Jazz & Rational Team Concert:
Development Solution for Collaboration, Automation and Extensibility
Agenda

The Project Reality

Jazz & Rational Team Concert Introduction
- Jazz Conception
- Source Code Management
- Software Change Management
- Build Management

Using IBM Solution to support project needs

Benefits of using the solution
The Project Reality
The Project Reality

People should collaborate

Reliable and up-to-date reporting

Traceability across the project is key for success

Unified Repository (SCM, WAM, BM)
The Project Reality

Different tools
  - Rational Application Developer / Eclipse
  - WebSphere Integration Developer
  - MS Visual Studio
  - ...

Different Teams
  - Customer team -> Requirements
  - Different Vendors -> Responsible for the Pieces
  - Customer teams -> Acceptance Testing & Change Requests

Different Processes

Different KPIs
The Project Reality

Costs
- Rework (teams are not synchronized)
- Bug fixing
- Administration

Deadlines
- Lack of synchronization among teams leads to time complications
- Time spent on communication affects the deadlines

Risks
- Reliability of project reporting?
- Changes impacting the projects?

Quality
- Unpredictable quality
Jazz Platform
Why Are We Doing Jazz?

The Development Landscape has shifted!

Late 90’s: Focus on **Point Tools**
Who can build the best Java IDE, the best C IDE, the best Web Tool,…

When we built **Eclipse**: Focus on **One Developer**
Seamless integration across a set of tools to improve the productivity of one developer

Today, we must focus on the **Team** and its **Collaboration**
- Geographically Distributed
- Accelerated Delivery Demands
- Agility with Predictability
- Innovation and Repeatability
- Increased Need for Transparency

Seamless integration across **All the Phases of the Software Lifecycle** to improve the **Productivity of the Entire Team**.
Jazz: Forces in Motion!

- Eclipse Experience
- Agile Practices
- ALM
- “Web 2.0”
Focus on Application Lifecycle Management

**Traceability** of relationships between artifacts

“Managing dependencies between high-priority change requests and ongoing application development efforts “sometimes seems like it isn't humanly possible.” (US healthcare company)

**Automation** of high level processes

“We had a consulting company define a methodology for us. We still have it on a shelf somewhere. A process needs to live in the tools we use if it's ever going to be followed.” (Transportation company)

“Automating our build-deploy-test processes would probably save each of our developers about an hour a day.” (Financial services firm)

**Reporting** provides visibility

“We do progress reporting the same way we've been doing it for 40 years. It's all manual: weekly status meetings, progress reports, demonstrations. We’d love to get test results from nightly builds posted somewhere instead of having to run people down to ask them whether things are working yet.” (Bank)
Focus on Application Lifecycle Management

- continuous testing
- continuous integration
- consume your own output
- end game
- always have a client
- component centric
- dynamic teams
- live betas
- milestones first
- API first
- adaptive planning
- retrospectives
- sign off
- reduce stress
- drive with open eyes
- transparency
- show progress
- update
- learn
- attract to latest
- enable
- validate

common Agile practices
common Open Source practices
scaling-up practices
Collaboration Obviously Means Complexity

Until now, building an integrated platform required a cross tools understanding:
Each integrated tool had to know about each other tool.
Jazz as an Integration Middleware

- With the Jazz platform, the tools communicate only with the platform:
  - By listening to normalized/standardized events from the platform
  - By sending normalized/standardized events to the platform

Team Collaboration Services
Jazz as an Integration Middleware

Jazz Collaboration Server
(Process Enactment)

Jazz Team Server

Jazz Repository

- Workflow
- Persistence
- Search and Query
- Collaboration
- Team awareness
- Versioning
- Security
- Events Notification
Rational Team Concert
Introducing IBM Rational Team Concert

Enables “real-time, in-context” collaboration for distributed project teams – making software development more automated, transparent and predictive

Integrates Source Control, Work Item, Reporting and Build capabilities which “think and work in unison”

Provides real-time Project Health information and Transparency of status through automated data gathering

Supports out-of-the-box and custom defined Processes which break down functional & organizational barriers

Allows choice of client tools and extends the value of ClearQuest & ClearCase in enterprise deployments
IBM Rational Team Concert

Team Concert provides an integrated Collaborative development environment
Using a single repository
Integrated Version Control, Work Item and a Build System
IBM Rational Team Concert

**Import**
- One way data migration
- Bugzilla repository
- Jazz Repository
- CVS or SVN repository

**Bridge**
- **SVN Bridge** links work items with change sets.
- **Mylin repository connector** (incubator)

**Connectors**
- Two way data synchronization
  - ClearQuest
  - Jazz Repository
  - ClearCase (Base and UCM)
Rolling Team Concert out to End Users

- **Eclipse Clients**
  - Jazz Client Extensions
  - Eclipse Platform

- **Web Clients**
  - Web 2.0

- **MS.net Clients**
  - Visual Studio

- **IBM Desktop Clients**
  - Rational Application Developer
  - Rational Software Architect
  - Lotus Mashup
  - WebSphere Integration Dev.
  - Eclipse Platform

- **Business Partner Clients**
  - Eclipse Platform
  - Visual Studio
  - Lotus QuickR
  - Microsoft SharePoint

- **Rational Team Concert**
  - Team awareness
  - Versioning
  - Security
  - Events Notification
  - Collaboration
  - Search and Query

- **Jazz Team Server**
  - Jazz Repository

- **Tooling Extensions**
  - Jazz Client Extensions
  - Web 2.0

- **Tooling Connectors**
  - Visual Studio
  - Lotus QuickR
  - Microsoft SharePoint
Applying Jazz & Team Concert
Benefit one: Unifying Source Code Mng.

One Single Repository for all Products’ Releases

Unified Source Code Sharing Principles

- Store different products or products component releases
- Track source control changes
- Get auditable repository of source code related to software change management process
Benefit two: Unifying Work Item Mng.

One Single Repository for all Auditable Work Items

Project Management Basis

- Address the process needs (planning, tracking, audit & reporting)
- Support key processes (Request Management, Change Management, Defect Management)
- Relate process elements (changes, defects etc.) to concrete release
- Leverage reliable reporting (tasks are basis for project management)
Benefit three: Unifying Build Management

One Single Repository for all Builds and Releases

Automating the Build Process across many technologies

- Build different projects compilations and get the relationships automatically
- Track builds and releases to changes, requests, defects and source control
Benefit four: Unifying Collaboration

Solve Issues Immediately with Right Person Across Projects
Document Your Communication in a Relevant Way
Summarizing Jazz Benefits and Vision
Outlining the Vision

**Rational Team Concert**
*Innovation Through Collaboration*
- Unify by “thinking & working” in unison with real-time project health

**Requirements Composer**
*Business Expert Collaboration*
- Elicit, capture, elaborate, discuss and review requirements

**Quality Manager**
*Collaborative Business-driven Quality*
- Coordinate quality assurance plans, processes and resources

---

**Best Practice Processes**

**JAZZ TEAM SERVER**

**Rational ClearQuest**

**Rational ClearCase**

**Rational Build Forge**

**Rational Requisite Pro**

**Rational Asset Manager**

---

© 2009 IBM Corporation
Jazz „Operational Benefits“ Summary

✓ Team Environment Unification
   Unifying different tools and teams by providing common team platform

✓ Real Time Collaboration
   Synchronizing teams by effective collaboration

✓ Project Traceability
   Brings reliable reporting based on traceability (in context)
   Collects the traceability automatically -> reduction of administration effort!!!

✓ Quality Management
   Enables to systematically plan and control the quality in context of project needs

✓ Very Effective Learning Curve
   Simplifies adoption of new technologies by using the same concepts
Jazz „Strategic Benefits“ Summary

✓ Getting Visibility into the Projects -> Risk Reduction
  Unifying different tools and teams helps to relate important informations -> traceability
  Setting up a reliable reporting based on real project data (in context)

✓ Rising Team Effectivity -> Costs and Time Savings
  Synchronizing teams by effective collaboration
  People can collaborate on data stored in a single repository or unified through Jazz
  Everybody has the right information in a right time

✓ Quality Improvements
  Key project decisions as well as daily routine work is based on right informations
  Quality problems are identified early
  Fixing defects is not invoking new defects
Meet the Jazz Community at http://jazz.net
Questions
Jazz & Rational Team Concert: Development Solution for Collaboration, automation and extensibility