

IBM Podcast

[MUSIC]

MATHENY: Welcome to this IBM podcast, Software Development Accelerators, Automation in Coding, Build and Test During the Development Phase. I'm Angelina Matheny with IBM. Have you ever been involved in a software project and thought, there must be a better way to do this?

Software delivery is a challenging proposition, but there are steps that can be taken to reduce costs and risk and still deliver applications and services on time and on budget. In this series, Five Steps To Reduce Your Cost of Quality, we'll examine five strategies software delivery teams can take to ensure application quality while maintaining a focus on cost and customer expectations.

The third installment is focused on how developers can use neat tips and tricks to accelerate code development without compromising on quality. In fact, using automation and code testing and profiling tools that ship with Rational Software Delivery Solutions can help dramatically improve the quality of the first code that goes through the door.

Automating the builds and repetitive processes for complex deployments can save time and allow development and test

teams to be more efficient and work with software as it's meant to be deployed. The discussion will include some easy methods to use during software development. And joining me for this discussion, it's Adeel Omer, Marketing Manager, Architecture Design and Construction. Thanks for joining us, Adeel.

OMER: Thank you, Angelique.

MATHENY: Let's jump right in, I have a few questions for you today. So, Adeel, where does automation come into play in software development?

OMER: Okay, good question. Automation is actually quite an overloaded term: different people use it to mean different things. And within software development, automation really comes into play in a number of different phases. It could mean the conversion of software models to code.

It could mean the conversion, the automated conversion of various types of models into other types of models -- that's the model driven development approach. Going the next step forward, it could mean you could take the code and deployment information and have it automatically generated from a more visual wizard or from annotations that create deployment information.

It could also mean that whenever you write your application and your code there is automatically some testing to adherence standards that takes place. There's code quality testing that happens in the background automatically. That's also a level of automation.

You know, we can mean the automation of test creation. So, based on your models or your embedded comments within your code you can generate test cases automatically based on the scan of the code. You can also, obviously, automate the execution of your test where as I'll talk about a few minutes, you can take a functional tester or performance testing and have those capabilities automated.

You can have the automation of your build. So, we're all familiar with nightly builds, we're all familiar with the ability of compiling code and getting applications together to where they can be run by your test teams or by your development teams.

And then obviously, there is the deployment automation which is to say, all right, now I've got my application, this huge enterprise application, how do I deploy it on this really complex IT infrastructure that I have instead of having a person from machine to machine installing various resources on it, how do I automate that?

So, there's different kinds of automation when we use that term automation. But although they might be different types of automation, the goal for all these automations are very much aligned. The end goal is obviously to produce high-quality deliverables, to reduce the number of mistakes that we have, to meet the original requirements that were set out when the project was undertaken.

The other goal is to reduce manual tasks and human involvement whether it's actually writing code or creating test cases, or as I said, walking from machine to machine re-creating test environments and reproducing bugs. We want to reduce all these menial tasks because humans are really so much better at creative problem-solving and machines are better at repetitive tasks.

And consequently, you know, you reduce the number of inadvertent errors when you pull humans out of the equation.

And ultimately all of these lead to a reduction in time to market of a better quality application. So, that really is the underlying business value of automation which is, it reduces the time to market of higher quality deliverables.

MATHENY: You mentioned a number of different opportunities for automation. What role does IBM Rational solutions play in each of those spaces, and can you give an example of how they help automate and bring about the

business benefits you outlined?

OMER: Sure. So, Rational has solutions for all the activities that are involved within software development. The very first that I talked about was the conversion of models to code and then model driven development. So, we obviously have Rational Software Architect and Rational Rhapsody in those spaces.

Rational Software Architect is used more for the software modeling business as well as the enterprise software development space, whereas Rhapsody is the modeling solution that we use for the system space. And both have the ability to take specific types of models and convert them to code that will either run on an application server or it will run on an embedded device or a complex system.

So, along with that you have patterns for custom development, you have existing patterns, you have existing transformations, but you can define your own based on the needs of your specific industry or your specific application.

Moving on from there, you have the generation of code and deployment information based on wizards and annotations, as I said. Those are specific capabilities that are available within the Rational Application Developer, the IDE that we

provide, as well as the Rational Software Architect product which also has some code development capabilities.

With this allows you to do is instead of specifying information around deployment descriptors or providing information that, you know, generating [stubs] for your application, creating the very basis of how your application classes are going to talk to each other, that information can be automatically generated.

And so all that's left for a developer to write is the business logic, and ideally that is all a developer should be asked to do because that is the creative problem solving aspect of creating an application. While this is being done, there is obviously static and dynamic testing of the code. There's a number of aspects to that, right?

One of them is, am I following best practices? Within the IDE, Rational Application Developer as well as specific products such as Rational Software Analyzer as well as our Appscan tools for Web Application Security.

There's a number of different rules and there's a number of various ways you can scan your code and provide feedback in terms of hey, you're not following these best practices or this is thought to be a bad practice or this is a vulnerability within your code. Those are things that can

happen automatically while you are creating your application or they can be included in your build process.

That level of automation provides you some degree of comfort to where if you get a report every time you do a build that says, your code looks good, or here is a glaring issue within your code, that's something that doesn't take a lot to implement, but it gives you a pointer towards what might be going on or better degree of comfort with either the security or the quality of your application.

Going forward, you have the automation of test creation based on documents, based on embedded comments or code models and that's a capability that we have in Rational Software Architect and its integration with Rational Quality Manager to where you can create or specify where you would like to have tests created.

This is a really interesting aspect, because when you are creating the overall architecture of your solution that is one of the places that you can specify the minimal requirements or the acceptance criteria.

And having that integrated with your quality management processes can be really, really useful in making sure that your product does meet the final needs that the architects always have in mind because they are feeding directly from

the requirements from the customer. So, RSA and RQM, Rational Software Architect and Quality Manager come into play there.

Then you have the automation of test execution. You know, you have performance tests, you have functional tests, Rational Functional Tester and Performance Tester obviously come to play there. You know, they tie in with the quality manager aspect of it all as well. So, you have quality management on the top and then you have the actual execution in specific ways that's being taken care of and automated by Functional Tester and Performance Tester.

And finally, where we have huge and really very, very strong capabilities in terms of automation is the automation of builds and the automation of deployments. And this is where the Rational Build Forge and the Rational Automation Framework for WebSphere come to play. Build Forge is obviously more than just a build tool, it's a whole engine that can consume workflows, it can consume scripts that can help you automate your build.

But the Rational Automation Framework for WebSphere is a specific solution for WebSphere Application Server and WebSphere portal customers. And customers who deal with WAS and Portal know that their targeted environments are extremely complicated.

You can have multiple boxes with multiple instances of the servers. They connect into various databases, they connect into different types of data sources, they connect into authentication services. It's a whole ecosystem. And so, as a tester or as the developer, you don't want to concern yourself with having to deploy your complex application at every iteration and waste time doing that.

And that's where Rational Automation Framework for WebSphere and Rational Build Forge really come to play and cut down the amount of time, they give you immense power over how your builds will be deployed.

And really the whole purpose of it all is for you to configure all of it just once and have your test teams and your development teams not have to worry about the logistics of deployment, but just work with problem recreation and passing that back to the development team to fix those issues.

So, those are the various places. There's obviously more, but those are the main places where automation can really, really help you out. You know, rational solution certainly cover the whole gamut of those automations.

MATHENY: It sure does sound like an end-to-end solution

for automation. Adeel, my last question today, what's the next step for our listeners to listen to any of these solutions in action or even try them out?

OMER: Okay, so, if you want to see these in action, obviously, that product pages for each of these solutions are a great way to get to them. You can just search for the individual product by name. Our developerWorks sites are the repository for a lot of technical information, for a lot of tutorials, for demos of each of these solutions in action. You can also find eKits, videos, tutorials that will demonstrate the capabilities and the integrations that I talked about here.

And then finally, if you like to get your hands dirty, in most cases you'll be able to find either an online hosted trial or a downloadable trial that you can use to actually get your hands dirty and play around with these solutions.

So, it's pretty much the same thing, but if you can't find what you're looking for, hit up the development teams on the forums on DeveloperWorks and they'll be able to get you in touch.

MATHENY: Adeel, as always, thank you so much for sharing your time today to discuss this podcast, Software Development Accelerators, Automation in Coding, Build and

Test During the Development Phase. We really appreciate it.

OMER: Thank you, Angelique.

MATHENY: We're also looking forward to the next step in the Five Steps to Reduce Your Cost of Quality Series. That was Rational's Adeel Omer, Marketing Manager, Architecture, Design and Construction. To share this podcast with your colleagues, or if you're interested in more podcasts like this one, check out the Rational Talks To You podcasts page at www.ibm.com/rational/podcasts.

We'll include a link to the Rational Software Integrated Development Environment Tools site, so check it out today. This has been an IBM podcast, I'm Angelique Matheny. Thanks for listening. Keep tuning in as Rational Talks To You.

IBM Podcast

[MUSIC] [END OF SEGMENT]