IBM

Moderator: Angelique Matheny
April 24, 2008
12:00 pm CT

Operator: Good afternoon. My name is Amanda and I will be your conference operator today. 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 would like to ask a question during this time, simply press star then the number 1 on your telephone keypad. If you would like to withdraw your question, press the pound key. Thank you. Ms. Matheny, you may begin your conference.

Angelique Matheny: Thank you very much. Hello everyone and welcome to this Rational Talks to You teleconference. Calling all Clear Case power users and those who would like to be. I am Angelique Matheny with IBM Rational and I'll be your host for today's call.

Joining us today is Paul Boustany, Change and Release Management Marketing Engineer and Mark Krasovich, Senior IT Specialist. Paul started in technical support where he assisted customer resolving complex problems with a focus on unified change management.

Paul was a product manager for UCM where he continued to work with customer who were adopting Rational Clear Case. With over eight years experience working with Rational Clear Case, Paul has worked as a technical evangelist for Rational and has given talks and presentations around the world on how to use Rational Clear Case as a power user.

Mark has been an IT specialist for four years, joining IBM through the Rational acquisition. Mark spent two years in Clear Case support and has also worked with various start-up companies as a systems administrator and support engineer.

In today's teleconference, Paul and Mark will be discussing how to become a Rational Clear Case power user and the tips and tricks for using Rational Clear Case and Agile.

Now you won't find any slides for this teleconference. These calls are really for you. We'll open up the lines and you'll get a chance to ask Paul and Mark your questions during the Q&A and anytime throughout today's teleconference, just press star 1 and the operator will break in and you'll get your questions answered right away.

We've already had some really good questions submitted. If you'd like to submit questions to our panelists after the teleconference, please e-mail us at askusnow@us.ibm.com. That's askusnow@us.ibm.com. Just put the title of this teleconference in the subject line.

Well I think you've heard enough from me so let's get started. Paul, take it away.

Paul Boustany: Thanks Angelique. So, as Angelique mentioned, this - the title of our talk today is a Rational Talks to You Teleconference. Well I actually think we'd rather call it you guys talk to us, because we'd much rather hear questions from the audience then just preach to you about all the great things about Rational Clear Case.

So, we're going to keep it very informal today. If you have questions at any time, please feel free to hit the star 1 button on your phone, and we've instructed the operator to break right in with your question, so don't be shy at all, feel free to join in the conversation.

We will start with a couple of tips and tricks or what we have put together as a top five implementation methods or top five tips to use Rational Clear Case as a power user, and so Mark's going to start us off with number 1. We're going to talk about unified change management or UCM. Take it away, Mark.

Mark Krasovich: Thanks Paul. This is Mark Krasovich and thanks Angelique for the introduction. What we're going to start off with today is some questions about UCM and I don't know how many of you are using UCM or thinking about going to UCM is something that we've seen growing in the industry and with the customers certainly that I work with, more and more of them are either converting from their base Clear Case environments to UCM or simply starting with UCM in the first place.

This is true of customers that are small from - I work with one particular customer who's got a five developer shop. They're pretty small, but they use a single-stream UCM in a pretty effective manner with Clear Quest.

All the way up to larger customers that have hundreds and sometimes more than that developers at varying sites and huge multi-sites and are pretty

successful with that. So one of the questions that we get quite a bit, we come into customer situations and people have various integrations working with Clear Case.

One of those things is they have maybe a Bugzilla or (Amantest) or something like that integration and one of the things that we like to talk about with them is why the Clear Quest integration is important and maybe why it might be a little more efficient and a little more productive than what customers are currently using.

A couple of the things that we like to point out are things like traceability, the ability to trace defects, enhancements from creation to release, and this goes from not only the defects side but when you start getting into the full traceability of the development life cycle and you talk about requirements and testing and build and release, we do cover that entire span.

Clear Quest acts as kind of the hub or the central data repository in a way to connect all of those, the spare tools, and it really gives you the ability for developers, for project managers and testers, whoever is interested in the project, to gain visibility into various projects at various levels of granularity.

This is done through a lot of the reporting capability, real-time data related to the projects for things such as trend analysis, to discover which developers are working on which particular projects and defects and perhaps some resource allocation types of issues.

It gives you the ability to analyze current project data, past data, as well as plan for the future, and try and determine what's happening, how it's going to affect your bottom line or your release schedule moving forward.

Another thing that we really like to emphasize is process maturity. One of the things that we typically find in different environments is while there may be a loose integration or a manual integration if you will, between Clear Case and certain types of defect or bug tracking tools, often times there is a lack of automation and that slows that entire process.

And there's a lot of yelling over the walls or handing spreadsheets back and forth, and there's data and bits of data that get lost in that. I have one particular customer who has an integration with, and it's actually a fairly sophisticated integration with a third party tool.

However, they have to maintain that third-party tool in four different sites in four different ways. They don't have a consistent and repeatable process amongst those four different sites because of some of these maintainability issues with the third-party application.

So they aren't able to get as granular a level of detail as they would like, even though they have a fairly sophisticated integration. Clear Quest is something that they're interested in purchasing and working with us to integrate into their Clear Case environment because it does give them that next level of sophistication in their process.

And it's something they're interested in because they can't scale the environment they way they have it right now, so this is something that is really becoming a large movement within the industry that we're seeing and that is that companies are, in order to maintain their marketplace, their place in the marketplace and their market share, they really have to mature that process.

Because other companies are doing it and you're competition is doing it, so process maturity becomes a pretty important piece. I'd like to open it up to any questions, are there any questions at this time?

Paul Boustany: And remember everyone, you can just press star 1 on your telephone and that will place you into the queue and the operator will break in. As we wait for that, I wanted to bring-up a couple of things that I was thinking about as Mark was going on about unified change management and especially with the reasons for using Clear Case and Clear Quest together because I get a lot of questions from customers that I meet at conferences or when I'm doing presentations or Webcasts like the ones we're doing today.

A lot of people want to know if they can use Clear Case and Clear Quest with other tools like Mark was saying, Clear Case with Bugzilla or some people like to use Clear Quest but maybe want to use CVS for their source code repository.

And we certainly want to - we don't want to discourage people from doing that if that's their preferred method of implementation. They may have reasons for doing that but it's important to remember that these two products were developed with the idea of integrating them together and as such, they have a very tight fit and especially when customers are using the unified change management process, the tie-in between the versions of the source code that you're changing in Clear Case with the defects and the requests for enhancements that you're creating in Clear Quest.

Operator: Your first question comes from the line of Kathy Kopczynski.

Paul Boustany: Great. Yeah, go ahead.

Kathy Kopczynski: Hi, what I'm getting to is the UCM question regarding rebase activities. We have our development streams with multiple activities going on at a time, as opposed to each stream being effectively one defect or one enhancement, and so we are often doing rebases.

When we go to deliver the next activity, we don't want to be delivering the entire rebase because there are dependencies on other activities and you get into this cascading event.

We move the versions in question from the rebase activity to the activity that needs to be delivered, so that we're only delivering what needs to be delivered.

Eventually our rebase activity becomes empty, because we've moved them all out into other areas, other activities to be delivered. Is there any harm in deleting those empty rebase activities?

Paul Boustany: Yeah, that's a great question, actually. So, we- these activities that you're talking about. They're the automatically generated ones that get created from deliver or rebase activity, correct?

Kathy Kopczynski: That is correct.

Paul Boustany: Okay, so while you could probably delete an empty activity and there's a check in the program that will prevent you from deleting an activity that has versions in it, in unified change management, what we tend to encourage people to do is rather than to actually delete something or completely remove it, there's a clear tool command and also a graphical equivalent that allows you to lock the activity with the obsolete switch.

Kathy Kopczynski: The problem with that is that when you go to deliver from that stream, you might have 10 or 15 rebase activities you have to go clear the little check box from, and so it just becomes …

Paul Boustany: It can become, right. There can be a lot of them in there and so all of a sudden, you've got this giant list.

Kathy Kopczynski: Exactly. Now if we had a clear all check boxes button, that would take care of the problem.

Paul Boustany: I think I've heard that request once before. That's a good point. Certainly, so again, the tool will actually prevent you from removing anything that has versions in it, so again, there's not going to be anything that will prevent you from doing it.

It's not something that we necessarily recommend …

Kathy Kopczynski: But is there any harm?

Paul Boustany: Not that I can think of just off the top of my head. Mark, can you …

Mark Krasovich: Well, not that I can think of. If the activity is empty, it's something that we'd have to play around with a little bit to be sure. I don't know of anything that would prevent further rebase activity or delivery activity. I shouldn't say activities.

Further rebases or deliveries based on an empty activity, and I can't see any problem, though I can't completely commit to saying yes, it's okay, with that, yeah.

Kathy Kopczynski: Well, we've been doing it without any problems so far, but we're just wanting to be sure.

Paul Boustany: So I think again, what we're trying to be as cautious as possible just because we don't want to cause any problems down the road, but thinking about it for a few more seconds here.

If you have an empty activity and it's one of these rebase or even deliver activities, there is the potential for a problem would be where the activity is getting included in a baseline that's being created automatically, so with the rebase action or with the deliver action, there's some baselines that get created that are used internally, basically.

Kathy Kopczynski: And that presupposes you can never delete any activity, even when created in error, if it's in part of a baseline.

Mark Krasovich: Yeah it would. What about, are you using Clear Quest as well.

Kathy Kopczynski: We are not integrated with Clear Quest, no.

Mark Krasovich: Okay, that's actually a good thing.

Paul Boustany: That is actually a good thing in this case because we wouldn't - because there is a tie-in between the activity and the defect. So what I was thinking, just to finish that thought quickly, so long as the activity is completely empty, so there's no versions within them.

They've been moved to other activities, it's totally empty. I - my short answer is that it shouldn't be a problem because removing the activity will also remove it from the internal baseline that gets created.

I'm remembering that there was maybe some defects related to reporting in cases where activities were deleted but I don't recall that there were any severity issues. I would say that's your best bet to be 100% sure is to follow-up with the Rational technical support.

Mark Krasovich: Yeah, that problem Paul is probably due to the integration with Clear Quest.

Paul Boustany: Right.

Mark Krasovich: I'll look it up real quick in the internal database and see if I come up with anything.

Paul Boustany: You know what you can do as well and I think if you'd like to get maybe a bit of follow-up from us that we could pose to the whole audience, if you could put your question in an e-mail and send it to the alias that Angelique mentioned.

Kathy Kopczynski: Okay, will do.

Paul Boustany: And we can do a bit more research for you.

Mark Krasovich: Yeah, I think that's a good question.

Angelique Matheny: That was askusnow@us.ibm.com.

Kathy Kopczynski: Thank you.

Mark Krasovich: And are you - never mind.

Paul Boustany: That was a great question and an excellent way to kick-off the call. We certainly would like to encourage people that have technical questions like that or if they have process questions or anything really about the tools that they'd like to ask us.

If we can't get you the answer, again, sending an e-mail to the alias, we can do a bit more research for you and get back to you.

Mark Krasovich: Yeah, that's a lot more fun than us reading a bunch of stuff to you, which we can certainly do.

Paul Boustany: So as we wait for any more questions to queue up, I did want to just finish my thought as what Mark was talking about …

Operator: Your next question comes from the line of Remal Alshammari.

Paul Boustany: Okay, go ahead. Hello? Yes, go ahead.

Remal Alshammari: Hey, hey. I was actually - I have first question was, I got in a little bit later, after the conference started, but the first one is, do we have a work class for this particular session here?

Paul Boustany: We do not.

Remal Alshammari: Okay. Is just a teleconference, correct?

Paul Boustany: Correct.

Mark Krasovich: Correct.

Remal Alshammari: And my second question is, what is the best branching way for example, for a project that has more - it is pretty much the same configuration for the control end but each project has different environments, and …

Paul Boustany: Are you trying to reuse some of the components?

Remal Alshammari: It is, yes, we are using them but for different projects, but the thing is the control is the only thing is being reversion, like the actual source, is being version in the same component all the time but there is different projects using that particular component.

My question here comes because I am trying to set-up a build automation for those projects and for streams to be configured correctly and using the same component, I'm not sure what kind of stream methodology should I be using for that, if that makes any sense.

Paul Boustany: That's a good question. As I'm thinking about this, I'm wondering if you could define the environment a bit or even the structure of your VOBs. Do you have multiple, single …

Remal Alshammari: Yes, yes.

Paul Boustany: So and are these …

Remal Alshammari: Some of our component VOBs.

Paul Boustany: So it's a single VOB with multiple components in the VOB.

Remal Alshammari: Exactly.

Paul Boustany: Okay. And how are your projects broken up? Do you have - are they more task-oriented projects and streams where you have a project for a specific piece of the software that you're building, or are they broken up into your groups, or other ways?

Remal Alshammari: They're broken up into projects where, for example, they're using a particular component but let's say those past projects using the same component, so we have those past projects with exactly same component or components because let's say the first project is for Windows and the second project is totally spended on Unix or (Solaris) so all these projects using the same components but the source is different.

Like it varies, I should actually say totally different, but it varies from one to another. So it's just like - and my question here becomes a little more complex about just the normal using of it because I'm also using the other side, it is a (request) integrated environment and we're using on the top of that a cruise control integrated with Clear Case.

So my issue here is having all these bells set-up in cruise control but it doesn't overlap in the configuration of the screen.

Paul Boustany: So you're building one environment at a time basically?

Remal Alshammari: Yes.

Paul Boustany: Do you have a common - let me see if I understand this. You have for all intents and purposes one gold baseline of your source code with projects branching off of that for let's say Windows, Unix, Linux, and you want to be able to build but the builds have to happen separately, correct?

Remal Alshammari: Right.

Mark Krasovich: Can you build those in parallel? For instance, if you build the source code, it's the same for all of them so that's built, and then you build the other three either at the same time or do they kick-off in a linear fashion, is that kind of going more towards your question?

Remal Alshammari: Yes.

Paul Boustany: So I think that's probably a good - you're getting down the right path Mark, so could we, would it be possible, or do you have streams set-up that are built on that particular golden baseline or that appropriate configuration that you could use to launch these builds separately, so you would have, you know, build streams that are rebased to a particular level, you know, to a particular baseline that's ready to release, and then those builds could be launched in parallel like Mark was suggesting.

Remal Alshammari: So are you saying that the streams - I only have what I have at this point is one project full build, it's particularly full build and it's what I do, I reconfigure manually the project and to do like for example the Unix or the Windows or whatever environment so I'm wondering if I can do a stream methodology or stream structure that would help me do that.

Paul Boustany: Yeah, exactly, so I think this is where Mark was going and I'm going to try and get this out and then Mark correct me if I'm wrong.

Mark Krasovich: No, go ahead.

Paul Boustany: I think what you want to do is to have your project that's a build project, and in that project you have all of the components available and potentially even

as read-only because you're maybe not making changes - okay, so you have all these components in a single project, that's your build project.

Now you should be able to make streams that are maybe this project has an integration stream that really doesn't get a lot of use but it has multiple development streams that each one of those streams handles building a specific component, and so each one of those streams gets rebased to a baseline from a different project that has built the golden baseline.

Once those streams are rebased to that appropriate point in the build project, you should be able to use a view that's connected to those streams to launch off of build, and again that could be done in parallel.

Mark Krasovich: Yeah, it could be.

Paul Boustany: It's not necessarily a manual process because you could even script the rebasing process once - if you have a naming convention with your baselines from the development project, you could have a job that looks for a certain name or version number or something like that in a baseline which triggers a or automatically causes a rebase to go on in your build project, which could then launch the build.

Remal Alshammari: Okay, you're recommending instead of just having that one integration stream that I'm trying to build on, to having the other one for the different environment and reconfigure those streams to my environment that's set-up for that particular project, correct configuration, okay.

Paul Boustany: Exactly. This would be a great instance for us to have a light board, right, because I think what we're envisioning is that you have multiple projects that

are development projects and they're creating new versions in all these different components and you're creating baselines against those versions.

Remal Alshammari: Right.

Paul Boustany: Then you have a single project that's your build project and in that project you make all of your components available as read-only. And now because we want to build out a lot of different things, we pretty much are ignoring the integration stream.

It's just there because it has to be there when you create a project, so but instead we create all of these development streams that are at the same level in the hierarchy underneath the integration stream.

Each one of those development streams just has the components that you want to build for that particular environment, so a Unix environment has only Unix components or a Java environment has only Java-related code or something, whatever makes sense to your configuration.

Then once stuff is done, once work is being done in the development project and baselines are being created that are considered the golden or the release baseline, you rebase those development streams across the project and then build from those newly rebased versions. So essentially you're just, you're not even changing anything in your build project.

All you're doing is setting those streams configurations to look at the newest baselines from the other project and again, that could all be automated with scripting so that as soon as you create this new baseline, and maybe what you would do is set a promotion level on the baseline so that once it reaches a release promotion or golden promotion, whatever you want to call it, it can

automatically trigger a rebase to those development streams and then in cruise control you can set-up something that will automatically build using a view connected to that development stream.

Mark Krasovich: How many times are you building a day, or is this weekly or daily or hourly process?

Remal Alshammari: There's different projects. The ones I'm talking particularly I'm talking about the build on weekly basis, but there's others like on an hourly basis or a scheduled, so it varies from one project to another, but these particular ones that I'm asking about, they're built weekly.

Paul Boustany: Okay, so even for the hourly ones, you know, because you're going to have a special project defined that's your build project, and it's essentially a read-only project or maybe it's not, but in the case of it being a read-only project, you could use this same approach for your weekly builds to your hourly builds.

I mean it really - all you're doing is changing the configuration of these development streams to look at the newest baseline that's created in your working project, you know, in your development project.

But I think that based off of what we were saying or what we understood from your scenario, it certainly is a doable configuration and one that could eventually be automated once you guys are comfortable with the process.

Mark Krasovich: One of the things that I would suggest is we had a customer that I worked with that had a similar situation. They had different environments that they needed to build on. However, they had one single stream of code that they were using for both builds, well, for all three builds in this case.

And that's pretty much what they did is they set that up in Clear Case although they used build forge as well. I don't know if you've seen that or looked at it.

Remal Alshammari: I know about it. I went to some testing classes, stuff like that.

Mark Krasovich: We worked with them for about three days and we were able to set-up their environment in such a way that it actually decreased their build time quite a bit and allowed them to - I can't remember the specific numbers - but it did decrease their build time by quite a bit and allowed them to do X number more builds per day, per week, etc.

But they used cruise control in a similar fashion. We just helped refine it with them, but I think that would work just fine.

Remal Alshammari: Thank you guys.

Paul Boustany: Thank you. That was really a great question and I think kind of illustrates the capabilities of UCM really. I think a lot of people see it as a rigid process enforcement method for Rational Clear Case but when you start to look at little more closely, you can really be creative with the ways that you're going to implement not only your project structure and your component structure, but the branching and the streams behind that, so it really is a powerful piece of Clear Case and it's a lot more flexible than I think people originally thought it to be.

As time has gone on and as we've developed new features and enhancements into UCM, I think that you'll find that there's enough flexibility in there to

really be able to implement and approach the type of development that makes the most sense for your environment, like the one we were just discussing.

Mark Krasovich: But it also illustrates a good point, Paul, in that build environments, what I'm seeing in the field and I'm sure what you're seeing as well, talking with customers, is the build environments are becoming an increasingly important strategic point or strategic avenue for customers to save time, save resources and increase their time or decrease their time to market.

A lot of the ways they're doing that is by creating more sophisticated process, whether it's through tooling or simply through process trying to incorporate things like build avoidance, reusability. I want to reuse build objects, I don't want to have to build them every time, and there's some of that that happens now.

Customers are getting more and more sophisticated in how they do those things, particularly with the tooling like a build forge or a similar type of tool. I know that there's other things out there that are somewhat similar in scope. One of the other problems that we see with building build environments is the knowledge transfer aspect of builds.

A lot of you on the phone probably have situations where you have one or two who know the build process and if those people leave the company or something happens where they're not available, that becomes a very arduous process to try and recreate builds.

A lot of time is lost, a lot of resources are spent trying to do those things, so building these things, building tools and process and specific functions around a build environment is becoming increasingly important. We're seeing a lot

of customers come to us with questions just like the one we just had, so it's a good question. Any other questions out there?

Paul Boustany: Yeah, again, remember it's star 1 on your phone there if you want to break in and ask a question.

Operator: Your next question comes from the line of Terry Bush.

Paul Boustany: Okay, great, are you there?

Terry Bush: Yes I am.

Paul Boustany: Go ahead.

Terry Bush: I've got a question about Eclipse, our developers use Eclipse with the Clear Case plug-in, and this seems to cause problems with triggers. I mean, I get the error that it doesn't support interactive triggers, which that one I kind of understand, but in other cases, the trigger just doesn't seem to fire, doesn't prevent them from doing something.

And I've got the paper, writing triggers for the Clear Case remote client, but that doesn't really seem to apply to my situation, and I was wondering …

Mark Krasovich: Are you using full Clear Case with Eclipse? Which plug-in are you using?

Paul Boustany: Right, because there's a couple of ways that you can use Clear Case in the Eclipse environment, so there is the Clear Case remote client which is actually a standalone interface that's built in the Eclipse rich client platform, but it can also be plugged into Eclipse, but then there is also, if you're using a full install

of Clear Case, if you have Clear Case on your desktop for example, you can just plug-in like a Clear Case perspective into Eclipse.

Terry Bush: No, they're essentially taking a snapshot of your (on oak) view from Eclipse and then working, you know, independent from Clear Case.

Mark Krasovich: Is it integrated with Clear Quest?

Terry Bush: No.

Mark Krasovich: And what types of errors are you seeing?

Terry Bush: Well, I see the interactive trigger and I kind of understand that, but I need some way to be able to enforce those things. But the other, the one that sort of baffles me is like certain triggers that should work anywhere don't seem to, you know, they don't even see them, it doesn't stop them from doing something and I really don't understand why that doesn't work.

Paul Boustany: Yeah, this is a tough one, maybe for us to troubleshoot in this format. I think that your approach has been good so far because there is that white paper that you talked about, you know, the one writing triggers for the Clear Case remote client.

I think the distinction that we were trying to make before though is that because we have a couple of different ways for people to work in Eclipse, the approach that you need to take with using triggers is different.

Now for example, with the Clear Case remote client, there is a specific way that you'd need to approach the use of triggers because you're operating in a remote environment whereas it's actually making connection to the servers via

the Web or via wide area network protocols, you know, versus a traditional Clear Case set-up which is oriented to the local area network, and if you were plugged into Eclipse via that way, the triggers would operate differently.

Mark Krasovich: Terry, you said one thing that I'm curious about. You said that they use Eclipse but that they download their files and then they work outside of Clear Case?

Terry Bush: Yes, that's correct.

Paul Boustany: Ah, I see, so potentially it sounds like you've got a snapshot view that has a local workspace directory with a bunch of Java files, and then you just add that workspace to Eclipse.

Terry Bush: Right.

Mark Krasovich: So where exactly - at what point in that process are you seeing the - either not seeing triggers or seeing the errors?

Terry Bush: Well, we would see the triggers when they reconnect to do the merge, you know, back to the parent branch, and the triggers either don't fire or you know you get the interactive trigger error.

Paul Boustany: And now are they using base Clear Case it sounds like, not UCM?

Terry Bush: Yes, base Clear Case.

Paul Boustany: So are they doing - they're just launching a merge themselves or do they have some other process that they do for merging?

Terry Bush: It's essentially the auto-merge. I mean, we have it scripted but underneath the covers, it's the automatic merge.

Paul Boustany: Okay. Yeah, I'm guessing and I'm just trying to think of all the various things at play here that might be causing some sort of problem with the interaction there, but just thinking about, you know, you've got a snapshot of your workspace directory and then it's being manipulated through the Eclipse IDE.

I'm guessing that there's some sort of disconnect that's coming through there. I'd be curious to know, since you guys are using already snapshot views, you could probably try and maybe in your test environment or something you could try to implement a couple of workstations that are using just the standalone Clear Case remote client and then you plug that into an Eclipse shell.

I'd be curious to see if by using triggers with the Clear Case remote client views, if you get better interaction there because that would be a good way - you're not going to lose any functionality by using the remote client since you're already using snapshot views and it might be a bit more familiar way for your developers to work.

Because then essentially you've got your Eclipse shell and whatever perspective they're doing development in, and then you've just got a Clear Case remote client perspective that can be used for the merging and can be used for loading new files or doing comparisons or so forth, so …

Mark Krasovich: I think that's a really good suggestion that may be a really good way to go, Terry.

Terry Bush: Okay.

Paul Boustany: Yeah. Why don't you - I'd love to think a bit more about this, so why don't send us in an e-mail, and Angelique, maybe you can remind us all of what the e-mail address or the alias is and maybe we can get you either connected with somebody in services or something, but I think the remote client is probably the right direction for you guys.

Angelique Matheny: Just put the title of the teleconference on the subject line, it's askusnow@us.ibm.com.

Paul Boustany: But that's a great question Terry. I appreciate that and keep them coming. This has been fantastic. We're really excited to be hearing from everybody.

Operator: You have a follow-up question from the line of Kathy Kopczynski.

Paul Boustany: All right. Kathy are you there?

Kathy Kopczynski: Yes, I'm still here, nobody else is asking. I've got lots of questions.

Paul Boustany: Great. Go ahead.

Kathy Kopczynski: I've got two and I'll let you take your pick.

Paul Boustany: All right.

Kathy Kopczynski: One is a UCM question where we have a situation where people will start a rebase and then they hit the close instead of the complete button, and then they keep going, and they check out and check in against the default activity and then they move their versions out of that and then they whine because they can't do a new rebase or whatever.

And then they run into errors when they're either trying to complete the rebase or cancel the rebase. So, the question is, are there any tips and techniques or triggers to keep people from doing stupid things like that?

The other question is that we are - you've got your pick here - that we are moving from Clear Case LP to full Clear Case, and eventually multi-site.

Paul Boustany: Great.

Kathy Kopczynski: We have about three years out of five years of VOB with a lot of abandoned streams and things that we need to clean the history out of, because we don't want to bring over all of that abandoned stuff when we replicate. Any suggestions for how we go about cleaning history out?

Mark Krasovich: This is Mark. So you have abandoned VOBs …

Kathy Kopczynski: Not abandoned VOBs, abandoned streams.

Mark Krasovich: Streams, oh okay.

Kathy Kopczynski: Projects, if you will.

Mark Krasovich: Yeah, and have you locked those obsoletes?

Kathy Kopczynski: Oh yeah. But they're sitting there and so when you look at your version tree, you can't see your version tree anymore. I mean, you can't figure anything out because there's so many dependencies. There's so much history there.

Mark Krasovich: Yeah, I don't know, Paul, you can correct me if I'm wrong here. One of the things that we usually suggest for customers who are making this kind of a transition is to bring over the latest and greatest …

Kathy Kopczynski: Not going to happen. We need to bring over history.

Mark Krasovich: You have, okay, you have to …

Kathy Kopczynski: I mean, we were going to convert everything to start with, but then we want to clean history after we get there, because you can't do anything in LP.

Mark Krasovich: Yeah, and are you guys using UCM?

Kathy Kopczynski: Yes we are.

Mark Krasovich: Yeah. And be clean, I am assuming you mean delete and get rid of?

Kathy Kopczynski: Yes. I mean, there's some stuff they went one way and then we sort of reimported into a whole another directory structure, if you will.

Paul Boustany: So again, we say clean, we mean removing not only - so you want to remove the versions on those branches, the branches themselves, and the streams?

Kathy Kopczynski: Yes.

Paul Boustany: So, my …

Kathy Kopczynski: I mean, we were told this was possible that we could clear out history.

Paul Boustany: It is absolutely possible, right, so but, I can imagine that my colleagues from my support days will cringe if I recommend that you do that, because the reason that we always - in support, we used to always heavily, you know, strongly encourage people to never remove any of the history, and it is there for a purpose.

It has its use for being there, but I can understand what you're saying that it can quickly complicate things.

Kathy Kopczynski: Slow things down.

Paul Boustany: Right, and slow things down, so there's - I think that there's two approaches, right, and I'm going to tread lightly here in respect for my former colleagues, but I think that the two approaches are one, either to when you do your upgrades of full Clear Case, you may want to do some sort of import of, you know, I'd lay down maybe some sort of baseline in the old VOBs that gets you all of the most current information that you want or even goes back a few or something - I mean, just lay down a baseline that's your most current.

And then I would probably approach it even by using the label type and importing that into a new VOB that's just a regular old VOB but get all that data into there by bringing, by using a label selector or something and getting all that information into the new VOB, and then maybe componentize that VOB and start new UCM projects based off of that label type.

Kathy Kopczynski: Well, the problem is we've got parallel development going on. We've got about five different releases being worked on at the same time that are related to each other.

Paul Boustany: Yeah, see, this is why it quickly becomes difficult to get rid of that kind of history because you obviously want to continue to march forward with your development.

I mean, I think what would make people cringe is to hear me say that all of this can be done, right, you can remove the versions, you can remove the branches, you can remove the streams and the activities, but there's a lot in place that would have to be removed and there's a lot of potential pitfalls.

And so that's why Mark's first question was the right one, which is did you block obsolete this stuff?

Kathy Kopczynski: Sure, it's blocked obsolete but it still shows up in version trees. And we're up to 60 gigs in one VOB and it's like we don't really need that.

Paul Boustany: Sure.

Mark Krasovich: This is me cringing. I've had this - I did work in support and in fact Paul and I worked together for a little while, but I've had this question a lot and I've always said, recommended the same thing and that is to just not - don't put it over. Don't bring that stuff over.

Kathy Kopczynski: But how do you not bring it over? I mean, other than importing from a single baseline and starting over? We need to keep our parallel development streams going.

Mark Krasovich: Yeah, and that's the crux. There's not an easy answer for this. Removing that stuff is possible. It's very, very painful, and the effort required to remove a lot of that data, if you've got a lot of those branches out there, probably more effort than it's worth and more painful than it's worth.

Paul Boustany: Right. The other thing to consider to and not to interrupt but the other thing to consider is that you don't have to do necessarily a one-time import/export. You can continue to bring information over. I mean the best approach would be obviously to start putting freezes on whatever development you can after it's been imported, but you could also continue to do this if need be.

It is, as Mark's saying, this is not a - this is a difficult situation and one that other customers have approached and we just want to be as cautious as possible because we don't want to get you into a situation where it will cause further problems by trying to remove things.

Mark Krasovich: Yeah, I think there's probably a way to accomplish what you need to do. It would take further analysis to see exactly what your branching strategy is and what you've got out there. I don't think we can answer the question …

Kathy Kopczynski: No, I just didn't know whether or not people had finally come up with an answer to this or whether it's just going to be painful and that's what we're going to be paying our consulting fees for.

Mark Krasovich: Yeah, potentially.

Paul Boustany: It's certainly something that you would want, either strategy that you choose, and it sounds like you know both of them pretty well. Either one, you're going to want to consult with the services.

Kathy Kopczynski: Oh absolutely, and unfortunately our users are not the sort that are going to say hi and we're starting over and oh, you can have a week to transfer everything.

Paul Boustany: Right.

Mark Krasovich: Well, and that's why Paul's suggestion is really good, that you do it in steps. You do that in phases and you …

Kathy Kopczynski: But I don't know how you do parallel development in phases.

Mark Krasovich: Well, I think is possible with, and again I don't know your environment, so I think it's possible if you design or you architect that movement properly, that is definitely possible.

Kathy Kopczynski: Well, like you start with your earliest one first and then do your second one?

Mark Krasovich: Well yeah, whatever works best for you and you want the guys to be working on the tip and productive but at the same time yeah, you've got to get that process straightened-out before you move forward.

Kathy Kopczynski: Yeah. Okay, so in other words, it's going to be just pain. All right.

Paul Boustany: It's going to be painful regardless, yeah.

Kathy Kopczynski: All right. Next question. I don't want to …

Operator: Your next question comes from the line of Ernest Hill.

Paul Boustany: Okay, are you there?

Ernest Hill: Yes I'm here. I was wondering if you had any tips on writing scripts to integrate RSA with Clear Case? For example, before checking-in a file, I'd

want to run the RSA code format or to reformat the source code to be in our standard format?

Paul Boustany: And so potentially you'd want to approach that as a trigger? What action do you want this …

Ernest Hill: Right, it probably is a trigger, pre-check-in trigger.

Paul Boustany: Okay, so a pre-check-in trigger. Let's see. And have you guys tried to just implement it as such, as a pre-op trigger on check-in that launches either a script or that particular command?

Ernest Hill: Yeah, I mean, that's what we're looking at. I just didn't know if there was any better ways or anything you had for integrating Rational software architect and Clear Case that would make it a little easier.

Paul Boustany: Yeah, I think that as of today the answer is a trigger is your best bet now.

Mark Krasovich: I do, Paul, I've got a customer actually who's doing something like that, and I haven't worked specifically with it, but if you send that question in to us, I do know the services resource who is very, very good and she has been working hand-in-hand with the customer to integrate RSA and Clear Quest and (recpro) for that matter, Clear Case, I'm sorry and (recpro) and I can talk to her and see if I can get a good answer to your question.

Ernest Hill: Okay, thank you.

Paul Boustany: Well that's great. Let me just continue my thought as well a little bit, just because I was saying that you know today, we have in-shell support for Clear

Case in RSA and in Eclipse overall, and so that for today, probably the answer is trigger and I think Mark can get you more information.

But going forward, I just wanted to share a little bit of strategic direction because you know we've been talking a bit about the jazz initiative and for those of you on the call that have heard a little bit about it, you know, that's our next generation right, that's the direction that we're heading with our strategy and the platform that all of our products will be built on.

And so what you'll start seeing is that products like RSA and products like Clear Case are going to be built on this common platform, this jazz platform, and so the integration pieces between them will be a lot more straightforward, a lot simpler, and hopefully a lot easier to use.

So today, I think Mark and the services guys will probably be able to give you a great answer to that question. I just wanted to share a little bit about some of you may have heard of this direction and where we're going.

Ernest Hill: Is there a timeframe on that?

Paul Boustany: You know, the first jazz-enabled product is due to launch in June and that is the Rational team concert product. I can't give any more specific information about dates for the other things we talked about, but jazz is, like I said, as a platform the first technology that's going to be jazz-enabled so to speak is the Rational team concert product.

Mark Krasovich: And that's available on jazz.net if you want to go out and check that out. You can do out and download it, play with it and it's available. We've got over 4000 downloads of it so far, well, that was actually a couple of months ago.

Paul Boustany: Exactly. It's a great little window into where everything's going and you can join the jazz community and you know, I'm forgetting the URL but I know that we have our roadmap listed out on the IBM Web site, so maybe if you do a little search for jazz and roadmap there, you can take a look at that.

I just want to do a shameless plug here at this point too for the Rational software developer conference coming up in June as well, the first week of June. There's going to be a lot more talk about jazz, about the future of these products, including the future of Rational Clear Case, where it's all going, and you'll be able to get some hands-on experience with the products.

You'll get some information about the 7.1 release that's coming for Clear Case as well as Clear Quest and it's a great opportunity for you guys to not only meet folks like Mark and I but to meet the developers, to meet the product managers, and to meet each other.

You know, there's probably, I think I've heard from a lot of customers that one of their favorite parts about going to the Rational software developer conference is that they just get to network with other Rational customers and you know, we think that we have a lot of good tips and tricks on how to become the power user, so to speak.

But really it's you guys that have a lot of the answers, and by networking with one another and joining the community and being a part of all that, it's really very valuable, so as we're drawing to a close on this call here, I thought it would be a good opportunity to do a little shameless plug for the Rational software developer conference. It's in Orlando from June 1 through 5.

Mark Krasovich: And I wanted to bring one more thing up. Kathy, you had another question about there about triggers. Please e-mail that question to us, and we'll get you information on that.

Paul Boustany: Yeah definitely, and any other questions that you guys may have thought of that you'd like to send in, ask us now and we would love to hear it. Angelique, do we have time for any more questions or are we at the end of the call?

Angelique Matheny: Well I think we're about out of questions, if you want to have some closing remarks or I have, I think, one really good question that came in earlier if you'd like to finish with that one.

Operator: I do have one more question from Dallas Hilliard.

Paul Boustany: Okay, let's take that question and then we'll have to wrap it up. Are you there, sir?

Dallas Hilliard: Yes I am.

Paul Boustany: Go right ahead, please.

Dallas Hilliard: We're working with a client that's been using base Clear Case and we're trying to get them to go to Clear Case, Clear Quest UCM. What good arguments would you put forward for someone to do that?

Paul Boustany: I can think of some things that I - but I know Mark has been working with customers and probably been trying to convince them of this.

Mark Krasovich: Yeah, definitely. I mean, I think there were a lot of things so on a high level, it depends. How big is their environment right now?

Dallas Hilliard: Oh, they're a huge company. They're probably 100 developers, maybe.

Mark Krasovich: Okay. And are they distributed?

Dallas Hilliard: No, but we will be.

Mark Krasovich: Okay. So, a lot of the things that you gain with Clear Case, Clear Quest UCM over simply a base Clear Case and some nebulous third-party bug-tracking application or some of the things that I described at the beginning of the call, and I'm not sure that you were on at the time.

But I would say the three biggest things are traceability, flexibility and process maturity, so the ability to trace those defects down to the activity level and moving to that activity-based change management where you're raising that level of abstraction, it helps the whole community of developers and project managers to be able to communicate with each other.

It solidifies process, it gets everybody doing something that's repeatable, that's reusable, and allows management or project management the ability to look at specific reporting, so you're going to be able to track trends, you're going to be able to get specific data on what your developers are working on, why things are taking so long or those types of things.

It's going to increase productivity. It's going to shorten your time to market. You're going to be able to get more things into releases, and these are things that we have seen over and over again.

The other thing I would suggest is looking at a (rut) process or something like that along with Clear Case, Clear Quest UCM. That has proven extremely valuable as well to a lot of our customers.

Dallas Hilliard: Yeah, one of things that we're getting pushback on is that they're saying that, in their eyes, UCM is very rigid and I know you keep saying it's flexible. Could you expand on some of the flexibility of UCM?

Mark Krasovich: Typically, and this is something Paul you can jump in too, we hear this a lot. It's rigid because we all know that as CM managers, developers like to have freedom to do what I really want to do.

Whether that means that they can go out and fix what they want to fix and push it out to production right away because they know it's right, or those types of scenarios which happens a lot, or they're going to go in and dig in the database and make changes just because they think it's right.

That's a difficult thing. It's a cultural change that needs to happen and one of the things I would suggest is when you're doing a conversion like this, select perhaps a smaller pilot type of project that you can go out and prove that (A) it's not as restrictive.

Look, you've got your own stream, you can do whatever you want on that stream, test, modify, change, whatever you need to do, that's your own sandbox. Once you deliver - all you're doing when you're delivering to the mother ship is you're delivering that perfect code that you've created.

But you have all the flexibility, all the traceability, everything you want is in your sandbox so it's not really restrictive in that sense, it's just more productive and it's a more refined process.

Paul Boustany: Exactly, and sometimes developers - sorry, Mark - but sometimes they don't even realize that they can establish their own subdevelopment streams if they wanted, because they have the flexibility to branch off and do - make their own testing environment where they have total freedom to work in an isolated stream.

You know, it's rigid because it's enforcing some kind of process, but that process itself doesn't have to be rigid. It could be totally flexible and it can match whatever you've been doing today. Often times there's not something in place and so people think oh well, UCM will put this rigid process over and enforce, make us do things a certain way, but that's really not the case.

Mark Krasovich: Well, I was just going to piggyback on that and say that one of the things that my team here in my area does is that when we go into a customer with a situation like this, we say look, we're not coming in here to mandate that you do things a certain way.

We're coming in here and saying how do you do things now, what works, what doesn't work, how can we improve the things that don't work, how can we streamline the things that do, incorporate them into this UCM process and make it work for you, so we're going to customize and tailor that process for your organization. We're not going to pull something out of a box and say you must do this.

Dallas Hilliard: And I think that's the pushback that we're getting is that they think that you pull it out of a box and that's fixed. That's what you use. Right, not that you can't tailor it to, you know, your operations.

Paul Boustany: And conversely, you know, it really can be used as a great out-of-the-box process enforcement model, but on the other hand, it has some flexibility and I think Mark's point is a great one for us to end on and that's the fact that when people go in and go to implement this process, they say how do you guys do it today because we can match that and we can make improvements where it makes sense.

Angelique Matheny: Well I hate to interrupt here but we are out of time.

Paul Boustany: Thanks for the question, that was a great one. Angelique, take us away.

Angelique Matheny: Thank you very much Paul and Mark. This is obviously a very valuable session and we appreciate your sharing your knowledge and experience. Be sure and check out our Rational Clear Case Innovator Series E Kit on developer works, www.ibm.com/developerworks/rational.

Just navigate to the Clear Case page and you'll find the link to the Kit in the latest content section. The E Kit provides a collection of materials that can help you become a power user of Rational Clear Case. The E Kit has also been recently upgraded to include a new white paper on implementing Rational Clear Case as a lightweight yet full-featured SCM solution.

This white paper is only available through the Innovator Series E Kit so be sure and take a look today. 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 the next few days on the Rational Talks to You site, www.ibm.com/rational/talks.

Our previous teleconferences are available there as well. I'd like to thank our speakers Paul Boustany and Mark Krasovich for being with us today to talk about calling all Clear Case power users and those that would like to be.

We would also like to thank your 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: This concludes today's conference call. You may now disconnect.

END