Envisioning a platform that can transform software delivery

An open technology initiative to transform how people work together to deliver greater value and performance from their software investments.

- Robust, extensible and scaleable
- Globally distributed, fluid and dynamic
- Community-based and open at Jazz.net

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

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

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

Real-time integration of people, process and projects across the lifecycle
Envisioning a platform that can transform software delivery

An open technology initiative to transform how people work together to deliver greater value and performance from their software investments.

- Robust, extensible and scaleable
- Globally distributed, fluid and dynamic
- Community-based and open at Jazz.net

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

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

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

Real-time integration of people, process and projects across the lifecycle
Translating words to actions

**Collaborate in Context**
- Collaboration
- Fewer meetings and status reports
- Reduction in late scrap and rework
- Reduced attrition of skills and greater leveraging top technical / business talent

**Right-size Governance**
- Clarity
- Easier to stay compliant with standards
- Earlier detection and accelerated resolution of defects
- Improved predictability of projects

**Day One Productivity**
- Continuity
- Project initiation in days not weeks
- Reduction in time to first demonstration
- Improved cycle times through automated patterns, services and architectures

**Open and Extensible Architecture**
- Community
- Viral, but Controlled Development
- Expanded Community
- Selectable integration characteristics
Delivering global team collaboration and visibility
Allows individuals to focus on what they do best

- Make confident go/no decisions based on facts, not estimates.
- Provision a new team member across the world in a day
- Build design reviews into your software lifecycle—no more chasing after rogue development streams.
- Track progress on the updates that your customers care about.
- Create great software, unencumbered by tedious status reports
- Know exactly what to fix when something breaks
“I need to know what was delivered in the release and what is in-plan for next release.”

“Is there something I can reuse?”

“Is the requirement implemented? With what quality?”

“We need to plan our work and implement requirements.”

“We need a single dashboard to view health.”

“We need to manage the End-Game determine when we are ready to release.”

“I need to consolidate team plans into a project delivery plan.”

“I need to run a central build, validation and delivery service.”

“I need to configure my test plans from delivered changes and report on test coverage.”

“I need a global lab to support my solution test schedule.”

“I need to automate my component builds and verification tests using central build services.”

“I need to plan our tests and trace to the release requirements.”

“I need to plan project iteration.”

“The requirement is implemented? With what quality?”

“The requirement was delivered in the release and what is in-plan for next release.”

“We need to manage the End-Game determine when we are ready to release.”

“We need to consolidate team plans into a project delivery plan.”

“We need a single dashboard to view health.”

“I need a single dashboard to view quality.”
Introduction to the Jazz Technology Platform

- **Update Iteration Plan**
- **Develop Component**
- **Build and Verify Solution**
- **Manage Solution Quality**
- **Deliver Solution**

**RAM**
- Requirement
- Sketch
- Reuse

**RTC**
- Plan Iteration
- Develop & Test
- Build & Deliver
- Validate Quality against Requirements

**CQALM & Interop**
- Triage Requests
- Plan Iteration
- Respond, Monitor Health

**CC Interop**
- Alternate UCM workflow

**Build Forge**
- Plan Test Effort
- Integrate & Build
- Rational Analyzer and WatchFire AppScan integrated

**Quality Manager**
- Submit defect

Final Distribution

Acceptance Criteria Met

All Work Complete With expected Quality

Baseline & Package

Test integration build

With expected Quality
Introduction to the Jazz Technology Platform

Jazz Architecture

- **Eclipse Client Platform**
- **Web Client Platform**
- **Visual Studio Client Platform**

Jazz Team Server
- Project and Team Structure
- Event Notification
- Process Enactment
- Security and Access
- Query
- Search
- Storage

Jazz Repository
- Defects
- Requirements
- Use-cases, ....
- Items and relationships
  - Event history,
  - Item history trends
- Builds
- Source code,
- Test-cases
- Test results
Extending the Jazz Platform

App Server

Jazz.war

Equinox Bridge Servlet

/jazz/*

/jazz/service/*

../IWorkltemService

../IFileSystemService

../IPlanRestService

/jazz/web/*

../WorkltemView.js

../PlanView.js
Jazz Server Kernel

- Team Repository Services
  - Provides the basics for querying, fetching, and updating repository data
  - Provides a platform for extensible AJAX Web UIs
  - Provides support for change events and feed rendering
  - Provides access to and manages component services
  - Provides the notions of projects, teams, and process
  - Runs in an OSGI environment
Express-C and Express are fully-enabled collaborative software delivery environments. They differ only on scale and “Enterprise” capabilities. Each Edition is additive – all capabilities in lower versions are available in higher versions.
Integrating Other Repositories with the Jazz Platform

- Import
- Co-Existence
- Connectors
Import

- ClearCase
- CVS
- Subversion
- ClearQuest
- JIRA
- Bugzilla
Problems with Import

- All projects cannot transition to a new repository at the same time
- Customers want to trial a new repository with option of easily coming back
Co-Existence

- ClearCase
- ClearQuest
- CVS
- JIRA
- Bugzilla
- Subversion

Jazz SCM

Jazz Work Items
Problems with Co-Existence

- Only provides loose (primarily GUI) integrations
- Limits the most compelling Team Concert strength/differentiator: rich/seamless lifecycle integration
Connectors
Connector Benefits

- Allows customers to use the best tool/repository for their role (best of both worlds)
- Provides interoperation between Team Concert islands (via ClearCase/ClearQuest/Multisite)
- Allows ClearCase/ClearQuest customers to trial Team Concert with the full Team Concert benefits but without isolating trial users from the rest of the organization
Architecture: ClearCase Today
Architecture: Addition of Jazz Repository

ClearCase VOB Server

UCM  Base

Multisite

ClearCase VOB Server

CCRC/Web View

Dynamic View

Snapshot View

Dynamic View

Dynamic View

Connector

Connector

Jazz Repository

Jazz Repository

CC Workspace

Workspace

CC Workspace

Workspace
Support for Rational Desktop Products

IBM Rational Team Concert

Rational Software Architect
Rational Systems Developer
Rational Software Modeler
Rational Application Developer

Jazz Team Server
Project Areas, Teams, Users, Process, Collaboration,
Chat, Presence, Events, Alerts, Dashboard, Project Health, Reports
Extend Team Collaboration via Variety of Clients
Extend Team Collaboration via Variety of Partners

- **Rational Team Concert**
- **WebLayers**: Manage business and architectural policies
- **QSM**: Track project actuals vs plan
- **sourceIQ**: Access to source code quality metrics
- **blackduck**: Manage software licensing risk
- **Legal Counsel**: Manage documents and workflows (MS SharePoint or Lotus Quickr)
- **Mainsoft**: Access project and team health information
- **IBM**: Software Architect, Functional Tester, Application Developer, Business Developer, Lotus MashUps, etc.
- **Web 2.0**: Integrate with Jira CM
- **CAST**: Project Coordinator
- **QA Manager**: Analyst
- **Executive**: Integration with Jira CM
In 2008

**Phase 2: Expanding Collaborative Application Lifecycle Management**

Rational Team Concert

*Innovation Through Collaboration*

Unify by "thinking & working" in unison with real-time project health

Rational Requirements Composer

*Business Expert Collaboration*

Elicit, capture, elaborate, discuss and review requirements

Rational Quality Manager

*Collaborative Business-driven Quality*

Coordinate quality assurance plans, processes and resources

**JAZZ TEAM SERVER**

Open Lifecycle Service Integrations

Best Practice Processes

Dashboards

Search and Query

Notificiation

Collaboration

Security

Rational Team Concert

Rational ClearQuest

Rational ClearCase

Rational Build Forge

Rational Requisite Pro

Rational Asset Manager

Rational Quality Manager

Rational ClearCase

Rational Build Forge

Rational Requisite Pro

Rational Asset Manager

Rational ClearQuest

Rational ClearCase

Rational Build Forge

Rational Requisite Pro

Rational Asset Manager

Rational ClearQuest

Rational ClearCase

Rational Build Forge

Rational Requisite Pro

Rational Asset Manager
Introduction to the Jazz Technology Platform

The road ahead
Phase 3 – Jazz offerings beyond 2008

Project Management
Manage global projects and resources

Method Composer
Dynamic process, enactment and management

ClearQuest, Build Forge & Requisite Pro
Native Jazz Implementations

ClearCase
Native for UCM users

Enterprise Reporting
Gain insight based on real-time and historical trend information

Best Practice Processes

Project Management

Dashboards
Search and Query
Collaboration
Team awareness
Events notification
Security

Method Composer

Jazz TEAM SERVER

Open Lifecycle Service Integrations

More dynamic integration leveraging Jazz 'Open Services for ALM" services

Rational Integrations
- Rational Software Architect
- Rational Application Developer
- Rational AppScan & Tester portfolio
- Rational enterprise modernization
  including system z and i support
Announcing Open Services for Lifecycle Collaboration

An initiative aimed at simplifying collaboration across the software delivery lifecycle

An open invitation to collaborate on a common integration architecture for software delivery

Barriers to sharing resources and assets across the software lifecycle
- Multiple vendors, open source projects, and in-house tools
- Private vocabularies, formats and stores

A proposed architecture for sharing lifecycle resources
- Described on Jazz.net through illustrative resource definitions, service implementations, and use cases
- Inspired by Internet architecture
  - Loosely coupled integration with “just enough” standardization
  - Common resource formats and services
- Innovation from the Jazz technology platform
What Value will Jazz Enable?

- Transform the way people work together, making software delivery more collaborative and productive
- Empower organizations to "right-size" governance to unique project, sourcing, and team needs
- Improve project predictability by providing transparency of status and trends through automated real-time data gathering and up-to-date reporting
- Through open and extensible architectures, enable teams to assemble their own software delivery platform, relying on preferred vendors and solutions
Transform the way people work together, making software delivery more collaborative and productive

- Bring clients, business professionals, architects, and practitioners together in a single integrated work environment that spans the product life cycle.
- Harness institutional knowledge through the capture and reuse of best practices, work product collaborations, and project histories.
- Make collaboration between stakeholders effortless and second nature by putting it in context and making it cognizant of the work environment.
- Simplify and automate the work required to start a project, form teams, join a team, and provision a work environment.
- Reduce training expenses with automated best practices and in context collaboration that is captured and archived.
Empower organizations to "right-size" governance to unique project, sourcing, and team needs

- Enact your processes so they provide active visible guidance to your teams in the context of their work
- Customize security controls based on roles and work product types to support your organization and sourcing decisions and compliance requirements
- Track work in process, artifact changes, team actions to provide for auditability
Improve project predictability by providing transparency of status and trends through automated real-time data gathering and up-to-date reporting

- True integration of artifacts and roles across the lifecycle
- Understand which work products are driven by which business goals and visa versa
- Perform impact analyses when business goals change or work in process varies
- Focus resources on the most important business goals
- Reports and dashboards to make this information accessible
Through open and extensible architectures, enable teams to assemble their own software delivery platform, relying on preferred vendors and solutions.

- Reduce system administration with a development environment and single integrations repository and data warehouse based on industry standards, that fit into your existing infrastructure.
- Enable integration with other tools and systems through open, standard protocols.
Open Commercial Development

- Delivering greater openness, transparency & customer participation in the products they depend on for software delivery
- Open transparent process is more than publishing the source code. It’s an open window into development, from requirements and planning through delivery for participation and feedback
  - Download, try out, and provide feedback on betas and incubators
  - Provide input on requirements
  - Access, Create, and update work items
  - Suggest source code changes for bugs or enhancements
  - Access milestone and component iteration plans and schedules
  - Access the development wiki
  - Participate in discussions on the development community newsgroups
Our Goals with Open Commercial Development

- Deliver higher quality products that more closely meet the needs of our customers
- Provide higher degree of responsiveness to customer needs and questions
- Provide new features, through betas and incubators, as soon as they are ready
- Create a healthy ecosystem of third party extensions and products
- Showcase a live demonstration of the latest technology being used by the Jazz development team to build Jazz and Jazz-based products
- Ultimately, to have happier customers
Summary

- **Jazz** is a vision and technology platform

- **Rational Team Concert**
  - Is the first Jazz based product
  - Source control, build system, work items

- Other Jazz-based products will follow
Additional resources

- Participate in the Jazz community and the open beta of Rational Team Concert and [http://jazz.net](http://jazz.net)