Innovation through Collaboration

Advanced source control using *Rational Team Concert*

Morten S. Madsen
IBM Rational Software
IT Specialist
msm@dk.ibm.com
+45 4120 5474

Agenda:

- What is Jazz and Rational Team Concert?
- Basic RTC SCM terms
- Collaboration and merging
- Branching – everybody's doing it!?
- Promotion
- Permissions and security
- Code consistency and preconditions
- Resolving ”normal” challenges using SCM

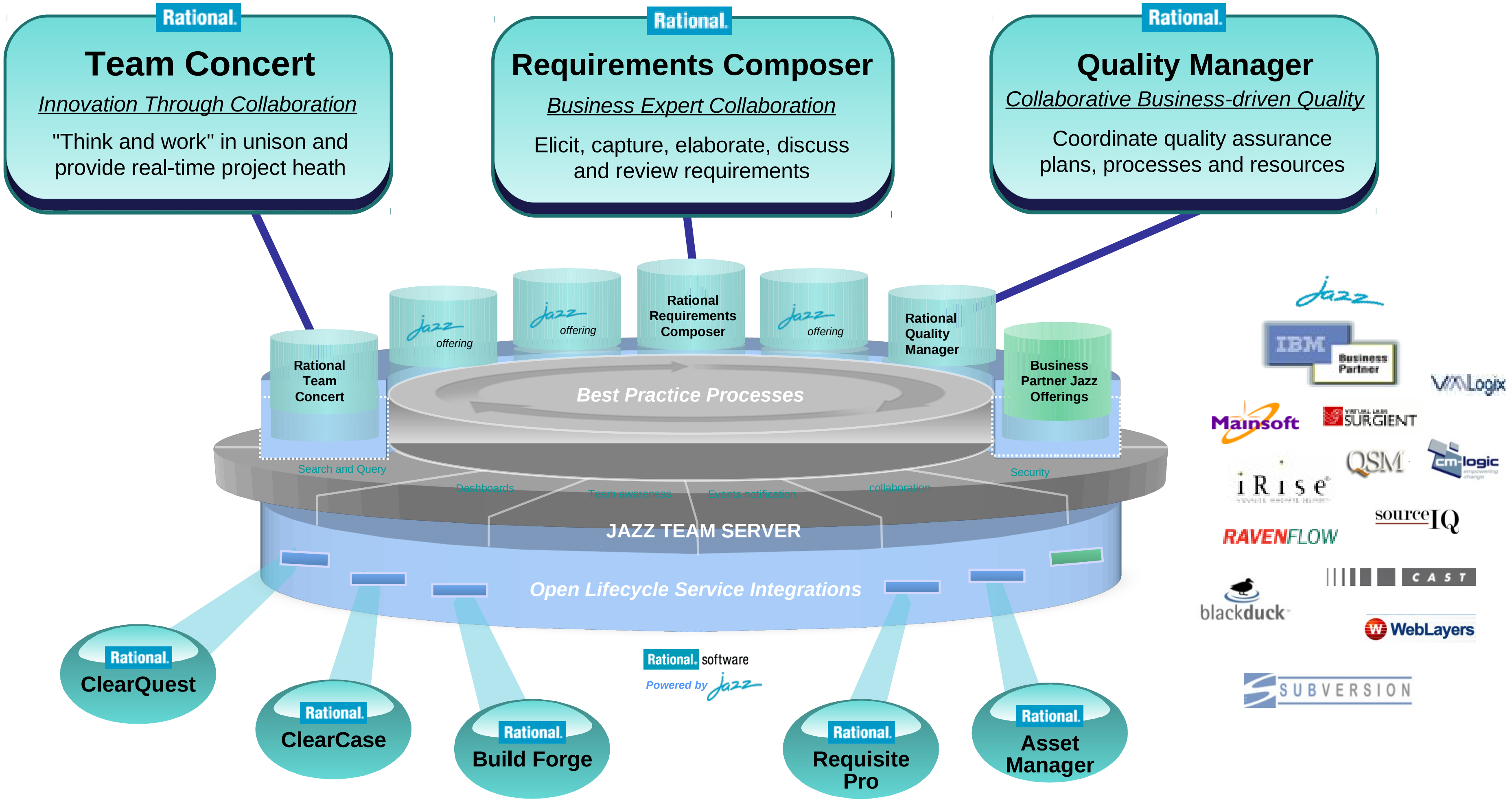
The Jazz project

People working together to deliver great software

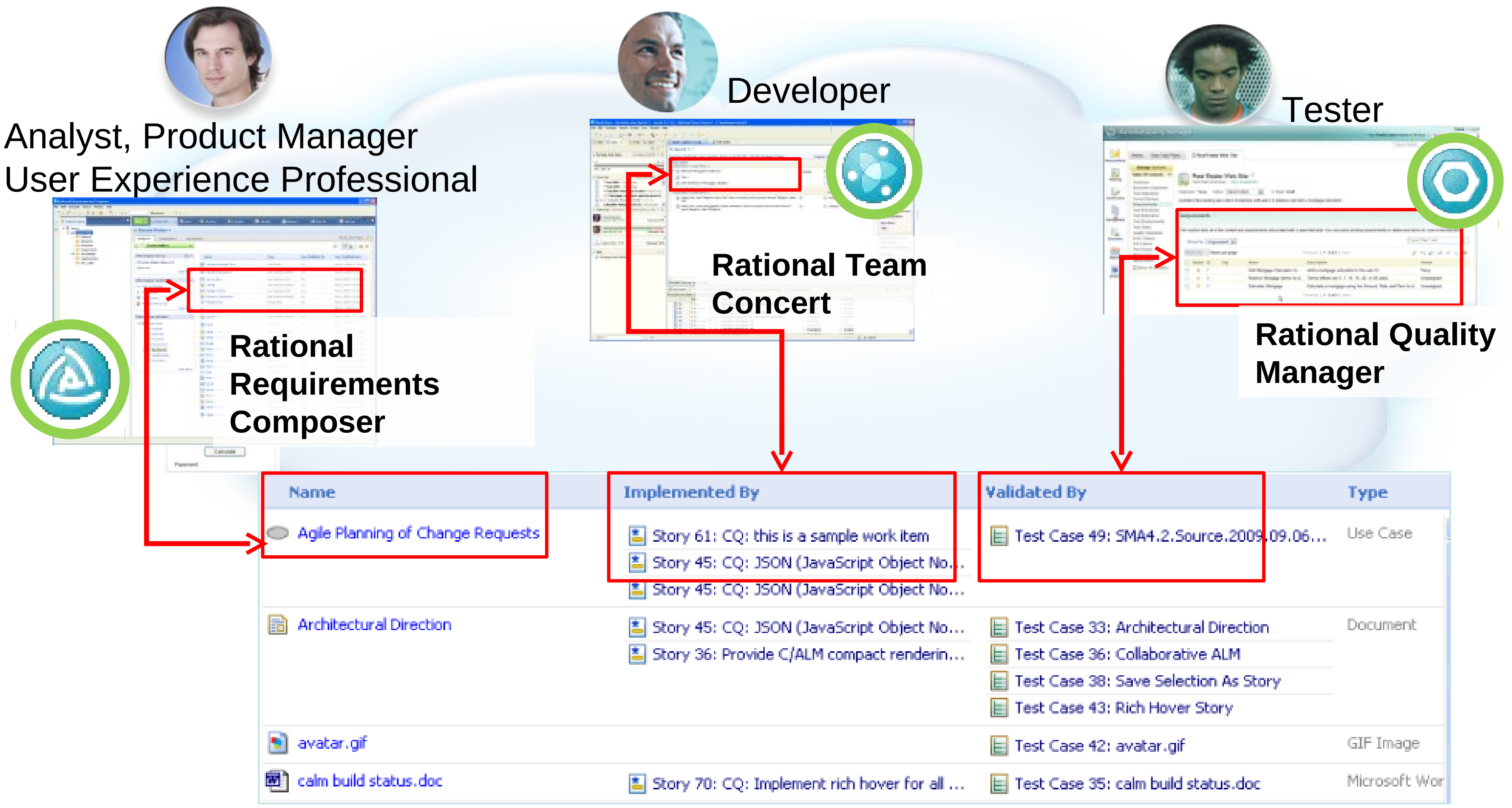
Jazz is a project and platform for *transforming how people work together* to deliver greater value and performance from their software investments.

- A major investment by IBM to create **a scalable, extensible team collaboration platform**.
- IBM's vision of the **future of software delivery**–globally distributed, fluid and dynamic.
- An **evolution of the Rational portfolio**, which will evolve to support Jazz technology over time.
- **A community at Jazz.net** – where you can see Jazz-based products being built.

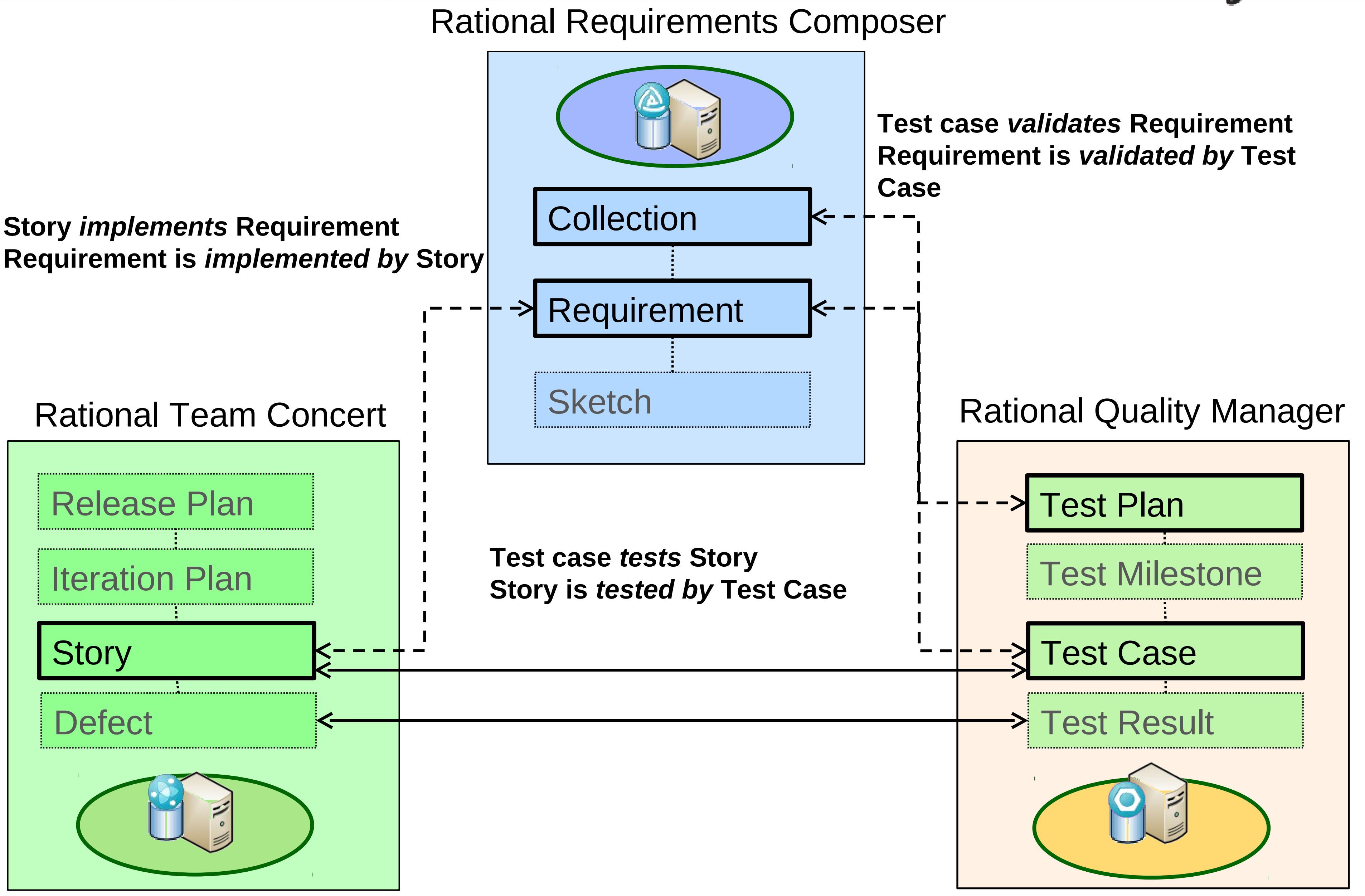
Introducing *Jazz*



CLM – Collaborative Lifecycle Management



Name	Implemented By	Validated By	Type
Agile Planning of Change Requests	Story 61: CQ: this is a sample work item Story 45: CQ: JSON (JavaScript Object No... Story 45: CQ: JSON (JavaScript Object No...	Test Case 49: SMA4.2.Source.2009.09.06...	Use Case
Architectural Direction	Story 45: CQ: JSON (JavaScript Object No... Story 36: Provide C/ALM compact renderin...	Test Case 33: Architectural Direction Test Case 36: Collaborative ALM Test Case 38: Save Selection As Story Test Case 43: Rich Hover Story	Document
avatar.gif		Test Case 42: avatar.gif	GIF Image
calm build status.doc	Story 70: CQ: Implement rich hover for all ...	Test Case 35: calm build status.doc	Microsoft Wor



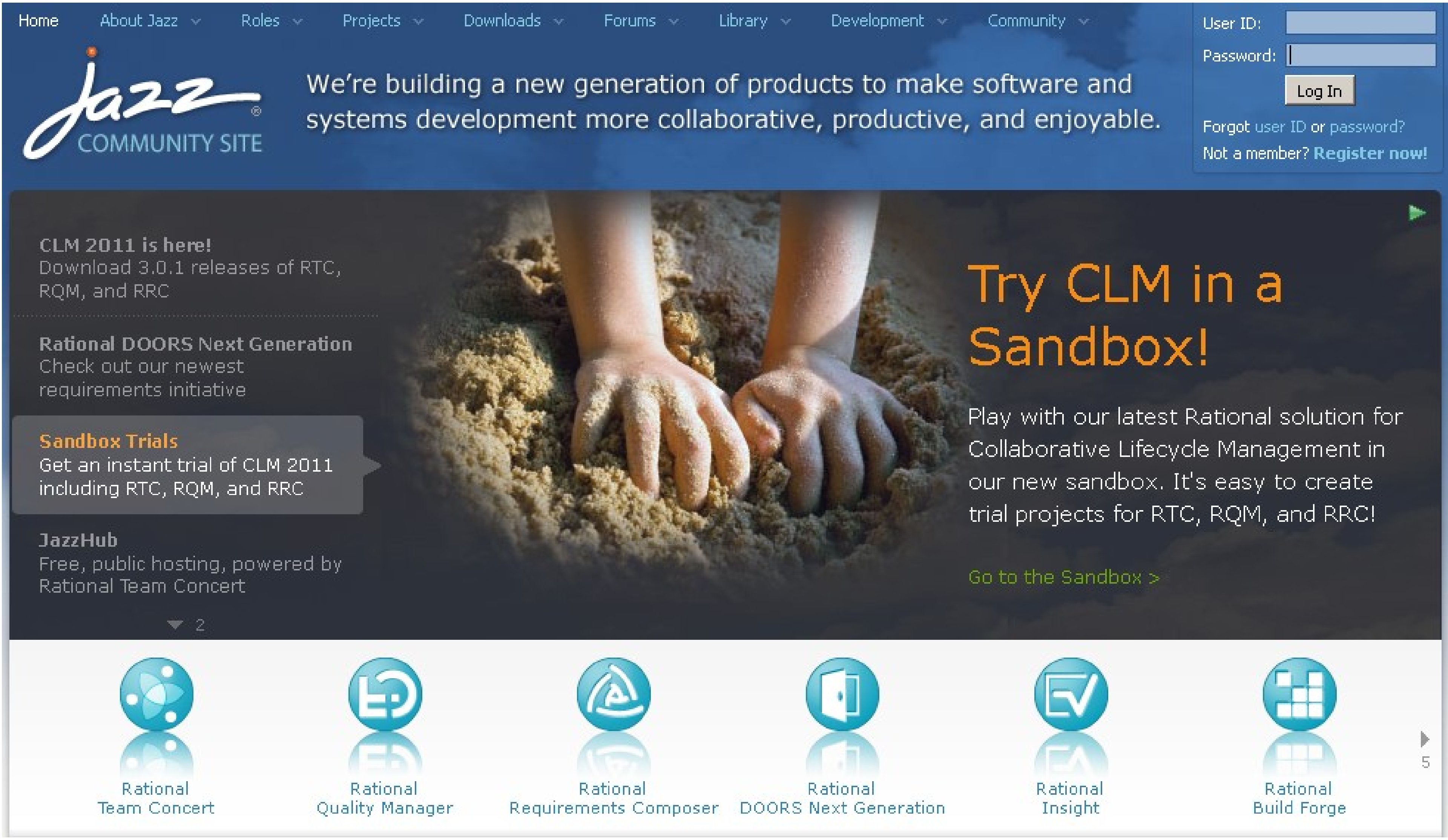
Test case *validates* Requirement
Requirement is *validated by* Test Case

Story *implements* Requirement
Requirement is *implemented by* Story

Test case *tests* Story
Story is *tested by* Test Case

Rational Team Concert

- Planning
- Process
- Change / task / defect management
- Source control
- Build management
- Dashboards



Home | About Jazz | Roles | Projects | Downloads | Forums | Library | Development | Community

User ID:
Password:
Log In
Forgot user ID or password?
Not a member? Register now!

Jazz COMMUNITY SITE

We're building a new generation of products to make software and systems development more collaborative, productive, and enjoyable.

CLM 2011 is here!
Download 3.0.1 releases of RTC, RQM, and RRC

Rational DOORS Next Generation
Check out our newest requirements initiative

Sandbox Trials
Get an instant trial of CLM 2011 including RTC, RQM, and RRC

JazzHub
Free, public hosting, powered by Rational Team Concert

2

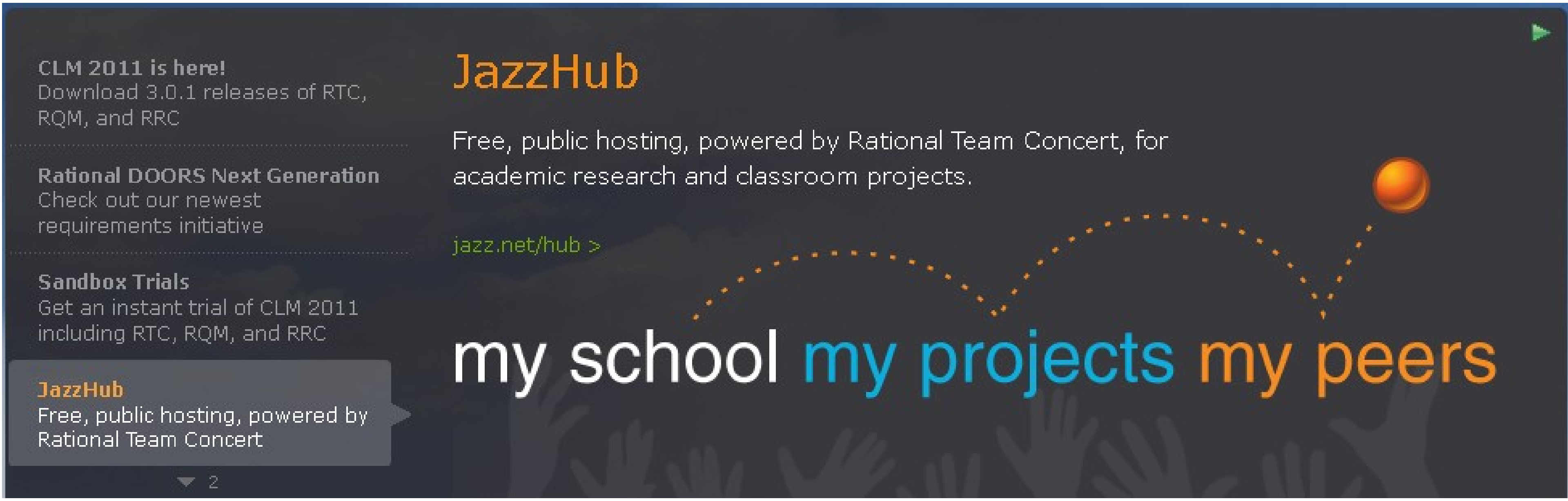
Try CLM in a Sandbox!

Play with our latest Rational solution for Collaborative Lifecycle Management in our new sandbox. It's easy to create trial projects for RTC, RQM, and RRC!

Go to the Sandbox >

Rational Team Concert | Rational Quality Manager | Rational Requirements Composer | Rational DOORS Next Generation | Rational Insight | Rational Build Forge

5



CLM 2011 is here!
Download 3.0.1 releases of RTC, RQM, and RRC

Rational DOORS Next Generation
Check out our newest requirements initiative

Sandbox Trials
Get an instant trial of CLM 2011 including RTC, RQM, and RRC

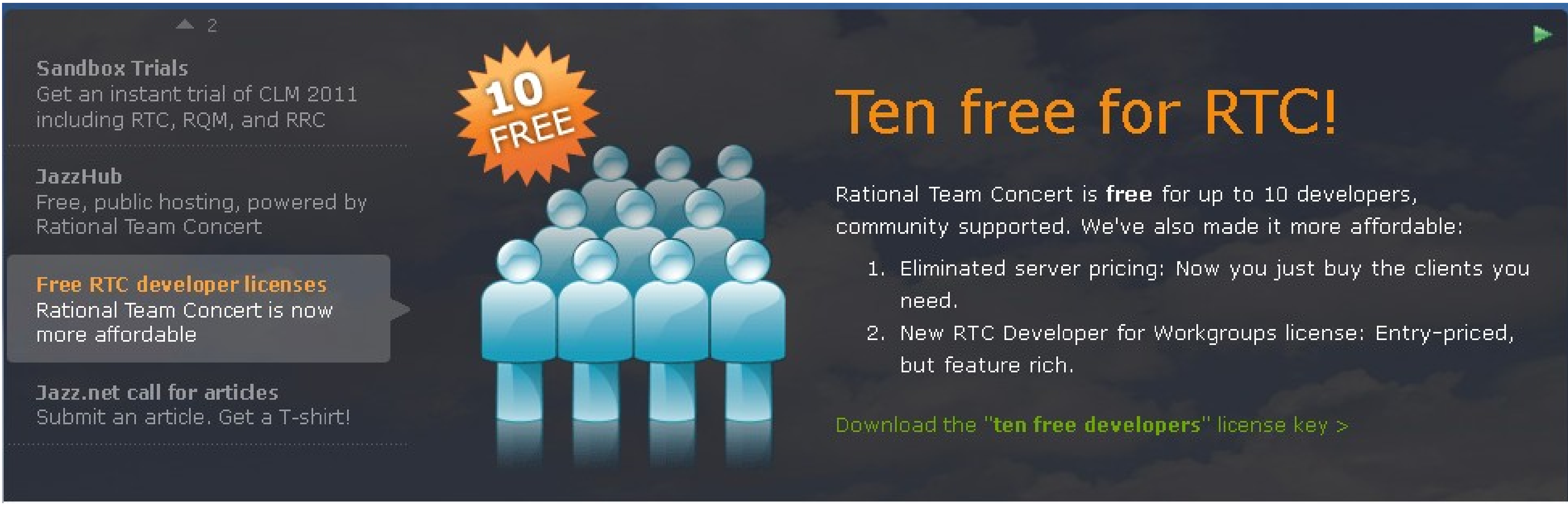
JazzHub
Free, public hosting, powered by Rational Team Concert

2

JazzHub

Free, public hosting, powered by Rational Team Concert, for academic research and classroom projects.

jazz.net/hub >



Sandbox Trials
Get an instant trial of CLM 2011 including RTC, RQM, and RRC

JazzHub
Free, public hosting, powered by Rational Team Concert

Free RTC developer licenses
Rational Team Concert is now more affordable

Jazz.net call for articles
Submit an article. Get a T-shirt!

Ten free for RTC!

Rational Team Concert is **free** for up to 10 developers, community supported. We've also made it more affordable:

1. Eliminated server pricing: Now you just buy the clients you need.
2. New RTC Developer for Workgroups license: Entry-priced, but feature rich.

Download the "**ten free developers**" license key >

Basic RTC SCM terms

- Workspaces

Basic RTC SCM terms

- Workspaces
- Streams

Basic RTC SCM terms

- Workspaces
- Streams
- Change-sets

Basic RTC SCM terms

- Workspaces
- Streams
- Change-sets
- Components

Basic RTC SCM terms

- Workspaces
- Streams
- Change-sets
- Components
- Baselines

The story – basic SCM

- A developer is working on a computer game.

The story – basic SCM

- A developer is working on a computer game.
- He wants to backup his work and keep track of changes to the source code.

The story – basic SCM

- A developer is working on a computer game.
- He wants to backup his work and keep track of changes to the source code.
- He uses a workspace for this.

The story – basic SCM

- A developer is working on a computer game.
- He wants to backup his work and keep track of changes to the source code.
- He uses a workspace for this.
- He can do checkins, roll -back / -forward and basically store and manage all changes.

The story – basic SCM

- A developer is working on a computer game.
- He wants to backup his work and keep track of changes to the source code.
- He uses a workspace for this.
- He can do checkins, roll -back / -forward and basically store and manage all changes.
- He can use baselines for storing ”labeled versions” so these always will be easily available.

DEMO

Collaboration and merging

- Workspace to workspace

Collaboration and merging

- Workspace to workspace
- Using a stream for collaboration

The story - Collaboration

- Another developer joins the project.

The story - Collaboration

- Another developer joins the project.
- Gets personal workspace.

The story - Collaboration

- Another developer joins the project.
- Gets personal workspace.
- Collaboration can happen directly between workspaces.

The story - Collaboration

- Another developer joins the project.
- Gets personal workspace.
- Collaboration can happen directly between workspaces.
- They use a stream for tracking the state of development and for standard collaboration.

DEMO

Branching – everybody's doing it!?

- Feature streams

Branching – everybody's doing it!?

- Feature streams
- Several personal workspaces

The story - Branching

- A third person joins the group.

The story - Branching

- A third person joins the group.
- They already have a current release v0.1.0.

The story - Branching

- A third person joins the group.
- They already have a current release v0.1.0.
- The decide to work on the following:
  - Release v0.1.0 maintenance.
  - Release v0.2.0 development.
  - New experimental feature.

DEMO

Promotion

- Workspace handled promotion.

Promotion

- Workspace handled promotion.
- Inter-stream promotion.

The story - Promotion

- Now challenges with propagation of changes arise:

The story - Promotion

- Now challenges with propagation of changes arise:
  - Release v0.1.x fixes needs to flow to all development streams.

The story - Promotion

- Now challenges with propagation of changes arise:
  - Release v0.1.x fixes needs to flow to all development streams.
  - Main development (Release v0.2.0) needs to merge into new feature stream.

DEMO

Permissions & security

- Role & Team based permissions

The story - Permissions

- They create a release stream to track the currently deployed baseline.

The story - Permissions

- They create a release stream to track the currently deployed baseline.
- They create the role ”release engineer”.

The story - Permissions

- They create a release stream to track the currently deployed baseline.
- They create the role ”release engineer”.
- They setup rules so only the ”release engineer” role can deliver changes to release stream.

DEMO

Consistency & preconditions

- Continuous builds (compile and run unit tests)

Consistency & preconditions

- Continuous builds (compile and run unit tests)
- Preconditions for a deliver operation

The story - Consistency

- A new ”hotfix” stream is created to handle emergency fixes to production code.

The story - consistency

- A new ”hotfix” stream is created to handle emergency fixes to production code.
- A precondition is added: ”no incoming changes at deliver” to ensure up to date streams.

The story - consistency

- A new ”hotfix” stream is created to handle emergency fixes to production code.
- A precondition is added: ”no incoming changes at deliver” to ensure up to date streams.
- An integration stream is created for tweaking and testing code.

The story - consistency

- A new ”hotfix” stream is created to handle emergency fixes to production code.
- A precondition is added: ”no incoming changes at deliver” to ensure up to date streams.
- An integration stream is created for tweaking and testing code.
- An automated build is setup to ensure quality in integration stream.

Resolving normal challenges

- Whoooops!! I shouldn't have delivered that!!!

Resolving normal challenges

- Whoooops!! I shouldn't have delivered that!!!
- Argh, who delivered this? It doen't run!

Resolving normal challenges

- Whoooops!! I shouldn't have delivered that!!!
- Argh, who delivered this? It doen't run!
- I'm in the middle of working on this task, but now there's a more important task with higher priority!

The story – normal challenges

- Joe accidentally delivers something to the hotfix stream.

The story – normal challenges

- Joe accidentally delivers something to the hotfix stream.
  - He reverses the change.

DEMO

The story – normal challenges

- Joe accidentally delivers something to the hotfix stream.
- Lisa delivers her new feature without checking if the code runs.

The story – normal challenges

- Joe accidentally delivers something to the hotfix stream.
- Lisa delivers her new feature without checking if the code runs.
  - Joe has to deliver her fix as hotfix in release stream (production).

The story – normal challenges

- Joe accidentally delivers something to the hotfix stream.
- Lisa delivers her new feature without checking if the code runs.
  - Joe has to deliver her fix as hotfix in release stream (production).
  - The hotfix is propagated to main development and from there to Morten & Lisa

The story – normal challenges

- Joe accidentally delivers something to the hotfix stream.
- Lisa delivers her new feature without checking if the code runs.
- Morten is currently working on a task, but another task is now needing higher priority

The story – normal challenges

- Joe accidentally delivers something to the hotfix stream.
- Lisa delivers her new feature without checking if the code runs.
- Morten is currently working on a task, but another task is now needing higher priority
  - The current changes are suspended

The story – normal challenges

- Joe accidentally delivers something to the hotfix stream.
- Lisa delivers her new feature without checking if the code runs.
- Morten is currently working on a task, but another task is now needing higher priority
  - The current changes are suspended
  - OR a new workspace is created

The story – normal challenges

- Joe accidentally delivers something to the hotfix stream.
- Lisa delivers her new feature without checking if the code runs.
- Morten is currently working on a task, but another task is now needing higher priority
  - The current changes are suspended
  - OR a new workspace is created
  - New task is done & delivered, then old task is resumed

DEMO

Thank
You