IBM

Moderator: Angelique Matheny
November 19, 2009
12:00 pm CT

Coordinator: Thank you for standing by and welcome to today's conference. All participants will be in a listen-only mode. Throughout today's presentation we will have multiple question and answer sessions. In order to ask your question or respond you may press star 1 on your touch tone phone. Today's conference call is being recorded. If you have any objections, please disconnect. I will now turn the call over to your conference host, Ms. Angelique Matheny. Ma'am you may begin.

Angelique Matheny: Thank you. Hello everyone and welcome to this Rational Talks To You Teleconference: Open-source, The Shining Aspects and the Limitations. I'm Angelique Matheny with IBM. Now these calls are really for you. You won't find any slots for these, for this teleconference. And we want this to be interactive. And to get these questions answered by our panelists today.

We'll open up the lines during the Q and A throughout the entire call. So as the Operator mentioned, press star 1 and jump right in. Write those questions down if you think of it. But please we want this to be interactive.

If you want to submit questions later after the conference is over, please email us at askusnow@us.ibm.com. That's askusnow@us.ibm.com. Just put the title of this teleconference in this subject line and I'll make sure it gets to the right person. Joining us today is (Pam Induni), IBM Rational Marketing Manager.

(Pam Induni): Thanks Angelique, I appreciate it and happy to be here. As Angelique said, "I'd also like to welcome you to today's telecom." And let me just take another moment to explain how today's telecom will work. We have actually assembled the Panel's (3S) experts who are here to bring up discussion topics and answer any questions or comments you may have.

Again, as Angelique said, "This call is really for you." So please feel free to be an active participant. We know this is a really relevant and very dynamic topic, so we're looking forward to a very interactive discussion today.

Let me now move into our panel and I'll introduce you to our panel. First, we have Rolf Nelson. And Rolf is a Rational Product Manager. He is responsible for Rational Team Concert, which is as many of you may know, is a Collaborative Development Solution from IBM and this ideal lean ALM.

Also, with us is Jean Louis Vignaud. Jean Louis, is a Senior Manager of Product Management, and he's also focused on our Change and Configuration Management product line. And finally we have (Don Boulia) and (Don) is a Director in WebSphere and he has responsibilities for WebSphere Open Source strategies, Web2.0 Development, and Cloud Computing. So, as you can see we have a really nice mixture of skills on our panel today.

So without further ado let me kick things off here, and I'd like to turn this over to our panel to make a few opening comments. So, Jean Louis let me turn it to you first and I'll tee-it up with this question. Open source has been a real driver of innovation, and so can you briefly comment on IBM's involvement, and the contribution to the development of this community?

Jean Louis Vignaud Yes, sure. Thank you Pam. For almost a decade IBM has been making significant contributions to the open source movement. Let me give you some examples. Ten years ago, IBM formed the Linux Technology Center LTC. The team of open source software development's IBM networks in competition, in cooperation, with the Linux open source development community.

Later on in 2001, IBM initiated the Eclipse consortium, an open source community that focuses on building an open source development platform for building, deploying, and managing software. And as part of Eclipse, IBM is leading several Eclipse projects. More recently, IBM has been leading the open (ethical) news Java Suite on XML Ajax initiative and partners with the (Dojoe Foundation). And now, Today, IBM has more than 1000 developers involved in over 150 open source projects.

Rolf Nelson: Yes, it's interesting. This is Rolf. I – it's interesting that you've mentioned Eclipse. Certainly in the development tools space, that's an area that keenly, that Rational focuses on. And a lot of the open source tools are built in the development tools space, certainly products like Subversion. What I find interesting about the adoption of Solutions like open source version control tools – is that if you look at what analysts have looked at what kinds of customers are using those tools, it's really a kind of a U-shaped adoption curve depending on the size of the company.

So we see small companies adopting open source certainly, like Subversion. But we also see very large companies beginning to incorporate subversion into a suite of products and build integrated solutions, what we call ALM or Application Lifecycle Management. So, it's kind of, I think it's interesting for the panel and for our customers to kind of speculate on why that adoption of open source varies between company size. And I think it's pretty clear that small companies are constrained by cost, and often adopt open source as a starting point for their development teams, and a place to get started.

And then you see as I've mentioned before – the midsized companies are really much more users of Commercial Software Solutions, and less so with Open Source Solutions. And we can kind of speculate on why that might be is that we're beginning to see larger customers incorporate open source into integrated ALM Solutions.

And what they're doing is basically they have the resources in their company that have a dedicated team that's responsible for integrating the tooling, whether they use Subversion and other products, like maybe a natural planning tool, or a defect tracker, or a Continuous Build Engine. And then they have teams that are building integrations, and managing those integrations to kind of provide some sense of standardization across your organization.

And so we see a combination where these very small teams and some of the larger teams are adopting it. So one of the things that I would like to propose is an area where we see, where it's more prevalent that you might look at a commercial solution as if you're trying to build that integrated ALM Solution. I, as a buyer of open source tools, or commercial products would kind of want to evaluate commercial products to kind of measure the costs, what does it cost me to sort of maintain those integrations. And also, what level of capability do I get in those integrations if I use a series of point products.

I'd like to think of it as kind of like when you're buying a system, like an airplane or something, you wouldn't buy a bunch of parts lying in closed formation but rather you would buy an integrated system. And the way that IBM has approached the ALM problem is to take a common platform approach, an open platform approach. And so that really allows you to basically have – if you think about it from the standpoint if you have 3 or 4 point products and you tried to implement those.

They each have a different security models. You have to add users to them, you have to implement role based security across them. And you start to look as your team size grows into things like Automated Traceability, or Common Metrics and Reporting across the tools. Or even of common process, and common security model across the tools. So, we sort of see that those companies, both midsized, and larger sized companies benefit from having a common set of services that they can use across the ALM tools that plug into that platform.

So, that's really where I think the sort of sweet spot for Commercial Solutions that integrate into a platform provide value over trying to integrate those solutions yourself, especially if you could use those resources elsewhere in your company to delver business value, rather than managing point product integrations, or getting less capable integrations across those tools.

(Pam Induni): So, Rolf, you…

((Crosstalk))

Rolf Nelson: …Those tools?

((Crosstalk))

(Pam Induni): Go ahead, I'm sorry – Was that Jean Louis or (Don)?

((Crosstalk))

(Don Boulia): I’m sorry, this is (Don). Yes, I just was going to add to that. That – we see a lot of the same types of things with our open source activities within the WebSphere runtime space as well, so. And in the middleware case we certainly have offerings that are in fact almost entirely open source, and certainly are good for a series of, you know, application types.

But as you look at consuming open source, or leveraging open source, you know, you really end up spending a lot of time, at least we do from a WebSphere perspective in applying what we would call that (Finnish) or that integrated experience as Rolf was talking about.

So, it is going to be in, naturally there. It’s something that as you consume open source, and add it to your offerings, and as we do with a lot of the WebSphere middleware – you have to build up Traceability, Serviceability, maybe even performance as part of that. And so, it’s still valuable to start from an open source perspective because that’s where a lot of the original capability comes from in the middleware space. But there’s a lot that goes into applying the overall finish for a commercial products.

Rolf Nelson: And you know, I think that it’s actually really key when you're looking at something like Software Best Practices or trying to implement Agile methodologies. And so, you know, Continuous Integration has been around or a long time but if you look at something like Continuous Integration it touches the (SDM) function, the Build function, the Defect Tracking System. You need to know, you know, what text went into this build and how does that trace back to the code changes.

And so, this kind of traceability, you know, being able to automate that. The same thing is true of the Agile planning side, which is you know, an active area right now, Scrum and Methods. Agile methodologies where you're managing Agile estimation and planning. And you want to link between those plans, and those timelines, and those stories, right into the tasks that are actually implemented into the change sets, into the build.

So the kind of example of the kind of thing you can do with Rational Team Concert, which is built on this common ALM platform. Is you can start to do things like, when there's a failed build in the continuous integration framework, we can actually reconstruct the workspace in the (SCM) tool so that another developer come, could come in and fix actually the problem that another developer – a previous developer may have created that caused the failed build.

So that's just one example but just, automated traceability and if you try to do Continuous Integration and you try to do Agile planning, it touched each of those tools. And so, you know, I would try to proffer out there you know, you can use open source or even commercial point products. And each of those point products, if they do version control, or agile planning, or defect tracking, or build – Continuous Build. – They work well in isolation. But when you want to use them together as an integrated whole. There's only limited support for that and you know, the integration capabilities are not as functional. And they're quite costly to maintain.

So it really comes down to, do I have the resources to maintain that. And I think when you see midsized companies not adopting open source tooling. A lot of it has to do with it, they don't have those resources to maintain those integrations. And I think as we look at tools like Team Concert, the functionality that you get around collaboration and the linkages across the tooling are so compelling that it really would cause you to step back and re-evaluate whether you should be maintaining an ALM integration across a bunch of point product tools.

(Pam Induni): Rolf, you and the Panel have thrown out some really I think, very interesting topics here. And a point – I'd actually like to open up the call right now to our listeners on the line to see if – A, they have any experiences that they would like to share, appropriate to what you just discussed. Or any questions for the team. So, (Katherine), could we open up the lines.

Coordinator: Certainly.

(Pam Induni): Thanks.

Coordinator: And now we'll begin the formal question and answer session. To ask your question, press star 1. You will be announced prior to asking your questions. If you would like to withdraw your question at any time you may press star 2. Once again, to ask your questions, please press star 1. And one moment for the first question.

(Pam Induni): And again…Not just questions – we're happy to have any comments you may have as well.

Coordinator: Once again, to respond please press star 1. We have no responses at this time.

(Pam Induni): Okay, (Katherine), well let's move on. And I think this might be a nice time to move into systems. Rolf, you mentioned airplane a little bit earlier. So, let me throw this one out to Jean Louis. I know you have a strong background in Systems marketing and development, and in the tools for the technical customers. Can you talk about the specific needs for Systems customers versus IT customers? That might make the open source decision versus commercial different from what we've discussed here already today?

Jean Louis Vignaud I'm not sure that she's different but that's a very interesting topic. You know, if I look at the systems space today, most of the companies need to meet the either regulatory compliance or industry standard. What I mean here – it could be the U.S. Food and Drug Administration, FDA. Or, it could be achieving a (CMMI Level 3), CMMI meaning Capability, Maturity, Mobile Integration, or it could be compliant with (DO178B). Or we definitely would have similar kind of performance with the IT industry with our (unintelligible) Compliance and with IT Infrastructures like IT.

And to me these products it may be necessary to monitor and report on the access control to that part. To implement electronic signature, to clearly separate school and user administration roles, to establish a policy on user permission. It could be, as well, to manage full and auditable trustability from requirement zone to the line of software code.

So, and the organization edition may also need to clearly demonstrate user (unintelligible) realities. And if we look from an open source tool, open source product perspective, you know, most of them are not designed to support all these things at first.

And if you're using an (Haddock) Solution complied of this code disparate open source solution as was the has well IT previously inefficient, you may not be able to provide the competency of all the features required for your industry. So certainly in the System industry customers will be watching and very find that the tools they use are open to comply with this kind of performance.

(Pam Induni): Excellent. Thank you very much, excuse me. So, let's – I want to go back to the audience for a question here and hopefully we'll get some responses here. We like open source be, you know, you could see all the defects, the SRC codes, et cetera. Rolf, I know you're high into this. Can you maybe rephrase that a little bit and open the question up?

Rolf Nelson: Yes, no. There are a lot of customers who like to use open source products, like Subversion or (Hudson), or other kinds of tools because they can see the development take place on the Internet. And they can participate in that, they can join customer community forums, they can get fixes to their defects, you know, quickly, because they can grab an intermediate, you know, fix or patch that might be post it up on a development site.

So when we looked at what customers were gaining from that, and kind of thought about that, like how can we deliver more value in the commercial software realm. We basically opened up our development effort for all of our Jazz based products on www.jazz.net. And other product – as a product manager for Rational Team Concerts, our customers get to see exactly what I see as a product manager inside IBM.

So you can see we've completely created a transparent development environment for Rational Team Concert, and the Jazz platform. And it's really unique. I mean in the sense that it takes all the benefits that you would achieve from having open transparent development of an open source project. But you also then have the visibility to the development team. You can communicate and converse with that team on the Forum. We even have our source code posted there. But you would see every intermediate milestone. You see the (unintelligible), you use our dashboards you can see the quality of all of our components.

You can have discussions with Engineering around the design in all of the work items, all of the defects, and enhancement requests. So it’s really kind unique, and you can actually get notified when a defect that your company has filed, is appears in an intermediate milestone or build. So, it’s really quite useful in that you have this open development model, transparent development model so that nothing is really hidden.

All of the information that you’d like to learn about that commercial product is up there on the site. And you can even see our product planning roadmap, our Agile planning backlog, you know, the way we use Scrum methodologies in our own development efforts. So, it’s quite different than you might be familiar with commercial products in the past from Rational. And we’re getting a lot of feedback from customers that this is the kind of thing they want to see with commercial products. They want to see that openness and transparency.

They want access to defects, they want to know when their defect is going to be fixed, how it’s fixed, and when it’s fixed they want to test it out on an intermediate milestone. So that they can validate that it’s fixed before the next revision comes out, or in a patch.

So, it’s really quite – it’s allowing us to really transform the way Rational Developed Software Products by connecting our development team directly with customers, directly with Agile methodologies around planning, and estimation. And exposing that to our customers for them to leverage and utilize in their own development efforts.

(Pam Induni): So, Rolf, if I could just jump in here. So you’re basically getting the best of both worlds, in essence. Because you're getting the openness plus the commercial development. And so, it’s almost like open commercial development, and you're calling this transparent development. Well, actually I'm wondering if any of the listeners on the line has even heard of this before? If they're familiar with Jazz and this transparent development.

(Katherine), it might be a good time to open the lines again and see if anybody out there – I'm just curious to know if they've heard about it because it does sound like the best of both worlds, and what they think about it. If this is something that would meet their needs. That, you know it, the title of the Telecon is, The Open Source, The Limitations And The Shining Aspects. This open commercial development, this transparent development almost sounds like the best of both worlds.

Coordinator: Once again to respond please press star 1. We have no responses at this time.

(Pam Induni): Okay, well that's Okay, we'll move on. This, I know this is probably new to a lot of people. Rolf, I'd actually like to talk to you a little bit about Rational Team Concert Express-C. Because I know that's our community product here, and that may actually fit into what you were just talking about with the transparent development as well. Can you just spend a couple of minutes on that?

Rolf Nelson: Yes, Express – Team Concert Express-C is a Community Edition. It's a very full featured ALM, you know, lean ALM Solution. It does have Agile Planning, work items, SCMs, Continuous Builds. And it's basically, comes with ten free developer licenses that are, that you can use perpetually. And they're basic – and then they're – it gives you community support. In other words, you use the Forums and the Forum activity that's going on at Jazz.net and those Team Concert Forums that discuss any questions and support options you might want to think about, or questions you might have to support your (Community Addition) deployment.

And, really what we have designed that for is really for two reasons. One, is when the small companies that have a cost issue, and want to get started with an ALM Solution from the beginning rather than (Point Product Solutions) can use Team Concert's Express-C edition to do that. And then the other model is if you're in a larger company, and you want to sort of experiment with Team Concert and see what the return on investment is. And really understand the collaborative value, and the traceability, and automated traceability, and capabilities to manage this common process across the tool set.

You can deploy it on a pilot project, maybe an Agile project that's doing Scrum, and report back to your management on the ROI of it, and use it as a way to show your management team the benefits of an integrated open ALM platform approach, which kind of gets me into another topic, which is, you know, some people will say, "Well, open source is really the most open of all – and how can IBM's platform be open if it's, you know, built by IBM."

And, "Yes, but if I use –" I hear this a lot from open source customers, they say, "You know, if I use all IBM's tools it all works great. But if I have to plug in other tools…" You know, "Then it doesn't work as well."

And what we've really done there, and this gets back to kind of contributions to open source. Instead of just sort of introducing a platform and saying, "Here's our API, this is how you talk to it, you know, build an application or an integration with it." We've really invited the community to participate in a definition and creation of specifications around Open Services for Lifecycle Collaboration is what we actually call it, OSLC.

And we have a separate Web site for that, www.open-services.net. And we have community-based open licensing model that allows you to take those specifications, implement them in your products, and have them communicate between tools across the Lifecycle.

So, a proof point for that is we've released a Change Management Spec, which is really like the work item or defect tracking spec for how Lifecycle tools would talk to a defect tracker. And then companies who produce open source products that supported open source solutions like (Myland) and a company called (Task Top), produces a (Myland) product that they support.

And they've integrated that with our Changed Management spec – and what it means is they write one integration, and it works with any of the Changed Management Solutions from Rational, including Rational ClearQuest, and Rational Team Concert's Work Item component, which is the Change Management component.

So it's really a way to write an integration once and have it work with many tools that implement a Specification. We have many companies including competitors and customers who have joined that OSLC activity to help build an open standard for how these tools will talk to each other.

And it's really, it benefits IBM because we have a lot of products that need to talk to one another. But it also benefits open source community and commercial vendors who are trying to build integrations with many tools. I mean (Myland) is perfect example. I think they support 20 or 30 different Change Management tools. And if everyone of those Change Management tools implemented a common spec, then they can, that one integration would work.

And the other benefits of this is, it's not just to promote a particular standard but really, we’re using (REST) style interfaces to do the specification and the purpose of that is really to allow these tools to be somewhat loosely coupled at the specification level so that when a version of the tool changes the spec – the integration doesn’t break.

So, this is really a real problem if you’re trying to build the ALM suite of tools and you’re using a combination of open source and commercial tools, if you – you don’t want to have to, every time there’s a new version of one of those 3 or 4 tools, have to kind of retest and qualify all of the integrations and all of that.

So, we see this as a real savings, and a real way, a real open possibility for any product to take advantage of this. It doesn’t have to be a Jazz product, or have to deal anything with Team Concert at all. The OSLC specifications are independent of Jazz and they can be implemented in any product, so. We are certainly implementing it in our requirements tools. We’re certainly implementing it in our Change Management tools. We’re looking at implementing it in our testing tools. And you can certainly find more out about that on the Web site open-services.net. Uh-hum.

(Pam Induni): Excellent, thank you Rolf. I’d like to change it up here just a bit. I know we’ve been talking a lot about development. We also have (Don) on the line with us. And, (Don) being the, from WebSphere, and the open source strategist there. I was just wondering if you could discuss with us a little bit how, you know, how should customers view the trade offs there with open source versus commercial in the area of middleware?

Rolf Nelson: Sure. So the middleware space is, you know, is fairly well established and mature, particularly in the (Java E) segment of WebSphere that I personally deal with on a regular basis. And, you know, there’s a couple of dynamics there. Certainly, we have a set of standards, activities, around Java Enterprise Edition that we follow from a WebSphere Application Server perspective. And there’s a lot of open source implementations of either parts of that Spec, or entire (Java E Servers) in some cases. And in fact there are multiple open source projects covering that same landscape.

So, it’s again, fairly immature and as a result of that maturity – there are multiple implementations out there. And so when we started talking about WebSphere we actually in fact, within the Lotus (unintelligible) portfolio have an open source based application server of our own. WebSphere Application Server Community Edition, which is derived almost entirely from the (Apache/Geronimo) project. And you know, the intent of that offering is really to give people the flexibility, and in some cases the advantages that they see with open source community based involvement and approach. You know, access to source code, in some cases it is relevant to people. And you know, and allow them, they use those in an environment, or in a model where you can get a level of IBM support.

We tend to recommend that for people when, you know, the needs are such that, you know, the speed of development is critical, and maybe they, the overall complexity of the application is not as much of a factor. As the applications get more complex and the needs, and the requirements, and the throughput requirements in particular for a runtime become more advanced you start to run into some of the bottlenecks that are present in open source software.

And, you know, not that these are not things that can’t be overcome. But, the difference starts to become and whether you as the consumer of the runtime become the person who manages all of these things versus letting a software product manage that for you. So, when we start talking about the commercial offering of WebSphere Application Server. And you start talking about clustered applications, highly available applications, highly scalable applications, so things that are going to truly run your business, whether that's internally facing or externally facing. Those concerns become much more important. And frankly those are the things where we find that customers tend to choose a commercial product to provide that.

There's the level of support that comes along with that. And then as Rolf was saying before there's the overall integration and consistency across the entire stack. The stack products are in the case of Lotus here are also, you know, of a commercially based set of things. So we have built on top of our WebSphere Application Server foundation a set of higher level offerings for things like business process management for example. Enterprise service buses, you know, commerce. There's a variety of things that are in that space and those leverage all those qualities of our Commercial Application Server underneath. So as you extend that into more needs, the commercial offering tends to be a better fit. But there's clearly a, you know, a class of applications out there that don't have those requirements and can benefit from, you know, the agility that is present in some of the open source projects out there.

(Don Boulia): But, that's interesting (Don) because it brings up a point on the Team Concert side, I mean, when we built Team Concert on top of an open platform built on open standards, we basically looked to supporting both Open Source Application Servers like Tomcat, as well as WebSphere Application Servers. So, when you move up to the Enterprise scale and you need things like high availability, or other kinds of capabilities like in the future clustering – we're leveraging the commercial capabilities built into WebSphere Application Server that we bundle with Team Concert for its use when it wants to implement those capabilities that (Don) just mentioned.

So, another question I get all the time from customers is, you know, can I use Team Concert or can I use the IBM ALM Solution for lean or Agile ALM, can I use that collaborative development environment with my open, existing open source tools. And I've got Subversion in-house, can I employ Team Concert, and take advantage of Agile planning and project dashboards, and managing all my work items in one place and in an integrated way and use the collaboration capabilities of Team Concert and the answer is yes.

We've built bridges that leverage actually some of the same OSLC interfaces we talked about. So the bridge between Subversion and Team Concert – and there's a bridge between (GET), which is another newly, fairly popular platform for version control. And if you think about Team Concert it has many components but there are two completely extensible components of Team Concert. One being the version control engine, you can use our Jazz SCM, which has advanced stream based and component based capabilities. Or you can plug in subversion or (GET) as a bridge that links to change that to the commit from those tools into the work items in Team Concert.

And then you can use the agile planning components, and the dashboards that are porting in metrics on project status in a consistent way. So, another thing that we noticed when we talked to analysts is that oftentimes companies have more than one (SDM) in their corporations for these reasons. They might have advanced (SDM) needs and certain projects, and maybe basic needs on smaller teams. But what they bring Team Concert in they can wrap that as a common reporting, and Agile planning engine for any (SDMs).

And we have bridges to ClearCase, and (GET), and subversion as I mentioned. And the other component that's completely extensible in Team Concert is the build engine itself. So it comes, Team Concert comes with the Jazz Build engine which would be sort of similar to a nice Continuous Build engine that’s (ANT) based or implemented using (ANT). But then you can also plug in Build Forge as the build engine which is a commercial product from Rational that does Enterprise Build capabilities like load balancing across Build Servers.

Or you can use a popular open source tool like (Hudson), and plug that in as your build engine to get the reporting status and metrics on project health and build health reported up through Team Concert. So, it really is an open platform in the broadest sense. And I think for this topic and for this customer set it’s not just an all open source or all commercial but rather, how can I use those commercial tools that give me value and wrap around the open source solutions that I’ve already begun to use in my company?

And we actually sell Team Concert in a couple of different ways to allow that to happen. If you want to buy the full Team Concert product without (SDM), or build capabilities, and wrap it around an existing open source tool – you can buy, there are Change Management component or the something we call the RTC contributor, which is essentially a planning and tracking and dashboard component of Team Concert that can be utilized with any (SDM) or build (unintelligible).

(Pam Induni): (Katherine), I’m wondering if we should open up the lines one last time here to see if there are any questions. I know, Rolf, you just kind of teed up, you know, why open source, or why commercial, or why a combo of the two. And I’d like to hear from anybody out on the line how they feel about this.

Coordinator: Once again, to respond please press star 1. And one moment for the first response. Once again, to respond, please press star 1.

(Pam Induni): I think we have a quiet audience today. We really thought that this topic would bring out some real strong opinions. But that's okay because I think a lot has come out today as far as the pros and the cons, in fact. If I could put the Panel kind of on the spot here in a summary. Can you talk a little bit about what would be, and I know we've talked about this throughout this call but maybe in the summary. What are the pros of open source, what are the cons – pros and cons of commercial development – and then when you're deciding what's right for your team, maybe a summary of, I think we've heard today, it's probably a combination of the two.

So, if I could throw this out to the Panel and see what you have to say.

(Don Boulia): Yes, so, this is (Don). I'll kick this off because this is a conversation that you know, we've certainly talked about a lot within the WebSphere community. So we are a consumer of open source in multiple ways. As well as certainly we see competitors in the open source, you know, arena, as well.

There's a few things that we look for, you know, in terms of what we would consider to be, you know, health fields, and source projects, and a few sort of dynamics that tend to play out. The first one is, that you know, we're pretty big on making sure that the open source communities that we've participated in are truly open. And so that's a piece of this as well – making sure that there is a true community around that. More people can participate. Because that's where you get kind of the best of the best when it comes to open source.

You know a lot of recent open source has really been open source in license, which means that you can really get to the code, but there's really not the open community around that license. And so, the amount of innovation that happens inside that open source project is now kind of stifled because it's, you know, controlled by one company or controlled by a couple of individuals.

So a lot of our work in WebSphere, and certainly Eclipse, and Apache fall into the category. You know, really revolves around making sure there's a good community structure around the open source, so that you can get the best of that. The other dynamic that I think is interesting to note if you're looking at open source is that, you know, there is typically a very disproportionate ratio of, you know, people who are consuming the open source versus people who are actually participating and contributing to it.

And so those, the contributors to open source tend to be fairly small. Most of the people are consumers of it. But those, whoever those contributors are really again, do determine company innovation that happens within that open source kind of community and project. So, we look for making sure there is a diversity there. That there is a good mix of people coming to the table with different viewpoints because again you get a better end result when we're the, when we're looking at open source from a sponsorship perspective.

We want to see those kinds of things happen. So I think that's kind of the best of what open source can bring to the table. Where open source doesn't tend to deal as well from an overall software perspective is in the problems that we come, you know, harder and harder to you know, solve in the sort of individual contributor case. And so an example is that most of, most open source projects have individual contributors usually it's a portion of their job or in some cases, even a hobby or a spare time. And so I mean, we start talking about more advanced function, it's harder to build, harder to test, it's harder to validate.

And also, it isn't necessarily in the mainstream of what everybody's needs. You don't tend to see people in open source focused as much on those kind of concerns. And so, the analysis we take away from that is, that's where we start to look to our commercial offerings. As really being where we can provide that with a commercial product offering – we can put a bigger test effort against a software. Put it, you know, features and functions in that address a, in maybe the higher-end needs of the market because there's a premium to be paid there.

So, there's a spectrum here but in general the community aspect is a good thing, and it's something we look for. And we want that to be healthy, but at the same time we're pretty aware of the fact that there's some projects that are, you know, going (unintelligible) to be very successful in open source because they solve, you know, a mainstream set of needs. There's going to be other things that really wouldn't be a very successful open source endeavor just because you won't find the critical mass of people to really contribute to it.

Rolf Nelson: Yes, and I think that's good summary I think of the, of some of the advantages that you see. You know, where open source would fit and where commercial products, or needs get more specific. Where you might look to a Commercial Solution. I think that the other point I would make is that in order to have the flexibility to use both Open Source and Commercial Solutions I think that, especially if you're looking at development Lifecycle or an ALM solution cycle.

Having an open ALM platform that is built around open source components and componentry. The people who built Team Concert and the Jazz Foundation are the same people who built Eclipse for example. The same development team that you know, brought Eclipse to the market and delivered it as an Open Source Solution for individual productivity, worked on Team Concert for Team productivity, and collaboration.

And I think that there's a synergy there, certainly between the ability for, "I want to grab an open source tool, let's say a product like AML, or (Myland), or any of the open source components that are part of my development effort. I want that to plug in seamlessly with the open platform that's my ALM platform for my reporting infrastructure, and my process infrastructure for the tooling.

So, I think it's really key when you making a decision to buy commercial products that you look for open commercial platform products that open source tools can plug into and extend the value of both your Open Source and Your Commercial Solutions.

(Pam Induni): That was perfect. Thank you. Let me go to questions one last time before I wrap this up. So (Katherine) can you see? I know we actually have quite a few people on the line today but they seem quiet. Can we see if there's any final comments from anybody, or, questions?

Coordinator: And once again, if you would like to ask your question, please press star 1. One moment for that first question. And we do have two questions coming through. (Brad Ebert), your line is open.

(Brad Ebert): Well, thank you gentlemen and ladies. It actually was a very good evening, even though the crickets are very quiet out here. It is something that does actually hit home with all of us. But the one thing kind of in the back of my mind. And this may not even be an issue anymore, concerning open source. For our companies that are going out there and developing solutions on their own using the open source tools. Is there any risk associated with the fact that that's public domain and the solutions they build?

Are they in public domain or even the products that they bill off of them become public domain. Is there any concerns there at all or is that sort of a fallacy of times past?

(Don Boulia): So, this is (Don). It certainly is going to depend on the license, and so that’s a piece of the puzzle here for people who are consuming open source, you know that, you do need to be aware, if you’re going to use it inside your product or if you’re going to use, you know, solutions based on open source to build your products. You know, generally speaking, there are, you know, commerce – so called commercial friendly licenses, you know. Eclipse as a license, falls into that category and so does Apache, in general. And either – there are still risks associated with open source when it comes to making sure that you have good governance over the IP that’s going into those open source projects.

So, you know, you don’t want people, you know, contributing to those – contributing assets to an open source project. That, you know, they don’t own because then you have a whole bunch of other downstream issues. Generally speaking though, a lot of the reason that IBM focuses in on, you know, projects and communities like Eclipse and Apache is a combination of both these community but also the fact that the licenses themselves don’t have some of those issues that you were talking about.

(Brad Ebert): Okay, thank you.

Coordinator: Our next question is coming from David. Your line is open.

Man: Yes.

(David): Good afternoon, good morning – wherever you happen to be. One of the things that I think that it’s very interesting about open source and I would love to hear the thoughts of the Panel. And that is respect to companies and the executives that require support of the software solutions that they use. For example, you know, they’ve got their assets in those, in these tools. And so therefore, we look a few back to when open source just started becoming prevalent.

You know, the executives very much shied away from open source because of the lack of support. And a good example of how that turned around, as we think about Red Hat. Where Red Hat took the open source of the Unix operating system and poured around the company around the supports and services. Do you see that IBM’s vision with Jazz and the things of that nature are thinking in that regard? I just want some thoughts with respect to that because that’s what caused open source to kind of, you know, get a rough start so to speak. Any comments?

Rolf Nelson: Yes, I think, this is Rolf. I think for, I do agree that open source products because of their interest level and the customer usage of them have created a sort of a secondary market. A very and actually the primary business piece around open source products is the support model, or building a business around, you know, commercial support for open source components.

I think that can work for some folks, I think can work at the operating system level. I think it can work certainly at, you know, point product solutions that – one – do one task well. I think it’s it gets a little bit more problematic when you're , when you try to sort of scale up, and then you have to buy commercial solution add ons. And you may have support for the commercial add on, but then do you have support for the other components. Or the combination of the integration across those tools might be different support contracts from different companies.

So it becomes more complex for a larger enterprise, certainly to roll that kind of support relationship out with the customer, which is one of the reasons I mentioned this sort of open transparent development. Because we’re trying to do there is really say, ”You know, open source has some great ideas and wouldn’t it be great if commercial products has the transparency of those open source projects” And yet you can get 24/7 support. You know, you can call us from, you know, from India or from other parts of the globe at any hour of the day. And you know, get a hot fix or a critical situation fixed. You know , when you're talking about mission critical application development where the, you know, your business is running on the development artifacts that are stored in the database. And if you can’t get at those your developers are down, and that’s costing you incredible amounts of money. So, you know, we really see the, you know, we provide of course, Enterprise support contracts with those commercial software products. And we see that you know, as a benefit.

We haven’t really extended, I don’t think Rational played a little bit around with like looking at what somebody wants to use in open source component, like Subversion – do we provide a support contract there when they’re using it with the Team Concert. And we haven't gone down that route today but certainly it’s something that if there was demand for it, you know, we certainly, you know, would be open to exploring that in the future.

But, as today we’ve kind of provide those support contracts for the commercial products, and certainly the open source components that we use, whether it’s Tomcat, you know, or Apache ,or other kinds of components that are built into Team Concert are supported commercially for customers. So when we actually bundle an open source component and use it in our infrastructure that’s part of a commercial support offering that you're essentially buying from IBM in that situation.

(David): Yes, that is essentially – thank you very much Rolf. That’s essentially what we end up getting is the best of both worlds -don’t we? With the collaborative efforts with an open source world with support of a – strong support of a company behind you.

Rolf Nelson: Yes, thanks.

Coordinator: We have no further questions at this time.

(Pam Induni): Okay, (Katherine). Well thank you. Well, I have, I really want to thank everybody today. Let me leave you by saying that we actually have a white paper coming out in about a week on this, it's titled, Open Source Commercial or a Combination of the Two: Deciding What's Right for Your Company. If you'd like more and more about transparent development. We talk about that in there, and nice leave behind, or pass over to your co-workers or managers. That I'd – like I said, will be available in about a week.

It's going to be posted with this Telecom replay on the site that you originally registered on. So we hope you come and read that. I think it will summarize a lot of what was discussed today. And if it's a very good paper, so we look forward to having you read that. So, with that said, Angelique, I'll turn it back over to you. Thank you.

Angelique Methany:: Thank you very much. This has been a really great discussion. And a thank you to all our panelists, (Pam), you – of course, (Pam Induni), (Don Boulia), Jean Louis Vignaud, and Rolf Nelson, for taking time out of your busy day to speak with us today. This is a valuable session. Very informative and we appreciate you being here for this Rational Talks To You conversation.

If you would like to listen to this conference again, or share it with your colleagues, this will be made available for replay in MP3 format in about a week or so. On the Rational Talks To You site, www.ibm.com/rational/talks.

Our previous teleconferences are available there as well. We'll post a link that (Pam) mentioned for the new white paper: Open Source or Commercial – Deciding What's Right for Your Needs. That's the name of it, so be sure to check it out. We would also like to think you, our audience, for your interest in IBM. We hope to see you back for another one of our events in the near future. Thank you very much. Talk to you soon.

END