

MATHENY: Hello, and welcome to this Rational Talks to You Podcast, IBM Rational Quality Management Portfolio update: What's new in 7.0.2.

The IBM Rational Quality Management Portfolio recently released updates to two core products: IBM Rational Performance Tester Version 7.0.2 and IBM Rational Tester for SOA Quality Version 7.0.2.

In this podcast, we will have a discussion with IBM Rational technology evangelist Brian Bryson about the new features and advancements that will help testers in their performance testing and SOA and Web services testing efforts. Brian, welcome to the podcast.

BRYSON: Thank you, Angelique, always a pleasure.

MATHENY: We've really got a lot to cover today, so let's get down to business and talk about the reason we have you here today. Tell us what's going on in the world of quality management.

BRYSON: Sure. Let's get right to it. Well, you covered a little bit of it in your introduction. The reason I'm here today is to talk to you about our latest 7.0.2 release of two of our core products: first, IBM Rational Performance Tester, and the other, as you mentioned, IBM

Rational Tester for SOA Quality.

What's happened is that on April 29th, we've released the latest versions of these tools and introduced some new technologies to essentially expand our platform support, increase our enterprise readiness, and focus on and improve our overall usability of these tools.

MATHENY: Okay, before you jump into the details, can you talk just for a minute or two about these two tools and how they fit into our strategy?

BRYSON: Sure. As you know, we have an entire suite of quality management tools. Our philosophical belief at IBM Rational is that quality management is more than just testing. It's bigger than testing; it's a continuous focus on quality throughout the software development lifecycle.

Now, the objective or the reason for this continual focus on quality is to get to a state where we have a governed software delivery -- or, put another way, you're essentially at all times aware of the quality status of your product and process and have the information necessary to make the right business decisions about your software delivery process.

I mean, I guess what we're really trying to do is get you to be able to answer the question, are we ready to release --

the ultimate question in software delivery -- to be able to answer that question at any point.

Now, to get to that governed state you need to continuous...continually focus on quality at all stages. And that means thinking about quality in your individual role, maybe as a business analyst, or as a developer, or obviously as a tester. So think about it from that perspective, and also communicating with the entire test team.

Now, in order to make that continuous process possible, you need automation. You know, quite frankly, there's an overhead associated with a collaborative quality management.

And the only way to really truly implement a continuous process is by powering it with automation. And that means kind of two things. It really means task automation and team automation.

Task automation is giving the developers, the testers, the business analysts tools that help those individuals do their jobs. And then for the team automation, that means providing tools that help those individuals collaborate with the larger software team for a team-based software delivery.

So effectively what we believe can happen is that with a continuous process where everyone is thinking about quality

and sharing data, and where that process is powered by automation, you get to a state where you're governing your software delivery.

You have all the data about the quality of your application and can make informed decisions about resources and feature implementations sort of based on that information. So it's a little fluffy, you know, admittedly. But that's our larger strategy.

The two tools we're talking about today are really part of our core testing tools and they would sort of fit into that task automation category. So they're tools that a tester would use to help them ensure the quality of their SOA and Web service applications, to ensure the scalability of the overall systems.

First, I guess, Rational Tester for SOA Quality -- to talk a little bit about the tools for a second -- is our tool for functional and performance testing of Web services.

With its wizard-driven interface, you can quickly and easily create tests for GUI-less or head-less Web services. And then you can play back those tests with multiple data sets to really fully validate the functionality and performance of your SOA application.

That's tester, for SOA quality performance tester on the other hand is a pure performance testing tool. With it, you can create tests that validate the scalability of your Web, Citrix or ERP -- so, like your SAP, and Siebel, and PeopleSoft, Oracle type systems.

Essentially with performance tester, you record yourself performing transactions on the system and then you load the server by sending multiple simultaneous replicas of that transaction to the server and measure the response time under load. You know, the idea, of course, is to identify the presence and cause of your bottlenecks, ideally before your customers do.

MATHENY: Right. Exactly. So, Brian, what's new with these tools?

BRYSON: Well, quite frankly, a fair amount. In fact, there's probably more than we can cover here. But let me give you the highlight reel today. That's really what I'd like to get across.

So our updates for this release kind of center around three areas or three themes: the expanded platform support, increased enterprise readiness, and just kind of a general improved overall usability of the tools that we try to kind of update in every release.

So let me start with the platform support. And essentially you know what we're talking about here is that we can now test more. For Performance Tester, in the ERP space, so the SAP Siebel type applications, we've updated our Siebel support to Siebel 7.9 and 8.0. And for SAP we've added a very important JCo batch input process support.

We've also added TCP socket support. And that's kind of a really interesting one, as it means we now have a way to test pretty much any type of system that communicates using TCP/IP, which is essentially the majority of all systems out there.

Some will be easier than others, but this is a very generic, very general mechanism, and it gives us support for some of the most common protocols out there, FTP and LDAP, e-mail protocols such as POP and SMTP, as well as any custom protocol testers you might encounter.

You know, if software teams are building in-house protocols, this gives them a general mechanism to do performance testing on those systems. That's on the performance testing side; for SOA tester or Tester for SOA quality, we've updated to SOAP 1.2 and we've added the ability to test systems that use two things, asynchronous Web service calls -- so kind of a new technology there -- and the ability to

test Web services that leverage MTLM attachments.

So I'm sure that all seems like a bit of an acronym soup. But you know, to testers, that's all critical information that essentially means...you know, it essentially means we can now test more stuff.

MATHENY: It may be alphabet soup but all very good stuff. Now, that covers the expanded environments; what about the scalability and usability?

BRYSON: Right. Scalability, a second thing, is really that amounts to a design improvement for Rational Performance Tester. With performance testing, it's a common practice to perform what are called stress tests. Stress tests are when you run your performance test for extended periods of times, like hours, days and even sometimes weeks.

And the thing is when you do that, and you know, when you're running tests for days and weeks at a time, you accumulate a lot of data. It's not unheard of for a team to emulate a load of a couple thousand users, making maybe, I don't know, 20 or 30 transactions each per minute. And as you know if you do the math on that, if you do 20 transactions per minute, that works out to about three and a half million transactions per hour, or 85 million or so transactions per day.

So measuring the response times for all that data can be quite tricky, just simply due to the massive amounts of data being captured, collected and analyzed as well.

Architecturally, what we've done is we've optimized our tool to be able to manage these large volumes of data.

We've made it possible for Rational Performance Tester to collect and analyze the data so users can identify the presence and cause of performance bottlenecks, even if they're buried deep in a week-long performance test.

MATHENY: This all sounds like finding a needle in a haystack.

BRYSON: That's exactly right. You're learning. You know, when you have a test that runs for a week and the performance problem occurs three days in, three days and two hours in, that's exactly what we're trying to do. We're going to make a tester out of you yet, Angelique.

MATHENY: And that would be great. That would be great.
 Okay, so I get all that. How about usability? Talk about that.

BRYSON: Right. Usability. So, you know, we do this in every release, and across the board we've done little things

to make both...both of the products more intuitive and just generally simpler to use.

You know, we've added some wizards, as an example, to Tester for SOA Quality, which make it simpler to record. And actually we've added some new reports which make it easier to view your test results and determine the quality of your SOA applications.

So we're really trying to help out at sort of the beginning and the end of the process there, in the early stages when you're recording and at the end when you're trying to find your issues.

For Performance Tester, we've added some new functionality to make it easier to debug your tests. You know, we don't talk about it too often, but when things don't go right with the performance test, it can be really tedious to find out where a problem occurred, especially if you think back to just a second ago where you're running these tests for days and weeks at a time.

So what we've done is we've added a viewer which lets you see the responses sent by the server exactly as you would if you were a real user with a browser. And we've also simplified our logs so it's easier to understand and find server-related problems. We're really going to save a lot

of time with that one.

And then, finally, we've added a nifty new feature called Smart Load. The idea with Smart Load is that you can preprogram as part of your schedule different user loads. You know, for example, you can say that for the first 10 minutes of your test you want to run with 100 users. And then for the next 20 minutes you want to run with 500 users emulating load.

And then for the 20 minutes after that, you want 1,000 users. And so what you essentially get is a very powerful feature that really speeds the process of finding your system breaking point.

MATHENY: Well, Brian, that's really cool. I know it sounds like a geeky statement, but that's great. And that's a lot. Is that it?

BRYSON: Yes, well, actually, you know, it is very cool stuff. And it does make performance testing better. You know, there's a lot more. These usability things, small little things that users of the tools have asked for here and there at client sites. And there is a lot more. But essentially, you know, those are kind of the highlights. I mean, I can probably keep you here all day here if I walked you through everything that we've snuck into this release.

MATHENY: I understand. And we do have a closing question: how do I go about learning more?

BRYSON: You know, perfect. You know, since we don't have time to cover everything here, let me give you three suggestions on how to get some more information. The first thing I'll do is I'll give you a link you can put on this podcast page where we have a document on our Web site that lists these key features that I've been talking about.

For people listening, it's kind of a long URL, but if you go to, just simply go to ibm.com/rational and then click on the quality management icon, you'll find the updated document there.

Then if you want to actually see some of the new features in action, we've recorded some new product demonstration videos where you can actually see the tools in action.

I checked this morning, the Performance Tester demo is live now. And then the Tester for SOA Quality demo should be live any day now. So, again, some pretty long URLs to get directly to them. But if you go to ibm.com/rational in the quality management section, all the content is there.

That's the...you know, those are kind of the quick and easy ways to do it.

The last thing I'll suggest is that if you head to ibm.com/developerworks, what you can do there is actually download trial versions of both Performance Tester and Tester for SOA Quality.

And it's really the best way to check this stuff out. The download is a 30-day trial, and that gives you the ability to try it out in your environment against your applications.

The proof is...as they say, the proof is in the pudding on these things.

MATHENY: That's right. Brian, this is a really great update, I wish we had more time. So, I thank you for taking the time out you did today to discuss IBM Rational Quality Management Portfolio, what's new in 7.0.2. We really appreciate it.

BRYSON: Oh, no problem. Always a pleasure to have the opportunity to broadcast the latest and greatest when it comes to all of our quality management portfolio.

MATHENY: That was Brian Bryson, IBM Rational technology evangelist. If you're interested in more information regarding today's topic, Brian mentioned this URL, www.ibm.com/rational, under the quality management section.

If you're interested in more podcasts like this one, check out the Rational Talks to You podcast page at www.ibm.com/rational/podcast. This has been an IBM Rational Podcast. I'm Angelique Matheny. Keep tuning in as Rational talks to you.

[END OF SEGMENT]