Making Jazz
Collaboration, Community and Design
in Open Commercial Software Development

Kimberley Peter
Jazz & Rational Team Concert UI Design
kpeter@ca.ibm.com
Topics

- Introduction: Overview of Jazz
- History & Motivation: Life before Jazz
- User Interface Overview: Principles in practice
- Community & Design: Inputs and influences
- Growth & Diversity: Now and ahead
Introduction

Theme > Definitions > Team and Contexts

- The essence of the talk is about …
  - how feedback has played an essential role in shaping the design of our software
  - how feedback comes from multiple sources throughout the cycle, from both the team developing it as well as a larger community using it
  - and that Jazz has a variety of input channels that make this flow of feedback possible
Introduction

Theme > Definitions > Team and Contexts

Open Commercial Software Development …
- Is about creating the software in the open, on Jazz.net
- Is distinct from Open Source in that the code is not shared, but the product of the coding is

Jazz …
- Is a technology platform for building integrated tools
- UI includes Dashboards (web), Team Central (Eclipse), and Administration (Web and Eclipse)

Rational Team Concert …
- Is the first product to be built with Jazz
- Provides an integrated set of tools to support the development team including a task management system (Work Items), planning for individual, team, and projects (Plans), integrated Source Control, Build, and Reports, with traceability across artifacts

Eclipse …
- Is an open source technology platform with an integrated development environment (IDE)
Introduction

Theme > Definitions > Team and Contexts

- ~100 developers

Location:
- Bangalore
- Yamato
- Beaverton
- Toronto
- Ottawa
- Lexington
- Raleigh
- Zurich
- St. Nazaire
Introduction

Theme > Definitions > Team and Contexts

- Eclipse Client
- Team Central
- Work Items

IBM Software Group | Rational
Making Jazz | Collaboration, Community and Design in Open Commercial Software Development
© 2009 IBM Corporation
Introduction

Theme > Definitions > Team and Contexts

- Web UI (available on Jazz.net)
  - Work Items
Introduction

Theme > Definitions > Team and Contexts

- Jazz.net
  Community Site for products, community, and partners
  - Projects
  - Dashboards
  - Wiki
  - Mailing lists
  - Downloads
  - Forums
  - Library
  - Community
  - Blog
  - Tweets

Jazz.net
Community Site for products, community, and partners
- Projects
- Dashboards
- Wiki
- Mailing lists
- Downloads
- Forums
- Library
- Community
- Blog
- Tweets
Topics

- Introduction: Overview of Jazz
- History & Motivation: Life before Jazz
- User Interface Overview: Principles in practice
- Community & Design: Inputs and influences
- Growth & Diversity: Now and ahead
History & Motivation

Social software > Eclipse > The next natural step

- **Collaborative development environments (CDE)**
  - “a virtual place where stakeholders of a project – often separated by time or space – can meet, share, brainstorm, discuss, reason about, negotiate, record, and generally labor together to carry out some task, most often to create some useful artifact and its supporting objects.”

- **Jazz research project: Team collaboration for the Eclipse developer**
  Research by John Patterson, Li-Te Cheng, Susanne Hupfer, Steve Ross of IBM Research in Cambridge, from ~2003 to 2005
  - Supporting the core team, peripheral awareness, conversation in context, and “concert” or activity awareness around artifacts

- **Eclipse open source technology platform**
  Started as an integrated development environment (IDE) for Java development by Erich Gamma and John Wiegand, among others, in 2001 …
History & Motivation

Social software > Eclipse > The next natural step

- The Eclipse inspiration
  - Rich set of development tools for the developer
  - Thriving community of contributors with a growing set of projects that extend breadth
  - **Eclipse Way process** enables continuous rhythm and feedback for teams
    - Agile: iterative, reflective, adaptive, incremental, inclusive of feedback
    - Open Source: community involvement, new & noteworthy
    - Scaling-up practices: component-centric, dynamic teams
History & Motivation

Social software > Eclipse > The next natural step

- The Eclipse Way process: It’s all about being continuous ...
  - Continuous iterative and adaptive planning
  - Continuous design/refactoring
  - Continuous integration/testing
  - Continuous delivering/demos
  - Continuous feedback
  - Continuous learning
  - Continuous health

- Many effective teams work like this, but there are pain points ...
History & Motivation

Social software > Eclipse > The next natural step

- Pain points
  - Static plans
  - Lack of early feedback on plans
  - Lack of awareness of team, build, project
  - Lack of process integration
  - Lack of traceability across artifacts

- The next natural step was an integrated toolset ...
History & Motivation

Social software > Eclipse > The next natural step

- Take the goodness of the Eclipse Way Process but address the pain points
- Build the Jazz team collaboration platform grounded on the following principles:
  - Collaboration (team first / awareness)
  - Process awareness
  - Transparency
Topics

- Introduction: Overview of Jazz
- History & Motivation: Life before Jazz
- User Interface Overview: Principles in practice
- Community & Design: Inputs and influences
- Growth & Diversity: Now and ahead
User Interface Overview

Collaboration > Process > Transparency

- Event feeds per team – scoped
- Person cards
- Team load awareness
User Interface Overview

Collaboration > Process > Transparency

- Team Central feeds tailored for individual
  - Inbox
  - Team events color-coded
User Interface Overview

Collaboration > Process > Transparency

- messages@me
- Online presence
- Linked artifacts and people
User Interface Overview

Collaboration > Process > Transparency

- Shared dashboards
  - Individual
  - Team
  - Project
User Interface Overview

Collaboration > Process > Transparency

- Process-neutral
- Team-tailored
- Enacted process, in context

<table>
<thead>
<tr>
<th>Operation</th>
<th>Precondition/Follow-up action</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate Team Invitation</td>
<td>Create initial work items</td>
<td>This follow-up action creates a set of work items for the tasks you need to perform to complete the process of joining the team.</td>
</tr>
<tr>
<td>Accept Team Invitation</td>
<td>Show work items</td>
<td>When you accept the invitation to join the team, a query is run to display your assigned work items.</td>
</tr>
<tr>
<td>Work Item Save Operation</td>
<td>Required properties</td>
<td></td>
</tr>
<tr>
<td>Source Control Deliver</td>
<td>Descriptive change sets</td>
<td></td>
</tr>
</tbody>
</table>

Agile Process
- The process is a mash-up of best practices of several existing agile processes.

Cloudburst Sample Process
- The process used for the Cloudburst examples and tutorials. Projects are created with this process will have a number of teams created with sample users assigned to those teams.

Eclipse Way Process
- The process originally developed by the Eclipse Development Team. The Eclipse Way is an agile, iteration-based process with a focus on consistent, on-time delivery of quality software.

OpenUp Process
- The OpenUp process is one of the processes created by the eclipse process framework project (EPF). It is included here as one example of how to integrate existing processes. For more details see http://www.eclipse.org/epf/

Scrum
- A process template for Scrum. Scrum is a popular approach to manage projects in an agile way.

Simple Team Process
- A simple process to get you started quickly. Team members are entitled to perform modifications of any kind, outsiders are not.

Work Items
- Required Properties completed successfully.
- Work Items must be filed against a specific category and a summary must be provided so that the team can quickly evaluate who is responsible for the issue.
- Why did this happen?
User Interface Overview

Collaboration > Process > Transparency

- Self-hosting transparency from the individual to team to project
  - Planning with dynamic plans
  - Development with automatic linking, build results/reports
  - Endgame with code reviews and verification
  - Process with team structure and team roles
User Interface Overview

Collaboration > Process > Transparency

- Dynamic plans
User Interface Overview

Collaboration > Process > Transparency

- Linked artifacts
  - Source control
  - Change Sets
  - Included Builds
User Interface Overview

Collaboration > Process > Transparency

- Endgame transparency
User Interface Overview
Collaboration > Process > Transparency

- Transparency for community on Jazz.net
  
  Join, watch and participate

  - Access project **Dashboards**, people **Profiles** and artifacts like **Plans** and **Reports**
  - Read the **Team Wiki**
  - Subscribe to, create, and comment in **Work Items**
  - Follow and respond to **Blogs**
  - Participate in **Forums**
Topics

- Introduction: Overview of Jazz
- History & Motivation: Life before Jazz
- User Interface Overview: Principles in practice
- Community & Design: Inputs and influences
- Growth & Diversity: Now and ahead
Community & Design

Anecdotes > Channels > Codifying results

- Collaboration mechanisms are a goldmine for designers
  - Scenarios – see what it will be
  - Self-hosting – see first hand
  - Community – see what others see

- Gathering feedback is both passive and active
  - Work Items and Forums (p)
  - Queries, Feeds (a)
  - Blogs (p, a)

- Story time …
Community & Design

Anecdotes > Channels > Codifying results

- Plan editor in Agile Planning
- Affordances in the Web UI
Community & Design

Anecdotes > Channels > Codifying results

Plan editor in Agile Planning

- Design phases
  - Three main phase, including pre-self-hosting to early-self-hosting to "open" on Jazz.net

- Feedback channels
  - Self-hosting, including UX and Design team evaluation
  - Community, including internal IBM channels, and external community via forums suggesting improvements

- Validation
  - Work items
  - Usability session within customer beta program
Community & Design

Anecdotes > Channels > Codifying results

Plan editor – v0: Word processor model with simple hierarchy
Community & Design
Anecdotes > Channels > Codifying results

Plan editor – v0: Word processor model with simple hierarchy – feedback

- Simple text list, no filtering or grouping
- Lack of distinction between parent and non-parent items
- No distinction between plan ideas and actual planned items
Community & Design

Anecdotes > Channels > Codifying results

Plan editor – v1: Hierarchy augmented with bullet semantics
Community & Design

Anecdotes > Channels > Codifying results

Plan editor – v1: Hierarchy augmented with bullet semantics
Plan editor – v1: Hierarchy augmented with bullet semantics – feedback

- Bullets look like radio buttons and do double duty to expand/collapse item
- Unstable location of ID on the right
- Using ID link to open item unexpected
- People did not expect clicking on the object icon would show item type menu
- Layout of three part “status” drop down control not understood as single entity
- All or nothing approach to showing description and details is heavyweight
Community & Design

Anecdotes > Channels > Codifying results

Plan editor – v2: Table layout, with richer rendering
Community & Design

Anecdotes > Channels > Codifying results

Plan editor – v2: Table layout, with richer rendering – feedback

- Self-hosting, customer pilot, and usability sessions revealed need for improved Scrum support

- The three big complaints from the community were:
  - Lack of Scrum terminology, e.g., Sprint Backlogs not Iteration Plans
  - Lack of a product backlog in 1.0
  - Lack of adequate Scrum reports

- Significant feedback provided by a certified scrum master at IBM
Community & Design

Anecdotes > Channels > Codifying results

Plan editor – v3: Scrum support added – backlog view
Community & Design

Anecdotes > Channels > Codifying results

Plan editor – v3: Scrum support added – taskboard view

![Taskboard Image]
Community & Design
Anecdotes > Channels > Codifying results

Plan editor – v3: Scrum support added – work breakdown view
Community & Design

Anecdotes > Channels > Codifying results

Plan editor – v3: Feedback from Beta customers

- Validation – The evolution continues!

“The scrum function is great to see, but we need guidance on how to get started and on understanding the different plan types and when to use them.” – from customer beta program
Community & Design

Anecdotes > Channels > Codifying results

Affordances in the Web UI

- Design phases
  - Two-three

- Feedback channels
  - Community, through Work Items
    - Drop down menu for “Create” button in Web UI not apparent

- Validation
  - Team usage
  - Work Item immediate response

- Outcome: New design for split buttons & line-level interactions pattern
Community & Design

Anecdotes > Channels > Codifying results

- Current

- Revised
  + split button
Community & Design

Anecdotes > Channels > Codifying results

- Work Items
- Forums
- Blogs
- Tweets!
- Usability sessions
- Customer Programs
- Surveys
- Interviews
- Observation
Community & Design
Anecdotes > Channels > Codifying results

- UI checklists and pattern guides on Jazz.net
  - Initially for Jazz and Rational Team Concert teams
  - Expanding coverage to support other products built or integrating with Jazz
  - Available to business partners
Topics

- Introduction: Overview of Jazz
- History & Motivation: Life before Jazz
- User Interface Overview: Principles in practice
- Community & Design: Inputs and influences
- Growth & Diversity: Now and ahead
Growth & Diversity

- Integration across the product development lifecycle: 
  Requirements + Development + Testing

- Increased scale and diversity of teams and artifacts: 
  Managing relevance with tagging, annotating, and targeted sharing

- Integrating Jazz with Lotus Connections and other social software tools
Summary

- Acknowledgement of need for collaborative development environments
- Success & pain points in developing Eclipse inspired an integrated toolset based on the principles of:
  - Collaboration (team first / awareness)
  - Process awareness
  - Transparency
- Self-hosting and developing in the open on Jazz.net provides a wealth of feedback to shape the experience
- Challenges ahead as the platform diversifies
Questions
kpeter@ca.ibm.com