How Jazz Team Server makes global software development a joy
IBM’s view of software development environment

Fredrik Ferm
ECAT Strategy Team Leader

The business landscape facing IT & business leaders
The need to innovate with less

Enable Innovation & Change

The changing global regulatory and business environment requires security, privacy and ongoing audit capabilities

Rising Cost of Operations

Application development & support budgets are shrinking at nearly 10% CAGR dropping from 50% to about 35% of total IT labor spend

Inability to Innovate

Bottom Line: Enable Business Flexibility...
Today’s reality: An intense focus on business outcomes

- Align IT investments with rapidly evolving business priorities
- Manage value and mitigate risks by improving project management
- Control costs & improve global operational efficiencies
- Ensure security and compliance in a changing global environment

Integrate value in organizationally diverse environments

Collaboration Challenges with Distributed teams

- How well are we currently progressing against our milestones?
- Why are some teams not following our agreed upon process?
- How are our fellow teams in other geos organized? Who is working on what?
- Why and when was this code change made?
- What builds include the new feature from the Beijing lab?
- The build was broken by a team in the US, how can our team repair w/o waiting

Distributed software development teams across the hall or spread around the world face similar challenges.
What is Agile?

- An iterative and incremental (evolutionary) approach performed in a highly collaborative manner with **just the right amount of ceremony** to produce high quality software in a cost effective and timely manner which meets the changing needs of its stakeholders.

- **Core principles**
  - “Fits just right” process
  - Continuous testing and validation
  - Consistent team collaboration
  - Rapid response to change
  - Ongoing customer involvement
  - Frequent delivery of working software

Challenges with Agile in the Mainstream

- **Geographical distribution**
  - Co-located → Global

- **Application complexity**
  - Simple, single platform ← Complex, multi-platform

- **Organization distribution**
  - In-house → Third party

- **Team size**
  - Under 10 developers ← 100’s of developers

- **Degree of Governance**
  - Informal → Formal

- **Compliance requirement**
  - Low risk ← Critical, Audited

- **Entrenched process, people, and policy**
  - Minimal ← Significant
IBM Rational Software Delivery Platform

Solutions to help customers achieve greater value and performance from their investments in delivering software

- Enterprise Modernization and Transformation
- Organizational Governance
- Skill Development and Community
- Implementation Services

Tools collaboration generally means exponential complexity!
Tools collaboration based on middleware services

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

Envisioning a platform that can transform software delivery

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

- robust, extensible and scaleable
- globally distributed, fluid & dynamic
- community-based & 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

Dynamic integration of people, process and projects across the lifecycle
The Jazz approach

Open Services for Lifecycle Collaboration

An initiative aimed at simplifying collaboration across the software delivery lifecycle

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
Jazz.net: Delivering greater openness and customer participation in the products they depend on for software delivery

- IBM is opening up the Rational Software Delivery Platform for greater ease of consumption, extensibility and integration to meet the unique usage needs of our customers
- IBM is providing transparent, collaborative customer participation in the development of new Rational technologies through an open commercial community

Introducing the first wave of new Jazz offerings
Jazz is not a product (it’s a platform)
Interoperability

- Jazz has overlapping capabilities to current Rational technology
- Interoperability is major design point
- Coexistence of "traditional" and "agile" teams
  - Connector – Data synchronization
  - Users do not have to care about other side

Unify distributed teams in mid-sized or enterprise settings
## Rational Team Concert

### Iteration Planning
- Integrated iteration planning and execution
- Task estimation linked to key milestones
- Out of the box agile process templates

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

### SCM
- Integrated stream management with flow relationships
- Component level baselines
- Server-based sandboxes
- Identifies component in streams and available baselines
- ClearCase connector

### Work Items
- Defects, enhancements and conversations
- Query results view and share queries with team or member
- Support for approvals and discussions
- ClearQuest connector
- Query editor interface

### 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

### Build
- Team advisor for defining / refining “rules” and enabling continuous improvement
- Process enactment and enforcement
- In-context collaboration shows team members and status of their work

### Build
- Work item and change set traceability
- Local or remote build servers
- Supports ant and command line tools
- Integration with build forge
- Build definitions for team and private builds

## Rational Quality Manager

### Quality Management
- Control the planning and ongoing status of the test project
- Responsible for effective execution of test and resources

### Test Management
- Develops automated tests
- 100s users
- Deep tool/programming language knowledge
- Talks technology language
- 100s users
- Web based UI
- No Java experience
- Knows application domain
- Talks business language

### Test Planning
- Specify test type
- Project requirements
- Platforms
- Team roles
- Test cases
- Quality exit criteria
- Schedules

### Test Construction
- Build test scripts
- Create manual tests
- Specify test environment

### Test Lab Setup
- Identify build location
- Specify lab request
- Pick test tools
- Select cases to run
- Deploy software

### Test Execution
- Execute UI test
- Execute
- Report real time progress
- Create log files

### Test Analysis
- Examine test results
- Recommend actions
- Code scanning
- Error analysis
- Run time analysis
IBM Rational Requirements Composer

Requirements Management and Definition

- **Requirements Definition**
  - Elicit, capture, review and discuss requirements using a variety of techniques and notations

- **Rich text capture**
  - Organize the web of information into documents
  - Create and link requirements across documents and external sources

- **Business processes**
  - Sketch Processes, Tasks and Decision points, and link to Use cases, Business rules in BPMN notation

- **UI sketches and storyboards**
  - Capture Sketches and Storyboards, and update over project iterations
  - Validate user experience requirements

- **Use Cases**
  - Create informative diagrams
  - Elaborate with rich document descriptions

- **Glossaries**
  - Promote consistency, clarity and correctness

- **Collaboration**
  - Attach comments to elements and run comment threads for conversational context

Thank You