



The IBM Rational Jazz cultural revolution

11 February 2009

Laurent Lachal

A decorative graphic in the bottom-right corner of the slide, composed of a grid of overlapping squares in various shades of gray and beige, creating a stepped, staircase-like effect.

www.ovum.com



Table of Contents.....1
The IBM Rational Jazz cultural revolution.....2
 Executive summary2
 Jazz connects people, not just tools.....3
 Agile design.....6



The IBM Rational Jazz cultural revolution

Executive summary

In a nutshell

The Jazz application lifecycle management (ALM) platform is project, team and process centric. It redefines not just the way developers relate to one another, but also the way enterprises relate to their software development teams. This cultural revolution is necessary but painful. It will require careful handling from IBM. IBM Rational Jazz products, however, go quite a long way towards minimising pain and disruption.

For more information on Jazz, see the Ovum reports entitled 'IBM Rational Jazz opens up' and 'IBM Rational Jazz: the story so far'. There are also two Ovum reports on native Jazz products respectively entitled 'IBM Rational Team Concert: Jazz springs forth' and 'Test and QA automation update: IBM Rational'.

Key messages

Jazz can adapt to your needs; can you adapt to Jazz?

ALM platforms redefine the way developers interact with both business users and IT operations people. Jazz, with its emphasis on Agile development, pushes this redefinition even further.

IBM claims that Jazz helps organisations get on a glide path towards the brand-new ALM world without requiring disruptive or impractical changes. Jazz is indeed well designed, but the issue is cultural, not technological. Many organisations are neither able nor willing to adapt to these changes. This resistance could prove the single most important obstacle to Jazz's uptake, and requires IBM Rational to focus on market education. Jazz needs to be carefully positioned for developers, IT managers and business executives to understand the benefits and constraints it delivers.

Jazz/Rational Team Concert makes agile development mainstream

The Jazz engineering team started with a bias for agile development methodologies, which is reflected in the way Jazz structures and manages projects, teams and processes. This design underpins Rational Team Concert, the first native Jazz product and integration hub for legacy IBM Rational products. It enables organisations to adopt select agile practices at the scale and timing that makes sense for them.



Jazz/Rational Team Concert encourages a hybrid approach to development

Besides endeavouring to get its customer base to experiment with agile development, IBM Rational is rightfully seeking a third way to mix Agile and more traditional approaches. Considering IBM's strength in model-driven development, Jazz/Rational Team Concert could also underpin the convergence between Agile and model-driven development.

Jazz connects people, not just tools

Jazz targets a variety of people

ALM platforms such as Jazz support not only increasingly geographically dispersed teams (with the practical and cultural issues that ensue), but also diverse teams at two levels.

- Software professionals: with business analysts, architects, developers and testers having to interact with one another (and with IT operations people). ALM platforms focus on the coordinated set of handoffs between team members rather than on throwing assets 'over the wall'.
- Enterprise departments: with software professionals having to interact with business stakeholders and other subject matter experts (such as lawyers and finance directors), both within the organisation and outside of it.

Jazz targets development professionals

Focus on team self-confidence

When it relates Jazz to developers, IBM Rational uses various metaphors (such as orchestra, football team and, at the Rational Software Developer conference in June 2008, circus acrobats) to link teamwork trust and reliance with individual virtuosity. It explains that developers, architects, analysts, project managers and QA professionals are gifted individuals that simply need a bit more coordination in an environment in which their individual talent and creativity can surface.

At the 2008 Rational Software Developer conference (RSDC08) the company focused on boosting development professionals' self-confidence and sense of self-importance. More implicitly than explicitly, the message was that by enabling more transparency, Jazz makes it easier for team members to get the recognition they deserve.

Tongue firmly in cheek, IBM Rational hailed the conference participants as 'R-Heroes', part of the 'League of Extraordinary Software Developers' that should be 'ready to save the day' to the tune of the 'Star Wars' movies soundtrack, complete with cartoon characters spattered around the conference centre. The keyword was 'league'. IBM was careful not to reinforce the old macho cowboy 'I-can-do-it-better-than-anybody-else-on-my-own' culture by pointing out that success



depends on teams rather than individuals. In addition to talking about 'gifted individuals', IBM Rational should concentrate more efforts on explaining how Jazz helps 'average Joe' team members.

Constraint versus empowerment balance

As part of its effort to prevent Jazz from being perceived by development teams as a straitjacket, IBM positions the platform as not just constraining (via process automation) but also empowering them at two levels.

- Their work: Jazz is positioned as the platform that enables teams to be more productive and deliver better software on time in the face of changing requirements.
- Their working environment: Jazz generates value for (not just out of) team members by relieving the pain of setting up a development environment, joining a project and keeping abreast of the context of the task they are carrying out.

Gifted professional versus blue collar worker balance

Danny Sabbah, general manager of IBM's Rational division, likes to compare Jazz to an enterprise resource planning (ERP)-like environment for development. At RSDC08 he asserted that there is too much 'excitement and unpredictability' around software delivery, which needs to become as 'boring' and predictable as 'accounts receivable'. This is not exactly consistent with the 'R-Heroes' message.

To combine the two perspectives, IBM Rational focuses on the themes of collaboration, automation and insight as well as the notion of 'right-sized governance' that seeks to balance the need for developers to make their own decisions, and for IT and business managers as well as IT operations people to keep an eye on developers' activities and intervene when necessary. 'Right-sized governance' does two significant things.

- It emphasises the need for painless automation, whereby the underlying Jazz platform gathers project information automatically and unobtrusively, without developers having to make extra efforts. (This is only half true: although the system keeps up with developers' activities, it also needs their active participation in the form of up-to-date data, not just passive connection, to deliver the right information for both collaboration and reporting purposes).
- It advocates flexibility of governance processes (avoiding the need to define everything upfront) and constraints (lower for early-phase experimentation projects, and higher for more mature projects).



Jazz targets business stakeholders

Focus on business issues

IBM does not just try to sell Jazz to practitioners, but also to business executives. To the latter, it emphasises Jazz's ability not just to lower costs (through process automation), but to involve them in software management (via right-sized governance) as well as development (with a focus on 'outside in' – for example, user-centric: Agile development).

During RSDC08, IBM Rational endeavoured to frame its efforts in as much a business as an IT context, based on the notions of business 'outcome' and 'value' and the need to make development more accountable to the business. Ultimately, the company asserted, even those development projects that deliver on time and under budget mean nothing without the right business outcome, measured with such criteria as dollars, customer retention and market share. We could not agree more.

RSDC08 itself hosted a business executive summit (BES) that attracted 200 people, up from 109 in 2007 and, out of the 300-plus sessions across 14 tracks, 130 were customers presenting sessions. During the last BES session, IBM Rational general manager Danny Sabbah emphasised the need to shift the IT/business relationship from 'adversarial' to an 'honest collaborative conversation'. Jazz is the platform that underpins this new type of conversation and implements one of the key agile manifesto principles: 'business people and developers must work together daily throughout the project'.

At RSDC08, IBM Rational's business-centric messages were too high level: too many 'conversations' and 'outcomes'; not enough connection with the actual Jazz platform. In over 30 sessions, a special evening event and 'Jazz Café' at which customers could download and try out Jazz products, RSDC08 seemed at times to be all about Jazz. At a time when IBM is doing its utmost to relate its business audience to topics such as service-oriented architecture (SOA), Web 2.0 or software-as-a-service (SaaS), IBM Rational could have done a better job of defining how Jazz pragmatically relates to these topics.

A difficult shift from 'command & control' to 'sense & respond'

IBM Rational should also distinguish more clearly between the relationships between developers and their manager(s) and business users, some of which may manage their manager(s).

The poor quality of both types of relationship explains why so many IT projects fail. ALM platforms do not guarantee quality, but make it easier for these relationships to develop. They also enable a shift in the nature of the relationships from a command-and-control environment, in which people are limited to the short horizon of their immediate responsibilities, to a sense-and-respond environment, in which people have visibility and are empowered to do what they need to do within the constraints of the processes they are involved in. This cultural shift is rather



difficult to implement and needs time to build up. IBM should be more upfront with its efforts and ability to support this shift, rather than simply repeating the traditional ALM mantra about the need to break down (organisational, process and tool) silos.

Jazz targets IT operations

ALM bridges IT development and operations

ALM platforms should not only deal with software development and delivery, but also software running in production environments. For example, they need to ensure that the code delivered behaves itself in production, based on defined policies, or that the change requests generated by IT operations people are given the right priority. This is an area that Agile development does not tackle at all.

Jazz needs to make more effort

The bridge between IT development and operation is where the Jazz story intersects the Tivoli system management story. The first product at this intersection is IBM Test Lab Manager, which automates the provisioning of test lab environments. However, IBM Rational does not make enough of an effort to explain how the two stories relate to one another. The key problem is IT silos. The infrastructure folks would not take initiatives originating from ALM seriously. IBM Rational has not figured out how to get the IT silos talking to each other and sell crossover tooling. Jazz will take several years to mature, and IBM can wait a little bit more to deliver a clearer set of messages around Rational-Tivoli integration.

The company is working on linking IT development and operation, not just from an ALM platform point of view, but also from a development methodology standpoint, with efforts such as the Enterprise Unified Process. This expands the Rational Unified Process (RUP) methodology to include production issues.

Agile design

Jumping on a new bandwagon

Agile development focus

Jazz enables IBM to piggyback on the current market interest in Agile software development methodologies and position itself against other vendors who have pushed support for these methodologies, such as Microsoft and Borland. Rational Team Concert, the first Jazz offering designed for the core development team, comes packaged with various Agile methodologies such as the Eclipse Way (used by the Jazz team to build Jazz itself), Scrum and OpenUp (an open source subset of IBM's Rational Unified Process methodology).



Development methodology neutrality

There may be an Agile Manifesto that lists a series of agreed principles, but when it comes to implementation, there are many specific Agile development methods to choose from. The design of Jazz and therefore, of Rational Team Concert, is method neutral. It can support any type of these methods as well as non-Agile ones. Teams are free to ignore the Agile templates provided with Rational Team Concert and start from scratch with their own approach to development. We expect that, over time, many other methodologies will be made available for Rational Team Concert by both IBM and third parties.

Project, team and process centric

Project- and team-centric repository

Agile methodologies require developers to collaborate with one another. Rational Team Concert enforces this collaboration via the repository component of the Jazz server (known as Jazz Team Server). This repository consists of project areas split into team areas. Project areas relate to deliverables, schedules and resources (such as code artefacts and roles). Team areas and sub-areas (for example, component X team area with coding and testing sub-areas) have their own artefacts, roles and processes. A user can be a member of more than one team. Each project artefact belongs to one team area but, over time, may move from team area to team area.

Project areas and team areas can be managed using either the Rational Team Concert Eclipse-based IDE or web user interface. Despite the focus on projects and teams that enforces collaboration between individuals, team members retain their personal space. For example, they have personal dashboards, not just team and project ones.

Process-centric repository with workflow engine for tool-supported process guidance

The Jazz Team Server repository manages any artefacts, but it is organised around work items and is process, not artefact, centric. Projects and teams structure themselves around processes, managed by the Jazz Team Server workflow engine, that organise and control the flow of work within and between team areas. Teams are managing processes, not just deliverables.

IBM Rational describes Jazz as a process-aware platform (all work happens within the context of a process) that supports the notion of tool-supported process guidance, whereby tools understand the process teams have decided to use and help them follow it. Processes kick in when a team member performs an action that triggers a process enactment mechanism. Jazz tool developers need to weave these triggers into their offering, which requires careful design.



Flexible process management

IBM Rational is careful to point out that the Jazz Team Server workflow engine is process neutral, that it is up to the teams to decide which process to enact and which level of rigour they need. For example, Rational Team Concert distinguishes between project-level processes that apply to every team, and cannot be customised to team-level processes that can be changed. It is up to the team to define its own processes, and capture as well as enact as much (or as little) of the process as is practical. Each team area may customise the specification and description inherited from its parent area(s).

Rational Team Concert processes do not just constrain, but help, in the form of 'team advisors' that provide process-specific advice on what to do next if a process rule has been breached (provided that organisations have the time, skills and resources to create these advisors, which have to be coded in Java). Instead of being blocked from completing the task, developers can be permitted to overrule some process violations.

Better process design capabilities to come

Rational Team Concert teams define processes by specifying permissions, preconditions and follow-up action for a variety of operations, such as saving work items, modifying team areas, delivering change sets to streams or starting builds. Follow-up actions can also be configured for events generated when changes are made to artefacts in the repository. Permissions, preconditions and follow-up actions can all be configured differently for each role, team and iteration.

Rational Team Concert includes out-of-the-box workflow templates, which teams use to generate process descriptions and specifications. Templates, descriptions and specifications are standard assets stored in the Jazz Team Server repository and, as such, can be mixed and reused in a way that is tool and language independent. Process descriptions document the specification and deal with the non-formalised aspects of a process. Specifications describe team roles and the rules that apply to them, as well as define which rules can be customised.

While templates help, process authoring in Rational Team Concert remains rather awkward and development centric, reflecting the fact that IBM Rational has built the Jazz Team Server workflow engine from the ground up. Process specifications are expressed in a declarative form (XML file), but kept in a compiled form to boost performance, while primitive actions need to be expressed as Java code. Instead of a nicely graphical process definition tool, Rational Team Concert offers a 'process configuration page' in the project area, team area and process template Eclipse editors. The page allows team members to edit a process without having to dive into the underlying XML presentation. They can also directly edit the XML, on top of writing Java code for 'advisors' and primitive actions.

Integration between Rational Team Concert and Rational Method Composer (RMC), as well as the Eclipse Process Framework (EPF) Composer, is still being developed. RMC 7.2, released in September 2008, features experimental code that allows the



generation of a formal Rational Team Concert process template from within RMC. The idea is to allow bidirectional flow of process definitions across the RMC and Jazz tools. This would enable organisations to create processes organically, from within Jazz tools, where they can later be documented in an RMC process website or to allow process authoring in RMC, which could then be actualised in Jazz tools and workflow.

In addition, process design needs to be improved. For example, missing a deadline does not necessarily trigger an event. Furthermore, the way Rational Team Concert relates team and development lines may be problematic for teams that work on multiple development lines. Rational Team Concert splits projects into development lines, each with its own objectives, deliverables, team, process and schedule. Each team area belongs to exactly one development line, with team members limited to accessing the plans and dashboards that relate to this line.

Structured processes and unstructured collaboration

Native Jazz products such as Rational Team Concert combine:

- structured processes based on process automation/workflow and rule technologies provided by Jazz
- ad hoc people interactions based on collaboration and communication technologies provided by Jazz (RSS feed-based alerting), as well as third-party products (instant messaging via open source Jabber or IBM's own proprietary Lotus Sametime offering).

IBM is weaving other communications and collaboration technologies into the fabric of Jazz, such as wikis, blogs and voice over Internet protocol (VoIP). Instant messaging (IM) is currently the most mature technology. During an IM conversation, developers can refer to development artefacts for other developers to click through and consequently, gain access to the artefacts and their context (for example, code with links to requirements and tests). We expect deeper integration between Jazz and the entire set of Lotus collaboration technologies eventually. We also expect more synchronous communications capabilities such as video conference; after all, one of the key Agile Manifesto principles is: 'the most efficient and effective method of conveying information to and within a development team is face-to-face conversation' (which Agile development prioritises over written documents).

Development intelligence

Communication and control underpin workflow and collaboration

ALM platforms underpin a democratisation of development intelligence, whereby reports and metrics are not just for managers but also for any team members to make their own decisions and adapt to new circumstances (which is key to Agile development). Development intelligence capabilities provide the mechanism for communication and control both within the team and with project managers and



business users; a mechanism that optimises the benefit provided by workflow and collaboration capabilities, in that:

- everyone is informed of team events
- each team member has a personalised workspace
- formal assignment makes responsibilities clear
- each person has his or her own task list
- management can monitor the progress.

Data warehouse and reporting engine

Jazz enables tools such as Rational Team Concert to provide team members and project managers with realtime information via its data warehouse and reporting engine.

The data warehouse complements the Jazz Team Server repository for storing aggregated, historical data. It retains a history of all artefact types, while the repository keeps a history of only a subset. IBM Rational is working at separating the data warehouse from the repository, to enable the data warehouse to live on a separate server and customers to use pre-existing data warehouses if these are available.

Native Jazz tools can feature out-of-the-box snapshots to aggregate and store the data they manage: for example, work items, source control and builds data for Rational Team Concert. The data warehouse is extensible, so third parties can contribute new snapshots that gather data from one or more Jazz repositories or any other sources. IBM Rational is working on a more declarative way of writing snapshots. Currently, these require Java code written against the Jazz application programming interfaces (APIs) and deployed on the server, which is rather unique and restrictive.

The current reporting engine is based on the Eclipse BIRT (Business Intelligence and Reporting Tools) project. IBM Rational is working on support for other reporting engines from, for example, IBM-acquired Cognos, and better access control from the web user interface (Jazz features a web portal for internal and external team contributors to have access to project health information). Currently, there are no access controls on any of the information in the data warehouse. Any report can view any information.

Views and dashboards

Jazz provides native Jazz tools with a variety of views and dashboards. A view consists of relevant practitioner data – namely, the right file versions of select artefacts – to enable practitioners to focus on a particular activity, and a dashboard refers to a 'cockpit' of correlated status indicators. The main views include 'My Work' view (which enables team members to see their assigned work and create a personal work schedule), Team Artefacts view (the central point of access to artefacts in the repository), Team Central view (for team members to see what is



going on in their team) and Jazz Development Dashboard (which measures project success).

Views consist of sections. Jazz provides a basic framework for writing sections, complete with special chart widgets. Users can dock multiple views, minimise them, configure which sections are visible and change the content of each section. Various views and/or sections provide different perspectives on the same subject. Team members may be overwhelmed by the wealth of views and sections on offer.

IBM Rational introduced project level dashboards to the two Express versions of Rational Team Concert in October 2008. Reports, as well as personal, team and custom dashboards, remain limited to the Standard edition, though. For more information on Rational Team Concert please see the Ovum report entitled 'IBM Rational Team Concert: Jazz springs forth'.

Alerts and queries

Team members can add alert and query sections to views to continuously track important metrics over the project or iteration lifecycle. Alerts and queries, like views, can be private or shared with the team.

Users subscribe to changes in specific items in the repository and to more general event types, to keep up with their own as well as other teams. As artefacts in the repository are modified, event notices are automatically sent out to the repository's feed service, accessible from any (Atom 1.0 or RSS 2.0) feed reader. Besides Jazz's own event feeds, users can subscribe to external feeds, such as external tools that generate events that must be monitored. An event log viewer lets them view events or feed items on a timeline.

Jazz enables native tools to provide users with packaged queries. For example, in Rational Team Concert, work item queries enable team members to monitor a set of work items – as a complement to subscribing to work item events – and look up all submitted defects that have been fixed in the iteration to ensure a fix has resolved an individual defect. (If not, the team member will reopen the defect.) Alternatively, developers can create their own.

Tagging and categorisation

Jazz groups artefacts use 'tagging' and 'categorisation' to make it easier for developers to find them. Tagging is a great concept that allows artefacts to reside in more than one structure. Coupled with search capabilities, it enables developers to group things virtually and dynamically. The problem is that as useful as search, tagging and categories are, they limit some native Jazz products. For example, Rational Quality Manager does not maintain a hierarchy of test cases or requirements. To group test cases, you use tagging and categorisation. The absence of a test case hierarchy – used by testers as a way of structuring their thinking – is a deficiency. We believe IBM will add hierarchies in the future.



Pushing the Agile bandwagon forward

Getting Agile into SMBs

The Agile approach to development is a good fit for the small to medium-sized business (SMB) space, although hardly any SMBs have taken it up, partly for lack of integrated collaborative environments such as the one provided by Rational Team Concert. On the other hand, IBM Rational has barely started to tackle the SMB market. For more information, see the Ovum report entitled 'IBM Rational Jazz: in progress'.

Getting Agile into large corporations

The Agile approach to development is increasingly seen as a good fit for large enterprises. A recent IBM survey showed that 88% of organisations were either using or (more likely) evaluating agile methods, including the majority of large enterprises. IBM Rational positions Agile practices as a mainstream process approach that yields practical benefits for any size or scale of development effort. However, the company also points out that while Agile can be used for innovative, leading-edge small development projects, traditional approaches such as Waterfall are still being applied to more large-scale and mature projects.

IBM Rational positions itself as the organisation that helps Agile development scale up to meet the needs of large companies.

- It enables Rational customers to leverage existing investments in tools such as ClearCase and ClearQuest, while supporting satellite teams experimenting with agile practices using Rational Team Concert. Most of IBM Rational's traditional tools are too heavy for agile development and too complex and entrenched to rip and replace. By providing them with a lightweight hub around which agile tools (such as Rational Team Concert and Rational Quality Manager) can thrive, Jazz enables enterprise users to move to Agile development at their own pace. For more information on how Rational Team Concert relates to Rational ClearCase and ClearQuest, see the Ovum report entitled 'IBM Rational Team Concert: Jazz springs forth'.
- It allows organisations to practice a mix of Agile and more traditional methodologies, which can learn from one another. For example, IBM Rational emphasises that the traditional, phase-based approach of its own RUP methodology can become more Agile by blending fast-paced, 'just enough' iterations into each of its development phases. It also pushes traditional approaches forward via Jazz tools, such as Rational Quality Manager, which weave testing throughout the development cycle instead of reserving it for the final weeks of a project. The organisation is less vocal than it should be about how Agile development can benefit from approaches adopted in more traditional methods such as risk-based management.

As Jazz matures and native Jazz tools evolve to replace legacy products, there will be less focus on mixing products while the 'complementary methodology' approach



will develop further. Besides talking about the way Agile relates to other development methodology, the company needs to relate Agile to development process best practices such as Capability Maturity Model Integration (which, admittedly, has hardly been adopted).

Agile and model-driven development convergence

We expect IBM Rational to develop a lightweight, Jazz-based tool for model-driven development, not because Agile does modelling (it doesn't), but because converging the two approaches would help scale Agile development. In return, the objective would be to make modelling more pragmatic by adopting the Agile 'just enough' approach to projects. In the meantime, the native Jazz tool called Rational Requirements Composer features lightweight requirement modelling capabilities.

Ovum does not endorse companies or their products. Ovum operates under an Independence Charter. For full details please see www.ovum.com/about/charter.asp.

For full details of Ovum's citation policy, see www.ovum.com/media/citation.asp.

Whilst every care is taken to ensure the accuracy of the information contained in this material, the facts, estimates and opinions stated are based on information and sources which, while we believe them to be reliable, are not guaranteed. In particular, it should not be relied upon as the sole source of reference in relation to the subject matter. No liability can be accepted by Ovum Europe Limited, its directors or employees for any loss occasioned to any person or entity acting or failing to act as a result of anything contained in or omitted from the content of this material, or our conclusions as stated. The findings are Ovum's current opinions; they are subject to change without notice. Ovum has no obligation to update or amend the research or to let anyone know if our opinions change materially.