

Integrated Change, Build And Deploy Is What Sets Rational Team Concert Apart

Gregory Sechuga
Manager, Rational Competitive Research
IBM Software Group
gsechuga@us.ibm.com

IBM Software

Innovate2011

The Premier Event for Software and Systems Innovation



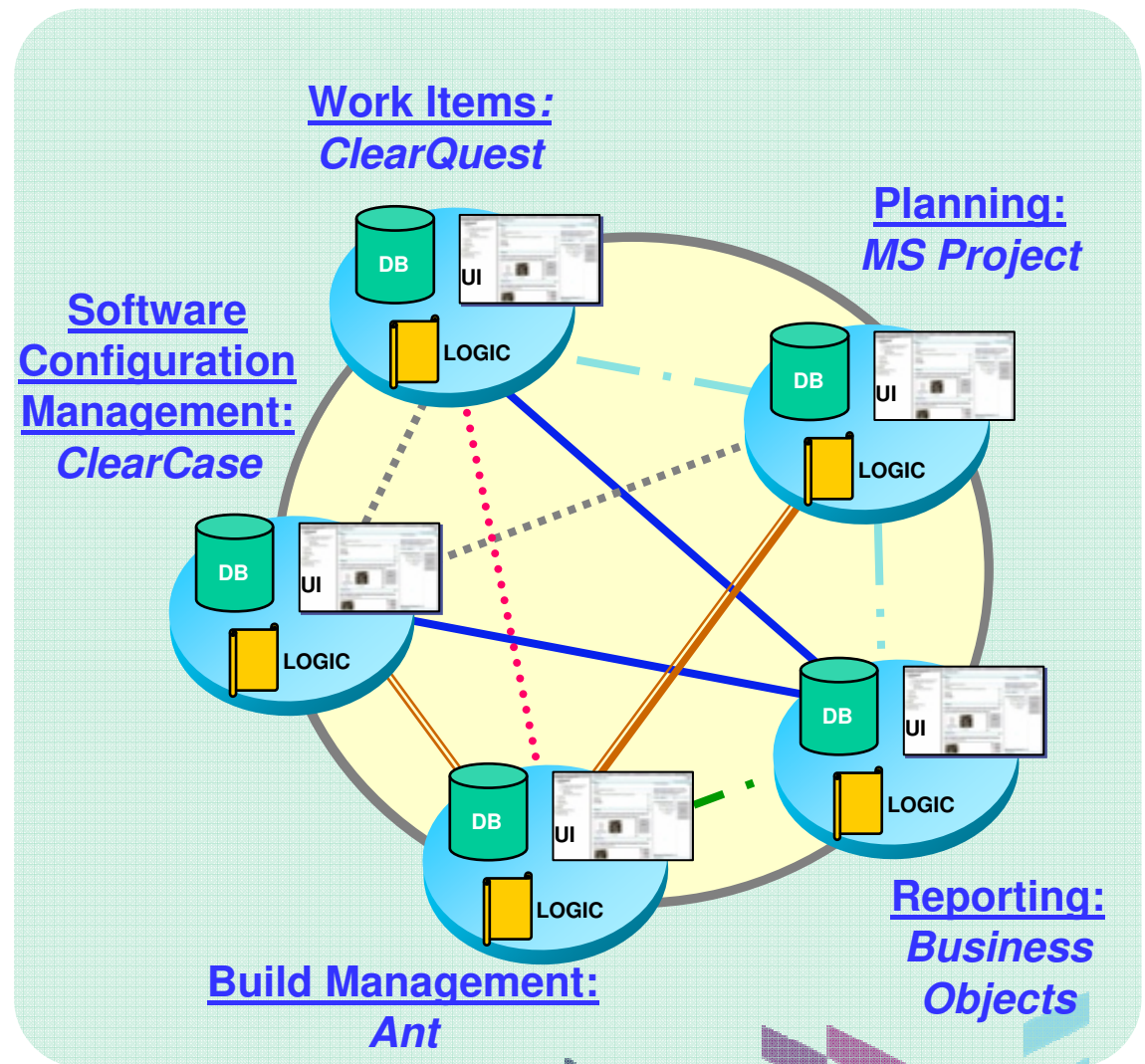
Software. Everywhere.

August 9-11, Bangalore | August 11, Delhi



A typical Change and Configuration Management Solution: a variety of standalone tools integrated point-to-point

- Multi-site, globally distributed development on different platforms
- Many different CCM methodologies/tools
- Each tool requires unique integration with other tools
- Each tool has own repository
- Point-to-point integrations become unmanageable
- Brittle/non-existent linkage to Requirements and Quality tools
- Teams find it hard to work together



Change and Configuration Management issues cross geographies and user roles

Analyst: “Are all the requirements tested?”



Project Manager: “Are we ready to release?”



Current Concerns

- No consolidated information hub
- LOB Analysts and Quality Testers feel detached from development team
- Project status reporting is labor-intensive



Developer: “What test uncovered this defect, on which environment and which build?”



Tester: “What defects have been addressed since the last build?”



Build Manager: “How can I speed up my builds across my platforms?”

A poor CCM process jeopardizes company's ability to develop high quality solutions

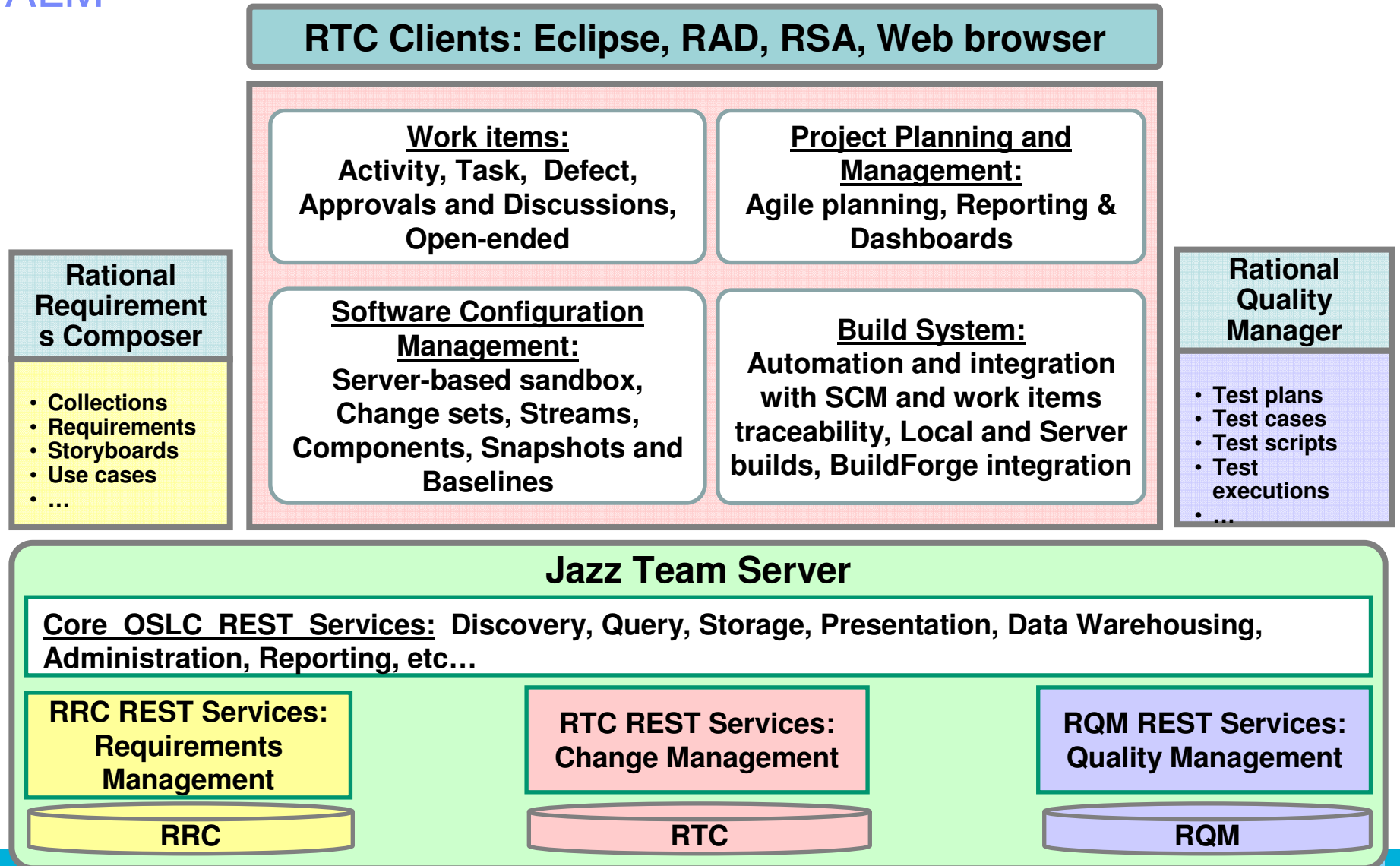
A better Change and Configuration Management solution is needed

1. Work Item Management
2. Linkage with Requirements and Quality Management processes
3. Software Configuration Management
4. Project Planning/Management
5. Automated Build Management

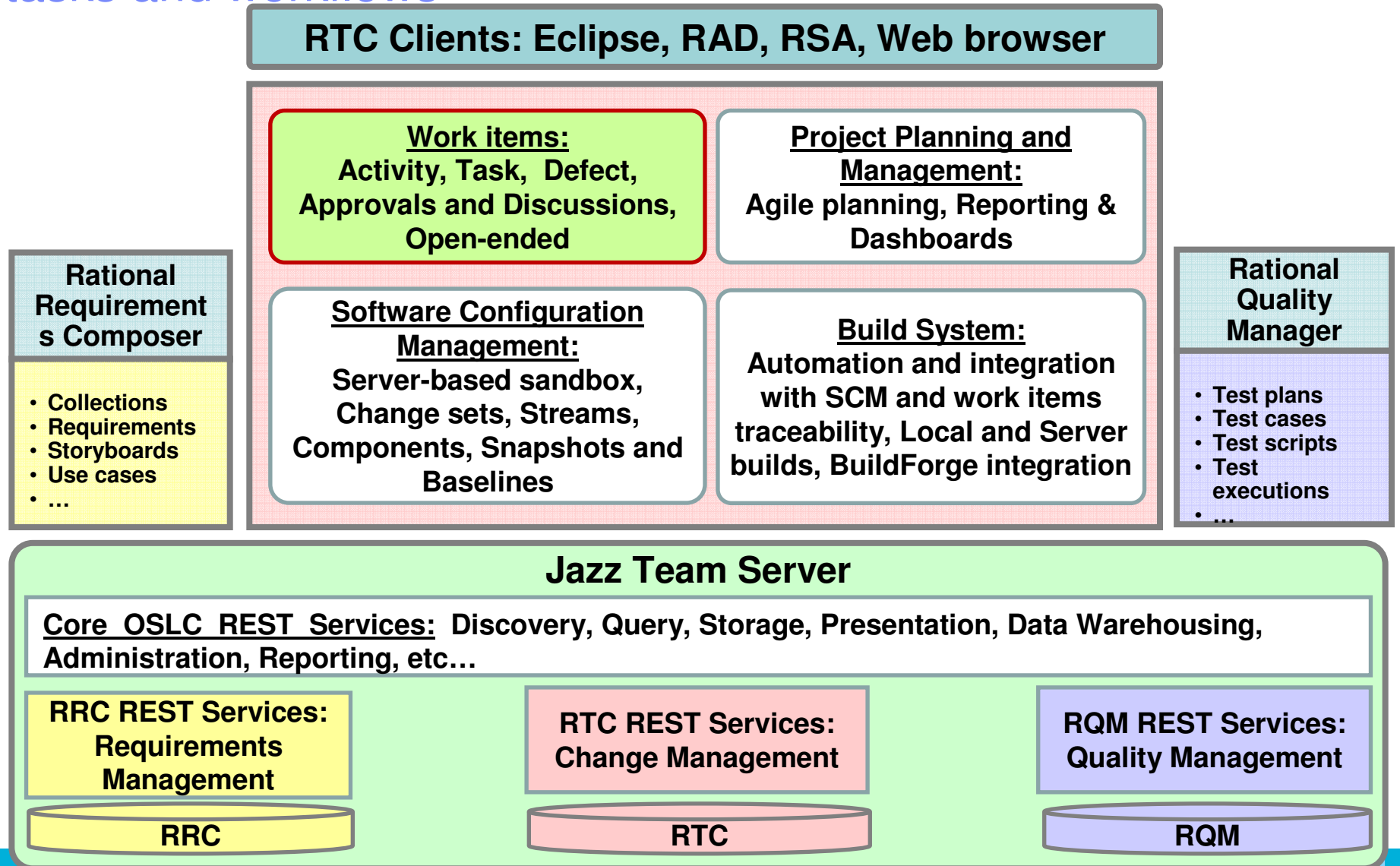
Rational Team Concert was designed to do this. It allows your team to collaborate on plans, tasks, code and builds, all in one place!



Rational Team Concert is the core product in IBM Jazz-based ALM



RTC Work Items are used to track and coordinate development tasks and workflows



Work Items are the fundamental artifacts in RTC

- Plan work item types are used to capture high-level plan elements
 - ▶ Epic
 - ▶ Story
- Execution work item types are used to capture the lower-level details and the work that should be completed in a single iteration
 - ▶ Task
 - ▶ Defect
- Work items are the basis for linkage between many artifacts

Work Item creation is easy and fast

- Work items that reflect repeating tasks in different iterations
 - ▶ Create by instantiating a template for each iteration
- Work item template wizard
 - ▶ Capture and create new templates

The screenshot displays the IBM Work Item Manager interface. At the top, there are navigation tabs for Work Items, Tag Cloud, Problems, Team Advisor, Progress, and Team Organization. Below the tabs, it indicates 'Found 198 work items - All Items'. A table lists work items with columns for Summary, Id, Status, and P. A context menu is open over a work item, showing options like Open, Change Status, New Work Item Template..., and Re-Run. Below the table, two wizard windows are shown. The first window, 'Create Work Item Template', has a 'Select Work Items for the Template' section with a list of work items: 169: Typo in cookbook: wrong package for JUnitCore, 131: Improve documentation for 4.4, 160: javadoc updates for @Ignore in 4.3, and 167: [Docs] Cookbook TestRunner section incorrect. The second window, 'Define Template Variables', has a 'Check Variable Attributes' section with checkboxes for 'Planned For' and 'Filed Against', both of which are checked. Navigation buttons like '< Back', 'Next >', 'Cancel', and 'Finish' are visible at the bottom of both wizard windows.

RTC allows users to collaborate with a real-time, in-context work item comment

The screenshot displays the Rational Team Concert (RTC) interface. The main window shows a work item titled "Story 74" with the summary "Allocate Dividends To Multiple Causes". The details section includes fields for Type (Story), Filed Against (JKE/BRM), Story Points (0 pts), Progress (0/0 h), and Team Area (Business Recovery Matters). A discussion section shows a comment from Deb dated May 13, 2011, at 12:38 PM, stating "We need to get this implemented in Sprint 2. Bob will ensure this happens".

Overlaid on the right side of the interface is a real-time chat window titled "Kevin (K.G.)". The chat window shows a conversation between Kevin (K.G.) and kartik@ca... regarding the work item. The chat content includes:

- Kevin (K.G.): looking at it now ...
- Kevin (K.G.): * Yes we are on track, the next build will have e of these issues addressed
- Kevin (K.G.): thanks
- Kevin (K.G.): np

A yellow callout box with a pointer to the chat window contains the following text:

- Shows team members and their online status
- Can start chat in context with work item

The bottom of the interface shows a table of work items:

Id	Status	P	Story Points	Summary
62	New	1 pt	1 pt	Organizations may apply with an ini
64	New	2 pts	2 pts	Customers can Nominate an Organiz
66	New	1 pt	1 pt	Organization must provide justificat
67	New	0 pts	0 pts	Donation by amount
69	New	2 pts	2 pts	Organizations can Apply
70	New	5 pts	5 pts	Donor Dividend Allocation Criteria
71	New	1 pt	1 pt	JKE Charity Coordinator will respond

Traceability helps team members understand what everyone else is doing

- What requirements did this iteration address?
- Are all the requirements tested?
- What defects are reported against which requirements?



LOB Analyst

- What requirements am I implementing?
- How can I recreate the last version?
- What test uncovered this defect?
- What changes occurred last night?



Developer

- What is the quality of the build?
- What has changed that I need to test?
- What defects did the last build address?



Tester

- How can I speed up my builds?
- Are build times getting longer or shorter?



Build Manager

Plan Item traceability improves quality and predictability

- Everyone's work aligns to requirements
- Team members have transparency to each others work
- As things change, the “new reality” is always available
- Team has insight when all work is done

Work Items >
Story 74 ?
 Summary: * Allocate Dividends To Multiple
 Overview | Acceptance | Links | Approvals | History
Details
 Type: Story
 Filed Against: * JKE/BRM
 Story Points: 8 pts
 Project Area: JKE Banking (Change Management) | Priority: Medium
 Team Area: Business Recovery Matters | Planned For: Sprint 2

Developers understand the requirements, test results and test criteria

Overview
95: Donor Chooses Multiple Organizations
 Description:
 Project: JKE Banking (Requirements)
 Created On: Mar 27, 2011 12:58:12 PM
 Created By: clmadmin
 Modified On: Mar 27, 2011 12:58:52 PM
 Modified By: clmadmin
 Type: User
 Comments
 Links (14)
 97: Dividend contribution - pag
 97: Dividend contribution - page 2
 55: Dividend Contribution
 122: Dividend Contribution

Teams collaborate and clarify the details of requirements

Development Items ?
 Change management items that are aligned with requirements
 Show All Items per page
 Summary
 Donor Dividend Allocation Criteria
 Quick Information
 Subscribers (1): B
 Children (1): 49
 Implements Requirement (1)
 Tested By Test Case (1)
 Description
 Donors should have the ability to... allocation options for the

Testers define and execute test cases with a clear understanding of requirements

Relationship views enable continuous traceability

- Find and respond to gaps as they surface through out the project
- Tracing throughout the project improves regulatory compliance

The screenshot shows the IBM Change and Configuration Management (CCM) interface. The main header displays 'JKE Banking (Change Management)' and 'BRM Sprint 2 (1.0) Plan' with 28 items (25 open, 3 closed) and a 10-day end date. A yellow callout box points to the 'Traceability View' which shows a table of relationships between requirements, test cases, and defects.

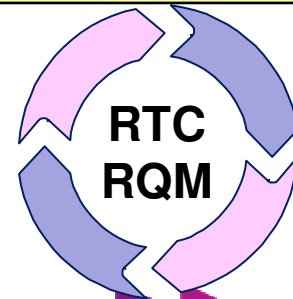
Traceability View shows linkage of requirements, test cases, and defects

Actions	Summary	Implements Requirement	Tested By Test Case	Affected by Defect
	Donors Can Choose to Support Multiple Organiza	444: Donor Chooses Multiple Organizati	97: Allocate Dividends to Multip	300: Failing Test Case "
	Donor Dividend Allocation Criteria	Donor Dividend Allocation Criteria	-	-
	Requests sent in form of email	-	-	-
	Frequency of dividend transfer	Frequency of dividend transfer	Verify dividend transfer frequen	301: Failing Test Case "
	Organization must identify how much money is de	Organization must identify how much mo	Organization must identify how	-
	Organizations may apply with an initial request	Organizations may apply with an initial re	Organizations may apply with a	-

Defect collaboration in the development – test cycle reduces costs and improves quality

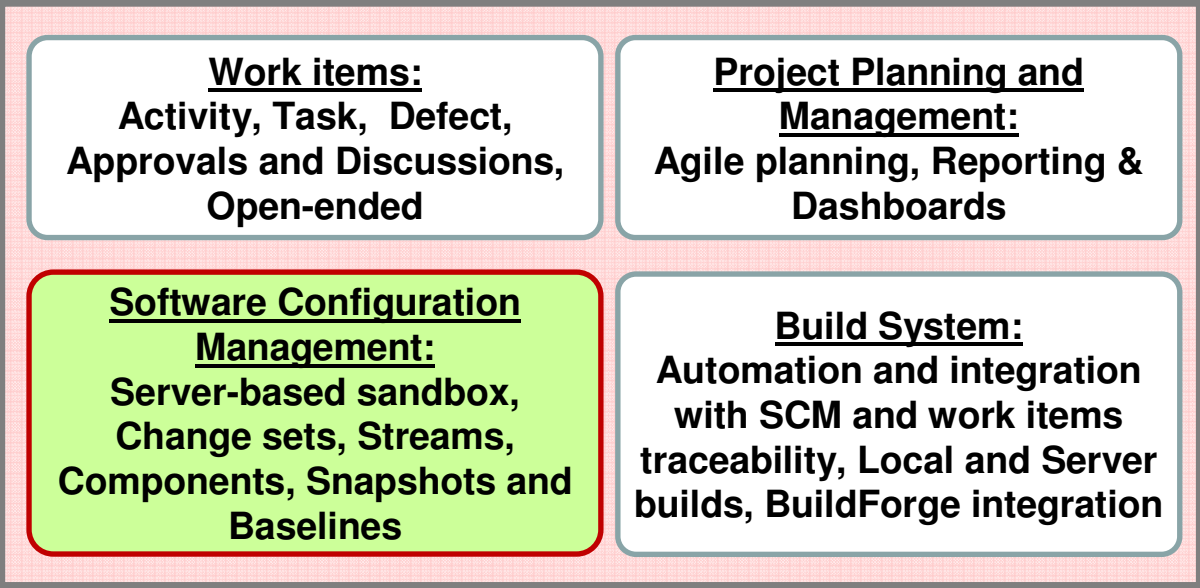
- Minimal # of clicks needed to submit a defect - automatically linked to impacted artifacts
- Test results recorded and linked to test cases and associated requirements
- Test results can link to software builds
- Everyone has visibility to defects, their impact, and actions taken to resolve them

The screenshot displays a software interface for defect management. At the top, an 'Execution Result' window shows a test that failed. A yellow callout bubble points to this window with the text: "Testers execute tests and submit defects found to RTC". Below this, a 'Defect 76' record is shown. A yellow callout bubble points to the defect record with the text: "Developers can see the exact test failure without having to ask for it." The defect record includes a summary, overview, links, and a list of related artifacts such as requirements, test cases, test plans, and plan items.



RTC Software Configuration Management consists of source control, change management, and version control

RTC Clients: Eclipse, RAD, RSA, Web browser

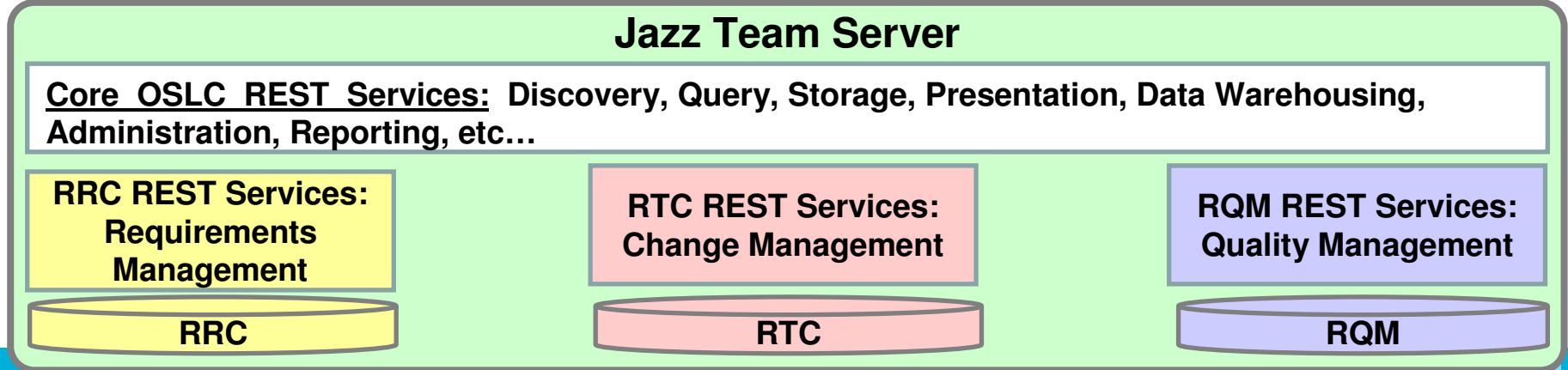


Rational Requirements Composer

- Collections
- Requirements
- Storyboards
- Use cases
- ...

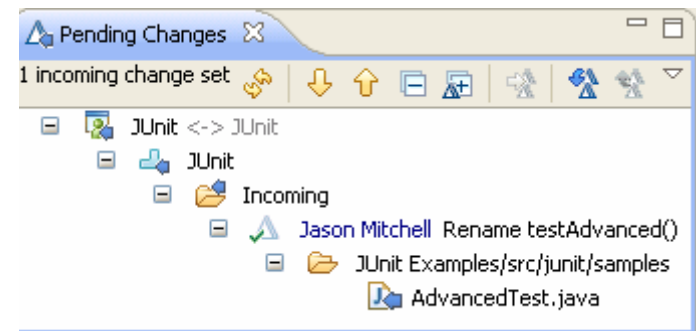
Rational Quality Manager

- Test plans
- Test cases
- Test scripts
- Test executions
- ...



Software Configuration Management tracks and controls changes to software assets

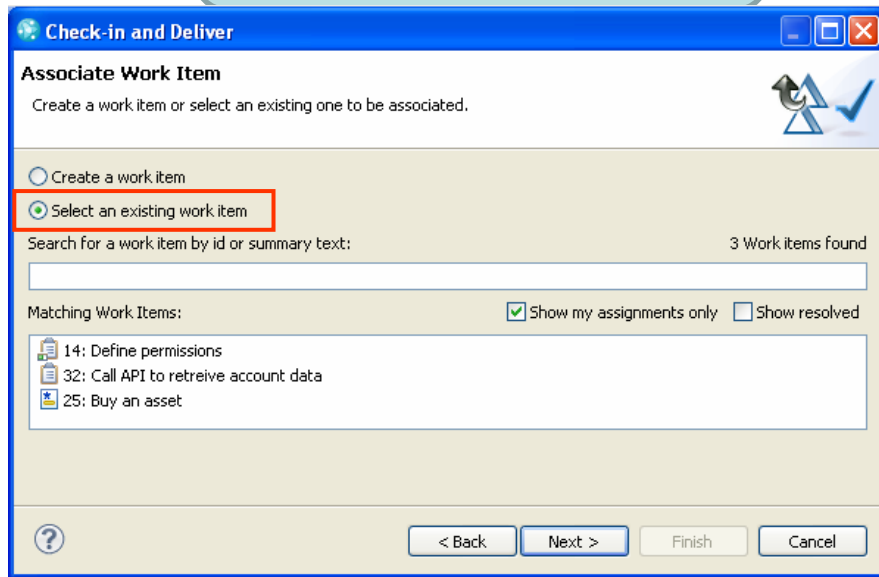
- Software Configuration Management (SCM) provides processes and methodologies for managing assets that
 - ▶ Are stored in a secure repository
 - ▶ Can be organized into versioned artifacts, components and subsystems
 - ▶ Can be baselined for milestones in projects, and tracked
 - ▶ Can be worked on by multiple developers at the same time
- RTC has its own indigenous source control that is built into Jazz for better integration
 - ▶ Storage model is based on the **change set** - a collection of changes to one or more files and folders



Integrated Configuration Management is critical for development

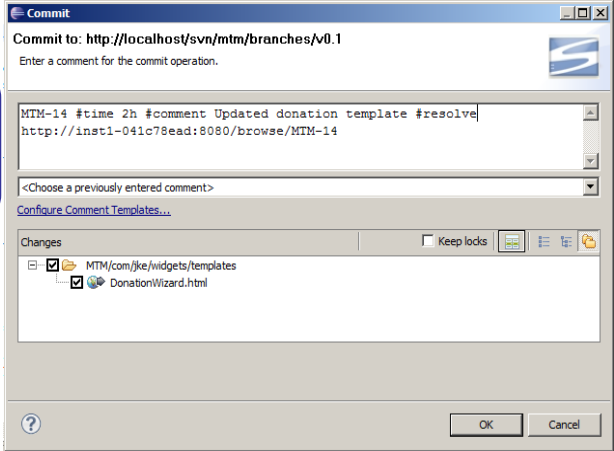
- RTC offers native source control allowing better integration especially when it comes to linking change sets with work items
- Many ALM suites integrate with 3rd party tools like Subversion for configuration management
- Results in poor integration -- in order to link a change set to a work item one has to enter the artifact ID in the comment field

Rational Team Concert
Planning, Work items,
SCM, Build, Deployment



JIRA or
TeamForge
or others...

Subversion



RTC uses a dual developer workspace design

- “Sandbox” – Developer work area where code changes made. In RTC:

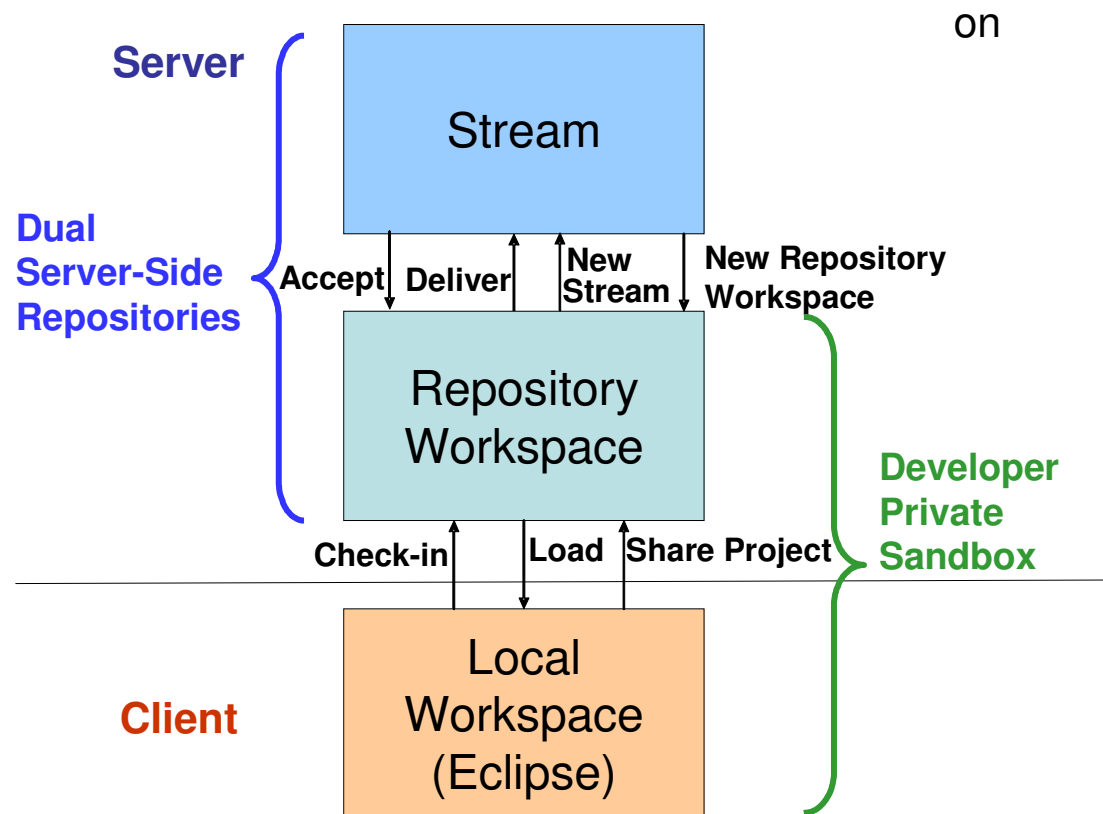
1. On the developer’s machine (Eclipse workspace shown), plus...
2. Personal repository workspace on the server

- Personal repository workspace

- ▶ Backed up with other server repositories
- ▶ Preserved in baseline and snapshots
- ▶ Searchable and collaborative
- ▶ Available to other team members to view and access

- Permits personal builds

- ▶ Test your code against latest team build before delivered to stream
- ▶ Avoids accidentally breaking team build with your changes



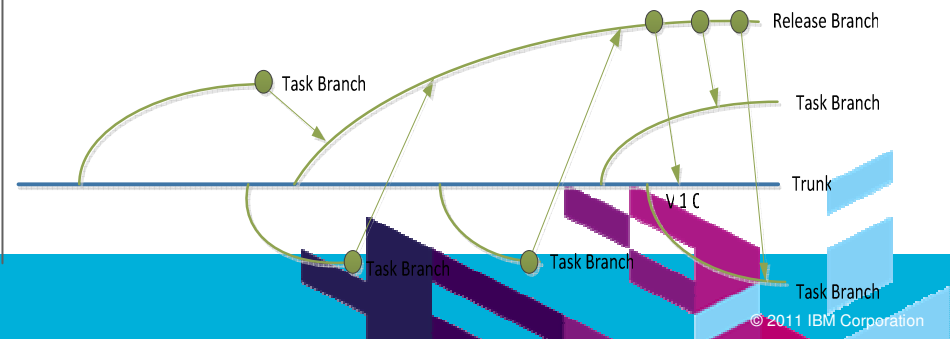
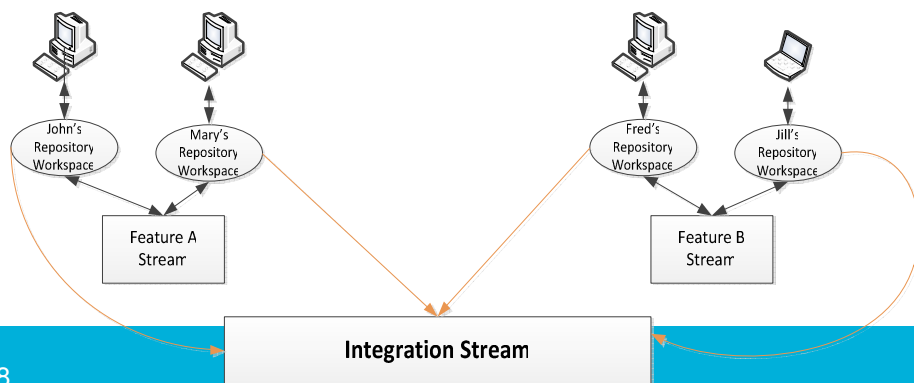
RTC Streams reduce the chaos that emerges from constant branching and merging

Change set: A collection of changes to one or more files and folders

Stream: A single configuration of source code

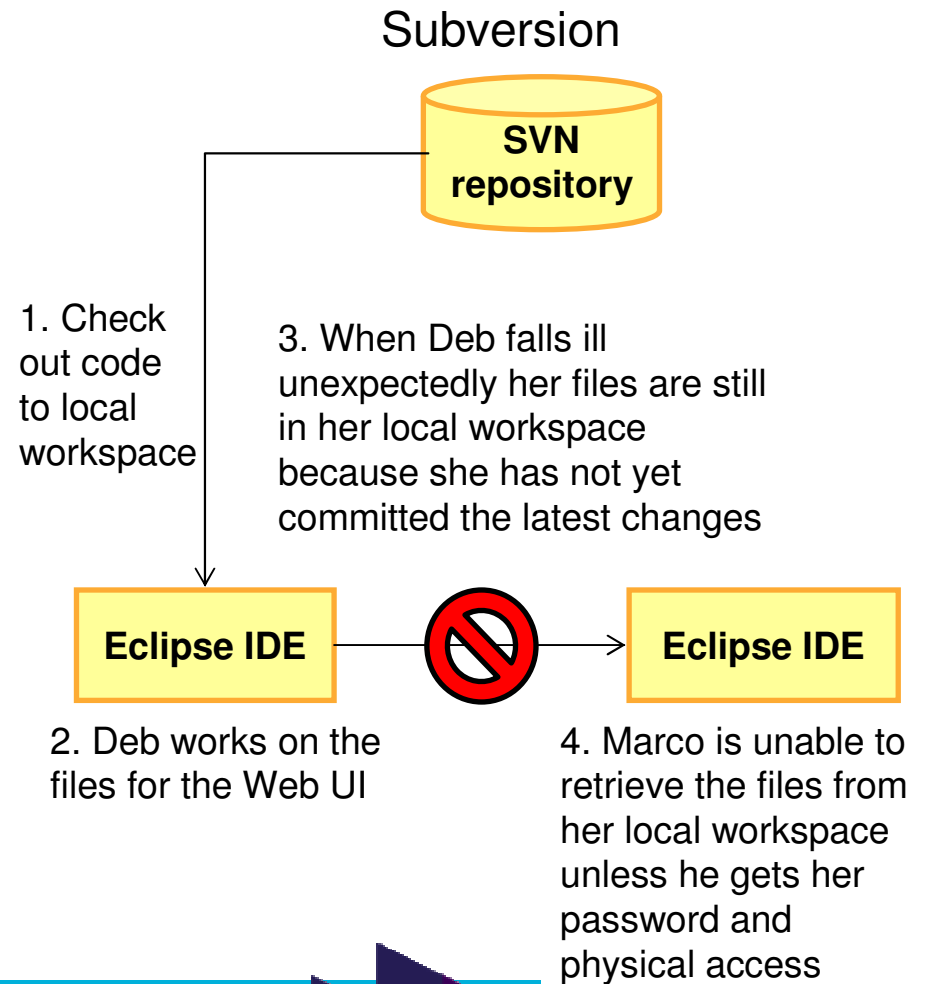
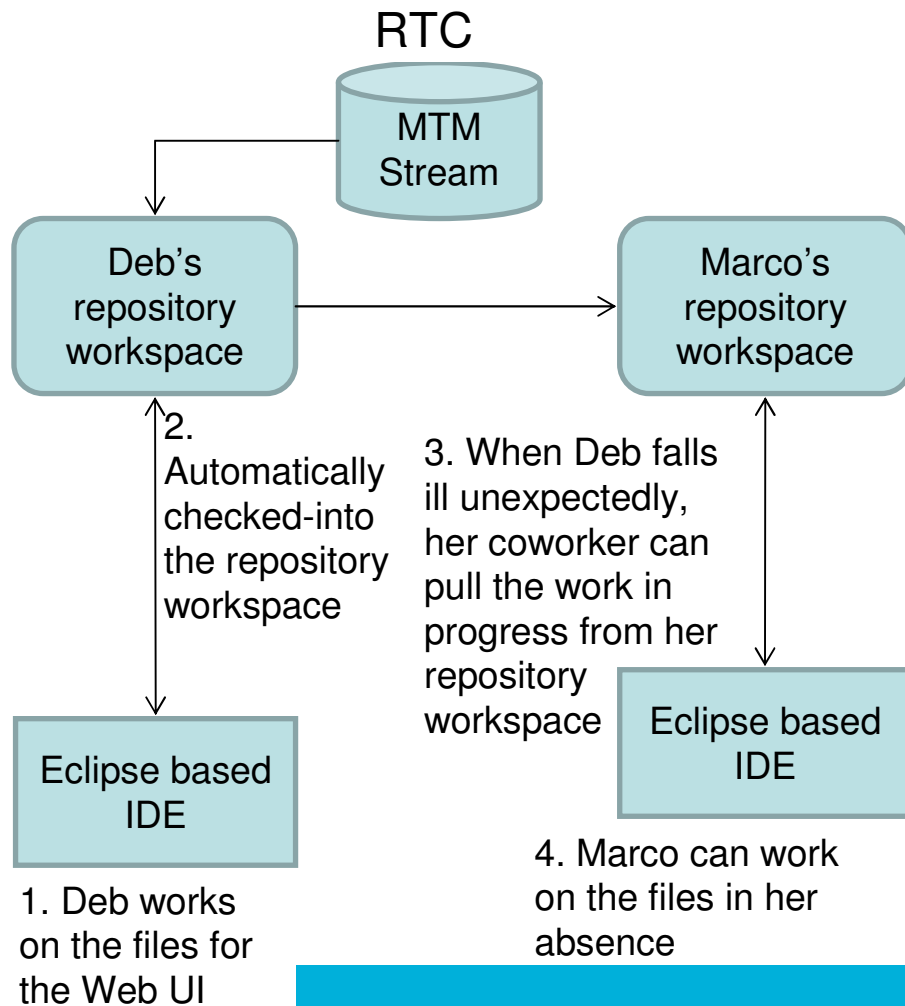
Team members can share change sets using streams or even in an ad hoc manner directly from each other's repository workspaces

- RTC's streams are integrated, organized and managed
- The repository manages searchable metadata about streams, components, and how they're linked
- The tool is the "integration agent"
- Subversion uses branches and merges, not streams
- Relies on conventions for naming and committing
- Depends on the underlying file system
- Developers need to understand that structure



Jazz Source Control allows a seamless hand off work between developers

- Example: A developer - Deb - is working on the website user interface. What will happen though when she suddenly falls ill?



RTC SCM allows mainframe programmers continue to develop using traditional tools if desired

ISPF Client For Team Concert

- Traditional ISPF programmers can continue to use familiar green-screen interface...
- ... but can integrate with Rational team services for software change management (SCM) functions
 - ▶ Use repository workspaces, change sets, link to work items, build requests, etc.
- Check out/check in code to native z/OS file system
- Facilitates phased implementation
- Reduces dependency on RDz deployment for mainframe programmers

```

Menu Help
-----
RTC/z Primary Option Menu
Option ==> 2_

0 Settings   Terminal & user parameters      ***** Logged in *****
1 Connection Work with Connection to source  Userid . . : robin
2 Workspaces Work with repository Workspace  Language. : ENGLISH
3 Edit       Work with source data          Server. . . :
4 Build      Work with Build options       Project . . :
X Exit       Terminate RTCz                   Workspace :
                                           Release . . :
    
```

ISPF SCM Client

```

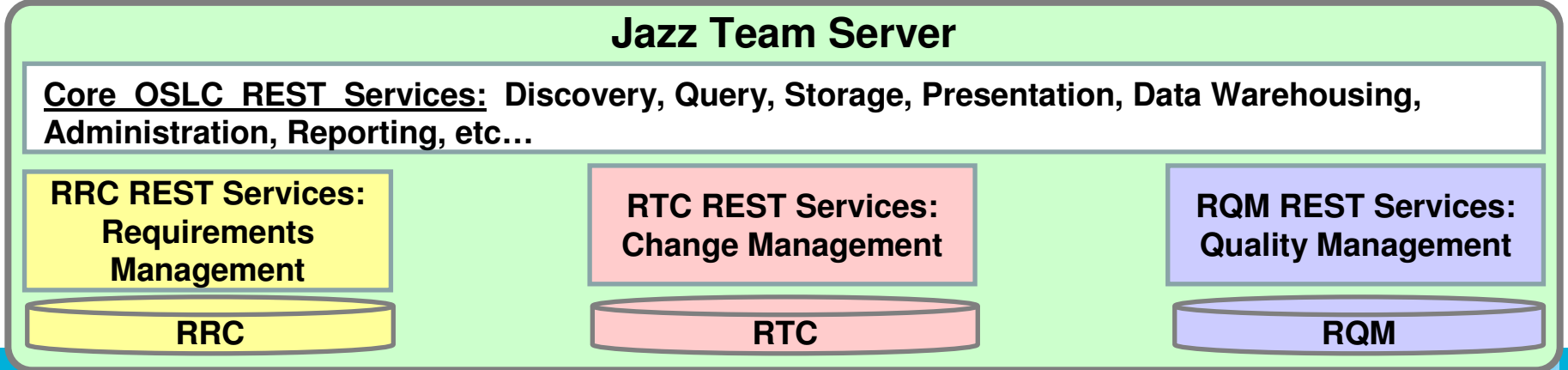
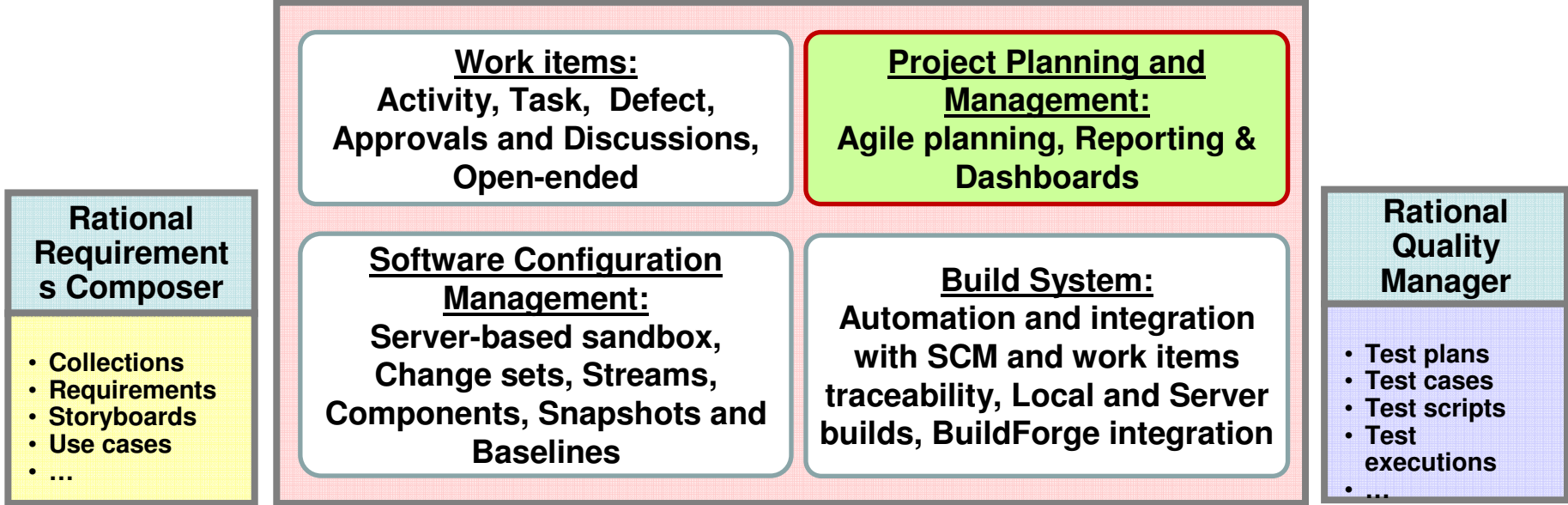
Menu Utilities Help
-----
Repository Workspaces                               Row 1 to 3 of 3
Option ==> _____ Scroll ==> CSR_

Enter new repository workspace name to create or "/" against existing
repository workspace for options

Names                                               Load location
-----
> Mortgage App Dev                                USER55.SANDBOX
- Test Workspace
- Weekly Integration Workspace
***** Bottom of data *****
    
```

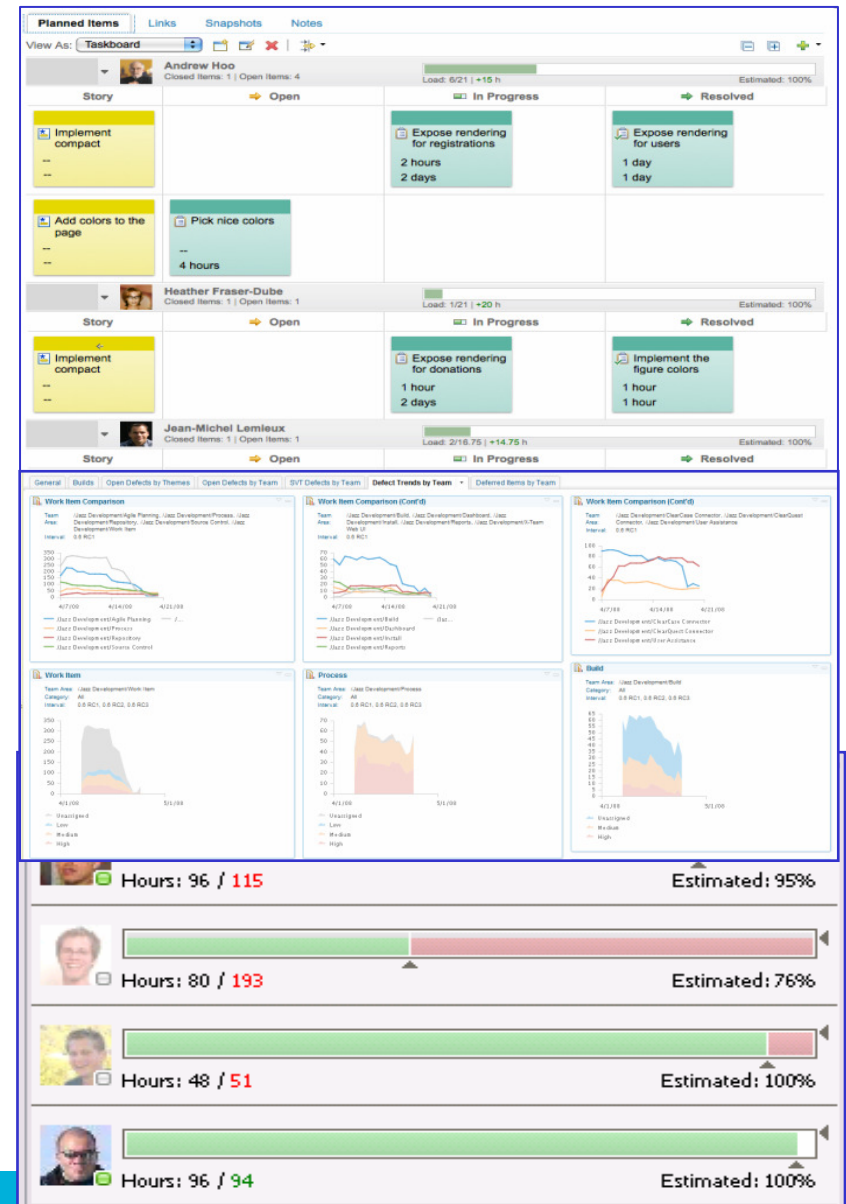
RTC Planning provides tools to assist with planning and execution of both agile and traditional projects

RTC Clients: Eclipse, RAD, RSA, Web browser



RTC planning and management is directly linked to execution

- Track progress during an iteration
- Balance work load of developers
- Determine schedule dependencies and constraints
- Make plans accessible to everyone on team
- Change plan dynamically over course of project to reflect team's position and direction
- Perform estimation and planning at daily, iteration, and release level
- Make plans and status Web-accessible to extended stakeholders



Planning across the entire team includes requirements and test team roles

JKE Banking (Change Management)

Project Dashboards ▾ Work Items ▾ Plans ▾ Source Control ▾ Builds ▾ Reports ▾

Release 1.0 Backlog ?

26 items: 19 open, 7 closed | Ends in: 24 days

▶ Plan Details

Planned Items **Links** ? Snapshots Dashboard Notes

Add: Contributes To Plan ▾

▼ Related Plans

Product Backlog [Product Backlog] 0/0 pts

▼ Implements Requirement Collection

▶ 344: Release 1 Planning

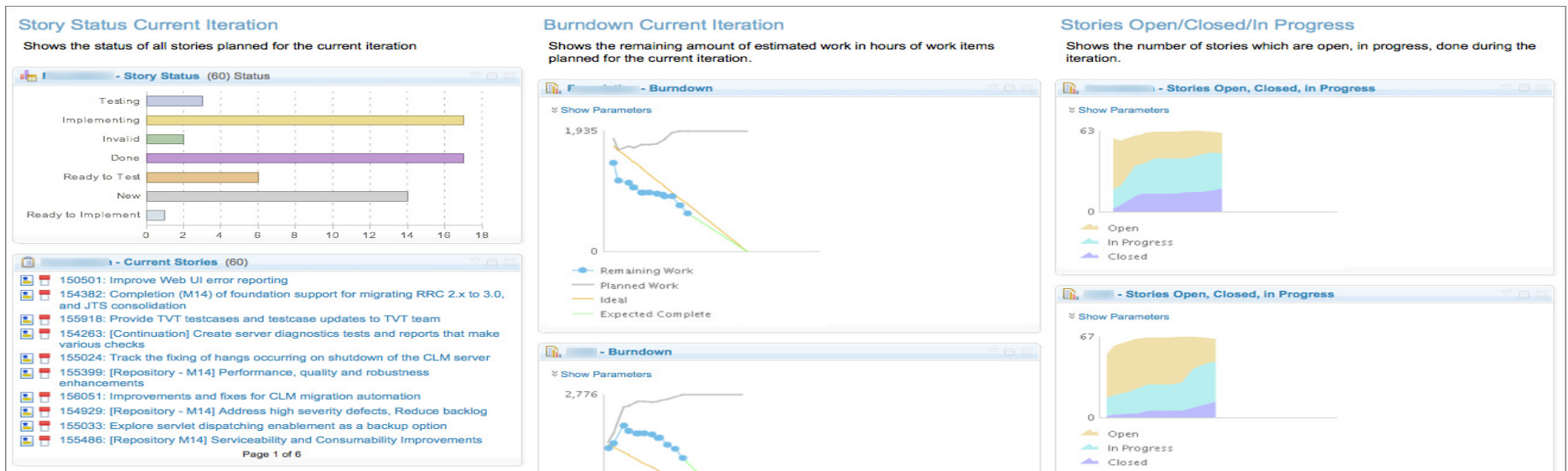
▼ Tested by Test Plan

10: JKE Banking Release 1

- Product Backlog
- Collection of requirements in RRC
- Test Plan in RQM

Reports and dashboards

- Keep tabs on project health
- Prevent unpleasant surprises
- Reports provide both real-time views and historical trends
- Ships with library of 50+ predefined reports
- Dashboards provide at-a-glance views on project progress
- Both customizable/configurable



Variety of views helps managers keep track of work items and changes

BRM Sprint 2 (1.0) Plan
27 items: 23 open, 4 closed | Ends in: 5 days

Plan Details

Planned Items Links Snapshots Dashboard Notes

View As: Work Breakdown (4 items filtered)

Resource	Effective Estimate	Progress	Status
Bob	4 hours	0/4 h	New
clmadmin	--	0/0 h	New
Deb	1 day	0/8 h	New
	2 hours	0/2 h	New
	--	0/13 pts 0/4	New
	--	0/8 pts 0/32	New

Work Breakdown Structure view has resource bars so that managers can determine who is overloaded. They can then manually level the assignments

Taskboard view is ideal for daily Scrums. At a glance everyone can see who is working on what in the pipeline

BRM Sprint 2 (1.0) Plan
owner: Business Recovery Matters | Iteration: Sprint 2 (3/27/11 - 4/9/11) | 5 Closed | 21 Open

To Do In Progress Done

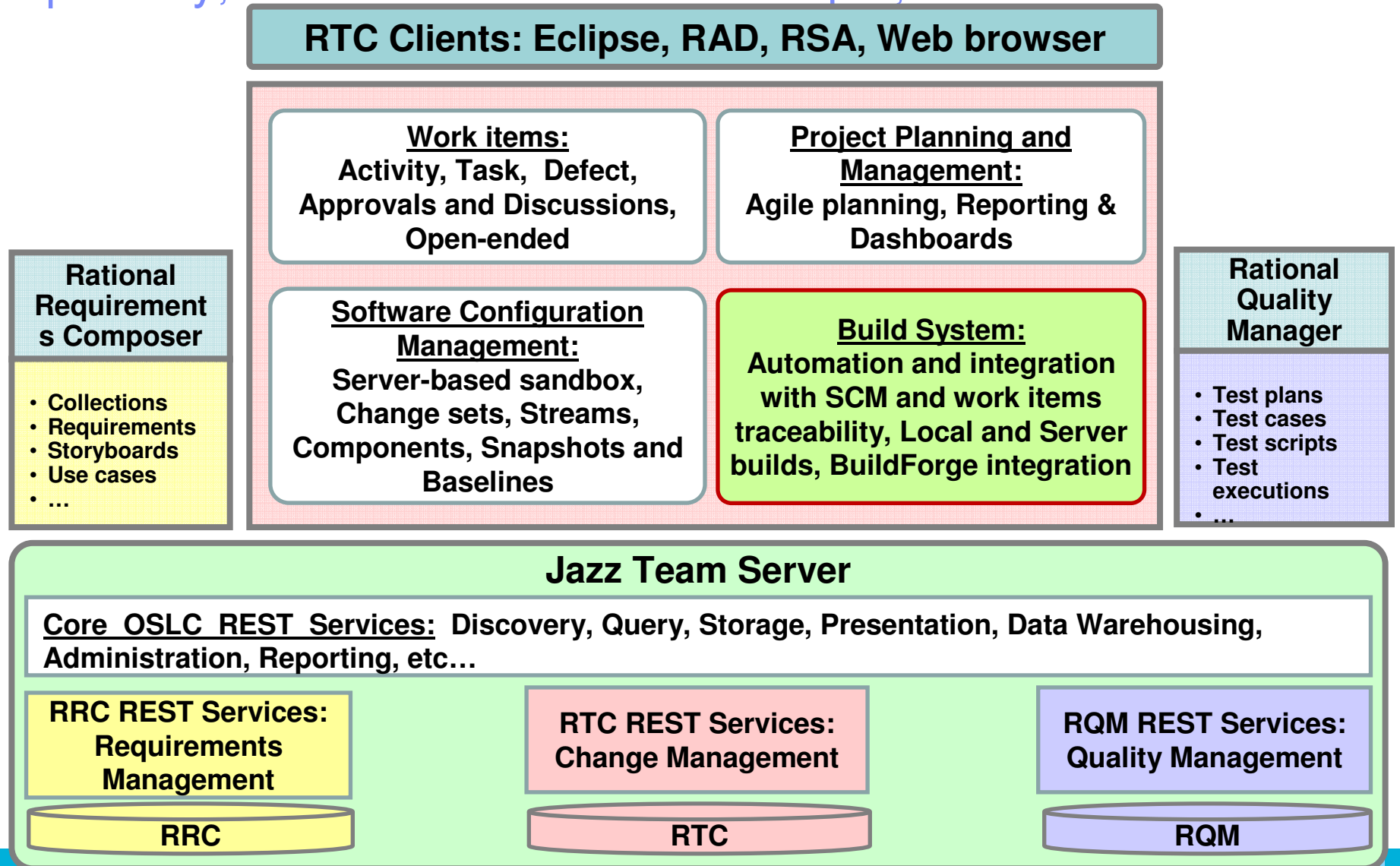
Item	Estimate	Assignee	Status
Implement - Donor Dividend Allocation Criteria	52		In Progress
JKE Charity Coordinator will respond to request in the website triggering	71		To Do
Implement - JKE Charity Coordinator will respond to request in the website triggering	53		In Progress
Organizations can Apply	69		To Do
Implement - Organizations can Apply	51		In Progress
Organization must provide justification for why funds are needed	66		To Do
Implement - Organization must provide justification for why funds are needed	49		In Progress

View As: Planned Time, Ranked List, **Taskboard**, Traceability, Work Breakdown

Actions: Re-sort

Exclude: Assigned Items, Empty Groups, Estimated Items, Execution Items, Expression, Theme Settings

RTC Build System provides means to retrieve work from repository, make available to build scripts, and deliver results



Automated builds save time and effort



- RTC Build System automates builds as part of the continuous integration process
 - ▶ Small changes are constantly applied and integrated
 - ▶ In some cases, builds are triggered on every commit
- Process of integrating early and often improves quality and reduces rework and cost
- The build can be fully automated
 - ▶ Compile the source
 - ▶ Generate documentation or supporting files
 - ▶ Package the binaries
 - ▶ Deploy
 - ▶ Trigger test execution
- RTC Build System tracks and monitors status of builds

Development teams can schedule and execute software builds efficiently

- Visibility to build engines, queues and schedules
- Results of each build displayed on status page
 - ▶ Any failures/errors highlighted
- Drill down for each build run:
 - ▶ Performance
 - ▶ Work performed
- Build results can link to change sets and work items
 - ▶ Traceability across the lifecycle
- Build results can deploy to servers for testing or production

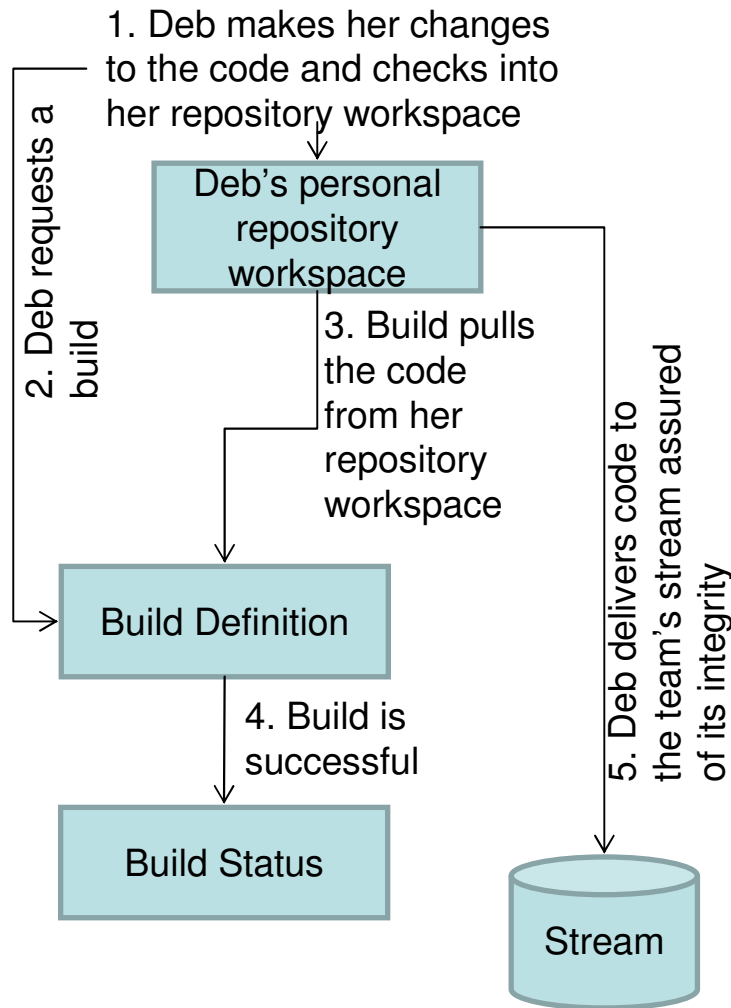
Work items, tests and change sets included in the build

Build Status

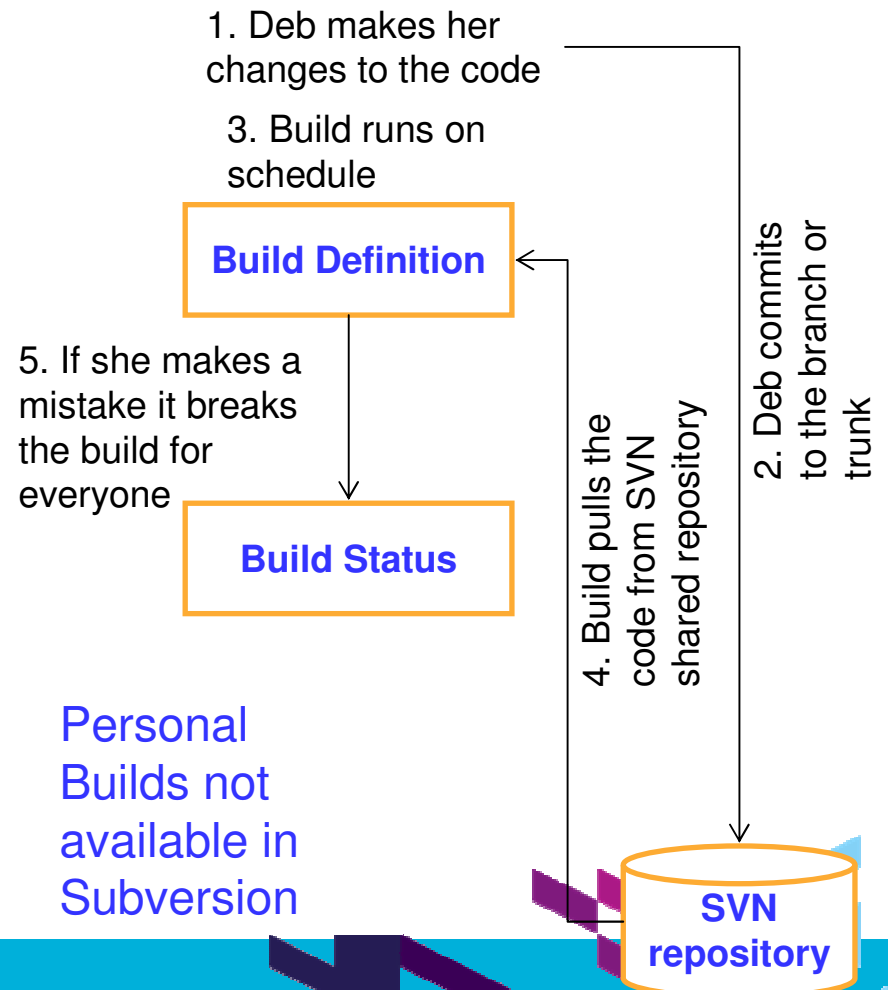
Build	Label	Progress	Estimated Completion	Start Time	Duration
✓ jke.dev	I20110306-1326	Completed		March 6, 2011 1:26:51 PM	6 minutes, 21 seconds
✓ jke.dev	I20110221-2047	Completed		February 21, 2011 8:...	6 minutes, 26 seconds

With personal builds you can test it out first before pushing it to the entire team

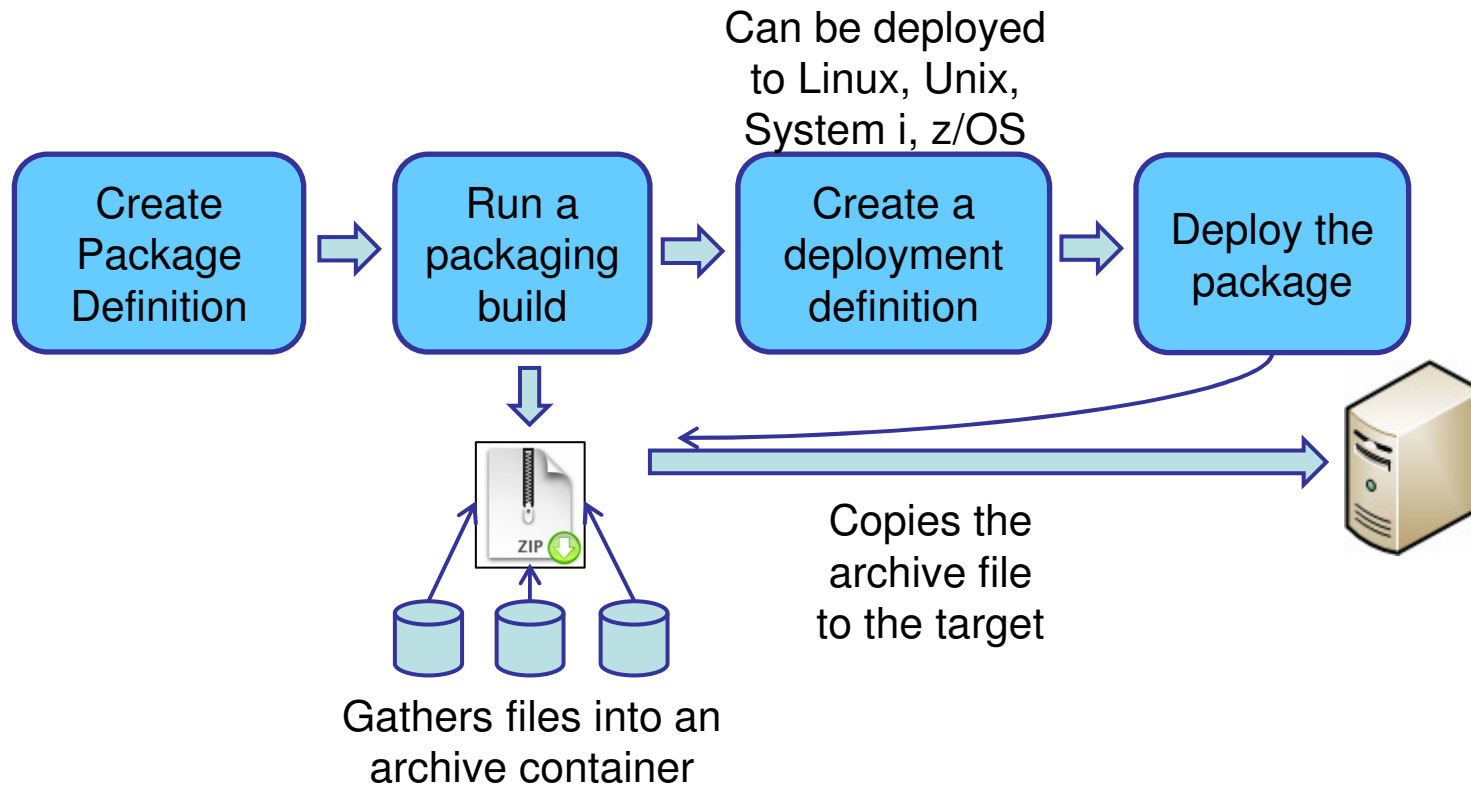
Running a personal build in RTC



Build process with Subversion + 3rd party build engine



After the Build, RTC creates the deployment package and deploys it to the target server



For organizations that require advanced integration, IBM Offers Build Forge

- Can execute centrally-controlled builds in parallel across distributed systems, speeding up the overall process
- Utilizes sophisticated scripting and scheduling of the steps comprising each build
 - ▶ Minimizes the amount of manual intervention required by a build engineer
- Support for a broad spectrum of platforms (including System z), scripting languages, and source code repositories
 - ▶ For usage as an enterprise standard build infrastructure
- Builds source code, but also automatically deploys the underlying software infrastructure (e.g. WebSphere Application Server, Portal) for the code to run on
 - ▶ This level of automation completely frees up the build engineer and is not found in any competing product
- Tight integration with Rational Team Concert and Jazz artifacts for end-to-end traceability

RTC offers tight integration with Rational Build Forge

RTC and Build Forge integration is ideal for customers who require:

- Enterprise-wide process automation across multiple platforms
- Software assembly for Jazz and traditional environments
- Build server load balancing
- Deployment process automation support
- Stringent compliance mandates for governance and traceability

Even when Build Forge manages and runs your builds, RTC will:

- Publish build results
- Show links to RTC-managed change sets and work items consumed by build
- Create metrics reporting/health status of Build Forge builds
- Start Build Forge jobs from the RTC User Interface

Summary

- Integrated Planning and Work Items management provides in-context collaborative environment
 - ▶ Plan is always up to date, developers and other team members track their work with zero overhead
- Life cycle traceability of work items
 - ▶ Coverage of requirements, development, builds and test results
- Productive Software Configuration Management
 - ▶ Integrated Jazz Source Control links code artifacts to work items and then to builds and build results
 - ▶ Enables developers to effectively work in parallel and to easily track versions of code artifacts for issue resolution
- Integrated and extensible Build System that seamlessly links to work items and code version artifacts
 - ▶ Can execute tests and automatically create issues linked to code artifacts and work items
 - ▶ Supports all platforms
- RTC creates and deploys packages on all platforms

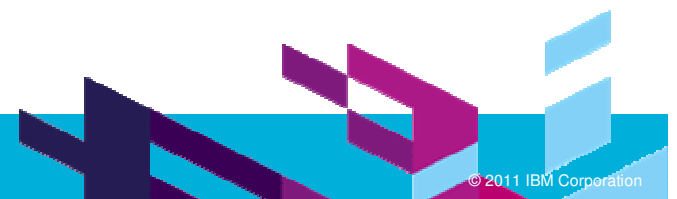


www.ibm/software/rational



QUESTIONS

www.ibm/software/rational





www.ibm/software/rational

© Copyright IBM Corporation 2011. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, Rational, the Rational logo, Telelogic, the Telelogic logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.