**IBM**

**Moderator: Angelique Matheny**
**November 20, 2008**
**12:00 pm CT**

Operator: Good afternoon. My name is (Andrea) and I will be your conference operator today. At this time I would like to welcome everyone to the conference.

All lines have been placed on mute to prevent any background noise. After the speakers' remarks there will be a question and answer session. If you'd like to ask a question during this time simply press star then the number 1 on your telephone keypad. If you'd like to withdraw your question press the pound key.

Thank you. Ms. Matheny you may begin your conference.

Angelique Matheny: Thank you (Andrea). Hello everyone and welcome to this rational talk to you teleconference: Quality in Action: Using Rational Quality Manager with Functional, Performance, and Web Service Testing products.

I'm Angelique Matheny with IBM Rational. I just want to make some quick introductions. Joining us today is Brian Bryson. Brian is technical evangelist for IBM Rational Quality Management. At IBM Rational Brian is a member

of the quality management team that plans and delivers IBM Rational Software Quality toolset.

Also Matt Holitza will be with us today. Matt is a marketing engineer for IBM Rational Quality Management as well. His current role focuses on providing customers with solutions and information to help them succeed in managing the quality of their software development efforts.

Today's discussion is focused on using the newly updated Rational Quality Management Functional Performance and web services testing products in conjunction with Rational Quality Manager.

Brian and Matt will explain how the new feature set enables you to better collaborate with the entire software team, use automation to accelerate delivery, and how our integrated approach provides you with the information required to govern your software projects, enabling you to answer key software project decisions.

Now you won't find any slides for this teleconference. These calls are really for you. We want this to be interactive and this is your chance to get your questions answered directly from our experts and discuss what's on your mind.

As the operator mentioned you should press star 1 and the operator will open up your line at any time so don't be shy. Also if you'd like to submit questions to our panelists after this teleconference please email us as askusnow@us.ibm.com. That's A S K U S N O W at U S dot I B M dot com. Just put the title of this teleconference in the subject line.

Well I think you've heard about enough from me so let's get started. Brian I'll turn it over to you.

Brian Bryson: Thank you Angelique and welcome everyone. My name is Brian Bryson. I'm with the product team at IBM Rational in the quality management division so the team responsible for all the tests and tools that we know and love.

I have with me today Mr. Matt Holitza, my partner in crime. We're going to have - I don't think we have an exceptionally long call today but we just wanted to follow up on some new announcement news and talk a little bit about what's new in our latest offering.

For the uninitiated on October 28 - so three weeks or so ago - we released a fairly significant update to our portfolio. In that update were new versions of our three primary automation tools so those would be Rational Functional Tester, Rational Performance Tester, and Rational Service Tester, formerly known as Rational Tester for SAO Quality. We tweaked the name a little bit there to be sort of fall in line with our naming convention.

And along with those updates were two significant new product announcements - those being Rational Quality Manager and Rational Test Lab Manager.

So the purpose of our call today is to just really discuss what's new in all these tools, explain a little bit how they all kind of work together and flow together. And to really open up the lines to any questions that anyone out there may have about the tools.

So at any time feel free to press star 1 to interrupt with a question. We'll - I think the way we're going to go about it today is we'll talk a little bit about

each product and after we talk about each product we'll see if there are any questions to run into.

Maybe at this point I will ask our operator (Andrea) - (Andrea) is there any questions that are actually sitting in the bucket as of - before we even get started?

Operator: No sir not at this time.

Brian Bryson: Excellent. And it is star 1, right?

Operator: Yes sir; star 1 on your telephone keypad.

Brian Bryson: Excellent. So at any time feel free to interrupt with any questions. And I realize it's kind of strange to not have, you know any slides or anything to follow along with.

But the agenda that we're going to go with today is we're going to start first talking about the two new products. I'll talk a little bit about Rational Quality Manager and then we'll unleash Matt to talk about Rational Test Lab Manager.

And after we cover the new products we'll talk about the updates to the existing portfolio; specifically Rational Service Tester, Rational Functional Tester, and Rational Performance Tester. And then we'll wrap it up and call it a day.

So let me begin then with the - probably the headline news for this latest release that we had on October 28. On October 28 we introduced the dawn of

a new era in test management for IBM Rational with a new offering called Rational Quality Manager.

So, you know, in its simplest sense Rational Quality Manager is a tool for managing the test processor - sort of a test management tool. It is the tool - a tool you can use to manage your requirements, your test cases, your test plans, and your defects. So kind of a full lifecycle from beginning to end requirements to defects management tool for managing your testing projects.

This is something new yet not new for us. If you've been a user of our tools before you'll know that we have had previous offerings that sort of had similar functionality - Rational Test Manager is a test manager tool we've had since the early 1990s so probably our very first generation tool.

A little more modern than that is Rational Clear Quest Test Manager so that's our most recent test management offering. And then now we move on to our next generation offering here with Rational Quality Manager.

So, you know, the area or the objective of the tool is really nothing new; wanted to offer something that gives teams an ability to manage a test plan, keep track of requirements and test cases so you can answer questions like what's best tested, what's not been tested.

Keep track of all your test results, run various kind of tests, functional manual performance, we service tests, security tests; any type of test really and consolidate those down to effectively a single report and, you know, answer the question that every QA team has to answer and that's, you know, are you ready to release?

I don't know of a single test team or test manager that doesn't have a weekly status meeting with, you know, the rest of the project and has to report on, you know, the quality of the application. It all boils down to that one moment when the project manager or the director or whoever's in charge looks over to the QA team and says, "How's it look? Is it ready to go? Are we ready to release?"

And the QA team needs to come up with an answer. And, you know, a lot of the times I think, you know, realistically and maybe unfortunately the answer's a bit of a gut feel based on how testing's been kind of going or a general feel for, you know, how many defects have been, you know generated or, you know, what you saw in the last build.

The idea of Rational Quality Manager is to quantify that and make it a little more justifiable; a little more quantifiable so you can actually say, "Well our defects have been increasing at a rate of X per day." Or we've, you know, we've got our requirements covered. Our requirements covered are now at 90%. Our test pass rate has been increasing and it's now at 80% or whatever the statistics are.

The idea is really to provide those statistics so that you can answer that, "Are we ready to release?" question.

You know, probably a lot of interesting things about Rational Quality Manager. I think probably, you know, the top one that most people are interested in is a web based tool so it's got a very slick web tool interface which gives you really two things.

One it gives you the ability to support distributed teams so you set up a central server that anybody around the world can access. So for teams that are

distributed across floors, across buildings, across cities, across states, across provinces, across countries, whatever - across the world - can all access and share the same data. And that's, you know, the key benefit of the web architecture and probably the number one reason we went with the new generation of tools is that our previous tools were not web based. There was sharing and replication but really we wanted to make sure we had a fully web based solution.

And then the second that enabled is a much more - want to say accessible interface. And by accessible I mean one that you can customize so you can - with Quality Manager as an example there's a dashboard that you see the moment you log in.

Well the content on that dashboard you customize. You have what are called little viewlets and you put in these viewlets have different types of information. You put the viewlets that you want on your dashboard.

These viewlets can be related to, you know, the actual information in Quality Manager or they can be (unintelligible) so you can pick up anything off the internet as an example.

So you could change the content and the layout. And as well in terms of sort of accessibility the tool has a very powerful sort of architecture that lets you pump data in and extract data out via (unintelligible) some very simple HTTP commands. So using HTTP post and get commands you can get data in and out of the tool which makes it quite accessible and very easy to integrate with other tools without having to, you know, write (unintelligible) code to some funky API.

Our method of, you know, import/export or our external access message our HTTP method which are, you know, make it quite simple.

And as well and, you know, and finally in terms of testability Rational Quality Manager is built upon the Jazz foundation. So the Jazz foundation is essentially an architectural nuance. It's the way IBM Rational is building all of its tools.

So you can think of it as similar to the way you think about an Intel processor in a laptop. It's not - we don't buy Jazz. You know, no one buys an L chip. What you buy is a laptop with an Intel chip in it.

Well same thing with this. You don't buy Jazz. You purchase IBM Rational Quality Manager which is built on Jazz the same way a laptop is effectively built on an Intel chip or a (unintelligible) or whatever the chip is on the inside.

But this Jazz foundation is the communication architecture and infrastructure between all of our products. So our entire line of tools is being redeveloped and redesigned to be built on this common foundation so as to better enable sharing of data and information between team players. And that's the idea of Jazz, right, you have a jazz band all collaborating and in sync and sharing and seamlessly sharing, you know, data and information - that's the idea with our foundation - our Jazz foundation server.

So quality manager is effectively the second offering on the Jazz platform. The first was Rational Team Concert which is tool for defect tracking, work item tracking, and version control or configuration management so kind of a developer focus tool.

Quality Manager is the second one for a tool for test management; integrates very tightly with Team Concert and integrates as well with the third tool which is Rational Requirements Composer. So the idea with Rational Requirements Composer is it is a tool for getting better requirements out of your stakeholders - out of your customers, out of your users, out of whoever the product is designed to serve.

We had for years a tool called Requisite Pro - a requirements management tool which really took care of the nitty gritty; the management of requirements - what's been done, what's been not, what are the requirements, which ones are high priority, which ones are not high priority, which ones are high severity, et cetera. So that's, you know, the management of it.

But the failing of that tool it's only as good as the requirements that are put into it; kind of a garbage in/garbage out scenario. And so the idea with Requirement Composer is to get better requirements in. And the way to do that is to extract better requirements from your customers. And the way to do that is to capture the requirements in the way that your customer elicits them.

And the idea with Requisite Pro is all requirements eventually were text - or ultimately were text. Well that's not necessarily the case. Sometimes the requirement is a screen shot. Sometimes the requirement is a mockup. Sometimes the requirement is a diagram. And that's what Requirement Composer let's you do. It lets you capture all these different formats and turn them into requirements. So speaking the language of the customer it captures the requirements then feeds them into the process.

So all of this to say our entire product line is being re-architected on the status foundation. Quality Manager or the task of quality management being so important in software delivery was the second tool built up on this platform

and is really becoming a central piece of communication between team members.

That's kind of the basics of Quality Manager. It's a test management tool, web took interface, built on Jazz, lots of nifty little features. What I would suggest if you want to take a look at it you go to our ibm.com website and view some demos or you can go to You Tube and see some demos. If you go to You Tube/Rational Tester you'll get a view of some of the interesting little features of Quality Manager that we won't really have time to jump in today but exist.

Now last thing I'll say about Quality Manager is that it recognizes something new that we've never done before in the area of test management and that's lab management.

So the idea here is that every test team has some sort of test lab attached to them where they actually run their tests. Maybe that lab is 5,000 difference PCs with different configurations. Or maybe that lab is three or four virtual images that are used to run tests.

Regardless of the size of the lab - be it a small virtual based or a large real based or some - or anything in between the need was identified to be able to manage those assets, schedule those assets, and be able to execute on those assets.

And that base capability's included in Rational Quality Manager but a more extensive professional lab management offering was also released and that's Rational Test Lab Manager.

So with me here today is our Test Lab Manager expert Mr. Matt Holitza. And maybe Matt I'll ask you to jump in and add a little more detail about Test Lab

Manager. Tell us what's - I guess all new with the product but tell us what's new with Rational Test Lab Manager.

Matt Holitza: Absolutely. Thanks Brian. And again I'll pose the audience if you have any questions about what Brian just talked about or QM it's star 1. But if no one breaks in then I will start talking about Rational Test Lab Manager.

The Rational Test Lab Manager is an extended component of Quality Manager. So what that means is it's an additional product that you can purchase to have more robust lab management capabilities than what's included in Rational Quality Manager.

The basics of Test Lab Manager are that it allows you to manage all of your test lab inventory like Brian was talking about - virtual or physical assets that you may have as part of your test lab.

And once - Test Lab Manager also allows you to do is automate the gathering of all the configurations that are on those lab machines. So once you have that, you know, what kind of operating system is it? What hardware's on it? What software is installed on these machines?

What this allows you to do is once you have that metadata in one repository your team can - your entire team has access to that now. And they can do - they can make requests, they can search the inventory, they can make reservations of the lab assets that you have placed in the inventory.

So this allows people to really work with the lab inventory without having to, you know, worry about if somebody else is going to distract them, if somebody else is going to execute a test while they're trying to run a performance test on a specific lab machine.

What Lab Manager also allows you to do is a lot of automation of your test lab processes. What we found in an IBM study of some of our customers is that upwards of 35% of a tester's time is spent testing up - or building up or tearing down lab machines. And so we wanted to take that time and give it back to the testers so they can spend more time, you know, doing what they do best.

So some of the things that you can do with Test Lab Manager is, you know, you can schedule the execution of tests on lab machines. You can - one of the neatest features I think is that you can actually deploy our test tools to lab machines.

So if you had an XP box that you wanted to run a certain test on - a functional tester script for example you could actually - and that machine didn't have it you can actually have Lab Manager go out and install functional tester on that machine and then run your test, you know, all in the background. You don't have to have somebody take a CD down to the lab, install Rational Functional Tester, then run your test. So, you know, just that capability alone adds a lot of value to, you know, adds a lot of throughput to your team.

And, you know, not only does it allow you to deploy your test and your tools but Rational Test Lab Manager's built using Build Forge technology. Now if you're not familiar with Build Forge that is our build automation solution that has been very successful as of late and provides a huge ROI for development teams.

Well now we have the capability in Test Lab Manager where you can actually do anything you can do in Build Forge with Test Lab Manager. So you can deploy builds to your machine. You can change configuration. Build Forge is

a very flexible tool and allows you to do a lot of different things. And basically you can look at your - the highest tend poles and really automate some of those processes that have given you some problems as of late.

Lastly Test Lab Manager allows you to do some reporting. So now you have all this information in your repository about what inventory you have, what lab machines are being used, which lab configurations are used the most. So we use - so for example we use and XP with Firefox and Rational Functional Tester is your most commonly used configuration.

Well now as a manager of the team - QA manager to the lab manager can go out and capacity plan for next year and see which ones I'm going to need in the future and really have an idea of what optimizing their test lab. So for next year we will be able to plan better, budget better for our needs and, you know, put the resources in place to fulfill those needs.

You know, one thing I'd also like to talk about is that Test Lab Manager is not kind of in a silo in Rational Quality Manager. So it really integrates with all the other functions of Rational Quality Manager.

So Brian mentioned, you know, the dynamic test planning you can do in Quality Manager. Well part of that test plan includes your test environment, your test lab.

So you say - I'm test planning. I'm not only going to worry about requirements in test cases and tracing those to my test plan but I also want to worry about the actual machines I'm going to use to execute that test plan.

So I can, you know, integrate that into my test plan and really track all those - all the different pieces now of a test plan to, you know, to my strategy.

And so that's really Test Lab Manager in a nutshell. So now, you know, one thing I didn't mention is that Test Lab Manager of course integrates with all of our existing products so Rational Functional Tester, Rational Performance Tester, Ad Scan tester edition, and Rational Test Real Time. So for embedded systems you can actually use Test Lab Manager to execute and install those tools.

And also Rational Service Tester which is formerly known also as Tester (unintelligible) Quality. So those are all of our tools and then you can also execute tests using a command line if you wanted to. So if you had some other tool that you wanted to use Rational Test Lab Manager with we can accommodate that.

So really Test Lab Manager is a tool for maintaining an inventory of assets, automating the test lab management task, and then optimizing your test lab utilization for future reference.

So now we're going to - we talked about kind of our newer offering. So now let's go into kind of some detail about some of the changes that came out with our other tools and we'll start with Rational Service Tester which Brian just talked about.

And again I'll ask the audience if you have any questions, you know, please hit star 1 because really this is about you and we want to hear your questions.

Brian Bryson: Operator (Andrea) anything in the question queue before...

Operator: No sir. But as a reminder if you'd like to ask a question you may do so by pressing star then the number 1 on your telephone keypad.

Brian Bryson: Lovely, thank you.

Okay yes don't be shy. We've got a fairly good size group out there today so I'm kind of surprised we're all quiet.

Let's motor on anyway and let's learn about some of the enhancements to the existing products. So three existing products that we have out there and have had out there for some time are Rational Service Tester, Rational Functional Tester, and Rational Performance Tester also got updated in the last release.

So I think, you know, we're not going to spend a ton of time on this but what Matt and I will do is walk through each of those. And then maybe just spend a few seconds describing it because I know not everybody uses every tool and then talk about what's new so maybe five or so minutes on each one. And then we'll stop for questions along the way or, you know, don't hesitate to interrupt - it's star 1.

Let's start with Rational Service Tester. So all of these tools - Rational Service Tester, Functional Tester, Performance Tester are all now at the C8 (unintelligible) which is up from the 702 level - the prior release I should say.

And on that release I guess there was a few - a fair amount of interesting things about it. Before I even jump into it I will point people to a resource where you can find additional information because we're not going to be able to cover everything here today.

But on developer works - IBM.com/developer works you'll find a product page for each tool - Functional, Service Tester, Performance Tester, Quality Manager for that matter.

And you'll see a what's new document in each of those where you can actually learn more about what's new in these things, see some screen shots, et cetera and sort of get a rundown of what's new.

So starting with Rational Service Tester for the uninitiated Rational Service Tester is our tool for validating the quality of SOA applications. And effectively what that means is this is our web service testing tool.

It is a tool that lets you - well the challenge with web services I guess at least to step back on step the idea with - the difficulty in testing web services is that for QA teams they're so interfaced. Unlike a regular application where there's, you know, some type of - maybe a browser interface or a (unintelligible) client interface there's buttons to click on, there's fields to type into, there's menu items to select.

But with the web service none of that exists. The web service is, you know, is only exposed as a function effectively where you have to know how to call that function, you have to know how to provide raw data to it. And then, you know, you get back some other form of raw data, you know, no pretty screens, no wizards.

And for QA teams this is kind of a bit of a challenge. It's not a paradigm or method with which they're familiar working. And so we got - especially with the advent of so much more SOAs - service oriented architecture development.

A lot of questions from testing's on, you know, how can I - I'm spending so much time working on these things how can I be more efficient with this process?

And so that's where Service Tester came from and maybe two or three years ago we introduced this tool. And what it does is it creates that interface on the web service and lets testers create tests against the web service.

So it actually will generate a web based interface for a web service based on, you know, what that web service does. It'll let you pump data into the web service and then it'll let you capture whatever the web service gives you back.

And the kicker, you know, what makes this so great for test teams is that the tests that are created have no code. So very much like Performance Tester if you've seen Performance Tester the actual test for a Rational Service Tester for testing a web service has no code in it - no Java, no Visual Basic - it's really just a tree view.

And the branches of the tree view are really called to the web service, return value, from the web service. Next call to the web service, next return value from the web service.

And that's it. It really simplifies it down to that level which really lets testers focus on the important part of testing web services which is the data - not the mechanics of calling them and dealing with all the background XML or XFB or any of that stuff. It really lets the tester focus on sending data and validating data.

So that's the, you know, that's kind of the overall concept of Rational Service Tester; very powerful tool. Although designed for QA a perfect tool to be used by developers, especially if you're working already in the Eclipse shell because this is an Eclipsed based tool - Eclipse being the Open Source development environment.

So you can have a testing perspective open in Eclipse to do your testing of your web services right next to your development perspective where you're actually coding your web services.

So with one mouse click you can go from a full fledged development environment where you're building your web service to a full fledged testing environment where you're testing your web service. Really nice feature - really nice benefit if you're an Eclipse day shop or you're a shop using Rational Application Developer or Rational Software Architect for development.

So then what's new? Let me tell you what's new in Rational Service Tester. The big news is we have redesigned our what's called the generic service recorder. And let me explain that a little bit.

So the idea is that, you know, the web service - market - the web service environment is changing and there's many ways to create web services. You can have them using, you know, your standard (unintelligible) and XSD type file, you can have just straight XML (unintelligible) web services. You can have web services that use message queuing software.

And all of these little nuances in the previous version of the tool had - were all supported but the way you created a test for let's say a regular web service versus an XML (unintelligible) web service was different.

And so what we've done with this version of the tool is we've created the generic service recorder and that enables you to use one single interface , test any kind of web service no matter what technology's used behind that web service.

So it's a very powerful and a very good simplification of the tool and it's called the Generic Service Recorder that enables you to do that. How does it work? You know, quite simply it's a wizard based thing where there is three - or really two steps: one where you invoke the service and one where you view the response of the service. So I guess it's really connect the service, invoke, and view the response.

Three step wizard driven process where you provide - you connect to the web service using whatever method is applicable for, you know, technology you're using. You send the data to the web service at the invoke stage, and then you view the data to view the response at the final stage.

And from that it generates a test. And the test looks the same no matter what technology you're using. The tests just show up as tree views of here's what you called and here's what you got back.

Now when you go into edit that test we've updated the test editor as well. So what you'll see is an overall effort with Rational Service Tester to simplify the process of web testing which is to make it as simple as possible.

First thing we did was change the way you record the test. The second is the way you actually edit the test. So a new test editor which really simplifies the process of working with header information such as authentication or reply to addresses for WS addressing or reliable messaging or coordination standards. So that was kind of tricky before; now it's a very simple method.

We've also put a new form based editor on the web service - on the editor I should say. And what this means is when you're now editing a test it looks

very similar to the interface you got when you were recording the test. In fact it looks identical.

And so the idea was that there really should be no difference between the process of recording the test and editing the tests. It's really all about populating data and viewing responses. So we now have essentially a single view into that with the new test editor. So that's a really positive aspect of simplifying the tool.

The last thing we did to really enhance the user experience and simplify the user experience is we've created what's called the service test persona. Now to understand this you need to understand a little bit about the architecture of Rational Service Tester.

With Rational Service Tester one of the great advantages of the tool is that it enables you to do both functional and performance testing on a web service. So you can, you know, you can create tests for web services that just run through one at a time, you know, validation the functionality of the web service.

But you can also take those web services and say I want to run 100 simultaneous calls or 1,000 or whatever the number is - 5,000 - hit the server, and not only validate that the response that comes back is correct but also measure the response time. So you're doing kind of functional and performance testing - great advantage of the tool.

Now in order to accommodate this we built the tool upon Rational Performance Tester. So really although you may or may not know this from looking at the tool behind the scenes, in the machinery a lot of the Rational

Service Tester code is shared with Rational Performance Tester - our tool for performance testing.

The feedback we got was that while this was very powerful the tools were so well integrated that it became confusing as to what was performance test functionality, what was service test functionality.

So in this release what we've done is a better delineation of service testing functionality. Now when you use Rational Service Tester everything - the only things that you see are related to service testing whether that's functional or performance. You see no more overlap of some of the rational performance Tester functionality that you used to see.

So it's just a greater separation on the user interface level. Effectively what this means is our menu items are much shorter. There's less items on the toolbar. It's a much simpler process of record, edit, playback. You know, we haven't deleted anything - we haven't removed any functionality. We just masked it to make it clearer and faster to create tests. We call that the service test persona.

Finally I think well let's talk about two last things. You know, with every release - and you'll hear Matt talk about this a lot since it's really important on the functional tester side - is support for different environments.

With web service technology evolving there's all sorts of different standards. So three of the key environment support updates for Rational Service Tester this time are Jason - support for Jason messages. So that's a - kind of a text messaging architecture for web services. So you now have access and support for that.

There is support for asynchronous communication so for web services that speak to you before you speak to them sort of thing. So for the type of web service that, you know, provides you information on a regular scheduled interval you now have the ability to test those. And then an enhancement in the security area there's now support for NTLM technology for secure web services.

So that's really the environment support. The last thing I'll say - and this one applies to Functional Tester, Performance Tester, Service Tester, is all of our tools were enhanced with an integration through Rational Quality Manager. I think we've already mentioned that.

But all of our tools now have the ability to integrate with the new Rational Quality Manager tool so that the idea being from Rational Quality Manager you can call any type of test in any one of our tools, have that tool execute the test - in fact if you're using Rational Test Lab Manager you can even have the tool installed - execute the test after that, and then gather the results so you have sort of consolidated test reporting in the tool. So that'll be common to all the tools. I'm sure we'll mention it along the way.

So that's the last thing I really wanted to mention about Rational Service Tester.

So any questions pop up along the way operator (Andrea)?

Operator: Not at this time.

Brian Bryson: Excellent.

Operator: I'm sorry we do have Philip Pinkowsky. Your line is open sir.

Philip Pinkowsky: Hi this is Phil Pinkowsky from Assurant Health.

Brian Bryson: Hey. How's it going?

Philip Pinkowsky: Good. Quality Manager's great; I've got the beta of it - love it a lot. We are extensively - we have probably ten man years worth of Rational rollout scripts. And I'm wondering if there - integration going to be available between that tool and Quality Manager.

Brian Bryson: Yes, yes, absolutely.

Philip Pinkowsky: Okay.

Brian Bryson: There is a robot integration. I'm actually surprised it's not working for you right now.

Philip Pinkowsky: Well I haven't tried it yet.

Brian Bryson: Oh okay.

Philip Pinkowsky: I'm trying to get up to speed on all the aspects of it to - so we can lay it out to our testers.

Brian Bryson: It's a significant change, you know, there's no question about that. And something that you don't want to just release it out having a little bit of a grasp on internally.

But yeah you should be able to run any type of test from Quality Manager including Functional Tester, Performance Tester, and Robot and J Unit and competitor testing tools.

Philip Pinkowsky: Great.

Brian Bryson: That's the architecture. That was the design philosophy behind it.

Philip Pinkowsky: Cool. Excellent.

Brian Bryson: Is the same Phil that's spoken at our user conference? Is that...

Philip Pinkowsky: Yes.

Brian Bryson: I thought so. I thought I recognized that name.

Philip Pinkowsky: Okay.

Brian Bryson: Spectacular. Glad to have you on board today.

Philip Pinkowsky: You bet.

Brian Bryson: Operator anything else before we jump into Mr. Matt and some Functional Tester?

Operator: If there are any other questions or comments you may press star then the number 1 on your telephone keypad.

And there are no further questions at this time sir.

Brian Bryson: Bring it on. So Matt I'll hand it over to you. Maybe we can talk a little bit about Functional Tester and then we'll finish it up with some Performance Tester after that.

Matt Holitza: That'd be great.

So for those of you don't know Rational Functional Tester's probably one of our more popular tools; has a lot of market share right now. And really Rational Functional Tester to put it simply is a tool that tests applications - automates testing of applications as users would test them on a GUI.

And so it's really designed for regression testing. And, you know, I like to point out one of the claims to fame of Functional Tester is that we really designed Functional Tester for both the novice and the expert tester.

So, you know, there's testers out there that are manual testers today; they need a tool that they can come to speed quickly on and something that's easy to use. Well Functional Tester's very easy to use. It's built using - has a lot of wizard base driven approaches. You know, when you go to record a test it gives you guidance - step by step guidance on, you know how to do different tasks like how to add something to verify that a order ID came up correctly.

And then it also gives you, you know, a wizard to say this is how you add a data pool. And for those of you who don't know a data pool is basically, you know varying the data you send to an application so you can test for different data conditions.

So, you know, Functional Tester you can have one script that will, you know, run for - you can use one script with multiple data variations and you get more time out of that test, right? You get more productivity out of that one test.

Functional Tester as are all of our tools are built on the Eclipse framework. And so what that allows is our script is not used for proprietary language like most other tools. What you get with Rational Functional Tester is the scripts generate Java or digital basic.net.

It integrates with Visual Studio so it's a plug in for Visual Studio. And what this does is it allows teams to leverage internal knowledge. So if you have developers that know Java they can, you know, help out your testers if they need to extend the scripts or, you know, add any conditional branching.

And so this really adds a lot to the tool because also testers get a lot out of it because then they're allowed to lean Java. They don't learn some derivation of Visual Basic. And, you know, it allows them to progress their career.

Rational Functional Tester also includes - as all of our tools do process guidance. So we have a built in process guidance wizard that basically is a mini version of our - of (RUP) which is just around functional testing. So it gives you step by step instructions on how to create scripts, how to add validations, how to add a new project, you know, just the basic things.

And then Rational Functional Tester is, you know, the one thing that's really nice about it is that it is a very complete in the coverage of different test environments. So - and I'll talk a little bit more about that in a second on what's new with Functional Tester.

But, you know, really a lot of teams, you know, they have varying technologies; not all teams have the luxury of building all their applications on, you know, one technology base.

So there's acquisitions, there's, you know, migrating from legacy applications that still are in place. So Rational Functional Tester allows you to test all the way from Web 2.0 all the way down to terminal base services.

So, you know, that's kind of Rational Functional Tester in a nutshell. And so let's talk about some of the new things that came out in the last release - the version eight release.

The first one I want to talk about is packaging. So prior to version eight Rational Functional Tester came with, you know, the basics - the Java, the Visual Studio testing. But we sold our other - some of our other technology support as extensions.

Well now Rational Functional Tester comes in its base license with the ability to test SAP, Siebel, and now terminal based applications as well. So really this allows teams that have those solutions to do more true end to end testing.

You know, the example I like to use is that you might have a Web 2.0 storefront and you have SAP as your ERP. And you may have a home grown solution for managing the distribution logistics. And then maybe you have a backend system (unintelligible) implementation for your financials, you know, that in the old mainframe.

Well now you can use Functional Tester to test all of those different applications, you know, even if you wanted to string together an end to end test in Functional Tester from one GUI - from one central console of Rational Functional Tester you can test all these different applications.

And, you know, the second thing is we added some environment support for different new environments, you know, the, you know, always changing

technology world for, you know, the next live version of Functional Tester came out with support for Adobe Flex 3.0, the newest version of Microsoft.net framework, newest versions of SAP and Siebel, and some of the newer environments supported (DoJo) which for those of you who don't know is an Eclipse plug in for developing Web 2.0 apps.

We also added support for Power Builder 11. So we're also looking at adding some of those more legacy applications support from Functional Tester to make teams more productive.

And then the final enhancement to Functional Tester was we added some usability improvements. So we added a new way to view your logs which is more Web 2.0 style viewer. And so this allows people to see their results in pie chart graphs, you know, nicely separates validations from the test steps. So people can see all - see what they need to see in one view.

And, you know, what this allows people to do is go in and, you know, if there was a problem, you know, they can do down and find it pretty easily in a tree view. And then if there are no problems then, you know, it's easy for them to see. And it's also easy for them to find information quickly and, you know, relay that to the developers so they can turn around their bugs faster.

You know, and Brian also talked about that Rational Functional Tester is tightly integrated with Rational Quality Manager. In fact Rational Functional Tester has a feature called key word testing which is, you know, widely known in the industry.

And but the neat thing about our implementation of key word testing is that in Functional Tester you can actually see the key words - we actually go with a

way of doing - of developing keywords that in Quality Manager you develop key words for manual tests.

So you can chunk - you can, you know, divide your manual tests into smaller chunks called key words. You know, the simplest example is log in. So you have these three steps that take to log into an application that you want to reuse across multiple manual tests.

So you can create these three steps - this three step keyword and share it amongst all of your other manual tests that need to log in. And the neat thing about this is that it's basically object orienting your manual test so that if a change is made to the sequence of log in, you know, if you had a different, you know, drop down box in your log in page and you change that in that manual step it propagates down to all the manual scripts that use that step - that use that key word.

And but the neatest thing about this is that when - what if you wanted to automate those? So Functional Tester is integrated with Quality Manager so that the users of Rational Functional Tester can see all those manual keywords. And they can go in and automate those. And once they're automated you can have the idea of having a partially manual and partially automated script in Quality Manager.

And then what this allows is for teams to gradually adopt, you know, automation and it allows another way to organize your scripts which is kind of nice because it's, you know, one level of abstraction above the script level. So, you know, one of the neater features of Functional Tester.

So, you know, again, you know, any questions out there please hit star 1.

Operator any questions in the queue?

Operator: You have a question from the line of David Daye.

Matt Holitza: Great.

Philip Daye: Actually it's Philip Daye from Citrix. And speaking about the key words currently that only works if the automated tests are done under Java. Is it going to be possible to do those under Visual Studio?

Matt Holitza: Yeah Brian and I had that discussion before. And we were in the understanding that the dot net implementation should have the same exact view of key words - at least in the latest versions that it does in the Java view.

Philip Daye: It doesn't in 702.

Matt Holitza: Okay. I'm almost positive (unintelligible), you know, but I can definitely follow up with you on that and, you know, get the answer for that.

Philip Daye: That would be great. Thanks.

Matt Holitza: Yep.

Operator: Again if you would like to ask a question or make a comment you may do so by pressing star then the number 1 on your telephone keypad.

And sir there are no further questions at this time.

Brian Bryson: Well we are coming up on time. I want to talk about - two things to talk about before we call it a day here. I want to give a quick update on what's up with

Rational Performance Tester. And then we've had a question come in from the ask us now email address so that's another way you can get in touch with us besides the star 1 is just send an email to ask us now - A S K U S N O W at U S dot I B M dot.com.

One question - so let's talk about Performance Tester and let's get to the question because the question's related to Quality Manager. So Performance Tester is our performance testing tool for throwing a large load of traffic at an application - a web based application, a Citrix application, an SAP application, Oracle, Siebel - server based application I might as well say in general.

It went through a major release last - oh February, March, April - I forget the actual timeframe. So not much development time between that and the release in October so not a huge amount of changes in Rational Performance Tester.

A couple of things - probably three things, four things worthy of note. One is we've cleaned up the Performance Tester interface. So this is in parallel with what we did with Rational Service Tester is we've really simplified the interface for Rational Performance Tester.

Now if you're a current user and like the interface we've left it there. You have the option - so now when you install you have two ways to launch performance tester. One is is the classic mode and one is the simplified mode.

And so you choose but for new users we're recommending going with the simplified mode. It's a streamlined interface with fewer menu options and less clutter.

Second thing - this kind of falls under the environment support - support for (unintelligible) authentication so a security method or algorithm now supported with Rational Performance Tester.

As with all of our other tools - excuse me - as with all of our tools Performance Tester underwent an integration with Quality Manager. So Quality Manager can launch Performance Tester, run the results, view the results of Performance Tester right in Quality Manager so that integration is there.

And then the last thing is the - excuse me - the scalability aspect. So for very large tests many people run performance tests over days. So, you know, five and six day performance test is not uncommon.

That accumulates a lot of data and we've changed the architecture of how that data is handled to really better support the transfer of that data from our old machine back to the central locations, to simplify and accelerate really that process. It's a much cleaner process, a much faster transfer and analysis of all that data.

So that's what's up with Rational Performance Tester. And let me ask one last time for questions before I go and answer the one that came in through the ask us now.

So operator is there anything in the queue for questions live?

Operator: At this time if you'd like to ask a question you may do so by pressing star then the number 1 on your telephone keypad.

And sir there are no questions at this time.

Brian Bryson: Okay. We'll give you a few seconds to get any last questions in and we'll just address this one that came in through the ask us now interface.

So the question is how does Rational Quality Manager relate to CQTM? Obviously someone who's using CQTM based on the acronym. CQTM is Clear Quest Test Manager, our previous generation - our previous technology test management.

So effectively the answer is here that Rational Quality Manager is our new strategic direction in test management. If you're a new customer, never used any of our products before, and came to us and said, "I need a solution to help me control my testing process, manage my test plan, link my test plan to requirements to defects," we would go and show you and talk to you about Rational Quality Manager.

However we have a significant install base of Rational Clear Quest Test Manager customers and that's for people over the last three or four years that have been using the Clear Quest Test Manager tool.

So it is a completely new product. However - I should say Quality Manager's a completely new product. However it has the ability to import data from Clear Quest Test Manager and as well as the previous generation Rational Test Manager.

So if you're an existing user of Test Manager or Clear Quest Test Manager and you want to take advantage of the new features in Rational Quality Manager there's an import functionality that will take all the data that you have in Clear Quest Test Manager, map it in and suck it in to Rational Quality

Manager so that you can leverage that data and move forward, you know, on day one in the new technology if you so choose.

If you're happy with Quality Manager - I mean with Clear Quest Test Manager by all means keep running with it. It's not end of life. There's no such plans for that. So it's, you know, you're welcome to keep going and using it. We know many teams that are quite tied to it and are experiencing a great success with it.

So all's good there, you know, status quo. But if you want to jump on board with the new technology it's a very simple task of import.

So with that we'll do one last call for questions and call it a day if not. Operator is there anything sitting in the queue?

Operator: No sir not at this time there are no questions.

Brian Bryson: Excellent. Well with that I will say thank you to my partner Mr. Holitza for the Test Lab Manager overview and the Functional Tester technical information. Thank you very much.

And Angelique I will hand it back to you.

Angelique Matheny: Well thank you very much Brian and Matt of course; always a valuable session when you guys are involved.

If you would like to listen to this conference again or share it with your colleagues this will be made available for replay and MP3 format in about a week or so on the Rational Talk To You site www.ibm.com/rational/talks. Our previous teleconferences are available there as well.

So Brian and Matt thanks for talking to us about quality in action using Rational Quality Manager with Functional Performance and web service testing products. We appreciate you taking time out of your busy day to be with us.

We would also like to thank you 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.

Operator: Thank you ladies and gentlemen. This concludes today's conference call. You may now disconnect.

END