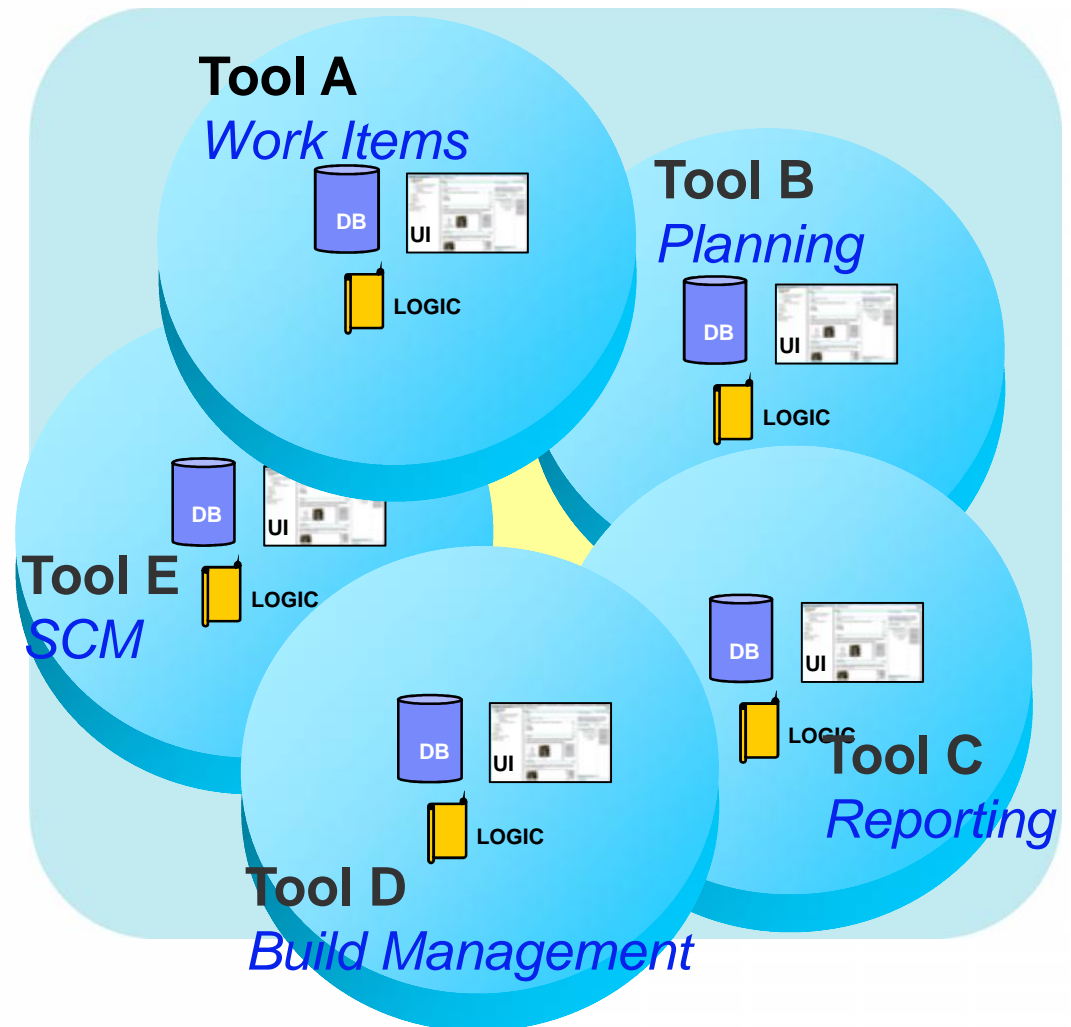
Organizations have invested in a diverse set of ALM tools...

Traditionally, each tool came with its own:

- Logic – Workflow, process, search, query, scale, security and collaboration
- Storage – Availability, traceability
- Privacy, backup/archive

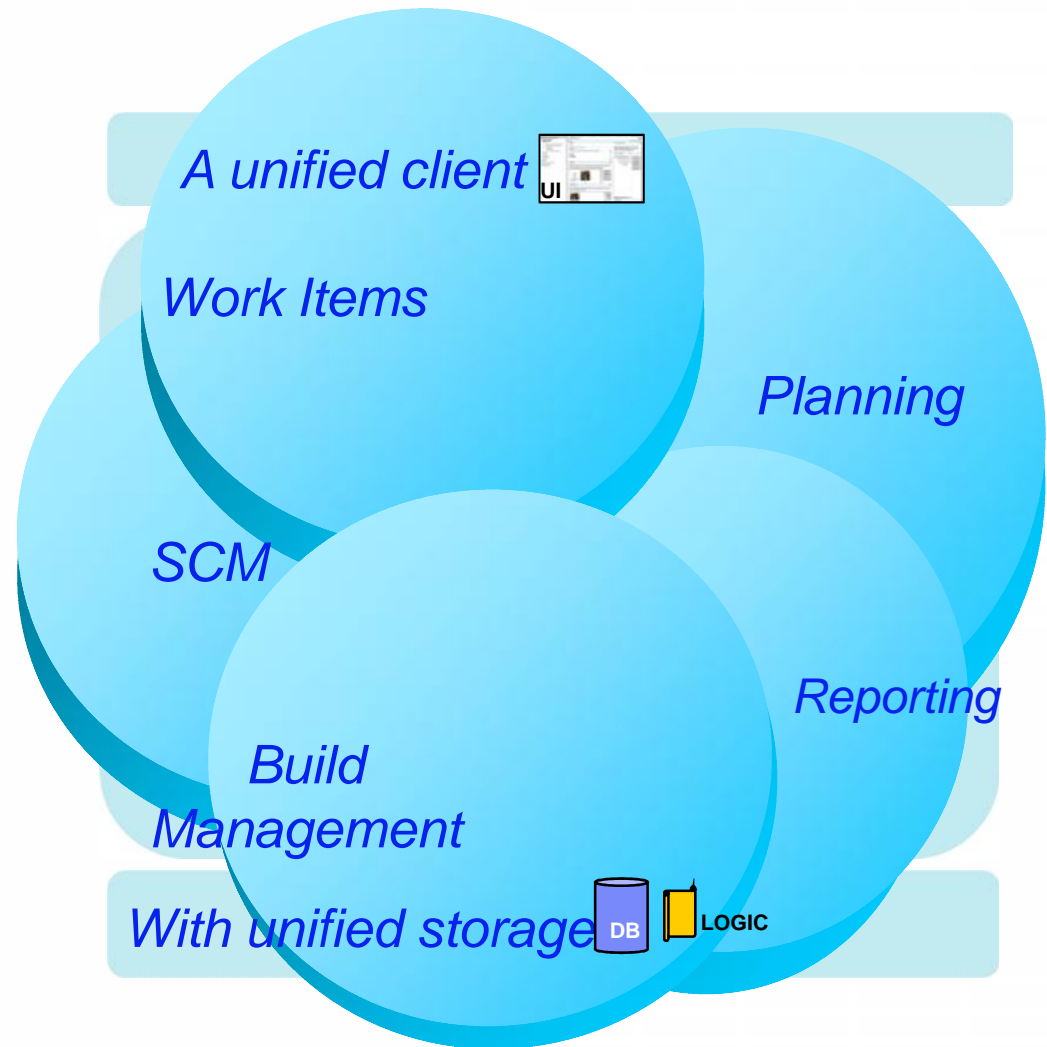
Resulting in:

- UI – Web and desktop presentations of views and tasks.
- Brittle integrations
- Silos everywhere
- High cost to maintain and administer
- Proprietary API's



How do you solve this?

- Unified Work Items, Planning, SCM, Build Mgmt, Reporting
- Single UI, storage and process
- Reduces Cost of Ownership
 - Administrative costs
 - Training/Adoption
 - Customizations
- Increases visibility & collaboration

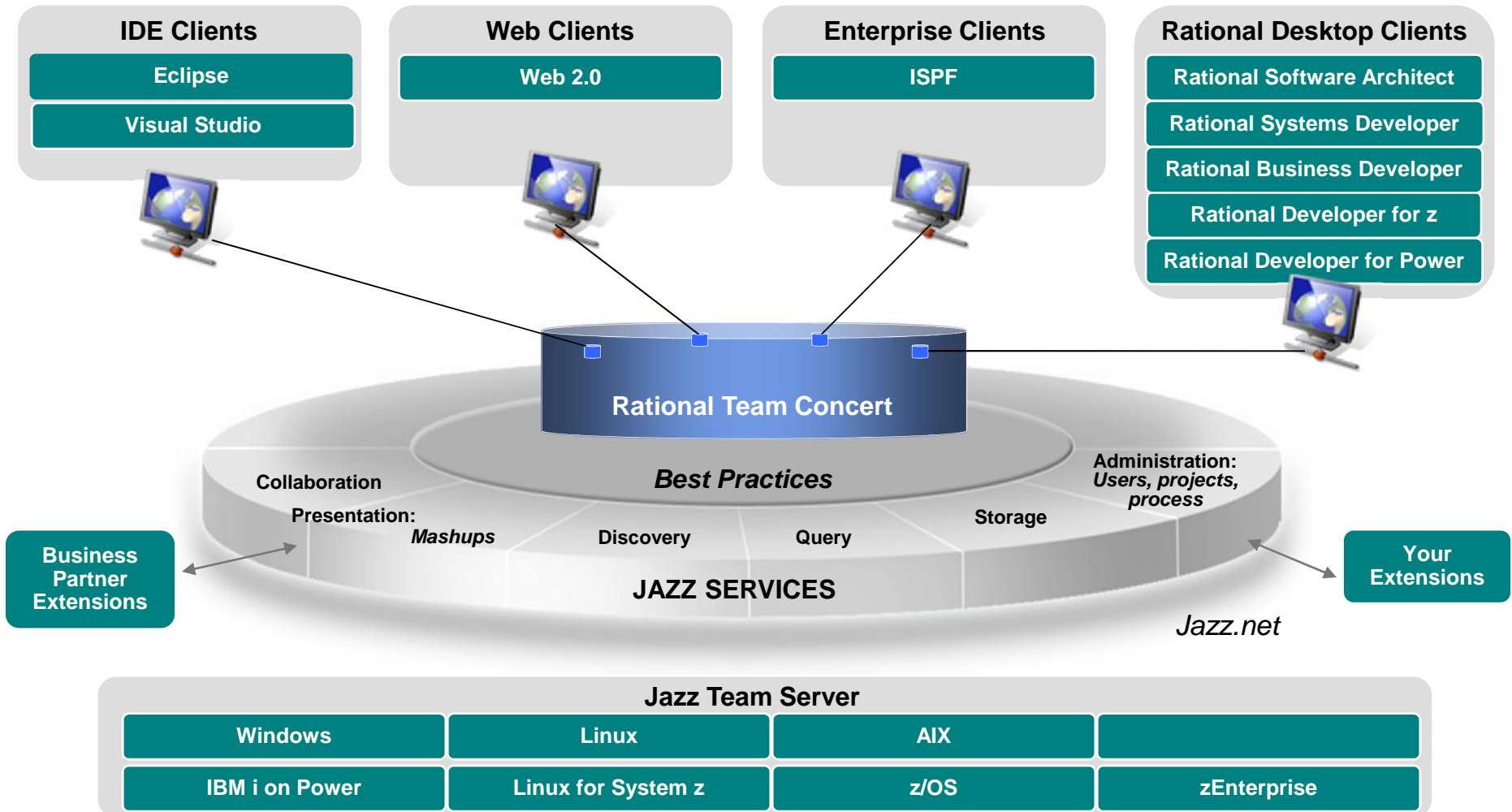


One Tool!

Rational Team Concert - an open, extensible architecture



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



RDz - Eclipse Based zOS Development



- Access resources on zOS
- Manage PDS (PDSe) – allocate, migrate, search, add members
- Intelligent editors for COBOL, PL1, HLASM, JCL, C/C++
- Access JES (submit jobs, see job log)
- TSO Command shell with content assist
- USS file access & command shell

The screenshot displays the IBM Rational Developer for System z (RDz) interface. The main editor window shows a COBOL program with the following code:

```
Line 31 Column 1 Insert
PURPOSE.
THIS PROGRAM IS DEFINED TO TEST A NUMBER OF THE
APPLICATION TESTING COLLECTION AND DEBUG TOOL FUNCTION:
...
CONFIGURATION SECTION.
OBJECT-COMPUTER.
IBM-370.
IBM-370.
...
DATA DIVISION.
...
* WORKING STORAGE
...
WORKING-STORAGE SECTION.
01 TDPARM1 PIC 99 VALUE 5.
01 TDPARM2 PIC 99 VALUE 2.
01 P1PARM1 PIC 99 VALUE 0.
01 BASE-YEAR PIC 99 VALUE 99.
```

The interface includes a Project Browser on the right showing a tree structure of files and folders, including 'ATCDEMS.cbl'. The bottom of the screen features a 'z/OS File System Mapping' table with the following data:

Mapping Criterion	Workstation File Extension	Transfer Mode	Host Code Page	Local Code Page
**MFS	mfs	text	IBM-037 (inh...	Cp1252 (inhe...
**COBOL	cbl	text	IBM-037 (inh...	Cp1252 (inhe...
**COBCOPY	cpy	text	IBM-037 (inh...	Cp1252 (inhe...
**PL1	pli	text	IBM-037 (inh...	Cp1252 (inhe...
**ASSEMBLE	asm	text	IBM-037 (inh...	Cp1252 (inhe...
**OBJ	obj	binary	IBM-037 (inh...	Cp1252 (inhe...
**LOAD	exe	binary	IBM-037 (inh...	Cp1252 (inhe...
**TEXT	...	text	IBM-037 (inh...	Cp1252 (inhe...

A Closer Look

Planning

- Integrated release/iteration planning
- Effort estimation & progress tracking taskboards
- Out of the box process templates: formal or agile

Project Transparency

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

SCM

- Component based SCM enables reuse across projects
- Change set based for easy addition or removal of features
- Server-based sandboxes
- Use RTC SCM or co-exist with a legacy SCM

Work Items

- Defects, enhancements
▪ and conversations
- View and share query results
 - Support for approvals and discussions
 - Query editor interface
 - Connects to ClearQuest or Change

Build

- Automated Work item and change set traceability
- Build definitions for team
▪ and personal builds
- Local or remote build servers
- Multi-level continuous integration
- Can integrate with BuildForge

Jazz Team Server

- Single structure for project related artifacts
- World-class team on-boarding / offboarding including team membership, sub-teams and project inheritance
- Role-based operational control for flexible definition of process and capabilities

- Team advisor for defining / refining “rules”
- and enabling continuous improvement
 - Process enactment and enforcement
 - In-context collaboration enables team members to communicate in context of their work



Working to standardize the way software lifecycle tools share data

What is OSLC?



Open Services for Lifecycle Collaboration
Lifecycle integration inspired by the web

- Community Driven – @ **open-services.net**
- Specifications for numerous disciplines
 - Such as, ALM, PLM and DevOps
 - Defined by scenarios – solution oriented
 - Inspired by Internet architecture
- A different approach to industry-wide proliferation



How does it work?



Inspired by the web



Free to use and share



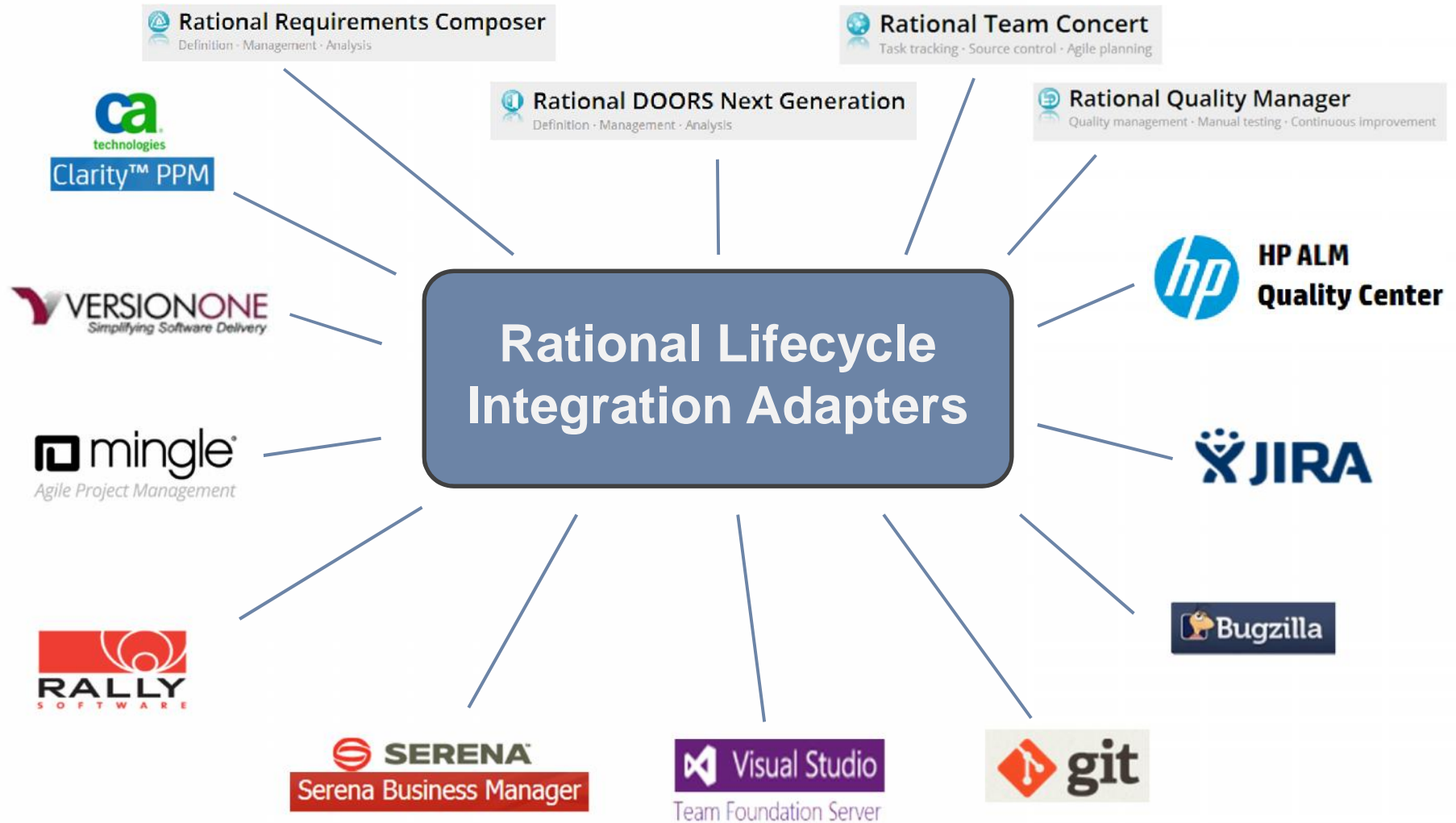
Changing the industry

GET INVOLVED AND CONTRIBUTE!

IBM Rational Lifecycle Integration Adapters



Integrate Rational with existing DevOps stacks



Work is done against a Work Item

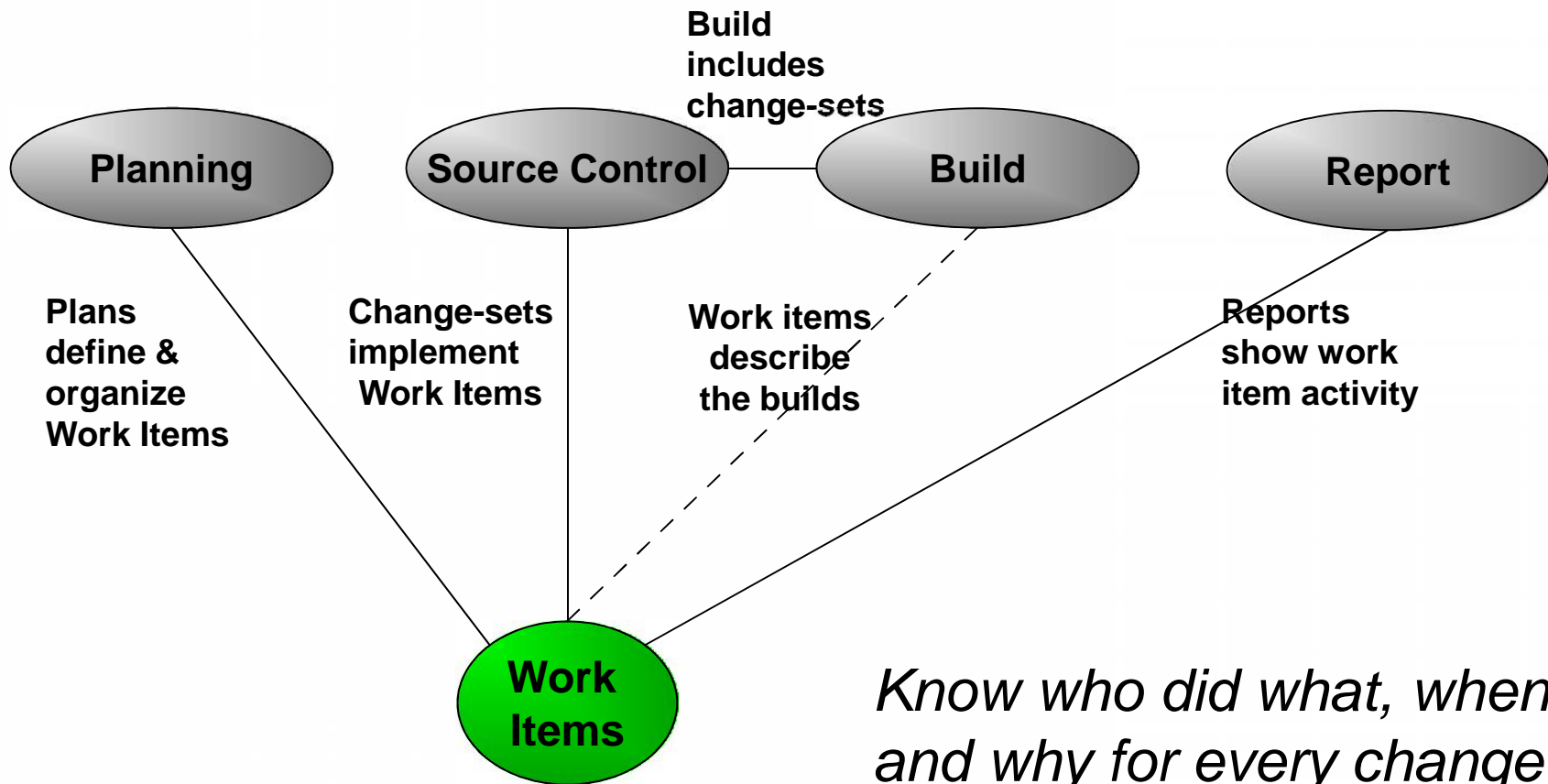


The screenshot displays the IBM Rational Team Concert web interface in Mozilla Firefox. The browser address bar shows the URL: `https://zserveros.demos.ibm.com:9444/jazz/web/projects/Bank Account Project 79#action=com.ibm.team.wor`. The application header includes 'Rational Team Concert', 'Project Manager 44', and 'Log Out'. The main navigation bar contains 'Dashboards', 'Project Areas', 'Work Items', 'Plans', 'Source Control', 'Builds', and 'Reports'. The current page is titled 'New Work Item - IBM Ration...' and shows a form for creating a work item.

The work item is titled 'Enhancement <22:09:03> *'. The summary is 'Enhance JK Bank application to support local currency'. The form includes the following fields:

- Summary:** Enhance JK Bank application to support local currency
- Overview:** Links, Approvals, History
- Details:**
 - Type: Enhancement
 - Severity: Normal
 - Found In: Unassigned
 - Creation Date: Now
 - Created By: Project Manager 44
 - Project Area: Bank Account Project 79
 - Team Area: RTCzPOT 79 Dev Team
 - Filed Against: COBOL Account
 - Tags: (empty)
 - Owned By: Project Manager 44
 - Priority: Unassigned
 - Planned For: Unassigned
 - Estimate: (empty)
 - Correction: (empty)
 - Time Remaining: (empty)
 - Due Date: (empty)
- Quick Information:** No Links.

The left sidebar contains 'Create Work Item', 'Create Query', 'Unsaved Work (1)' (listing the current enhancement), and 'Shared Queries' (listing predefined queries like 'All Open Stories', 'All Stories', etc.). The bottom status bar shows 'Done'.



Team and project breakdown



Easily re-balance and re-organize.

Plan: Sprint 3 [Sprint 3] - IBM Rati... Rational Team Concert - Call Centre

Rational Team Concert Jean-Michel Lemieux | Log Out

Dashboards Project Areas Work Items **Plans** Source Control Builds Reports Admin | Call Center

Plans Related to Timeline 'Main Development [Project Timeline]' >

Previous | 4 of 5 | Next

Sprint 3

Owner: Call Center | Iteration: Sprint 3 (11/30/09 - 1/15/10) | 5 Closed | 19 Open

Overview | **Planned Items** | Charts

Progress: 36/171 | -51.75 h Estimated: 79%

View As: Work Breakdown

Team Member	Task	Duration	Status	Priority	Estimate
Jean-Michel Lemieux Closed Items: 1 Open Items: 6 Load: 57/132.75 +75.5 h Estimated: 83%					
	Backport call hang-up fix to Sprint 1	-	New	High	157
	Fix up unit tests to avoid having to re-initialize for each suite	2 days	New	High	156
	Update blocked listed table	5 minutes	New	High	155
	Prepare sample data for call demos	0/27 h	New	Medium	144
	Problem with build I20091217-1336	4 hours	Done	High	143
	Adding new content to the server	2 days	New	High	140
Jennifer Ginness Closed Items: 1 Open Items: 6 Load: 32/132.75 +100.75 h Estimated: 67%					
	Prepare sample data for call demos	0/27 h	New	Medium	144
	Allow adding features via addons that can be added to a running call center application	32/64 h	In Progress	High	87
	Allow home handlers to call in and take calls	-	New	High	84
Markus Kent Closed Items: 0 Open Items: 3 Load: 44/132.75 +88.75 h Estimated: 100%					

Transparency and Status for Everyone

Every stakeholder knows project status without having to ask

Jazz Project Description

Process enabled team collaboration platform

The goal of the Jazz project is to build a scalable, extensible team collaboration platform for seamlessly integrating tasks across the software lifecycle. The new technology is called the Jazz Platform. The Jazz project is defining the overall architecture of the Jazz Platform, and building an initial set of Jazz components that plug in to the Jazz Platform.

The work in the Jazz project is divided into development work and maintenance work. Development work produces a consecutive line of releases. Maintenance work produces bug fix releases of the released produced by the development work.

Development as well as maintenance work is overseen by the PMC. The team members listed in this project area constitute the PMC.

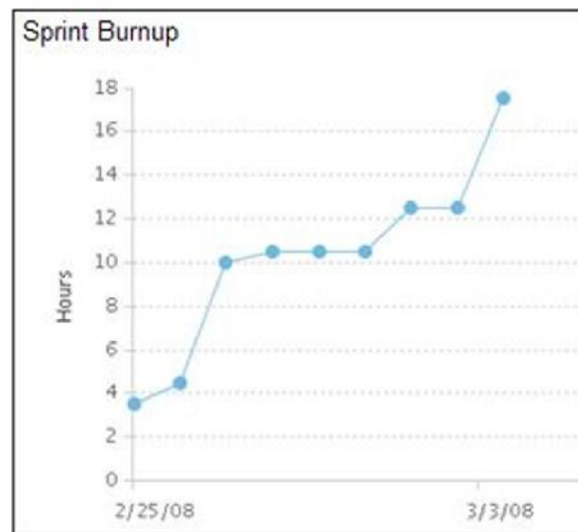
Jazz Project Members (11)

CQ Connector Account	connector
Dejan Glazic	dashboardadmin
Erich Gamma	dashboardadmin, projectadmin, PMC, contributor
Jean-Michel Lemieux	dashboardadmin, projectadmin, PMC, contributor
John Vasta	connectoradmin
Kai-Uwe Maetzel	PMC, contributor
Kevin Haaland	PMC, contributor
Kim Moir	projectadmin
Peter Klenk	PMC, contributor
Scott Rich	PMC, contributor

Page 1 of 2

Jazz Project Teams (50)

- ▼ Jazz Development
 - Agile Planning
 - Build
 - ▶ ClearCase Connector
 - ClearQuest Connector
 - Community Site
 - Dashboard
 - ▶ Developer Test Team
 - Improv
 - ▶ Incubator
 - Install
 - Jumostart
 - Process
 - Release Engineering
 - Reports
 - ▶ Repository
 - ▶ Source Control
 - System Test
 - UI Design
 - ▶ User Assistance
 - Web UI Foundation
 - WorkItem
 - X-Team Web UI
- ▶ Jazz Maintenance



Development dashboard

The screenshot shows a web browser displaying the Rational Team Concert development dashboard for a project named 'Call Centre'. The browser address bar shows the URL: `https://localhost:9443/jazz/web/projects/Call%20Center#action=com.ibm.team.dashboard.viewDashboard&`. The dashboard interface includes a navigation menu with options like 'Dashboards', 'Project Areas', 'Work Items', 'Plans', 'Source Control', 'Builds', and 'Reports'. The main content area is titled 'Call Centre' and has tabs for 'Overview', '1.0 Planning', '1.0 Development', and '1.0 Testing'. A left sidebar contains sections for 'Recently Viewed', 'My Dashboards', and 'Public Dashboards'. The main dashboard displays several widgets: 'Open Track Build Items (3)' with a list of integration tracking items; 'Call Center Builds' showing a list of build statuses (Succeeded, Failed) with timestamps; 'Integration Build Health' with a line graph showing the 'Number of Tests' over time; 'All Defects (19) Status' with a horizontal bar chart showing 'Resolved' and 'New' defect counts; and a table of build results for 'callcenter I20091223-0906'. A tooltip is visible over one of the build entries, showing a 'Completed' status with a red 'X' icon and the following details: Start time: Dec 23, 2009 9:06:29 AM; End time: Dec 23, 2009 9:06:51 AM; Duration: 23 seconds; Tags: <none>; Time in queue: 1 seconds. The tooltip also includes an 'Open Item' link.

Establish the rules of the road & help your team work with them

The screenshot displays the Rational Team Concert interface. On the left, a tree view shows the project configuration structure, including Roles, Project Configuration, Permissions, Configuration Data, Team Configuration, and Development Lines. The main area shows the configuration for the 'Deliver' operation, with a table of preconditions and follow-up actions. A red box highlights the 'Preconditions and follow-up actions are configured for this operation' checkbox and the list of preconditions: Clean Workspace, Descriptive Change Sets, and Require Work Item Approval. A blue callout bubble points to this area with the text: "Configurable Rules can be run to enforce team or organizational standards. Can be refined 'on the fly' for an individual team for continuous improvements".

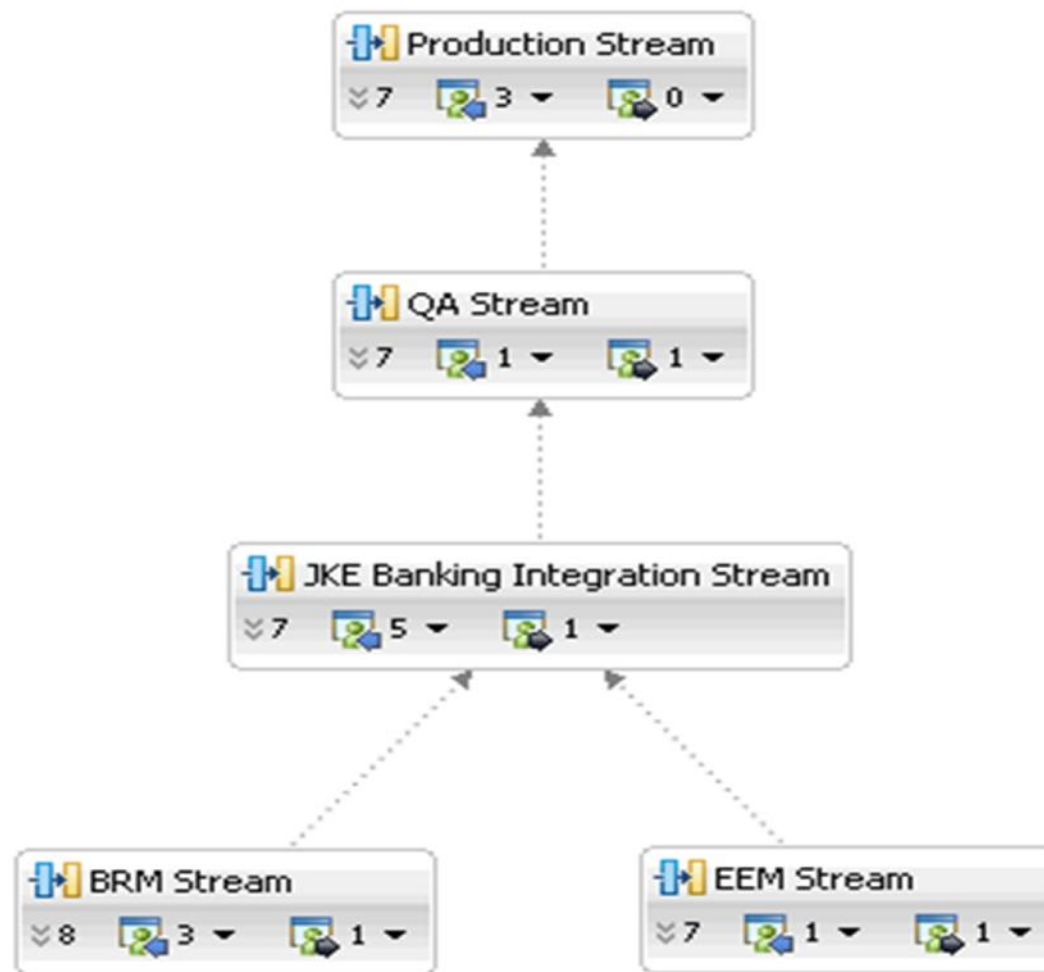
On the right, the Team Advisor window shows a list of operations: Deliver (1 error) [client] 11:38:49 AM, No Unused Imports, Clean Workspace, Descriptive Change Sets, and <No Description>. A red 'X' icon indicates an error. A red box highlights the 'Deliver (client)' operation. A blue callout bubble points to this area with the text: "Quick Fixes" can be specified to simplify corrective action. Out of the box processes include Scrum, iterative, formal or any process. Below the error message, three quick fix options are listed: Associate Existing Work Item, Associate New Work Item, and Associate and Try Again (experimental).

At the bottom, another Team Advisor window shows a successful 'Deliver' operation [client] 11:52:46 AM. A green checkmark icon indicates success. A blue callout bubble points to this area with the text: "Deliver: Checks whether the selected change sets are ready for delivery according to the team's development process. If the team defines strictly enforced advice which hasn't been followed, delivery will be blocked. Source Workspace: TC Workspace Target Stream: Eval Result: no problems found".

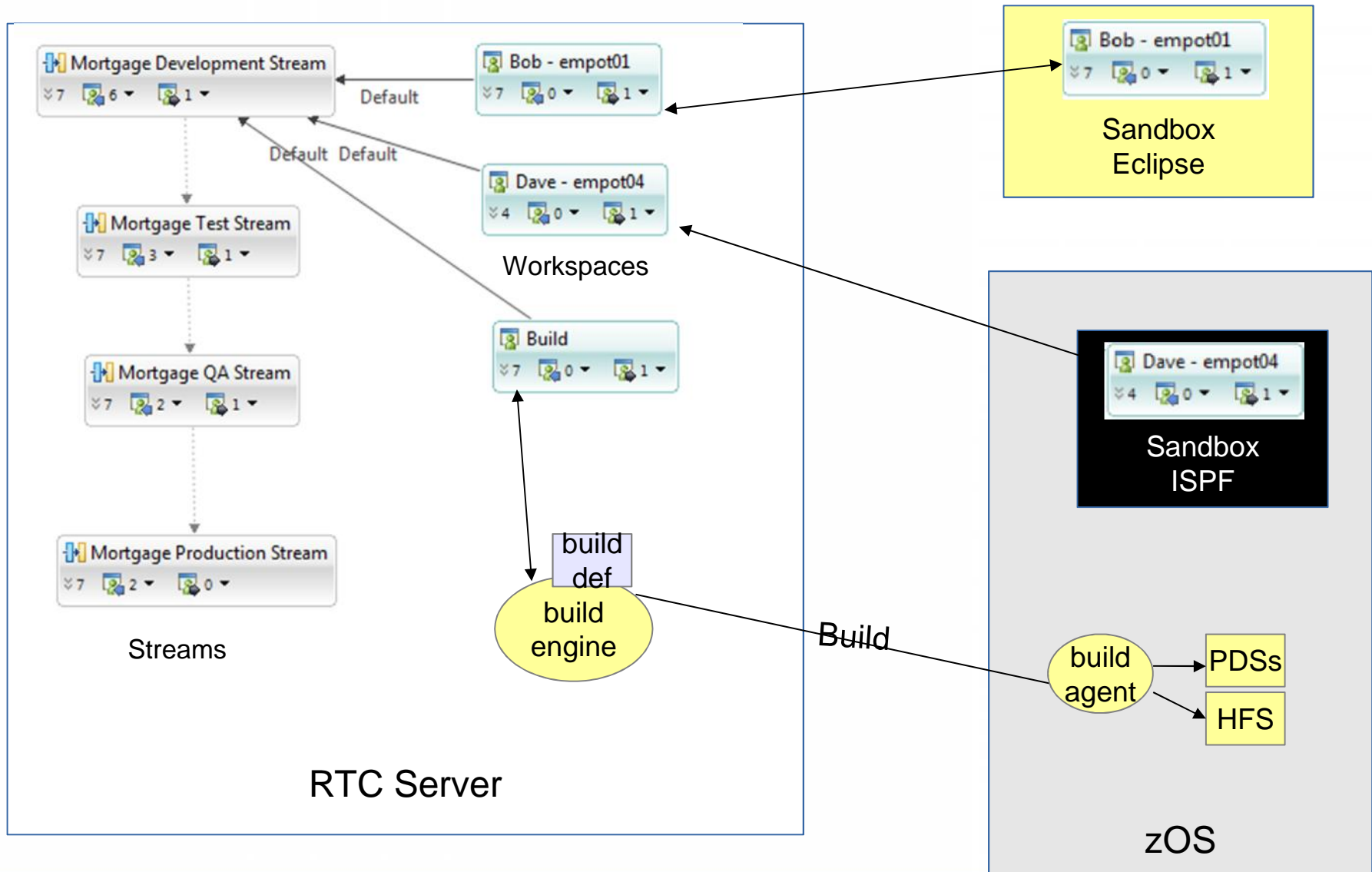
Streams



Provide flexibility to meet any process



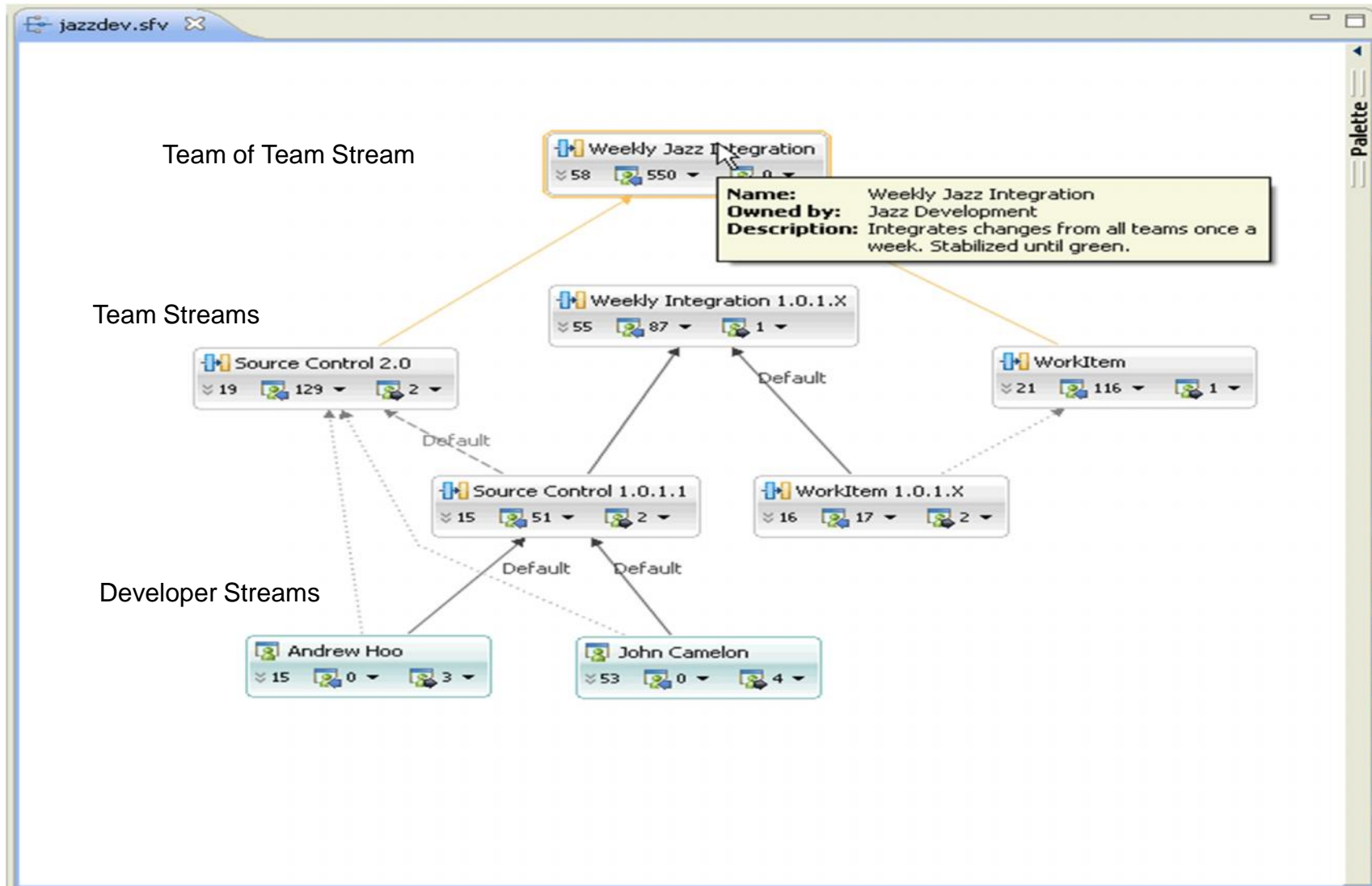
Facilitates simultaneous development



Work on multiple product versions and teams



and maintain order



Parallel development needs visibility, control, and assistance to merge code.

The screenshot displays the IBM Rational Developer for System z interface. The main window shows a comparison of two COBOL source files, both named `EPSMPMT.cbl`. The left pane is labeled "EPSMPMT.cbl (editable)" and the right pane is "EPSMPMT.cbl @ Deb BRM Stream Workspace (read-only)".

The source code in both panes is as follows:

```

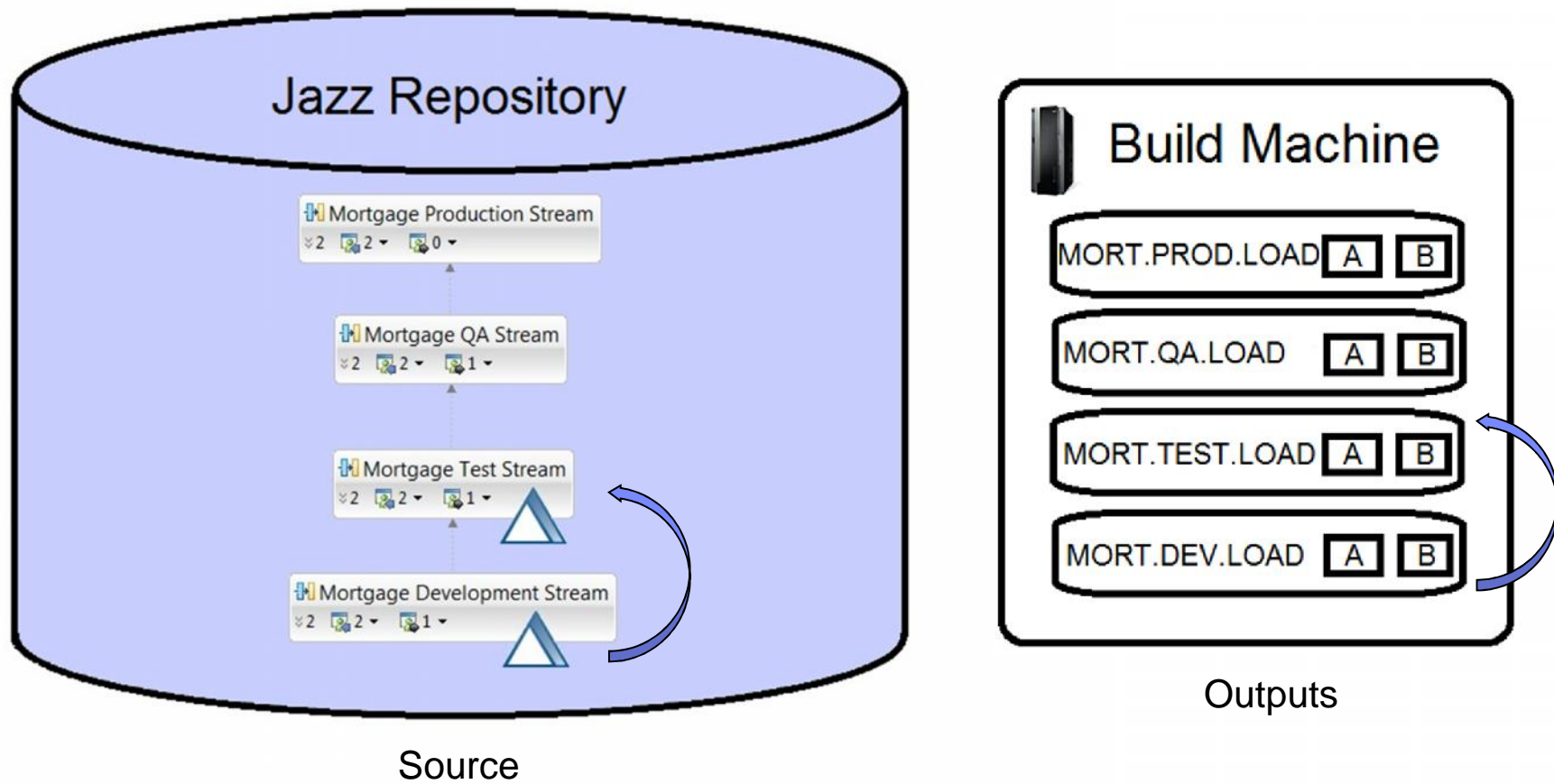
*
* (C) 2008 IBM - Jim Hildner
* Bob's Change.
* Deb's Change.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. FLEX-ES.
OBJECT-COMPUTER. FLEX-ES.
  
```

The comparison highlights the difference between the two versions: the left version has a line for "Deb's Change." which is missing in the right version. The right version has a line for "ENVIRONMENT DIVISION." which is missing in the left version.

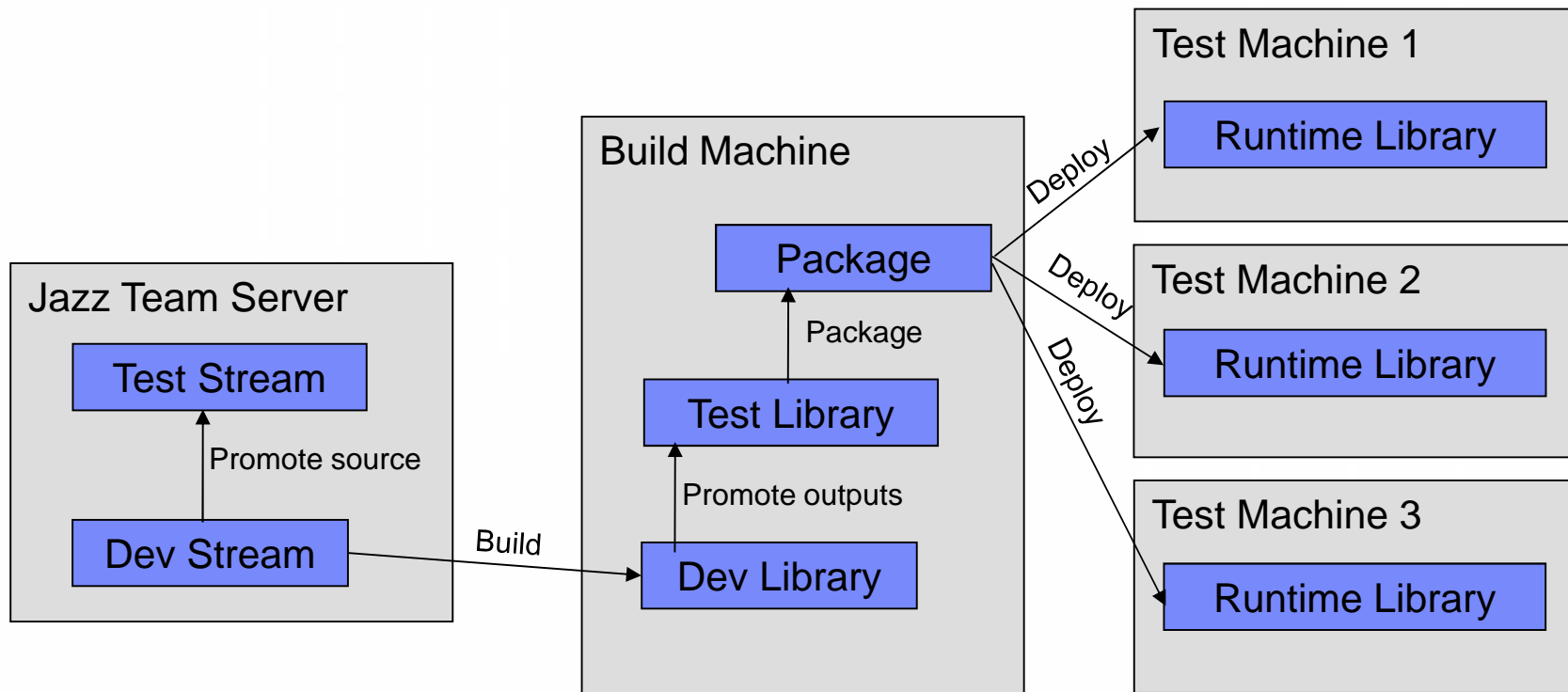
The bottom of the interface shows a "Work Items" panel with a tree view of the project structure:

- 1 outgoing change set
 - Deb BRM Stream Workspace
 - BRM Stream
 - Banking Logic
 - Build
 - C# UI
 - Cobol (1 merged)
 - Outgoing (1 merged)
 - 78: Implement back end (cobol) - investors specify minimum criteria (1 merged)
 - MortgageApplication-EPSMPMT/zOSsrc/COBOL (1 merged)
 - EPSMPMT.cbl (Merged)
 - 78: Implement back end (cobol) - investors specify minimum criteria
- Database

- Flow source code changes and build outputs through the development hierarchy



- Package build outputs and deploy to various runtimes



Still want a green screen?



Don't abandon your ISPF experienced developers!

```
Session C - [24 x 80]
File Edit View Communication Actions Window Help
Host: mvs114.rtp.raleigh.ibm Port: 23 LU Name: Disconnect
Menu Help
Rational Team Concert Primary Option Menu
Option ==>
0 Settings Terminal and user parameters ***** Logged in *****
1 Connections Work with connections to source User ID . . . : rebecca
2 Workspaces Work with repository workspaces Server . . . : 9.65.245.31:9
3 Edit Work with source data Project . . . : JKE Banking (
4 Build Work with build options Release . . . : 4.0 RC2
X Exit Exit client
F1=Help F2=Split F3=Exit F4=Expand F7=Backward F8=Forward
F9=Swap F10=Actions F12=Cancel
MPC C 04/014
Connected to remote server/host mvs114.rtp.raleigh.ibm.com using lu/pool TCP01330 and port 23
```

- Full function SCM client
- Uses usual ISPF editor
- to edit files and follows common ISPF panel standards for all
- panels and actions
- Designed to be used with the Web client for planning and work items



www.ibm.com/software/rational

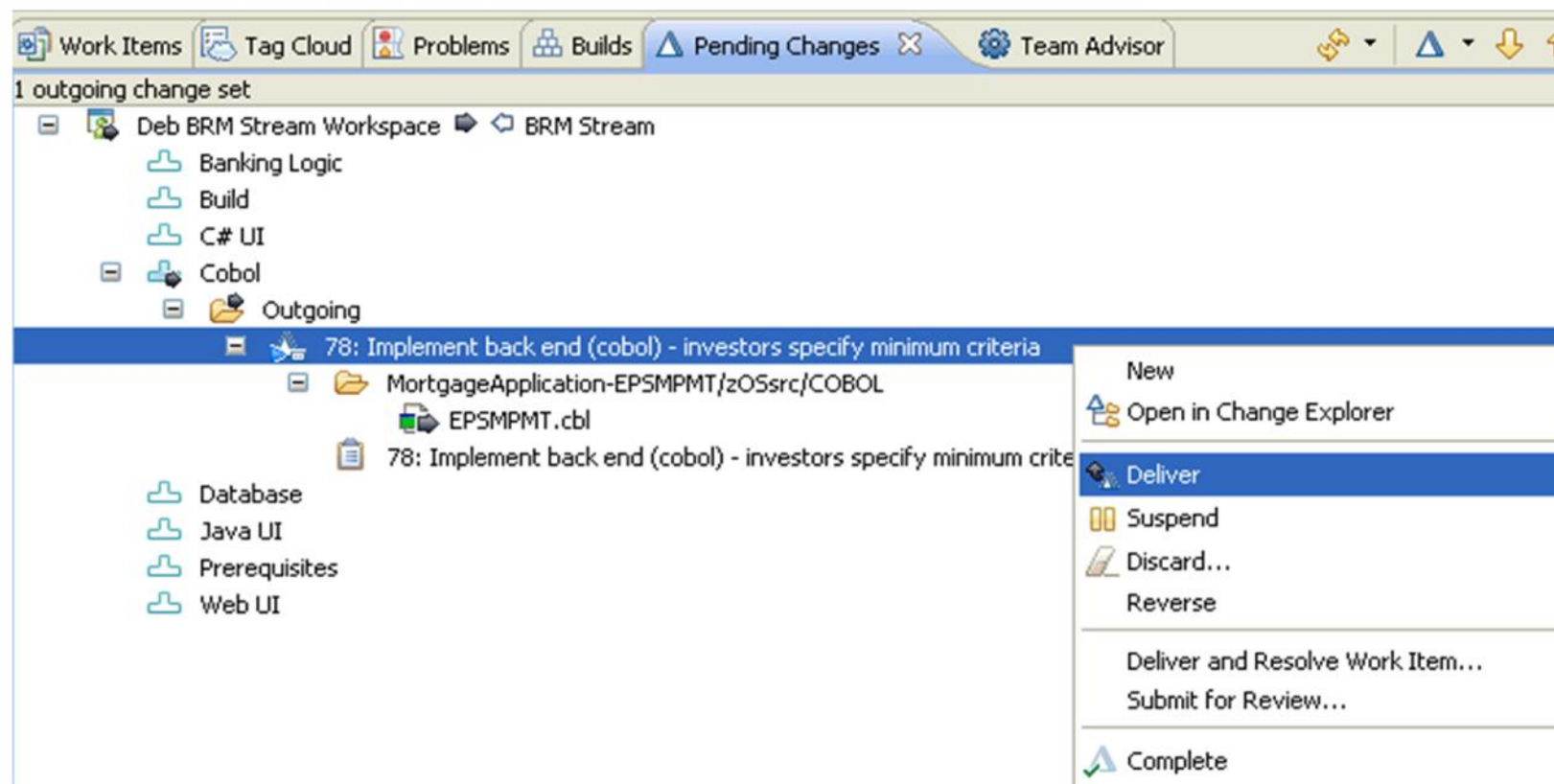
QUESTIONS

The word "QUESTIONS" is written in large, white, 3D-style capital letters. Each letter is hollow, and inside each letter, there is a small, square photograph of a different person's face. The people shown are diverse in age and ethnicity, all appearing to be in professional or office settings. The letters have a slight shadow beneath them, giving them a three-dimensional appearance.

www.ibm.com/software/rational

Backup Material

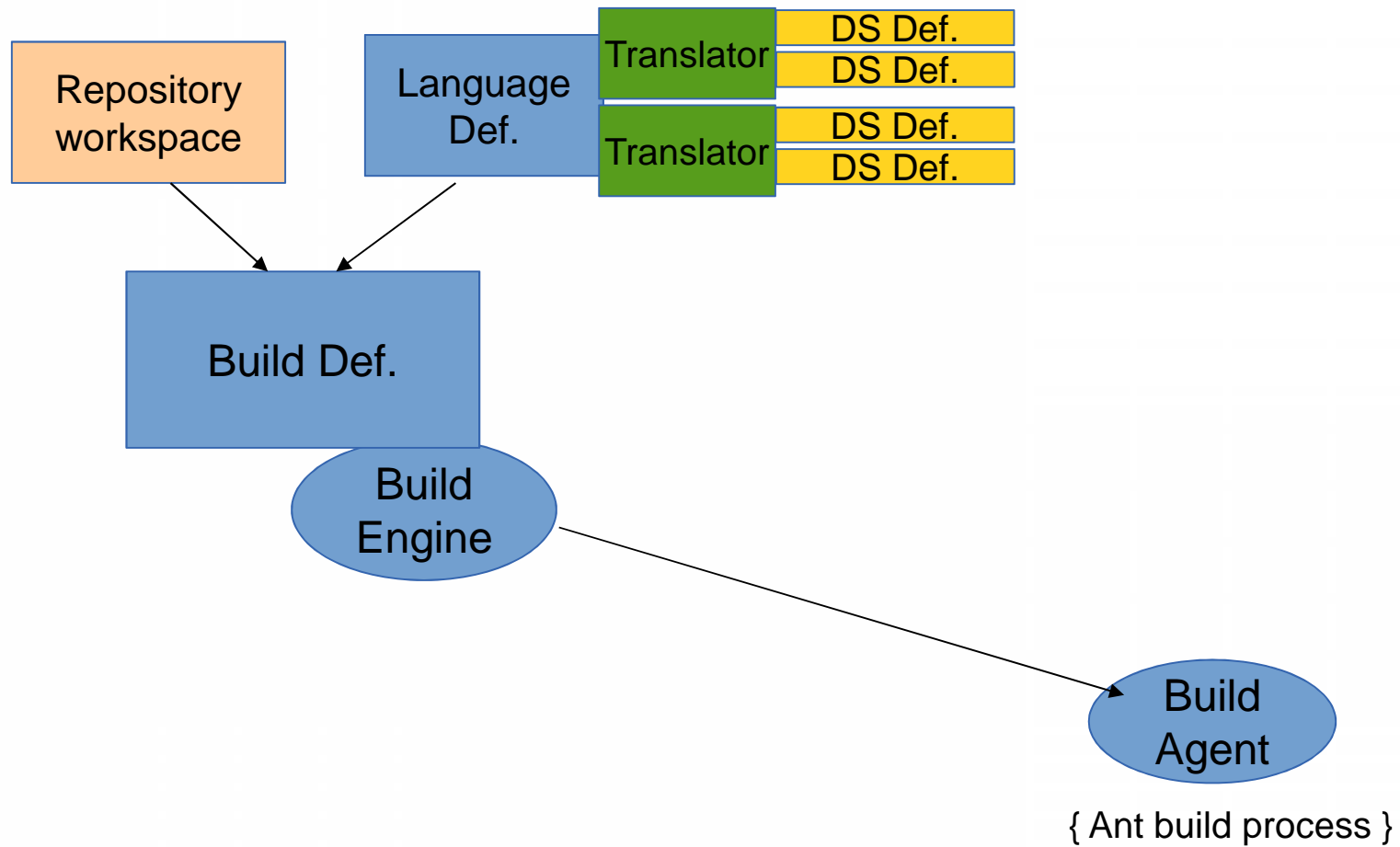
Delivery to Stream



General Compilation Structure

1. Define Data Definitions
2. Translators
3. Language Definitions
4. Build Definitions for the build

Build process details



- Build Administrator
 - Creates Data Set Definitions
 - Used to describe the characteristics of the MVS data sets that will be used in the build
 - Creates Translators that describe some type of processing to be done on a buildable file
 - Similar to a JCL job step
 - Defines
 - Compiler or translator to be used in this step
 - DD allocations needed by the compiler
 - Maximum return codes
 - Creates Language Definitions
 - Collections of translators that can be applied to a buildable file
 - Creates a Build Definition describing the build

Data Set Definitions



- Container for information regarding a data set on the z/OS system.
- All datasets referenced by a build process must have a corresponding Data Set Definition.

Data Set Definition Save

Name:

General
Specify general information about this data set definition.

Description:

Usage

- Destination data set for a zFolder
- New data set used for build
- Existing data set used for build
- Temporary data set used for build

Data set name: Member:

Add data set prefix from build definition to data set name

Data Set Characteristics
Specify the characteristics to be used for the new data set.

Data class:

Storage class:

Management class:

Volume serial:

Generic unit:

Space units:

Primary quantity:

Secondary quantity:

Directory blocks:

Record format:

Record length:

Block size:

Data set type:

General

Translators



Translator Save

Name: JKE COBOL compilation (CICS&DB2)

Call Method

Called program

Data set definition: JKE COBOL compiler Browse...

Default options: LIB,DYNAM,CICS,SQL,ADATA,EXIT(ADEXIT(ELAXMGUX))

DD names list:

ISPF command or exec

Command/member:

TSO command or exec

Command/member:

Maximum return code: 4

Data Set Properties

DD concatenations:

DD Name	Data Set Definitions
SYSLIB	JKE Copybooks,JKE CICS.SDFHCOB
TASKLIB	JKE COBOL.SIGYCOMP,JKE CICS.SDFHLOAD,JKE DB2.SDSNL...

Add... Edit... Remove

DD allocations:

DD Name	Data Set Definition	Member	Keep	Output	Publish
SYSIN	<INPUT>	no	no	no	no
SYSLIN	JKE Temporary file (object deck)	no	yes	no	no
DBRMLIB	JKE DBRM library	yes	no	no	no

Add... Edit... Remove

- Describes an operation to be performed on a file during a build
- Supports variable substitution for data set definitions
- Supports build-level and file-level overrides for compiler options and ISPF/TSO commands

Language Definition



141: Change Request for Mortgage Application Updates | JKE COBOL compilation (CICS&DB2) and link-edit

Language Definition

Name: JKE COBOL compilation (CICS&DB2) and link-edit

General

Specify a language for this language definition. Optionally, you can specify file extensions to automatically associate files with this language definition.

Language:

File extensions:

Separate extensions with a comma; for example, "cbl,cob".

Description:

Translators

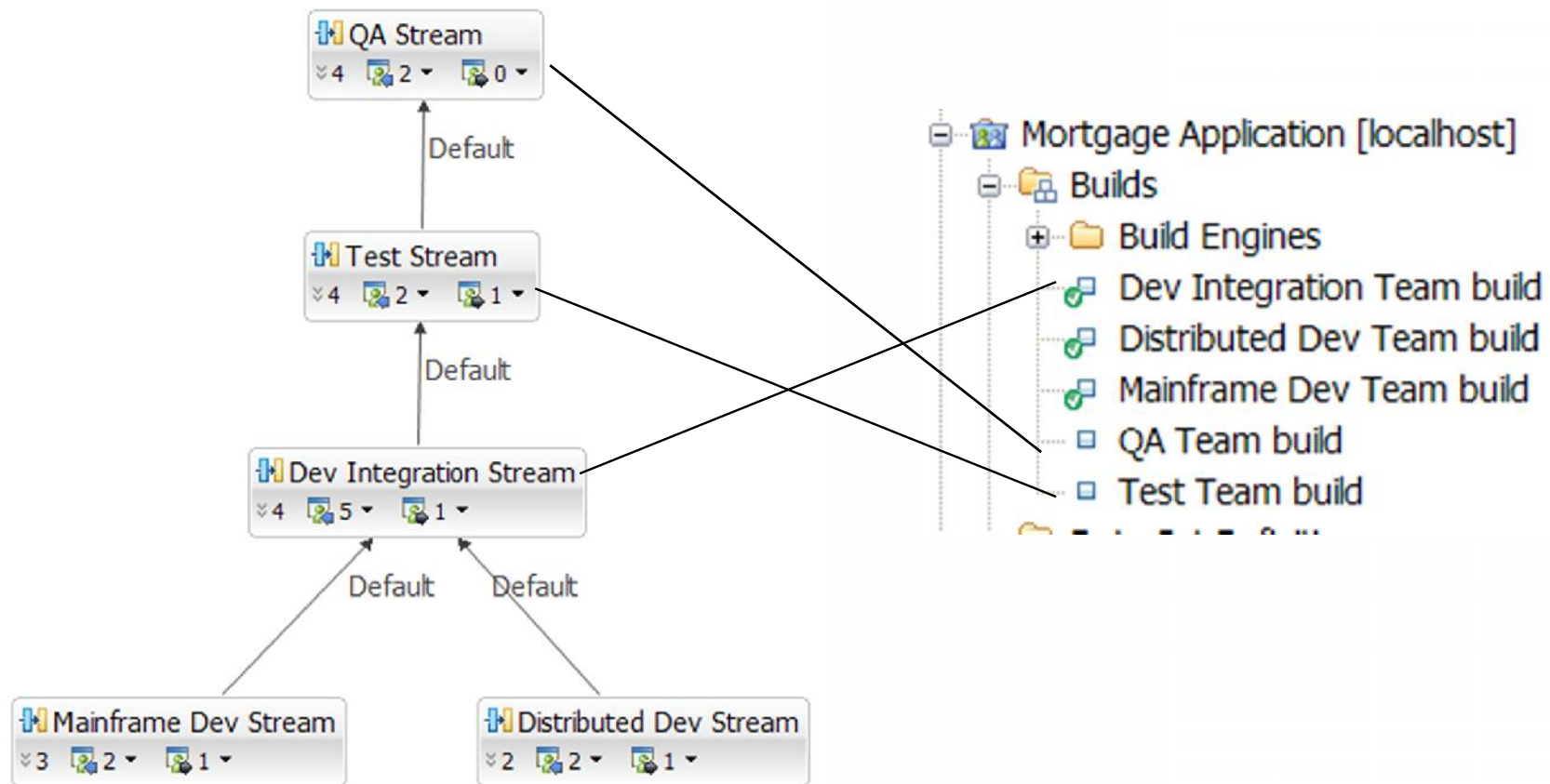
Specify and order translators for this language definition. Translators run in this order during the build.

JKE COBOL compilation (CICS&DB2)	<input type="button" value="Add..."/>
JKE Link-edit	<input type="button" value="Edit..."/>
	<input type="button" value="Remove"/>
	<input type="button" value="Up"/>
	<input type="button" value="Down"/>

General | Scanners

- Collection of tasks to be performed on a file
- Tasks defined using Translators
- Example: DB2 Preprocessor, CICS Translator, COBOL compile, and link-edit

Setup build definitions related to streams



Dependency Build



- Build changed and impacted programs only
- Options:
 - Build subsets
 - Personal build
 - Preview build
 - Ignore changes
 - Rebuild

Build Definition

ID: mortgage.dev Project or Team Area: JKE Bank

General | Dependency Options

Build File and Targets

Generate a build file from these language definitions

JKE BMS map processing	Add... Remove Move Up Move Down
JKE COBOL compilation	
JKE COBOL compilation and link-edit	
JKE COBOL compilation (CICS) and link-edit	
JKE COBOL compilation (CICS&DB2) and link-edit	

Use an existing build file

Specify a custom build file and the targets to be invoked. Properties can be referenced using \${propertyName}.

Build file:*

Build targets:

Trust build outputs

Choose to build all programs or a custom subset of buildable artifacts.

Build workspace

Build subset

Browse...

Build changed items only



- On delivery of code into the SCM a scanner parses the code and mines for code dependencies
- The scanner is associated to a Language Definition and the Language Definition is associated to the file being delivered
- The dependency data is stored in the repository
- Custom scanners can be contributed to this framework and will also be invoked
- Their output will also be stored in the repository

Source Code Data Save

Namespace:

▶ **Jazz SCM Properties** <http://www.ibm.com/xmlns/prod/rational/rtc/metadata/sourcefile/scm/>

▼ **Dependency Properties**

The following properties were produced by the associated scanner and may be edited. To add properties or to create additional instances of existing properties use the Add button in the User-defined Data section.

dependency ✖

dependencyLogical...	dependencyFileType	dependencyPath	dependencyRefere...
JKEPDATA	INCL	SYSLIB	COPY

fileType:

language:

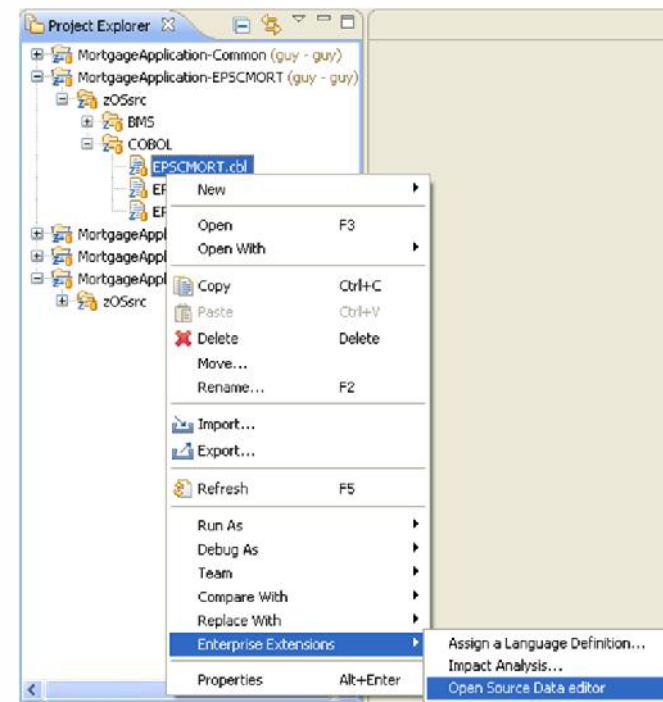
logicalName:

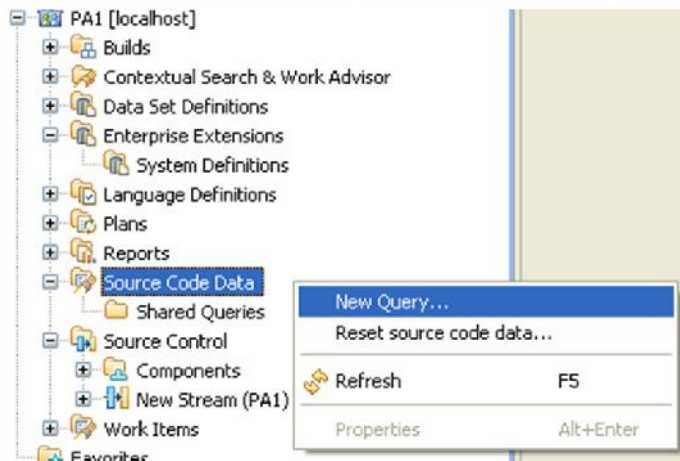
▼ **User-defined Data**

User-defined data is not produced by the source code scanners and will persist even after resetting the scanner data.

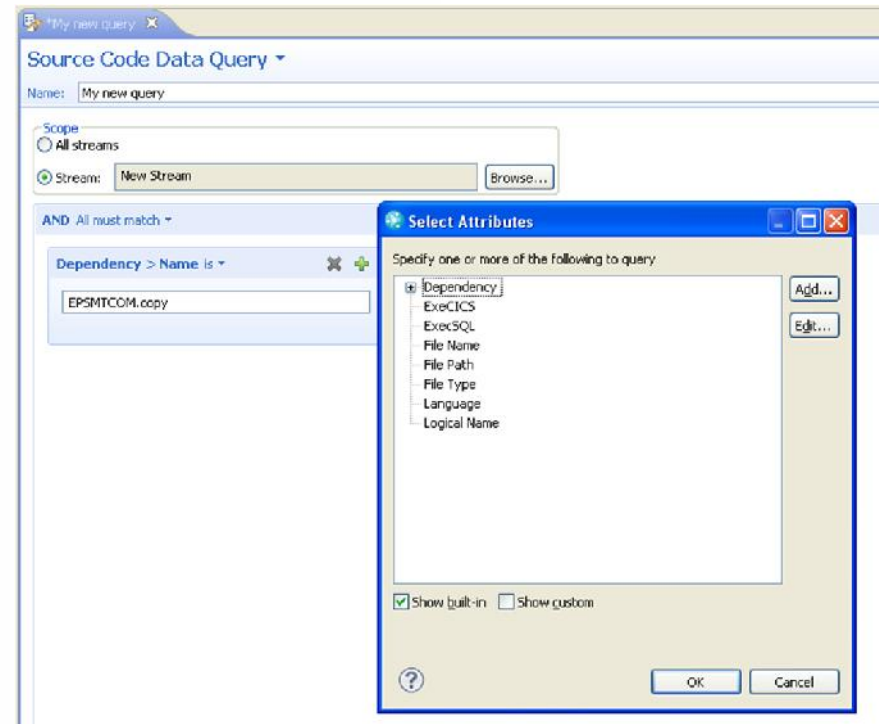
Overview

- The Source Code Data editor allows you to view the data collected by all the
- scanners run against the file.
 - You are also able to manually add
 - or edit data that the scanners have
 - collected





- We have a fully functional query editor



Works out of the box against the data that the default scanner

- produces
- Attributes - user defined scanners
- produce can be added

Queries can be

- saved and shared



Impact Analysis



The screenshot illustrates the process of performing an impact analysis in IBM Rational Developer for System z. The Project Explorer shows the file JKECMORT.cbl selected, with a context menu open. The 'Impact Analysis...' option is highlighted. The 'Impact Analysis Search' dialog box is open, showing the search type set to 'Show files that this file depends on'. The search results are displayed in the 'Tasks' pane, showing two files that JKECMORT.cbl potentially depends on.

Impact Analysis Search

Search for files that this file depends on, or for files that depend on this file.

Search Type

- Show files that this file depends on
- Show files that depend on this file

Stream: Mortgage Development Stream

Advanced

- Apply search path

Language Definition: JKE COBOL compilation (CICS&DB2) and link-ed

Build Definition:

Tasks | Source Code Data

Found 2 files that JKECMORT.cbl potentially depends on. (Logical def

fileName	filePath	componentName
JKENBRPM.cpy	/MortgageAp...	Mortgage Common
JKEMTCOM.cpy	/MortgageAp...	Mortgage Common

Promotion Build Definition



□ Promotion Definition ▾



ID:

Project or Team Area:

Enterprise Promotion

Define enterprise promotion configuration properties.

Source build definition:*

Target build definition:*

Select components from the source build definition workspace to promote.

- Mortgage
- Mortgage Common

Skip timestamp check when build outputs are promoted

Build File and Targets

Choose to generate a build file or specify an existing one

Destination directory:*

The destination directory is the location on the build system where intermediate build files will be placed.

Generate build file

Use an existing build file

Build file:*

Build file path should be relative to the build machine.

Build targets:

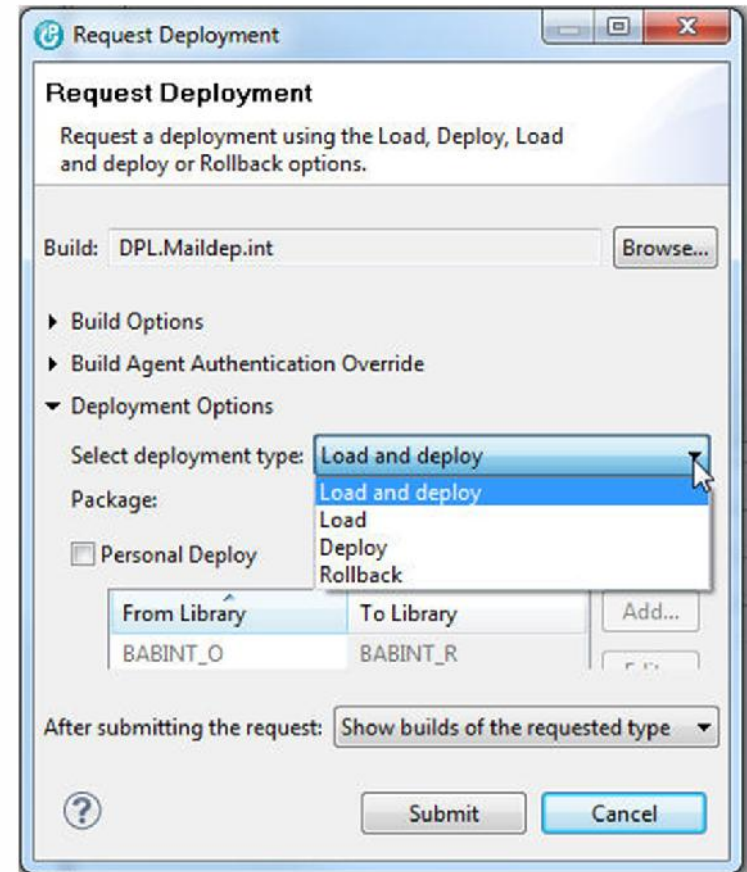
Overview | Properties | Job Output Publishing | z/OS Promotion

- Installs packages to target system
- Copies the binaries
- Backup n-1 version for deployment for back out
- Full back out of packages
- Must have build agent on all target systems
- Pre and Post processing is supported for deployment

Submitting a Deployment Request



- There are 4 types of requests you can execute on a Deployment Definition:
 - **Load:** just load the specified package from the Build Machine to the Deployment Machine (to be deployed later)
 - **Deploy:** deploy a previously-loaded package
 - **Load and Deploy:** load the specified package and deploy it immediately
 - **Rollback:** reverse the changes made by the last Deploy request.



- Deployment reports produced from deployment requests contain detailed information on what was modified

The screenshot shows a web-based deployment report interface. The title bar indicates the deployment is for 'Maillist.int.deploy' with ID '20120330-1148220648'. The report is titled 'Build Maillist.int.deploy 20120330-1148220648'. The summary section provides details about the host name, package directory, and a work item ID. The main section, 'Deployed Objects', contains a table listing various objects that were created during the deployment.

Summary

Host name: i7adt10.torolab.ibm.com
Package directory: /home/babineau/deploytest/loadspace//_IllJoG6tEeGXz-X84oufjA/20120329-1148330604
Summary Work Item: [138](#)

Deployed Objects

Double-click to open work item

Name	Type	Subtype	Library	Last Modified ...	Type of Change	Work Item
BUILD	*PGM	CLP	BABMLIRUN	March 12, 201...	Created	
EVFEVENT	*FILE	PF-DTA	BABMLIRUN	March 12, 201...	Created	
MAILLIST	*PGM	CLLE	BABMLIRUN	March 15, 201...	Created	
MLGINQD	*FILE	DSPF	BABMLIRUN	March 12, 201...	Created	
MLGMNUD	*FILE	DSPF	BABMLIRUN	March 12, 201...	Created	
MLGMSTL	*FILE	LF	BABMLIRUN	March 12, 201...	Created	
MLGMSTL2	*FILE	LF	BABMLIRUN	March 12, 201...	Created	
MLGMSTL3	*FILE	LF	BABMLIRUN	March 12, 201...	Created	
MLGMSTP	*FILE	PF-DTA	BABMLIRUN	March 12, 201...	Created	
MLGMTND	*FILE	DSPF	BABMLIRUN	March 15, 201...	Created	
MLGNAMD	*FILE	DSPF	BABMLIRUN	March 12, 201...	Created	
MLGNAML	*FILE	LF	BABMLIRUN	March 12, 201...	Created	
MLGREFP	*FILE	PF-DTA	BABMLIRUN	March 12, 201...	Created	

Overview | Activities | Downloads | Logs | Properties | Deployment Summary

- Run queries for your packages and deployments to discover the state of your test and production machines

A dialog box titled "Search for Deployed Packages" with a search icon in the top-left corner. The title bar includes standard window controls (minimize, maximize, close). The main area contains the text "Search for Deployed Packages" and "Specify information in the fields to search for deployed packages." Below this are four dropdown menus: "Deployment definition:" with "Maillist.int.deploy" selected, "Package definition:" with "Maillist.int.pkg" selected, "Build agent:" with "i7adt10.sean.rba" selected, and "Packages:" with "*" selected. At the bottom right are "OK" and "Cancel" buttons.

A table window with a toolbar at the top containing icons for "Work", "Tag", "Probl", "Pend", "Tea", "Cons", "Build", "Prop", "Sour", and "Selec". The table title is "Found 1 matches". The table has five columns: "Package Result", "Deployment Result", "Package Definition", "Deployment Definit...", and "Build Agent". The first row contains the following data: "20120329-1148330604", "20120330-1148220648", "Maillist.int.pkg", "Maillist.int.deploy", and "i7adt10.sean.rba".

Package Result	Deployment Result	Package Definition	Deployment Definit...	Build Agent
20120329-1148330604	20120330-1148220648	Maillist.int.pkg	Maillist.int.deploy	i7adt10.sean.rba

Automated dashboards

- Role based dashboards - Personal, Team, Project dashboards
- Dashboards populated with data from across the silos
- Review status on-line – Stay informed – Automated Project Status w/o having to ask.

- Embrace and use techniques in appropriate way
- Automation enables and enforces process changes

Plans now feature a **Kanban Taskboard**. Configure your view mode to use *Kanban* as display type, define limits, and off you go. By default, limits use counts, but you are free to choose Complexity or Estimates. Deviations from the ideal are shown as a small box in the column header. The further your goal is away, the more red the box gets.

The screenshot displays a Kanban Taskboard interface. At the top, there is a search bar containing the text "Kanban". Below it are two tabs: "Options" and "Columns". A control bar shows the "Display" mode set to "Kanban" and a summary of items: "Open: --, Ready To Test: 3, Testing: 2, Closed: --". A dropdown menu is set to "Count". A "+ More" link is visible below the control bar.

The taskboard is organized into three columns:

- Open:** Contains two task cards:
 - CompositeRunner.filter incorrect if child throws NoTestsRema
 - javadoc updates for @Ignore in 4.3
- Ready To Test: 3 / 3:** Contains two task cards:
 - assertThat signature does not match Matcher
 - timeout doesn't work properly for >=2 cases in junit4.3?
- Testing: 0 / 2:** Contains one task card:
 - assertEquals throws NPE while comparing null elements

Package Definition



Package Definition ▾ 💰 Save

ID: Project or Team Area: Browse...

Package Options

Build definition: Select...

Package root directory:*

Package pre-command:

Package post-command:

z/OS ISPF gateway script:*

Include		Exclude
<input type="text"/>	Add...	<input type="text"/>
<input type="text"/>	Edit...	<input type="text"/>
<input type="text"/>	Remove	<input type="text"/>
<input type="text"/>		<input type="text"/>
<input type="text"/>		<input type="text"/>
<input type="text"/>		<input type="text"/>
<input type="text"/>		<input type="text"/>
<input type="text"/>		<input type="text"/>
<input type="text"/>		<input type="text"/>
<input type="text"/>		<input type="text"/>

Overview | Properties | z/OS Packaging | Ant

Deployment Build Definition



Deployment Definition ▾

Save

ID:

Project or Team Area:

Specify the PDS packaged, and where to deploy them in the restore mapping table.

From PDS	To PDS

Deploy pre-command:

Deploy post-command:

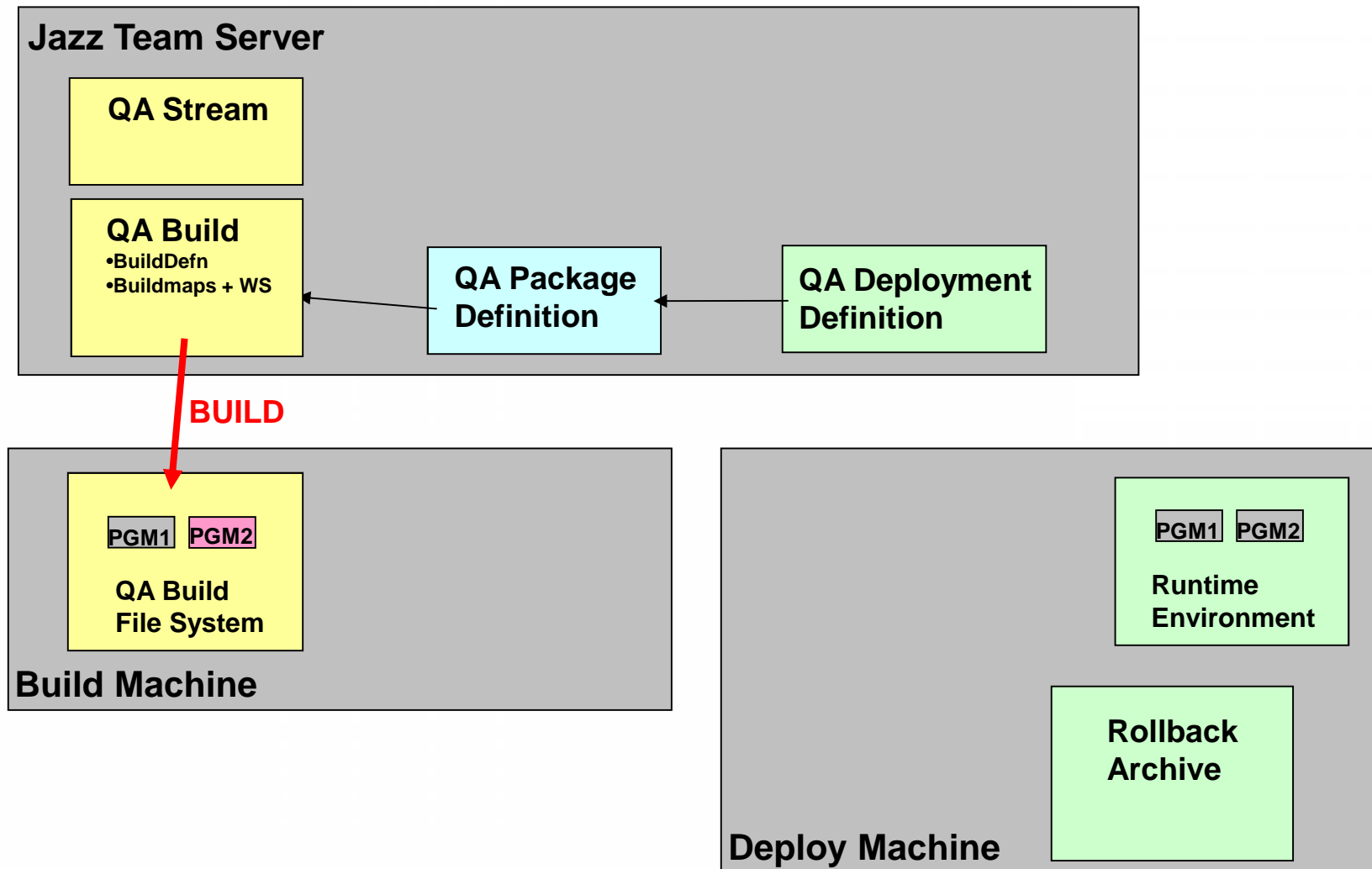
Rollback pre-command:

Rollback post-command:

Deployment by steps:



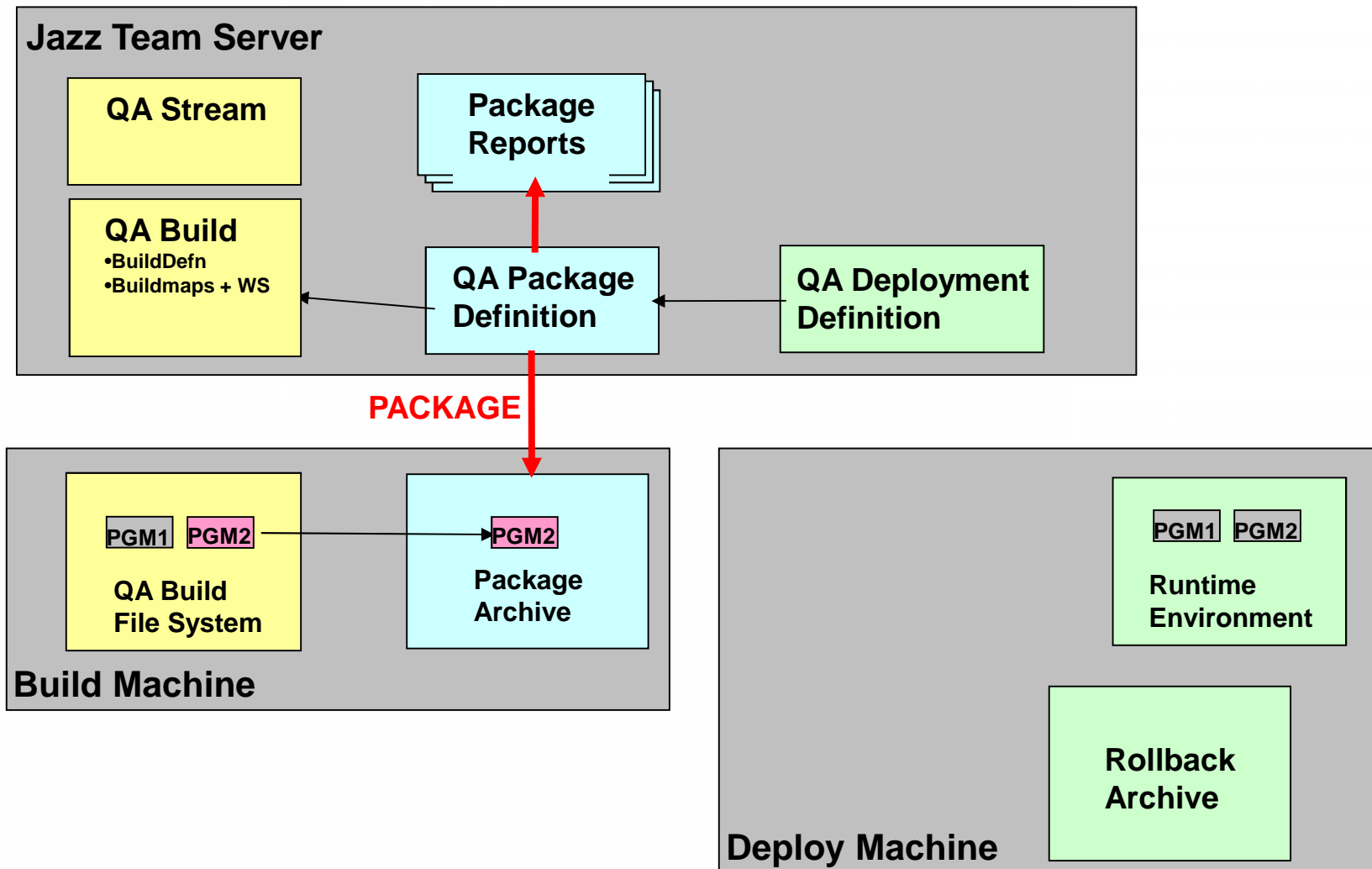
Step 1: something has been Built or Promoted



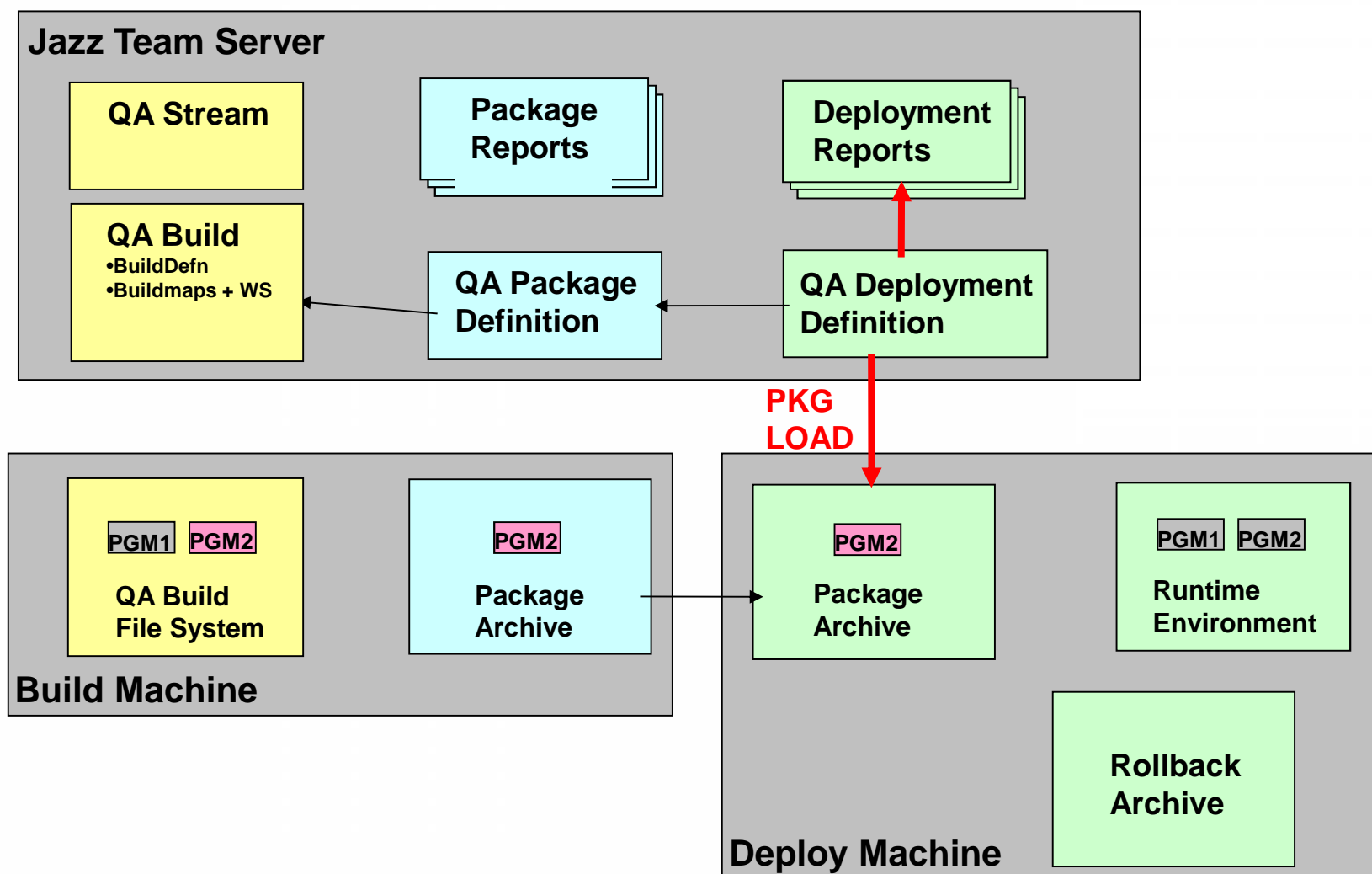
Deployment by steps:



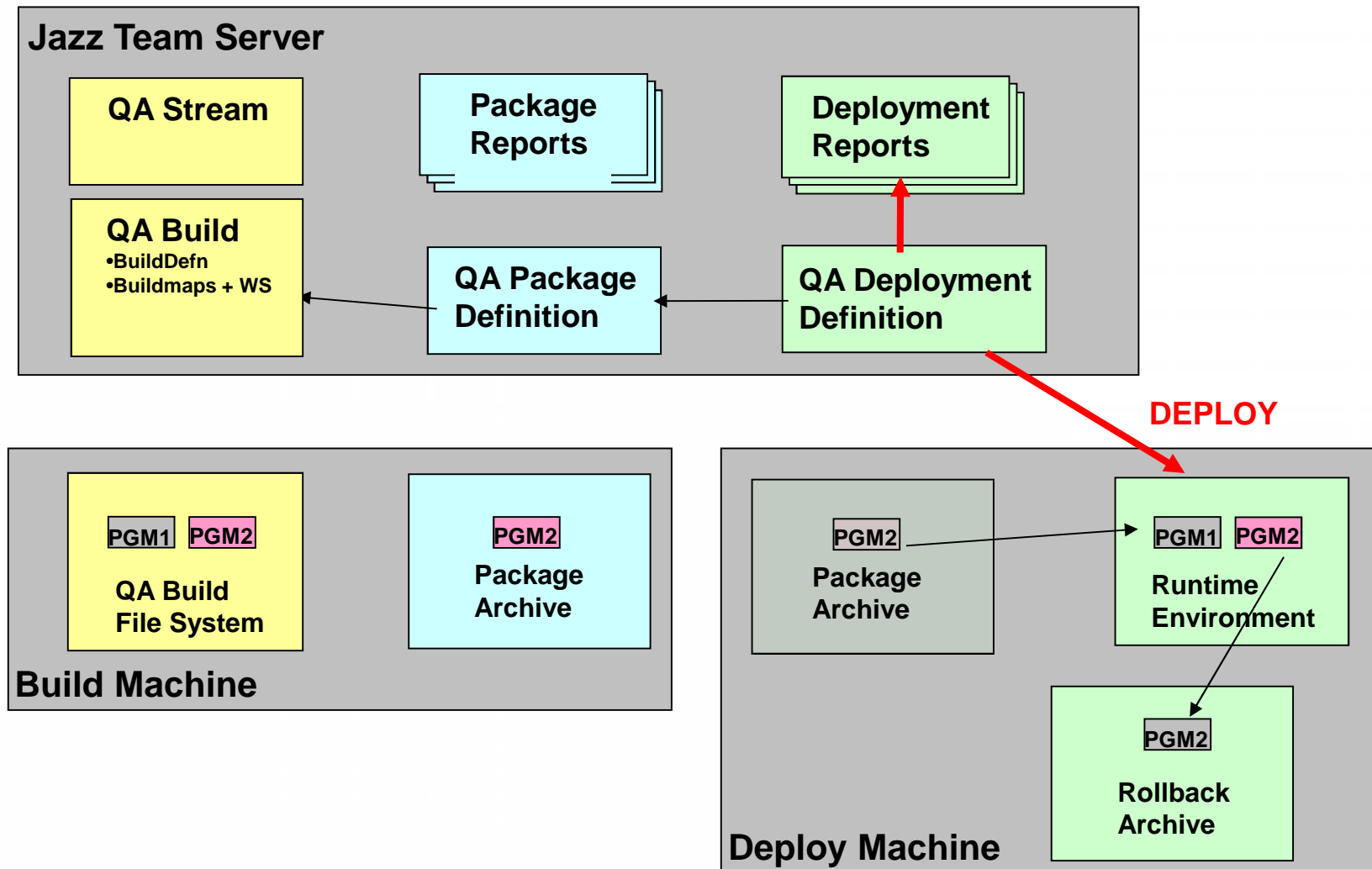
Step 2: A Package is Requested



Step 3: Package is Loaded to Deploy Machine



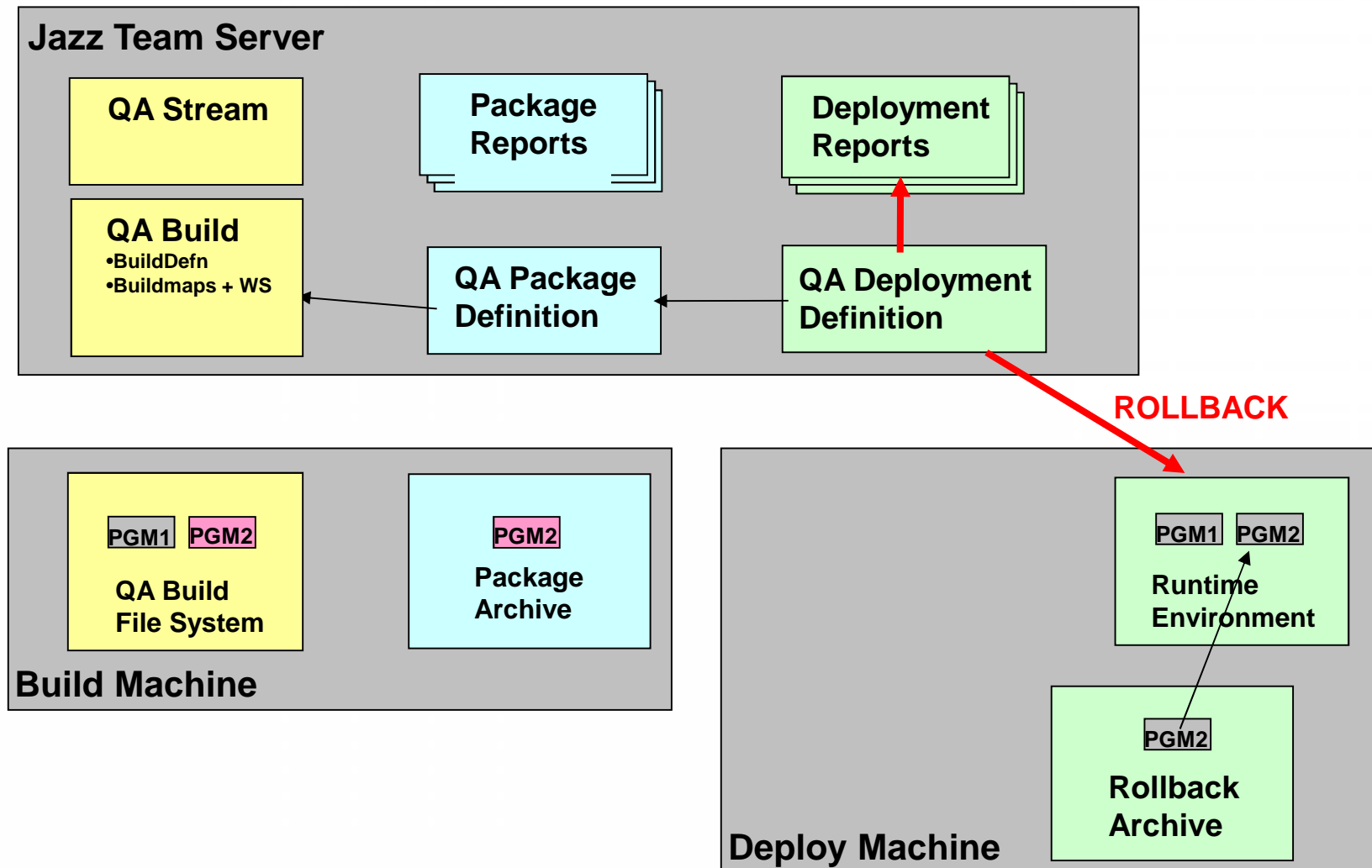
Step 4: Package is Deployed



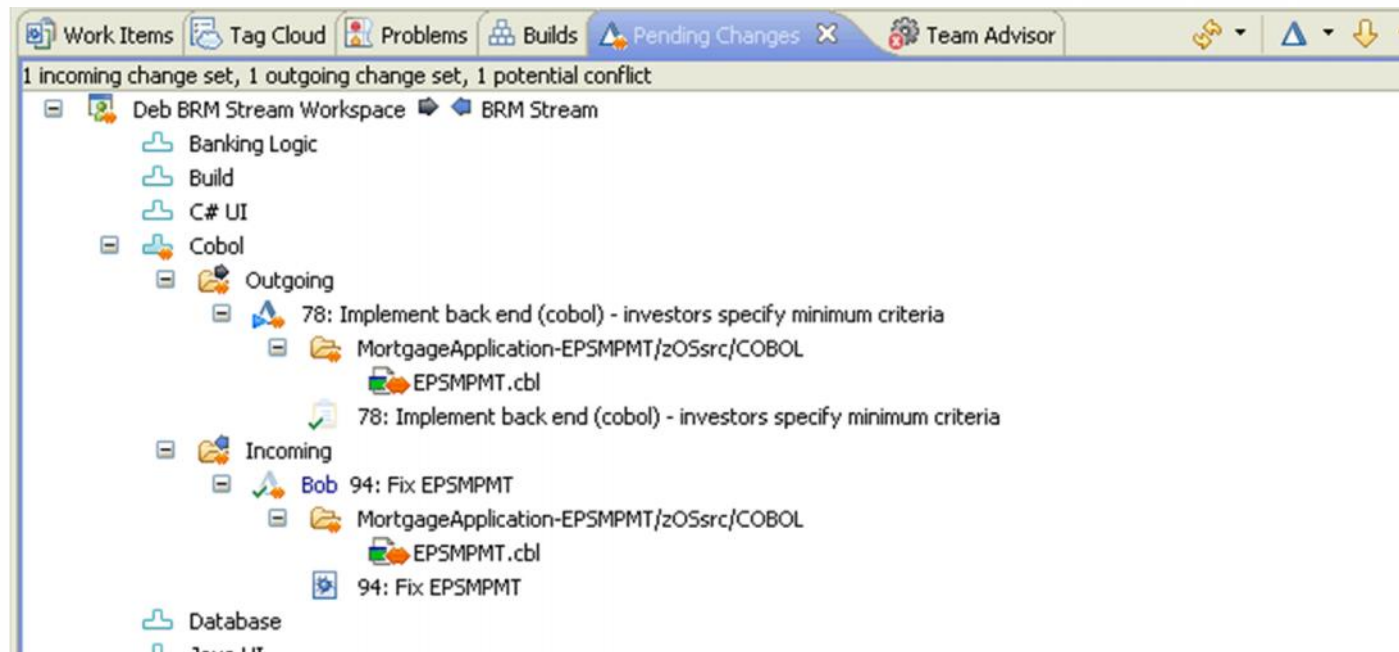
Deployment by steps:



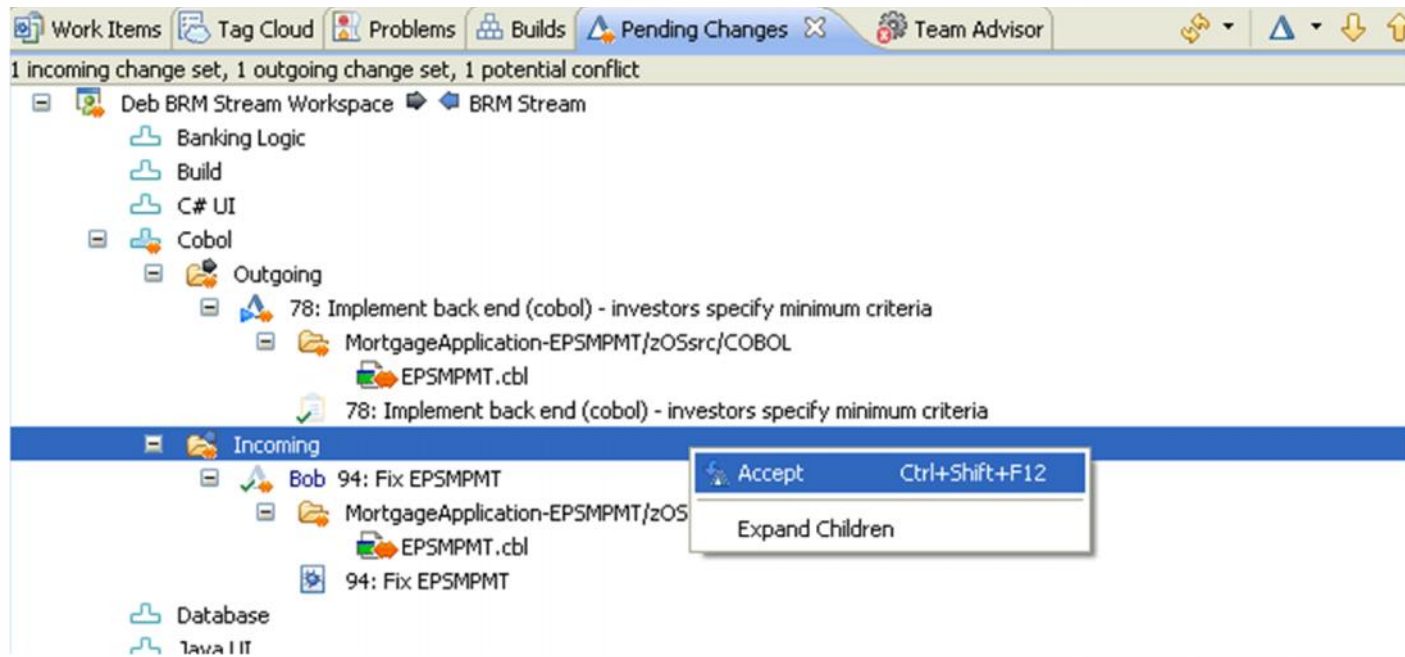
Step 4 minus 1: A Package is Rolled Back



Merging Code



Merging Code



Merging Code



The screenshot displays the IBM Rational Developer for System z interface. The main window is titled "Bob's BRM Stream Workspace" and shows a comparison of the COBOL source code for the program "EPSMPMT.cbl".

COBOL Structure Compare:

- PROGRAM: EPSMPMT
- ID DIVISION.

COBOL Source Compare:

The comparison is between two versions of the source file:

- Left Pane (Editable):** EPSMPMT.cbl (editable). The source code includes:
 - * (C) 2008 IBM - Jim Hildner
 - * Bob's Change.
 - * Deb's Change.
 - ENVIRONMENT DIVISION.
 - CONFIGURATION SECTION.
 - SOURCE-COMPUTER. FLEX-ES.
 - OBJECT-COMPUTER. FLEX-ES.
- Right Pane (Read-only):** EPSMPMT.cbl @ Deb BRM Stream Workspace (read-only). The source code includes:
 - * (C) 2008 IBM - Jim Hildner
 - * Bob's Change.
 - ENVIRONMENT DIVISION.
 - CONFIGURATION SECTION.
 - SOURCE-COMPUTER. FLEX-ES.
 - OBJECT-COMPUTER. FLEX-ES.
 - DATA DIVISION.

The bottom of the interface shows a "Work Items" pane with a tree view of the project structure:

- 1 outgoing change set
 - Deb BRM Stream Workspace
 - BRM Stream
 - Banking Logic
 - Build
 - C# UI
 - Cobol (1 merged)
 - Outgoing (1 merged)
 - 78: Implement back end (cobol) - investors specify minimum criteria (1 merged)
 - MortgageApplication-EPSMPMT/zOSsrc/COBOL (1 merged)
 - EPSMPMT.cbl (Merged)
 - Database

Merging Code



■

The screenshot shows a software window titled "EPSMPMT.cbl in Stream BRM Stream - Cobol [robin.raleigh.ibm.com]". The window has a menu bar with "Help", "Properties", "Repository Files", and "History". Below the menu bar is a toolbar with various icons. The main area contains a table with the following data:

Merges	Comment	Creator	Date Created
	78: Implement back end (cobol) - investors specify min...	Deb	4:37:45 PM (5 minutes)
	94: Fix EPSMPMT	Bob	4:19:45 PM (20 minutes a
	Share projects	Deb	Oct 4, 2011 11:38 AM

See and manage the big picture for releases, roll up progress and work load

Plan: Call Center 1.0 [Release 1.0] ... Rational Team Concert - Call Centre

Rational Team Concert Jean-Michel Lemieux | Log Out

Dashboards Project Areas Work Items **Plans** Source Control Builds Reports Admin | Call Center

Plans Related to Timeline 'Main Development [Project Timeline]' >

Previous | 2 of 5 | Next

Call Center 1.0 Related Plans (2)

Owner: Call Center | Iteration: Release 1.0 (10/20/09 - 11/29/09) | 15 Closed | 23 Open

Overview **Planned Items** Charts

View As: Iterations

Progress: 121/327 pts Estimated: 100%

Item	Points	Hours	Status	Assignee	Points
Release 1.0	Closed Items: 15 Open Items: 25		Progress: 121/327 pts	Estimated: 100%	
Prepare sample data for call demos	5 pts	0/27 h	New	Unassigned	144
Allow home handlers to call in and take calls	100 pts	--	New	Jennifer Guinness	84
Send thank you notices to Facebook accounts	8 pts	--	New	Markus Kent	115
Allow voice note talking for customer accounts instead of typing in messages	20 pts	--	New	Bill Cassavelli	116
Allow concurrent modification of non-conflicting users settings	20 pts	--	New	Unassigned	131
Provide bundle options based on customer loyalty	13 pts	--	New	Unassigned	132
Need more build servers for continuous integration			New	Jean-Michel Lemieux	101
Report doesn't consider dial tone differences			Done	Unassigned	400
Call forwarding broken in latest build			Done	Unassigned	90
Allow adding calls to the dashboard	0 pts	0/0 h	New	Unassigned	158
Task 1			New	Unassigned	159
Sprint 1 (1.0)	Closed Items: 4 Open Items: 0		Progress: 71/71 pts	Estimated: --	