The IBM Rational Jazz Strategy

for Collaborative Application Lifecycle Management

Liz Barnett
EZ Insight, Inc.
July 2008

Collaboration and the Development Landscape

Software and systems are essential to nearly every business. Global organizations require software solutions that span geographic boundaries, cultures, and business units and that comply with corporate governance mandates. Such global businesses must collaborate effectively and leverage existing investments, or lose out to competitors. This collaboration nearly always involves business partners and customers in defining—and sometimes even creating—software products. To compete in this climate, organizations must rethink the ways that they build and deliver software solutions.

In today's economy, global organizations pose two new sets of challenges to software delivery. First, team members must have easy access to each other and to the diverse set of assets comprising their software projects in order to collaborate and achieve their goals. This is complicated by the fact that members of a project team, including developers, managers, business partners, and customers, as well as the project assets, can be located anywhere in the world.

Second, while team members must collaborate on a shared set of deliverables, each individual has his own role and responsibilities and thus his own view of the shared environment. Developers work with testers during the development lifecycle, yet their lens is one of code and defects, while a tester's is one of functionality and coverage. On the same project, managers must continuously monitor status, balance the use of resources, and manage risk. Their customers pose requirements and feedback, and track progress according to agreed-upon project plans. In order for each individual to be productive, the environment must understand the context in which he works.

Fortunately, the global technology infrastructure—specifically the use of the Internet and Web-based services—provides proven and scalable solutions for the challenges of distribution and collaboration. New technology-driven ways of working, such as those leveraging Web 2.0 concepts, are familiar and have transformed users' expectations. For example, worldwide procurement via eBay or Amazon is an everyday occurrence. Consumers understand that when they use a URL to request a service or an application, they do not need to know where the requested service or data resides; the infrastructure provides "late binding" and resolves the location. The power and value of collaboration among those building software is also well understood. Open source initiatives, such as those producing Linux and Firefox, have become household names. Business and lay people

appreciate the benefits of tapping into knowledge around the world to achieve breakthrough solutions.

So why not implement these same approaches when developing and delivering business software solutions? Project teams are essentially a microcosm of the global business world: team members and the assets they produce are geographically distributed, and unique skills and resources can reside at any location. Successful application lifecycle management (ALM) environments must address this complexity by connecting all members and components of a project and then providing each individual with a role-based view. By coupling these two concepts—***global access*** to a common project environment and an ***individualized context*** in which to work—teams have the potential to change the ways that they develop and deliver their products.

People produce high quality software. Tools and processes are enablers. With the Jazz project, IBM Rational is providing integrated tools and processes that help people working on distributed teams become more effective in producing software solutions. It calls this type of environment *Collaborative ALM*, and defines it as a "new paradigm for transforming the software lifecycle so that it is more collaborative, transparent, and productive."[1] The Jazz project is Rational's vehicle for achieving collaborative ALM.

The Jazz Project encapsulates three distinct but related initiatives that will enable collaborative ALM:

1. Rational's ***vision*** of an open integration architecture that can achieve truly collaborative, productive, and transparent software delivery in a global environment.
2. Jazz ***technology,*** a set of core services that enable ALM tools to add value and interact with each other.
3. A ***community*** in which Rational developers, customers, and business partners can collaborate on developing Jazz services and associated tools.

[1] Source: John Wiegand, IBM Software Group, Rational, Distinguished Engineer and Rational Chief Architect.

Jazz provides tools and automation that fit with the way successful teams work.[2] The intention with Jazz was not to invent completely new concepts for lifecycle management, but rather to exploit current technologies and provide a rich collaborative development solution for global organizations. Just as Eclipse provides rich integration and productivity for individuals, Jazz will do so for teams.

This paper discusses the broad vision for the Jazz project, the Jazz technology and associated products from Rational and its partners, and the community that will participate in all of these initiatives. It includes experiences shared by early users of Rational Jazz tools as well as those of partners building complementary Jazz-based products.

[2] In this paper, the term "Jazz" is used synonymously with the "Jazz project," referring to the overall project. Specific initiatives, such as the Jazz Team Server software or the Jazz.net community, will be clearly noted.

The Jazz Vision

When it launched the Jazz project in January 2005, Rational set out to create an entirely new collaborative environment for building software, leveraging the architecture of the Internet and its expertise in producing ALM and development tools. Some developers working on Jazz came from the Eclipse project, bringing techniques from highly distributed, collaborative, open source development. Others drew upon knowledge gained on other Rational product teams, where they had built tools capable of supporting enterprise-wide requirements. Jazz developers also drew upon their experiences as users of previous generations of ALM tools, on projects ranging from small research initiatives to global product development. It is this mix of leading-edge thinking and scalable systems architecture that has led to innovations within the Jazz project.

As Rational takes this leap in providing next-generation technology, it is essential that it also provides an evolutionary path for current customers. This is no small feat, given the breadth of existing ALM tools, development processes, and customers' investments. Rational has committed that in addition to delivering future products on the Jazz technology it will upgrade most existing Rational tools to the new environment. Thus, throughout this process, customers will be able to take advantage of Jazz capabilities and benefits on an incremental basis, as appropriate for their projects.

Achieving the Jazz vision is a tremendous undertaking and one that is burdened by the history of the ALM marketplace. Previous ALM solutions which approached integration from the perspective of the tools or processes across the lifecycle were only moderately successful. These cobbled-together ALM environments were useful, but they were brittle—they relied upon point-to-point product integrations and required updates with every interface change or product upgrade. In addition, few ALM solutions provided seamless front-end or back-end environments. Users lacked a single interface for all development activities. Relationships among data created and used by various tools were not managed cohesively. Analytics rarely drew upon data from multiple repositories.

The Jazz architecture and tools differ from other ALM solutions in that they approach development from the point of view of the *people* who are delivering software, rather than that of the tools or processes. Team members can work in an environment that is cognizant of the activities and issues that they typically confront during the lifecycle. Following a specific event, such as executing a test or fixing a defect, an individual on a team needs to understand what happened, who is

or should be involved, and what needs to be done in response to the event. The tools and processes support his ability to collaborate with other team members, regardless of their locations, in order to respond to the event.

To achieve the goals of the Jazz project, Rational has been developing three sets of technology-based deliverables: the Jazz Foundation, new Jazz tools that leverage this foundation, and the Jazz.net collaborative community. The Jazz Foundation comprises the Jazz Team Server plus additional building blocks for implementing and integrating Jazz-based products. This multi-year initiative is still in its early stages; technology releases in 2008 and beyond will include additional new products, extensions to existing Rational ALM tools, and enhancements to the Jazz Foundation (see Figure 1).

Jazz Project Timeline

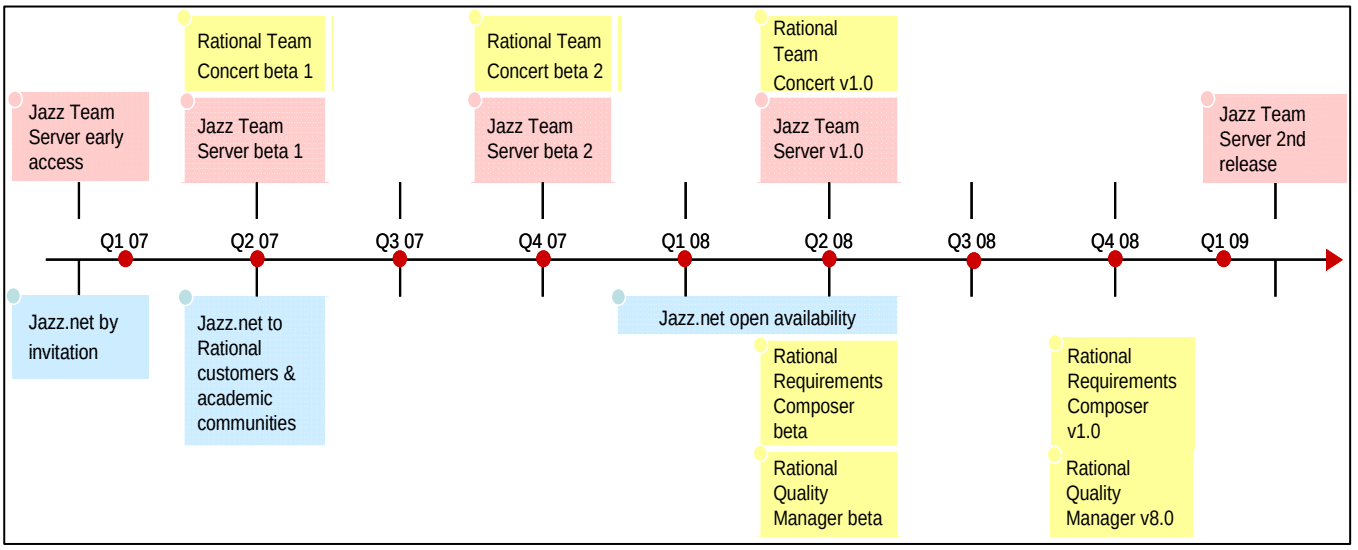

Figure 1: IBM Rational Jazz Project Timeline
Source: EZ Insight, Inc.

Key Benefits of the Jazz Project

Successful collaborative ALM requires innovation in collaboration, process, and transparency. These concepts that underlie three of the key benefits of the Jazz project:

1. **Collaboration in Context.**
   Diverse and geographically distributed teams work towards a single set of goals, as defined by a project. However, each participant on a project has his own role, his own responsibilities, and therefore his own context for working. "Collaboration in context" refers to a team member's ability to work with others on shared deliverables in the most productive way for the individual task at hand.

   In the Jazz environment, team members are aware of who else is on the team, what each team member is working on, and the relationship of others' work to their own. Jazz facilitates team member communication using instant messaging, persistent discussions, and other Internet-based tools. But the Jazz technology goes a step further and treats all interactions as valued project assets, storing them as associations with the specific artifact (e.g., defect, test case) that was the source of the interaction. Since the repository workspaces can be saved periodically, a remote team member can reproduce the exact problem that someone else has encountered. This is especially important for teams working across time zones—they do not require real-time collaboration to resolve issues, and thus individuals can be effective working asynchronously.

   This approach to collaboration is unique to the Jazz offering and is perhaps its greatest differentiator. In a global environment, teams need to be able to draw upon resources regardless of their location. The greater the information available to a team member, the lesser the need for real-time interactions, reporting, and other forms of overhead. Individual productivity improves, as does the quality of team collaboration.

2. **"Just Enough" Process.**
   No single development or deployment process is appropriate for any organization. Companies require a portfolio of processes to support the myriad projects that they run. An environment that facilitates and automates the use of best practices without impeding individual or team productivity is a necessary component of collaborative ALM.

The Jazz architecture supports the ability to define and implement a range of processes, from lightweight approaches used by small iterative teams to highly-structured, rule-driven processes necessary for corporate or regulatory compliance. In order to add value, all types of processes must be integrated seamlessly into the team's tool environment.

This is particularly important for Agile teams, a key target for the new Jazz products. Some organizations have viewed (and may still view) these projects as being unruly and unmanageable. Nothing could be further from the truth. Agile teams that follow a specific process like XP or Scrum are highly disciplined. Adherence to practices such as continuous integration and build management is critical to a team's ability to achieve short iterations and high quality deliverables.

At the same time, projects that are subject to strict compliance requirements must follow—and prove that they follow—a clearly-defined and pre-approved process. ALM tools must support the team in achieving compliance and provide visibility into all phases of the project's progress.

Process support in Jazz-based products is seamless and unobtrusive. Jazz services provide the essential components of a development workflow, such as event notification, awareness of and connections to team members, validation of team-defined rules, and automation of basic tasks such as status tracking and reporting. The resulting process can be implemented in Jazz-based tools using a very lightweight or a very restrictive workflow, depending on the team and organizational requirements.

One benefit of this passive process support is the ease of onboarding of new team members. New projects can be launched quickly. New staff can join established projects and immediately participate in the team's work-in-process. Role-based permissions, access to previous team collaborations and best practices (saved as permanent project assets), and pre-defined definitions of project metrics and workflows all run behind the scenes and contribute to a simplified learning curve.

3. **Global Transparency.**
   Transparency is a fundamental component of any collaborative ALM solution and a requirement to satisfy governance initiatives. Global teams must have the same degree of visibility at all locations. Effective transparency has two distinct aspects: the *availability of accurate information* and the achievement of *passive governance*.

   Traditional attempts at transparency within the development environment achieved the opposite results. Compliance was commonly achieved by intrusive project or organizational audits, imposing additional work and distractions for team members. To produce required metrics, team members used supplemental tools or tedious manual processes, and so much of the information given to management lacked accuracy. The combination of lagging productivity and erroneous information led to many project failures.

   Jazz addresses both of these deficiencies with a tool-based environment that unobtrusively generates and communicates accurate project metrics based on the organization's unique guidelines. On distributed teams, developers, managers, customers, and other project participants all require real-time knowledge of a project's status, issues, and risks. Jazz accomplishes this by providing automated user-defined processes, generated project metrics, and global access to dashboards at the individual, team, and portfolio levels.

In addition to these benefits, Jazz offers a future infrastructure for the many Rational ALM tools in use at large organizations. Jazz is designed so that shared services and data can be accessed via standard open interfaces and Web protocols. Initially, Jazz-based products provide bi-directional connectors for integration with existing tools.

Jazz Technology

Jazz technology encompasses a rich set of services, a shared repository for all development assets, and the interfaces with which ALM tools can add value and interact with each other. In essence, the Jazz architecture provides a new approach to tool integration and a means to accomplish collaborative ALM.

The Jazz Foundation and Jazz Team Server

The Jazz Foundation implements the underlying architecture for a new generation of Rational tools. The foundation includes the Jazz Team Server technology, as well as a set of frameworks that aid in the construction Jazz-based products. It provides an evolutionary path for existing ALM tools, so that customers can adopt Jazz capabilities while retaining investments in development assets. The Jazz Team Server is the core infrastructure for deploying and integrating tools. It includes support for in-context collaboration, real-time project health, event notification, process enactment and enforcement, global search and query, security, role-based access, automated traceability, and a distributed repository for all development assets (see Figure 2). These server-based features can be leveraged by a range of client platforms, including Rational and third-party tools.

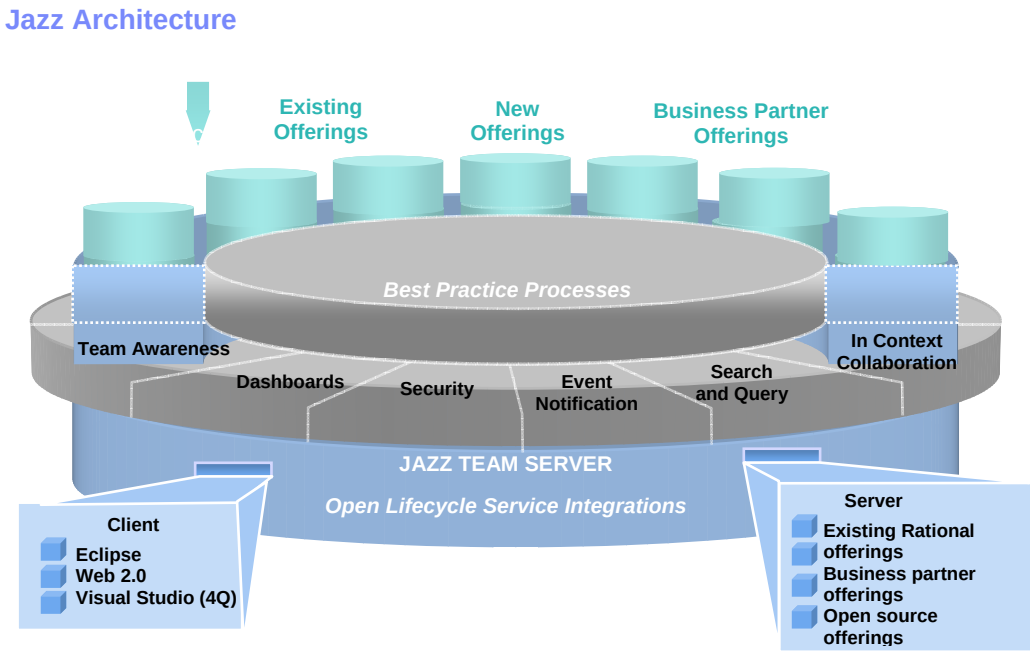

Figure 2: The IBM Rational Jazz Architecture
Source: IBM Rational

But haven't we seen this architectural vision before? IBM's AD/Cycle and prior Rational tool integration initiatives also articulated these goals. Why will the Jazz Team Server make this vision possible? The answer boils down to two factors: open design and scalable global infrastructure. First, the architecture has been designed from the ground up as an integrated distributed environment that acknowledges, but is not limited by, a suite of heterogeneous investments. The new breed of tools is designed intentionally with a collaborative, user-centric model and open interfaces for tool integration. Second, the Jazz architecture leverages the Internet—a model that is proven to reach global communities and to scale for the largest organizations.

Eclipse for Individuals, Jazz for Teams

There is a strong relationship between the Jazz and Eclipse environments, yet the two are distinct and can run independently. Eclipse addresses the challenge of **individual** productivity. Prior to Eclipse, ALM products had limited integration and thus limited ability to enhance productivity. Fragile solutions required constant maintenance to support evolving products. Further, no single interface could access the breadth of ALM capabilities—developers had to switch from IDEs to issue and bug tracking to build and release environments as they progressed through the lifecycle. The Eclipse desktop and associated tools have significantly improved an individual developer's ability to build high quality code and extend his own environment.

Jazz addresses the challenge of making **teams** more productive. This means not only integrating tools across the entire lifecycle, but also providing collaboration each step of the way. Team-centric capabilities, such as team and individual awareness, persistent collaboration, automated data collection, and passive process support, differentiate this approach and complement the benefits provided by Eclipse.

Jazz servers support a variety of clients, including a Web browser, command line interface, and IDE interface. The current release supports the Eclipse IDE; Jazz will also support Microsoft Visual Studio clients later this year. This distinction is important because it is often assumed that the Jazz technology is a superset of Eclipse code. Use of Jazz technology does not require use of Eclipse and the reverse is also true.

Rational Jazz Products

The Jazz Foundation provides a set of services and interfaces on which collaborative ALM tools can be deployed. Rational has developed new team-centric tools on this foundation, and has begun upgrading some of the existing Rational ALM tools.

Anyone using the Jazz Foundation to build a new tool or extend an existing tool must determine the most appropriate type(s) of user access. For some products, as is the case in Rational's forthcoming quality management tool, a browser interface may be all that is necessary. Other products may be deployed best as Eclipse plug-ins or offer a choice of interfaces. For example, in Rational's new collaborative ALM environment, developers may choose to maintain work items directly from their Eclipse IDE, while their managers may view status of those work items through a Web browser.

Rational has announced three new products based on Jazz technology. Each is discussed below.

Rational Team Concert (RTC)

RTC is the first family of collaborative ALM tools delivered using the Jazz Team Server. The goal with RTC was to think first about how people worked together on a distributed software development team, and then to design a tool to support those work styles and roles. Initially, RTC targets small- and medium-sized distributed teams, particularly those using Agile processes for development or team management, however RTC can be used with any development process.

RTC provides three major capabilities: integrated software configuration management (SCM), work item management, and build management. [3] Teams can deliver software solutions using RTC's features, and they can also use Jazz-based connectors to access enterprise-scale capabilities and assets in Rational ClearCase, ClearQuest, and BuildForge. Future versions of these Rational products will leverage Jazz technology and be incorporated into the enterprise version of RTC.

[3] For detailed descriptions of RTC's SCM, work item management, and build management capabilities, go to http://www-306.ibm.com/software/awdtools/rtc/.

RTC uses the Jazz Team Server for developers' and managers' activities (see Figure 3). For example, as a work item progresses through the lifecycle, RTC maintains current status data such as who is responsible, what triggered the event (e.g., failed test case, new requirement), which assets in the repository are related, and the team's workflow rules. All of this information can be fed into a management dashboard or other reporting capability, visible to all project participants.

Rational Team Concert Capabilities

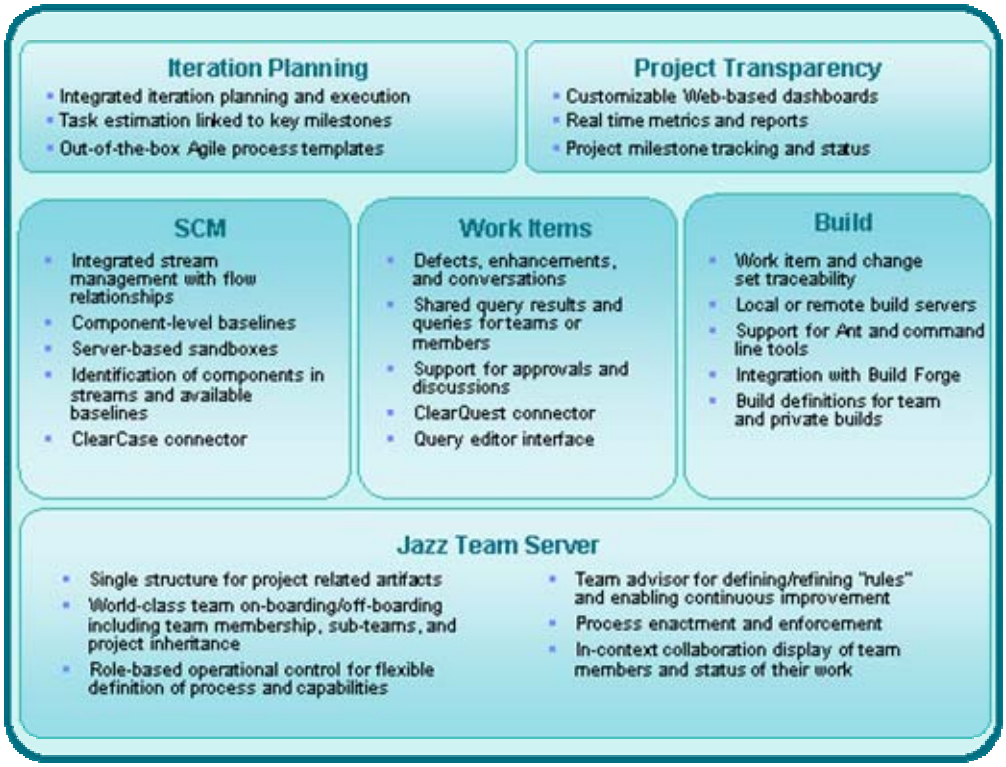

Figure 3: Overview of Rational Team Concert (RTC) capabilities
Source: IBM Rational

RTC is a family of products; the different editions essentially reflect team size:

- RTC Express-C, running on Derby and Tomcat, supports teams of 10 users and is geared towards entry-level teams. The Express-C edition can be downloaded at no charge from Jazz.net for up to three users.
- RTC Express has the same functionality as Express-C, but also runs on DB2, Oracle, and WebSphere Application Server (WAS) and can support mid-sized and departmental teams of up to 50 users.
- RTC Standard, a superset of RTC Express targeted to corporate teams, supports up to 250 users and includes connectors for ClearCase, ClearQuest, BuildForge, and Subversion as well as built-in project dashboards.
- RTC Enterprise (not yet available) will provide enterprise-scale security, provisioning, and governance, as well as versions of the ClearCase, ClearQuest, and BuildForge tools running on Jazz.

RTC implements the essential components of a development workflow and allows the team or organization to determine the appropriate type of process to meet its requirements. Initially, RTC includes process templates for Scrum, OpenUP, and the Eclipse Way.

Within a project, business users and customers can access the repository and project dashboards directly. This can be both a blessing and a curse. On the positive side, there may be little or no need for custom management reports—customers can see the project work items and dashboards, understand the status of the project, and directly add their own issues or change requests. However, this also means that customers can closely monitor a team's progress and potentially scrutinize decisions made within the team. Teams learn to take advantage of this visibility, building relationships with their customers and demonstrating their quality work through frequent releases. As customers become accustomed to releases that are weeks rather than months apart, they become more tolerant and flexible. They learn that with short iterations, issues that are not resolved in a particular iteration may be delivered in the next one, only a few short weeks away.

RTC has been available in beta since early 2007, and version 1.0 general availability as of June 2008. Over the next 18 months, Rational will provide additional milestone releases, including enterprise support and further integration with existing Rational tools. In addition, RTC is available at no charge to academic institutions and select open source projects.

Rational Quality Manager (RQM)
RQM, currently available in beta, is a new Web-based test management portal built to the Jazz architecture. RQM will help teams drive quality from a business perspective and better align development, testing, and delivery teams' activities. General availability of this new product is expected in the second half of 2008.

Focused on addressing the needs of business analysts and QA professionals, RQM employs a test plan-centric view of testing assets. It provides the ability to use different perspectives for accessing and viewing testing assets, based on the user's role. For example, managers can review timelines and status reports for testing cycles, while business analysts can concentrate on test coverage for business requirements. In addition, test planning assets can be related to specific testing executables stored in Rational functional and load testing tools.

In addition to the new test management offering, Rational announced a plug-in to RQM for test lab management. Users will be able to inventory all of their test lab assets (e.g., CPUs, middleware, databases, applications), derive utilization rates for those assets, schedule a specific configuration to use for a testing session, and then physically deploy that configuration.

Rational Requirements Composer (RRC)
In the area of requirements definition and management, Rational is offering new visual tools and is providing integration among requirements, SCM, testing, and other ALM tools. RRC was also announced in open beta in June 2008, with general availability expected in the second half of 2008.

RRC provides graphical modeling, storyboarding, and sketching tools for eliciting and defining requirements, and uses a wiki-like platform for enhanced collaboration among stakeholders and development teams. RRC can generate the textual requirements assets that are stored in Rational RequisitePro.

Users can also relate this rich requirements content through RequisitePro to other lifecycle assets, including test cases created in RQM, and thus be able to determine test coverage and traceability for requirements. Over time, Rational intends to evolve its requirements management portfolio to provide a unified set of capabilities based on the Jazz architecture.

In addition to the announced products, Rational is working on a number of incubator projects—technology in a pre-beta state that may or may not result in commercial software products. The first incubators that are available for download address code coverage and static analysis functionality.

The Jazz Community

Jazz represents not only new technology but also a new way for Rational to develop commercial products. Through the Jazz.net community, customers and business partners have the opportunity to observe Jazz technology being developed, to learn how it works, and to validate that it scales by witnessing its use by Rational's large and distributed teams. This transparent model of commercial software development will directly benefit customers and Rational itself.

Jazz.net

Customers can use Jazz.net to interact directly with Jazz development teams. They can raise questions, submit defects and requests for features, and offer suggestions and even solutions. Rational calls this model "open commercial development"—meaning that it is developing Jazz-based products in an open and transparent environment, but that its intention is to create, sell, and support commercial products. This is not the same as an open source initiative, where both the source code and resulting software are freely available. While third parties can create Jazz extensions and products, only Rational developers can commit changes to the Jazz Foundation. At this time, Rational has determined that Jazz technology will not be open-sourced, but will be available at no cost to qualified open source projects.

For customers, the Jazz.net community may provide a faster path to Rational-supported capabilities, as well as a lower cost model for the customizations that they submit. At the same time, Rational's developers gain frequent and early feedback from a broad set of users, which makes it possible for them to deliver continuous improvements of the technology and tools.

There is certainly some risk for Rational in exposing its product development to the outside world—including its competitors. This is a very different type of relationship between a vendor and its customers. Customers have access to all information: open issues, priorities of the team, how Rational is responding to their specific requests or issues, how long resolution takes, etc. But by doing so, Rational is actually improving its credibility with current and future customers, and demonstrating its ability to deliver on its promises. The gamble is already paying off.

RTC users and Rational partners building complimentary products are making use of the Jazz.net environment. Early beta customers are thrilled that they *really do* have direct access to the Jazz development teams. Primarily, these customers have used Jazz.net to submit work items (e.g., defects, enhancement requests) and then to track the progress of the work items' resolutions. One user noted the *excellent*

responsiveness of the Jazz developers—as fast as three hours for some questions, and no more than two or three days for more complex issues. Another beta team found the Jazz.net wikis to be great source of information. In fact, even the Rational post-sales support teams use information from the wikis to assist their customers.

Open Services for Lifecycle Collaboration

Along with delivering new technology, Rational is continuing its work with the software tools industry to provide open and flexible solutions for heterogeneous customer environments. Rational has begun an initiative called "Open Services for Lifecycle Collaboration," aimed at enabling tool interoperability across the software delivery lifecycle. Instead of publishing an API toolkit and asking partners to build to that API, the Open Services initiative is focusing on defining common formats and protocols for requirements, test cases, and other ALM assets, along with a practical approach for interoperability using Web architecture standards. The hope is that vendors—both partners and competitors—will be more likely to support this approach as it is not reliant on any single vendor's representation of an "ALM server" or repository.

This initiative, which was announced in June 2008, is still in its early stages. To elicit discussion, the Rational team has published a reference architecture, sample designs for assets in the quality management and requirements management domains, and sample reference code (available on Jazz.net).

Rational Business Partners Adopt Jazz

Rational has created a community of business partners around Jazz, including ISVs and services providers, called "Ensemble." These partners are taking the lead in investing in, expanding, and promoting Jazz technology. Through the Ensemble program, they can take advantage of Rational-provided technical assistance and sales and marketing services, and collaborate with Jazz developers and each other via Jazz.net.

As part of IBM's "Ready for IBM Rational Software" partner program, IBM business partners have begun building extensions and complementary products around RTC, RRC, and RQM. Eleven partners demonstrated early work at the IBM Rational Software Developer's Conference in June 2008; these and other products will become available beginning in the third quarter of 2008.

Customers using the new Rational tools will be able to extend their ALM environments with partner products in a number of different areas, such as:

- Compliance and policy management, by managing open source and third-party software and automating policy definition and management (Black Duck Software, WebLayers).
- Application portfolio and quality management, by providing metrics to support management and compliance reporting (CAST, SourceIQ).
- Project management, by supplementing work item management with estimation and project control analyses (QSM).
- Collaboration, by enhancing team-based workflows and document management capabilities through integration to enterprise collaboration systems (Mainsoft).
- Change management, by providing integration and synchronization between work item management and third-party tools (CM-Logic).
- Requirements definition, by complementing graphical requirements definition capabilities with application simulation and visualization services and requirements validation (iRise, Ravenflow).
- Virtual test lab management, by extending RQM test asset management to include virtual test assets (Surgient, VMLogix).

Over time, partners will be developing Jazz-based solutions for the entire lifecycle, including IDEs and project management tools. Any tool that creates artifacts in a team-based environment can take advantage of the versioning and change management capabilities within Jazz. In the future, Rational may also license the Jazz Foundation on an OEM basis.

Getting Started With Jazz

Rational's aggressive plans for the Jazz project will result in multiple product releases in 2008, as well as beta and incubator projects for new products and extensions to the Jazz Foundation. Existing Rational tool users will have the opportunity to take advantage of Jazz capabilities on an incremental basis and ultimately run on the same architecture and repository. This is clearly a huge step for the Rational product base, as Jazz is an architecture that will take its users far into the next generation of their software initiatives.

"Jumpstart" teams have been playing the frontline role in supporting early customers' implementations, supplementing the Jazz development teams and post-sales technical support. The Jumpstart teams provide on-site training and initial customizations, getting initial customers up and running and helping them absorb all the capabilities in the new products.

Early RTC Experiences

Since some customers have had access to early versions of RTC since 2007, they have already begun to build a body of experiences and best practices. In many organizations, RTC adoption grew from the bottom up: developers were drawn to the environment and then began selling its benefits to their managers. In particular, developers like the tool's ease-of-use, its integration with Eclipse, and its ability to support their teams' lightweight processes. More than one user noted the "helpfulness" of the built-in processes; none complained that their productivity was constrained.

Managers approach RTC somewhat differently. Even if they are not totally sold on the use of Agile processes, many have been willing to launch RTC-based Agile pilot projects. It is their confidence that RTC will be a Rational-supported product, and the knowledge that it will be integrated with existing Rational tools and development processes, that sets the pilots in motion. Further, the fact that many Jazz developers came from the Eclipse project gives managers confidence in the strength of the underlying technology.

Among the many features in RTC, beta teams highlighted several stand-outs:

- *Quick start-up*. Training for the core work item management capabilities is minimal. Some teams worked with Jumpstart staff to conduct a brief three hour training class; others just "figured it out" without formal training. RTC's pre-defined role definitions and

workflows made it easy to add new team members, without administrative overhead. (Rational call this "day one productivity.")

- *Seamless integration with the Eclipse IDE*. With RTC, teams work from a single client environment and have access to all their development and collaboration tools—developers no longer need to leave the Eclipse IDE to access other ALM functions. For example, one customer noted that in the past, teams had to use three tools and switch from the Eclipse IDE to manage code in CVS and issues in Jira. This alone has been a big incentive for them to use the new product. Further, teams had little difficulty learning to use the new tool, since it leveraged the Eclipse interface.
- *Little need for reporting overhead*. By providing customers with a Web browser interface to the work item system and team dashboards, teams don't need to spend time creating additional status reports. Also, providing 24x7 direct access to team progress led to goodwill and far fewer surprises for customers.
- *Simpler installation and management*. Beta teams noted that maintaining project assets in the Jazz repository cuts management overhead and is far simpler than maintaining separate SCM and defect tracking environments.

Users also recommended a number of best practices that they have developed in their early use of RTC:

- *Address SCM training up front*. Teams using any type of Agile or iterative process rely on seamless SCM and build integration to achieve short iterations. Traditional development teams may use code management systems, but they do not always have experience with sophisticated SCM environments. To truly take advantage of RTC, teams must learn SCM concepts, such as using workspaces and integrated stream management. That said, users found the learning curve for RTC's SCM capabilities to be quite short. One customer required only one additional day of training for SCM, on top of a three hour general class for all developers, and then relied on some on-the-job training.
- *Create custom work item types for customer communication*. By providing customers with specific artifacts for entering requirements or defects, the teams could manage and report on these items without exposing

their internal work item issues. This was a big improvement for some teams that had relied historically on email for customer input. It was also easier for teams to add new work item types than to modify the RTC-provided repository. Changing the artifacts or processes without sufficient training (e.g., understanding search capabilities, workflow rules) could be problematic.

- *Adopt the Eclipse Way process*. RTC includes the Eclipse Way process as one of its pre-defined workflows. This approach appeals to Agile teams and simplifies daily activities by allowing team members to work off lists of tasks. Behind the scenes, the tool provides traceability and a record of what has been done, without intruding on the developer's tasks.

The biggest challenge cited by beta customers is keeping up with all of RTC's milestone releases. The incentive is obvious—milestone releases contain enhancement requests submitted by these very users—yet teams are not always thrilled at the notion of stopping work to upgrade their tools and repository. As RTC and the other Jazz-based products evolve to general availability, the frequency of these milestone releases will obviously slow.

Achieving Strategic Benefits

With the Jazz project, Rational has developed breakthrough technology and is poised to set the standard for collaborative ALM. Those companies that adopt Jazz-based tools will be able to transform their software development organizations. Most notably, teams will be able to:

- Deploy global teams of developers, managers, business users, and customers to produce leading-edge software;
- Improve team productivity and quality through collaboration and seamless tool integration;
- Provide real-time visibility into accurate project metrics for all project participants, regardless of their location;
- Employ Agile and iterative development and management practices to address ever-changing business requirements and to shorten delivery cycles;
- Achieve compliance with corporate governance and regulatory requirements without impeding team productivity;

- Leverage development assets created and processes used by existing Rational ALM tools; and
- Interact with Rational developers, other customers, business partners, academics, and anyone using Jazz.net to enhance the Jazz environment and participate in creating industry standards.

Jazz offers a great opportunity to initiate Agile projects and provide distributed teams with a productive collaborative environment. Small- and medium-sized teams can get started with the new Rational Jazz tools immediately. Given the current business climate of "deliver more with less much sooner," organizations have no choice but to rethink the ways that they deliver software solutions. The Jazz project will help teams achieve that goal.

About the Author

Liz Barnett is the Principal Analyst at EZ Insight, Inc., an analyst and consulting firm focused on global software development issues and technologies, founded in 2005. In 2006, she also launched the Agile Journal, an online publication providing in-depth analyses and case studies for Agile developers and managers, and was Editor in Chief for its first two years. Previously, Liz spent 10 years as a Vice President and Research Analyst at Forrester Research, joining Forrester as a result of its acquisition of Giga Information Group. Liz has held management positions at Accenture, PepsiCo, Atelier Research, and New Science Associates (subsequently acquired by Gartner Inc.). Liz earned her B.S. in operations research and industrial engineering at Cornell University.

This report is the result of research underwritten by IBM and developed by EZ Insight, Inc. The analyses and findings are derived from EZ Insight's independent views.