IBM

Moderator: Angelique Matheny
August 27, 2009
12:00 pm CT

Coordinator: Welcome and thank you for standing by, at this time all participants lines are in a listen only mode for the duration of today's presentation.

Throughout, we will conduct a question and answer sessions. At that time, if you would like to ask a question or make a comment over the phone, please press star 1. Please unmute your phone and record your name clearly when prompted.

Today's conference is being recorded. If you have any objection, you may disconnect at this time.

Now, I would like to turn the call over to your host Angelique Matheny.

Angelique Matheny: Hello everyone and welcome to this Rational Talk To You teleconference, A Conversation with Grady Booch.

I’m Angelique Matheny and I will be your host for today's call. Now these calls are really for you, you won't find any slides for this teleconference and we want this to be interactive and to get your questions answered. We will open up the lines during the Q and A portion of the call, so as the operator mentioned, press star 1 when it is time.

Write those questions down and get ready. And you never know, Grady might ask you a question. Also, if you would like to submit questions after this teleconference, please email us at askusnow@us.ibm.com, that is A-S-K-U-S-N-O-W @ U-S.I- B- M.com. Just put the title of this teleconference in the subject line.

The software delivery team of the future will be more global, collaborative and transparent than ever before. How is internet culture and social media both shaping and responding to this evolution? How are software teams responding to Wikinomics and the democratization of ideas? How are Open Source initiatives changing software economics?

This is your opportunity to engage in a dialog with our thought leader, who is equally versatile in debugging code and predicting the future. And today's distinguished guest, Grady Booch, is recognized internationally for his innovative work on software architecture, collaborative development environments and software engineering.

Our renowned visionary has devoted his life's work to improving the art and science of software development. Grady served with the chief scientists of Rational Software Corporation since its founding in '81 and through its acquisition by IBM in 2003. He is now part of the IBM Thomas J. Watson Research Center, where he mentors and leads various software engineering projects that are beyond the constraints of immediate project horizons.

One of the original authors of the unified modeling language, Grady has served as architect and architectural mentor for complex software intensive systems in just about every domain imaginable.

Grady, welcome and thanks for joining us today.

Grady Booch: Grady, thanks for having me. I appreciate it. Let me have the opportunity to just chat for a few minutes and then really this call is for those of you on the call, I am here to respond to any questions, concerns, comments you might have.

Many economists are saying now, that we are in the midst of what is called the Great Recession and it is likely it will be protracted in its recovery. The markets lost easily eight trillion in value in just the last year and there is still considerable jitteriness in the market place.

And if we even project out further in the coming months, there’s a lot of uncertainty with regards to where the G20 is looking on issues of international trade and climate and things like that, so, tremendous uncertainty.

And at the same time, because of this economic malaise, the common reaction in such circumstances is to retreat, to step back, to shrink things. We must realize though that in our business, software, we as industry have contributed an incredible amount to the wealth of the world and literally changed the way that people and companies and countries interact with one another.

So, we have been a source of tremendous wealth in the past and my observation is that we in our industry can be an important contribution to the recovery and indeed, future innovation.

The common reaction in these circumstances, and I see this from company to company I go, is that you view software development as just another cost center, and so you cut it like you would do anything else. But rather, I would like to make the case that software development, except for the commodity pieces that we do, is really an asset of strategic importance and therefore, should be encouraged.

Those companies, right now, who are retreating, who are viewing their software development teams as just a cost center, when the recovery comes, and indeed will come, history has certainly shown that to be the case, they are going to be in a terribly bad position. Because there will be others who I speak to now and who I haven't spoken to, who are certainly trying to seize the initiative and view software as a means of pulling themselves out of this malaise.

So, in that regard, there are a number of forces outside of our control that are pushing what is going on in our business. Yet, there are some things we can control. We can go back to the fundamentals of how we deliver, operate, deploy and evolve software systems. We, as the professionals in this space, can work to improve the ways we deal with predictability and repeatability and quality. Those are the things we can control.

The other point I would like to make, is that one of the great things our industry has is the realization that a lot of what we produce comes from a cognitive surplus. We have tremendous visions for the kinds of things we would like to do and our task as software professionals is to turn that into raw, running, naked code.

And what trips most of us up are things that are kind of on the cusp of the technical and social. How does one best architect a system? How does one best architect of the organization? And there is this wonderful dance that happens between the two.

Jim Coplien in his marvelous book, “Organizational Patterns of Agile Development” points out, there is this wonderful interplay between implementation and architecture, and each of the two imposes restrictions and imposes innervations upon the other. And insofar as I could have a dance between the two, then you will find yourself in a position of really using your software systems for a strategic advantage.

Now back to those external forces. Those external forces are leading us to materially do some things that we wouldn't have had to in the past years. Our software development teams are by and large far more distributed than they ever were. And I don't mean distribution, just geographically, but I also mean temporally. The geographic distribution is obvious.

Most of us are building systems which we will outsource pieces or near-shore pieces and certainly we are building systems that are far larger and more complex, in general, that can be done by just a small team sitting within a room. So, we are having to deal with systems of systems and that by its nature deals with geographic distribution.

The temporal distribution attends to the issues of legacy, in that we will build something and we slowly accrete that code over time. So although software has no weight it does have mass that creates an inertia for us. And it is that inertia, through that accretion over time that makes it very difficult for us to continue to innovate in the face of having to serve our old systems.

So, we see the changes coming in processes, thus the pendulum swinging towards more agile methods. For a while, it was swinging sharply in the direction of agile methods. But I think the pendulum is moving back to a more sane spot that realizes there are some principles of agile that make a whole lot of sense, one also has to deal with issues of scale. It is also addressing the issues of how teams work together with one another. Where Jazz and rational team concert, for example, are headed. It is a great example of how one can build these portals, if you will these virtual meeting places for organizations that are geographically and temporally dispersed.

Same thing is true with regards to the economics around Open Source. I mention this notion of accretion of software over time and there are known things you can do with old software. You can give it away, you can ignore it, you can put it on life support, you can harvest from it, you can transform it.

Open source is actually one of those kinds of things, because it means if you are contributing to the Open Source world, it means that you have as an industry, have extracted the economic value you can find from it and yet you believe there is fundamental value still, that was in your domain, you can't extract, so you put into the Open Source world.

Similarly, as a user of Open Source kinds of things, you realize there is a commoditization of software out there that makes economic sense for you to draw from, but I think even more compelling is the notion that many of the Open Source efforts that are happening right now, create a platform that is an abundant place for creation. Thus, back to the notion of systems of systems, it means now I can focus upon the value-added I have and not worry about all these infrastructure kinds of things.

So, despite where we are in this space, in this economic space right now, I really am utterly convinced that now is a great time to be in software, because we have the opportunity to continue to deal with the fundamentals but also continue to innovate at the same time. And so we in our industry need to focus on those fundamentals so we can help, not only our industry, but also help the economy at large.

There are so many wonderful things yet to be done and everything I can envision, everything anyone can envision for that future state, involves software that we haven't written yet, software that you and your organizations will be creating.

So (Robin), that's enough of me talking. I want to hear from folks and see what interests them and what we would like to talk about. Again, I am open to fielding any question, so, let's open the line, shall we. Operators are standing by, as they say.

Coordinator: Thank you. At this time if you would like to ask a question or make a comment over the phone, please press star 1. Unmute your phone and record your name clearly when prompted. To withdraw your request you may press star 2. Once again, to ask a question or make a comment over the phone, please press star 1. One moment please.

Angelique Matheny: Grady, we have a shy audience today.

Grady Booch: We have a very shy audience. Well, it may be time for me to ask a question, I will give them like 10-15 more seconds and then I will ask a question.

Coordinator: Once again, to ask a question, please press star 1. Please unmute your phone and record your name clearly when prompted. Please stand by for our first question.

Our first question comes from Mr. Jason Steiner from Credit Suisse. Sir, your line is open.

Jason Steiner: Hi. Good afternoon, Grady.

Grady Booch: Hello Jason.

Jason Steiner: So, one of the conundrums. I work for a large financial firm, one of the conundrums within the Open Source immunity is that there is a heck of a lot of good software developers that work in finance.

Grady Booch: Yes.

Jason Steiner: They sit at their desks all day, they bang out code, you know making the financial industry tic. Sometimes, we like to use Open Source code, sometimes we have to use commercial products. But a lot of times though developers go home and they want to contribute to Open Source projects that may either further their own interests or potentially for their own firms name.

How does one make a good balance of this in trying to set potential policy for these types of groups? I mean, we have personal lives, we like to go out and contribute, how do we balance that?

Grady Booch: That's a great question and you know IBM is a good example to look to in terms of policies, because we encourage our folks to work on Open Source kinds of things and we ourselves even, you know, have people dedicated to do that.

So let me tease a part of your question and I think the base question is, “What kind of policies for an organization such as yours makes sense?” So, to answer that, let me tease a part what I think the issues are.

One is that of intellectual property, I believe, because you and your company certainly want to insure that there is not a leakage of IP, that sneaks out into the Open Source world and similarly back in, because of many of the licenses with regards to Open Source kinds of things, you have to be very careful about the use of that Open Source code within your products, especially if you resell and redistribute them.

In the latter case, you know this is one where your legions of lawyers need to get involved, but practically, what IBM tends to do and again, this is just a broad brush here -- we can certainly put you in touch with the people who set those policies -- is that we make very clear whenever we are doing any research or any product development, what is sort of in the box and what's outside of the box.

We don't let individuals go out and say, I would like to use this or that and build things, because that does complicate and muddy the waters for the IP. So I think there has to be a degree of accountability, with regards to what's imported into your organization and the mechanisms you need to have in place are not just the policies, but you also need to have architectures aboard that ensure the fidelity of the software that comes in.

As for what goes out, frankly, I think increased transparency helps you. At IBM we encourage people to blog and there is nothing between my blogging and the world, there are no lawyers that get in the way or anything like that. And so to encourage the transparency of the people who want to contribute, I would even encourage them as part of a policy to say, not only do we want you to do these kinds of things, but please make it visible to us, so that we in our company can gain some value for the kinds of things you are doing for the Open Source world.

Make an internal Wiki or encourage internal blogs, that allow people to at least tell other people within your company what they are doing, because there may be opportunities and for serendipitous connections along the way.

So, that is the broad brushing answer I would offer you and I would certainly say I'd encourage it, because it does add value. So often within companies, our windows become mirrors and we never look outside and by having your people contribute to the Open Source world, I think that is an incredible opportunity for inspiration. The bottom line is, you have to protect yourself from issues of IP leakage and that is where your legal policies can come into playing.

And (Robin), help me out here, but you might write me a note on this, but, we could certainly put you in touch, Jason, with the people inside IBM that are really in charge of the policies that make it tic and I am sure they'd be happy to share, because I know that publicly, we have said some things about how we do this.

Did that answer your question?

Jason Steiner Yes it did. I thoroughly would enjoy engaging IBM further on this topic, as it is very hot within my firm at this time.

Grady Booch: Got it.

Jason Steiner: Thank you.

Grady Booch: You’re welcome, Jason. Thanks, good question.

Anyone else? See, that wasn't so painful, now was it?

Coordinator: Once again, if you would like to ask a question or make a comment over the phone, please press star 1. One moment please.

Grady Booch: Thank you.

Angelique Matheny: Grady, I think you’re going to have to get them going.

Grady Booch: I'm going to have to ask a question. So, let me ask the following question, which is sort of an economic one. In the current economic malaise, I'm curious if you have seen your upper management and many of you may be that upper management, are you at this point shrinking your teams or are any of you really trying to seize the initiative and a view this as an opportunity for taking the high ground? Where do you land on that? Do you find yourself shrinking or do you find yourself taking some bold action?

Coordinator: Once again, to answer the question or to ask a question, please press star 1. Please unmute your phone and record your name clearly when prompted.

Grady Booch: I may ask another question. Maybe no one wants to say, “Oh yes, we’re firing people left and right. Would you like my résumé?” I'm not looking for those kinds of answers.

Then, I shall ask another question. It doesn't sound like we are going to have a response on that one.

Does anyone have a story they would be willing to share with regards to how they are currently saddled by Legacy.

The short story I will give you now to maybe seed your ideas is, increasingly I have been doing some work for federal agencies and it astounds me, not just the amount of COBOL I find out there, but the amount of assembly language I find out there.

There is one agency in particular that built systems starting back in the 60s, and there is still code from that living around, about a half a million lines of IBM assembly language and they have legions of people working on that. So, maybe there is a direct question. How many of you are dealing with Legacy in those older languages, be it Assembly, or Fortran, or COBOL or heavens, it might even be JOVIAL or Ada, or something like that?

Coordinator: Our next question or comment comes from (Bill). Your line is open.

Grady Booch: Hi, Bill.

(Bill): Thanks. I wasn't quite quick enough to get in before your second question, but I'll address that...

Grady Booch: Sure.

(Bill): ...a little bit as well. Our company is taking a -- I work for a telecommunications firm -- and we’re taking different steps in different areas. We're looking really hard at the Legacy stuff to see whether it’s still cost effective to support it, because our quarterly earnings demand that we have to cut costs somewhere and that does, inevitably, end up effecting software developers...

Grady Booch: Yes.

(Bill): But the other thing I’m really pleased about is we are looking very hard at all the opportunities that exist with the internet and the content developers of all sorts that provide, in particular, over the top type of applications and what kinds of additional services that can be provided to them.

As far as it applies to Open Source though, we do use a lot of Open Source, but, there are two issues that come up for us, which I would like your comments on.

The first is having a good feeling -- we just recently had a policy issued about this -- a good feeling about the security of the Open Source software you get, because it is done in a collaborative method and the ethics of different people in a group are going to have different levels.

And the other thing is historically, Open Source has been much more difficult to configure. If you buy a product from a vendor, it tends to have -- and I'm not saying that for the big software applications there aren't installation issues -- but they tend to have a much more straightforward and much less flexible installation process, so there are fewer, there's less room for error.

Where as Open Source tends to be, you can configure just about everything, because it is so flexible, it's designed to run on multiple platforms and for that reason, it can require more effort to install and to maintain. So, I am curious about, in particular the security issues of it and also just the ease of use, where we stand and where're we going.

Grady Booch: Great question. So the security issue scares me as well, too, because, you know, it's certainly possible for, you know, some of the bad guys to come in and sneak in code in something that might be used as mission-critical and it could be a Trojan horse in some ways.

For that matter, we have that same kind of fear, the government in particular, has that kind of fear, for chips that are manufactured in countries that are not our own. Because of their complexity, we simply can't, you know, really know and so there is a level of trust and so it is with Open Source software.

So, my recommendation to mitigate that risk is to say that insure that you are not using, you know, the latest and greatest thing for your operational use, because you'll have decreasing trust, the fresher that code is. So you need to have somebody that insures that the system that you're bringing in from the Open Source world, they're baselined, they're trusted, they're tested against and you have, you know, a reasonable understanding with what the rest of the world is doing in that case.

So that means you probably need a gatekeeper between what you import in the self and source space and what you use in production. And that goes hand in hand with my earlier response about making sure you have some degree of accountability in that regard.

It may happen. It's certainly a risk. It is possible, and if such a thing happened, it wouldn't impact you, but it could impact lots of folks. But, one has to be careful about the danger of that risk, I mean I might get hit by a meteorite, as well too, but I don't obsess over it. So, one has to, I believe, trust in the many eyes looking at a little bit older code, not the latest and greatest release, so that those kinds of things, the community itself will discover.

And again, the notion of this open transparency, the greater transparency decreases the risk of that kind of thing happening. So, short answer on issues of security, you need to have some policies and mechanisms in place as a gatekeeper for the things you import, as opposed to just using everything that might be out there in the (unintelligible) itself.

The second question, you asked, let me see if I remember it. It dealt with, remind me please, it was in my head, but it popped out.

(Bill): Yes, it’s ease of installation...

((Crosstalk))

Grady Booch: Ease of installation. Yes.

((Crosstalk))

(Bill): ...installing of Open Source.

Grady Booch: Yes

(Bill): ...to be a lot more of roll your own.

Grady Booch: Exactly. The Open Source community has really been beat up by that of late. There have been some threats on Slashdot about usability in Open Source spaces, because you know, frankly, that's just not a concern of many and there are many of the Open Source projects for which there are developers contributing, but it is not like there has been great designs on the human interfaces side.

The result of it is that you've seen this third party community set up, people like Red Hat and others, who are the ones who package those things for you. And you might consider, you know, finding such a group to do that for you, or if you have a large enough use within your own organization, within the company as a whole, maybe it says, I need to set aside people that do that for me and package it up, so that I act as my own packager. And that could serve as the same body as the gatekeeper in the first case.

The other thing I would recommend is feedback to the Open Source community, if you have particular things that you say, this just doesn't work, then contribute to fixing it, as well, too.

(Bill): Good point. Could I follow-up, just a...

Grady Booch: Absolutely.

(Bill): ...very quickly on the security issue. Um, it occurred to me while you were talking, are there organizations that will certify it, analogous to what Sun, at least, used to do, I don't know if they still do, with the, you know, 100% JAVA certification?. Do you know what I’m talking about?

Grady Booch: Yes, I do know what you mean.

((Crosstalk))

(Bill): ...are there organizations that address that, like does Red Hat certify the security of its...?

Grady Booch: Right, you would have to ask Red Hat, I really don't know...

(Bill): ...yes, okay.

Grady Booch: ...but I do know that the Apache Software Foundation does do such things. There is a thing called FUSE, which I forget what it stands for, but it's basically Open Source software components which are certified releases. So, there are such things out there. And there is a whole, go do a Google search for FUSE and you will see some things on it, so at least those kinds of things.

(Bill): Great. I haven't heard of that. Thank you.

Grady Booch: Yes, yes, yes, um, there are a few others that come to mind - let me, keep talking and I will see if I can find them in my notes here.

(Bill): Keep talking. I'll let somebody else.

Grady Booch: Okay, very good.

(Bill): Thank you.

Coordinator: Once again, if you would like to ask a question or make a comment over the phone, please press star 1. One moment please.

Grady Booch: You know, as I think out loud here, does IBM do that? I don't know. I mean there are, heck, there are almost half a million of us, so there might be somebody somewhere in some office, but nothing immediately comes to mind.

Coordinator: The next question or comment comes from Daniel.

Grady Booch: Yes.

Daniel Thompson: Hello, this is Dan Thompson, can you hear me?

Grady Booch: I can.

Daniel Thompson: Hi Grady. Thanks for having me on. I actually work for the sales department of Rational Software at IBM.

Grady Booch: I've heard of that group, yes.

((Crosstalk))

Daniel Thompson: I was kind of curious of what your thoughts are - this is somewhat of a general question - but around the Cloud computing, as it may impact sales organizations? How we might structure sales to customers? How it’s going to impact customers’ way of working with applications? What do you think customers’ perception is of the value of Cloud. And is the knowledge basically out there and people are moving on it quickly, slowly, how quickly will it be adopted. I know it is all kind of crystal ball type stuff, but just your general thoughts.

((Crosstalk))

Grady Booch: Sure, so here is my take on the whole Cloud thing. Let's consider what the economic pressures are that are pushing organizations to even consider Cloud in the first place. And I think there are two of them. The first is just the sheer cost and complexity and madness of managing your own multiple kinds of data centers. That there's obvious opportunity for commoditization in that regard and therefore, economies of scale. So, it's just, the biggest one is just reducing our costs in the infrastructure.

The second one is, I guess, a separation of concerns that if you go way back in time, the kinds of software systems we built were deeply tied to the machines which ran them and vice versa. But many of the kinds of things we are doing today, we don't really care about the operating system, we care about a set of services and a set of protocols and it can run anywhere.

So, the separation of concerns that has become the dominant design of many enterprise kinds of systems, leads us to say, let's separate the platform execution from the applications themselves. And so from an architectural perspective, that pushes you in that direction, as well, too. But be careful is my reaction, because the following issues raise themselves.

I had a conversation with some folks in DC recently, where there was a push from Obama's new CIO to move toward Cloud, but some of them were coming back and saying, “Whoa, wait a minute. We actually have federal law that requires us to control our data in certain ways. And if it goes into a Cloud, we can't even know or trust where it is.” So, physical location and physical capture of the data itself becomes a topic which must be considered as one moves to the Cloud.

It is also the case of just performance, because, you know, Amazon talks to their Cloud, Google talks to their Cloud, IBM talks to their Cloud. What I would urge people to look at and say, “Well, wait a minute, I look outside my window and there are not just clouds, there are Cumulus, and Nimbus and Cirrus and all these kinds of things and some days I even have fog all over the place. What kind of Cloud are you actually giving me?

Because one person's Cloud may not be another person's Cloud and there are characteristics that are different among them. I've seen some statistics recently, again, there was the threat on Slashdot. I get a lot of news from Slashdot, you ought to take a peek there, if you don't swing by there every now and then. Where someone did some performance tests against Amazon's Cloud and other Clouds and what they discovered, and I am not bashing any of these, you have to be careful about the performance you get from them, because it varies from time to time. It's going to be different if you're going to be using a public Cloud, than if you are using a data center that you control yourself.

The other - the third thing I would mention is, if you move toward a Cloud, there are certain economic benefits, but you also have two ask yourself, what are you giving up in the process. There may be some things you can optimize in the infrastructure that save costs that way, but you may also be losing an edge, in terms of performance. And if everybody moves to Cloud or we're all in the same space, then that means you are going to have to compete on the application side, as opposed to the system side.

So to reiterate here, what are the issues? The comeback from this, I would recommend you to say is, when you think of Cloud, ask yourself, “What kind of Cloud of is it?” And second, “What are the forces I'm trying to resolve and attend to by even considering Cloud, in the first place?” So those are the big things that come to mind, for me.

Daniel Thompson: Right, well thank you and one follow-up question...?

Grady Booch: Sure.

Daniel Thompson:...regarding the security of people's data, do you foresee a potential for vendors to come up with ways to product services to sort of assure security?

Grady Booch: Well, see I personally think, and this is my opinion, not that of IBM's, I think we should just hand over all of our data to the government, because I utterly trust their ability to keep my data secure, so, I think that is what we should have.

The real answer is, yes, I do believe that the marketplace will lead us to the emergence of those kinds of things. In fact, and I am not, you know, pushing product here, but where IBM has been headed with its Cloud is less interest in the public Cloud, because that is a commodity that we just don't play in well, but more so on the notion of the private Cloud, so the captive Cloud for organizations.

And once you're in those circumstances, then you have a lot more trust as to what you are doing with the data itself. So, as an example of what you're saying, “Are there companies going to do that?” Well that's what IBM is trying to do in the Cloud space.

(Bill): Okay, thank you.

Grady Booch: And by the way, you can tell from the response to that question, a little bit of my political leanings and my cynicism.

(Bill): We were already aware.

Grady Booch: Okay, you were already aware of that, okay.

I am hearing a knock on the door and all of a sudden there are black helicopters flying around...

Any other questions today?

Coordinator: Once again, if you would like to ask a question or make a comment over the phone, please press star 1. Please stand by.

Angelique Matheny: Well, I think we have another question coming.

Grady Booch: Very good.

Man: Hi Grady. It was very good to have you on the call. In the present economic situation of formulating architectural guidelines, what are the different compromising levels that can on looking to, while looking into that nonfunctional requirement of the application.

Grady Booch: Let me see if I understand your question. The premise of it was, in our current economic situations, you're asking what kinds of architectures do we need to attend to or what kind of processes that allows us to deal with nonfunctional requirements, is that the essence of your question?

Man: Yes, that’s got it.

Grady Booch: Okay, wow, that’s an interesting one. I'm not sure that economics' impacts that as much as you might believe, because as I look at nonfunctional requirements, or maybe it does, maybe it does overwhelmingly. Hang on, let me step inside my head here and see where I am on this one.

Okay, as I look at the nonfunctional requirements, we're talking about things like, you know, issues of (unintelligible), the availability, the performance and those kinds of things and usability, as well, too.

While, certainly, economics does impact that, because, if I want to have a system that has, you know, 99.9, lots of 9s uptime, it is going to cost me incrementally more for each of those further digits beyond the right side of the decimal point, than if I allow myself to have a little bit more flexibility and looseness in it and thus the difference you see in the costing of military systems, or human essential systems, like medical electronics versus others.

So, I guess, the short answer, I'd say is as one evolves an architecture, you have to view it as an engineering process, so there is no perfect answer to it, but rather it is a resolution of those kinds of forces. And I think the best thing that can be done, to attend to those nonfunctional issues, is you need to have a process in place that allows you to make reasoned decisions about them.

There are a number of meltdowns that I have seen in software projects where that wasn't done. The IBM, not the IBM, but the FBI, I mean, another three letter and acronym. The FBI case file management system is an example. The FAA's AAS, their advanced automation system of a few years ago. Both of those are examples are examples of, I think clear failures, certainly in the FBI case and as I look at it from the outside, the number of systemic problems, one of which, in both of those cases, is they simply lost control of the requirements process.

So, requirements were thrown in, everything was viewed as equal and they're certainly not. So you need to have a process that looks at those changes and as (unintelligible), as looks at the implementation and the architecture and the dance between them, and also considers what's the cost reach of those kinds of things. Because there is nothing that is free, there are requirements that are not as equal as others and you have to make reasoned, intentional decisions about that.

So, as I have always recommended, the process that works in all these circumstances, is you need to have one that grows your architecture to be incremental and iterative delivery of testable executables. And of course, the three major parts of that are the most important artifact you produce is an executable. So having a rhythm of release is important and that's, you know, an essential principle of agile methods. Of course, it also has to be testable along the way, too.

The second principle of that is you do it incrementally and iteratively. Which means, even those nonfunctional requirements, you cannot know a priori exactly what they are. So, you need a process in place that allows you to judge them and evolve them over time. The mere presence of the system will change the way you view the world itself and that's why incremental and iterative is necessary.

And the third point is using architecture as an artifact is a governance mechanism. You throw use cases against it, you throw your tests against it, you make reasoned decisions about the costs of things, and you keep control and accountability for the significant design decisions you've had. So I think that process, which is really at the core of the rational process and many other processes for that matter. I think that's the way one attends to the dance between architecture and nonfunctional requirements.

Great question, by the way.

Man: Thank you.

Grady Booch: Anyone else?

Rick Tobias: Grady, this is Rick Tobias...

Grady Booch: Yes.

Rick Tobias: At the Oregon Department of Revenue. I've got a two-part question for you. How much influence do you think the Open Source movement has had historically on interoperability? And then what do you think the impact of interoperability and openness would be if someone was to, for example, purchase Java or some other key Open Source tool and take it private?

Grady Booch: Well, okay. Two interesting ones. Let me attend to the first one, then I need to ask you a question about the second part of that.

Interoperability is an interesting beast, because many of the early Open Source projects really didn't care about interoperability, except in as so far as they had to deal with other protocols that were out there in the world. So, I think the first and second, or so, generation of the Open Source projects we saw, weren't really caring about interoperability, because they were dealing with more point kinds of projects, or products themselves.

I am beginning to see a sea change in that space, but I am not certain I would characterize it, necessarily, as an Open Source movement, as rather as a domain specific or industry specific movement. The two examples that come to mind and I will start a background process in my head, to see if I can find one in your space.

The two that come to mind are in totally different domains, there is a thing called (Auto-Saw), which is an attempt by a consortium of auto manufactures to devise a common architecture, both hardware and software for in-car electronics. And there is a similar effort afoot by the consultive consortium of space data systems to come up with a common architecture for space communications. And as we see more and more commercial space-flight kinds of things, that becomes incredibly important.

Is that Open Source? Well, sort of, kind of, a little bit, not, but it's really an industry coming together and saying, we are going to put out there common architectures that sort of define the ways we play well with one another so we can interoperate. And it will become open, but not in the traditional sense of the Open Source world. It's just a matter of creating these abundant platforms upon which we can build.

Now, I am thinking out loud here in your space, and I just came back from the IRS, and I know that there are some efforts afoot, especially through Vivek, Obama's new CIO, to try to push, you know, open standards in that space. What's been happening with gov.org and there's, I think, data.gov is another one. There's a real fierce attempt on the government's part, the federal part, to encourage the creation of those kinds of standards. And I am aware, at least, of some common data standards of being forged in that community and some common architectures, as well.

Is that traditional Open Source? Not exactly, but I think, especially, in your world, you know, you're the tax man kind of person. I think there will be some things emerging. I could probably put you in touch with some of the folks I know on the fed side that are at least thinking in that space. You guys can begin to have a dialogue. So, maybe as a follow-up, have your people talk to my people, except that I don't have any people, send me an email and I can get you connected.

So, I think that answers the first question. The second one, I wasn't quite clear on exactly what your question was saying. It was the notion of, you know, buying some Open Source sort of thing and then turning it into a product. Can you repeat that question, or clarify it for me?

Rick Tobias: Yes, in a nutshell, looking at Oracle's attempt to purchase and Larry Ellison's comments that Java was the prime piece in that pie, that you have to ask the question, well if they take that private, Java being one of the key tools in the Open Source movement, what would happen to the Open Source movement?

Grady Booch: Well, I think there would be utter rebellion were that to be the case. Mr. Ellison, I did not read his comments, so I don't know exactly what he was saying there. Were I in his situation, I would also say, yes, one of the family jewels that they bought was the IP around Java itself. But taking that cat, which is out of the bag and putting it in the bag, I'm not sure that's possible, from both the practical, as well as a legal perspective, for one to make Java private. I just don't know and again, I'm not a lawyer in that regard, so my reaction, my personal reaction is that the scenario you described, I don't see happening in the Java space.

Rick Tobias: Okay, thanks a lot, thank you.

Grady Booch: Now not to say that Mr. Ellison is not going to find ways to leverage his purchase, you know, certainly, he's going to do that.

Rick Tobias: I would have to agree with you. I think the UNIX movement from the 60s and 70s is probably a better example than not.

Grady Booch: Yes, yes and we know what happened with, you know, people like SCO and others who have tried to assert, you know, rights that closed it down a little bit. Go look at Groklaw and see what's happening there. But, I would not be concerned about that happening.

Rick Tobias: Thank you.

Grady Booch: There’s an arrow of time here and it would be very difficult to undo it.

Any other questions?

Coordinator: (Bill), your line is open.

(Bill): Thanks. I didn't expect the opportunity to get to ask two. Back to where you were reading the abstract and thinking back to how you started about software deliveries, not just more open, but it's more global and more collaborative. And I’m curious, along with our company and certainly much more global and collaborative than it used to be, and so now we are dealing with things like conversations over here take place while the other half of the team is sound asleep in Asia, and we have to wait 12 to 24 hours to get turnaround time.

Collaboration ends up being done by multiple people in multiple countries with varying commands of the English language, which, at least in our company is the common language.

I'm just curious, it's not all it's cracked up to be and it's not always cheaper, because the overhead in trying to manage global development can be significant. So, I'm curious, first of all, just your general opinion of what my comments are and also, are there software solutions which make it better. I mean the Wikis are in a sense, at least in one sense, no more than use-net news groups, with so many people "collaborating," that it's hard even to get a consensus and some things are done better with small teams who are all working together and see each other every day and so on, so...

Grady Booch: Absolutely...

(Bill): What do you think?

Grady Booch: I wish you had been with me yesterday in Denver, because I gave...

(Bill): Oh.

Grady Booch: ...an address to the IBM share conference on that very topic.

(Bill): Oh...

Grady Booch: And (Robin), help me out here, but I have a slide deck which can be publicly distributed. Is there a way we could, you know, you guys could post that and make it available to people...

(Bill): Oh.

Grady Booch: Because it addresses your questions directly.

((Crosstalk))

(Bill): It might be very interesting.

Grady Booch: Sure.

Woman: We could do that.

Grady Booch: So here is the short answer on that. First, I am incredibly sympathetic to the things you’re describing. It’s not just, you know, finding the cheapest person to do something, but there are the human issues involved with this. It is an issue of share communication, and how does one even, you know, talk across cultures, because you don't have the same level of trust if that person's in the room.

There is also the challenge of serendipitous communication. If I've got somebody in my office that I work with, I have the opportunity to, you know, meet them for breakfast, or meet them at lunch, or do whatever and on a walk, and I have the opportunity for a watercooler conversation, where I can build trust with them, but then along the way we may say, “Oh, I've been thinking about this.” and you find opportunities for solving problems you could not have thought of before.

The downside of it, the reality, of course is that there are a variety of economic pressures that push us to bad distributions. So I think the cat's out of the bag on that one, as well, too.

So, the question for me, from a technical perspective, “What can one do to build a virtual meeting place? To provide the same kinds of meeting forums that I would have were I to bring all of these people together.

Windows 2000 is an interesting study, in that it's the first operating system release that Microsoft had in which their development staff no longer fit within the same room. So you saw them increasingly doing net-meetings, they would have lotteries where people would say, we've got a stand up meeting, but you will sort of get a number and if your number comes up, you can attend the meeting physically, otherwise, you have got to do a net-meeting. So, they toyed with those notions, as well, too.

So the good news is there's a whole lot of stuff that's out there. There's, you know, blogs, and mailing lists, and message boards, and chat rooms, and dare I'd say, even social networks. IBM itself, because of our vast size, we have our own social network, called Beehive, which is a means of getting those serendipitous kinds of connections.

Well, I ought to point you to the Jazz folks, because the product answer, and I’m not a product person, so I'm not here to push you on it, but this is a notion of the shift we've called from extended development environments, to collaborative development environments.

And (Robin), that reminds me, there is a paper I could post on this, as well, too. So the idea is, is there a platform we could produce that allows teams to work in distributed ways and so Jazz, the Open Source platform and Rational Team Concert, the product manifestation of that, is one idea in that regard.

But, I would urge you to take a look at other domains, as well, too. We are not the only ones that have that. Do a Google search for a place called Gehry Technologies. This is Frank Gehry, the architect and he has a company that provides tools for collaboration in building buildings, a thing called the Digital Project Designer, which does some very similar things for architects, as to what Jazz and Team Concert does for software developers.

For that matter, if you want to go really wild out there, take a look at the kind of communities that are built in the World of Warcraft or in Eve. And we're actually researching that, to see if we can learn from those kinds of communities.

Lastly, and this is a little further out and you will see this in the slides, IBM has been vesting in virtual worlds, so we, IBM have a product called Same Time 3D, which allows you to have meetings in virtual spaces. We are eating our own dog food. Our last Academy of Technology meeting, Sam Palmisano said, “Can't afford it, given the current economic climate, because it costs us a few million.” We bring 3,000 people from around the world, so instead of flying people everywhere and paying for their hotel rooms, as we did in the past, we met in a virtual world.

And there is an interesting study by Linden Lab and IBM that showed that we spent one fifth of the cost and had just about as many meetings. So ask me for the set of slides, it will give you an idea of where I see the spectrum, the landscape of stuff here. IBM is doing some things in this space from a product perspective and I think there are some cool things on the horizon, as well.

And by the way, one last thing, in Second Life, I also eat my own dog food. I hold regular office hours in Second Life. Every other Thursday you can come meet me in Thornbridge in Second Life and meet with me. So, another example of how we might collaborate. I have been doing lectures to, in fact, in the forthcoming month, I’ve got a lecture to Lima, Peru, to Chicago and to Moscow. And so that’s one of the ways that I can shave off time and culture by meeting in this space.

(Bill): Thank you. So, how do we, how are you going to make the...

((Crosstalk))

Grady Booch: I have no idea.

((Crosstalk))

Grady Booch: (Robin), can you help me figure out how to do that?

((Crosstalk))

(Robin): Yes

(Bill): Like I've got an IBM passport ID, I've got IBM notification, so...

(Robin): We'll probably post it on the Talks to You site, so we'll have those links on where you registered for this teleconference, we'll post them there for you.

(Bill): Oh, okay, thank you very much.

((Crosstalk))

Grady Booch: I do have to tell you that AT&T apparently reported a cut fibre cable somewhere in upstate New York, so I have been off the grid for 48 hours. I'm starting to feel withdrawal symptoms. So, until it's up, I won't be able to get those slides to (Robin).

(Bill): Okay.

Grady Booch: It's really weird not having any Web access, it's frightening

Angelique Matheny: We have one more question, Grady, before we let you go...

((Crosstalk))

Grady Booch: Sure, thank s.

Angelique Matheny: I know that you have a hard stop, so if we can go ahead and open up the last question line. Thank you, (Bill).

Angelique Matheny: Are you there?

Man: Yes, hello.

Grady Booch: Yes, we hear you.

Angelique Matheny: Go ahead.

Grady Booch: Please.

Man: Okay, great. Sorry, I joined late, so I don't know if this has been discussed before and so I will have to go through the recording. But, my question was, I just heard about this new announced consortium of IT software quality that's sponsored by the CMU, SCI and Object Management Group. Can you tell us something about that?

Grady Booch: No, because I don't know anything about it. Sorry.

Man: Okay.

Grady Booch: I know of its existence, that's about all I know. But I really don't know, you know, what their intent is or where they're headed. So, sorry, but you asked me a question for which I have complete ignorance.

Man: Okay, all right, thank you.

Grady Booch: But, I think it's a good idea.

Man: Well I guess maybe I could follow-up...

Grady Booch: Sure.

Man: ...as things, you know as these procedures and methodologies that we use for developing software mature and our understanding of what makes them work matures. You know people are always, especially, I am in the contract world and some of our government clients are very adamant, and ask, for example, having CMMI certifications and such...

Grady Booch: Yes.

Man: ...you know and we invest millions of dollars getting certified, and then, you know, a new way of looking at the problem comes along. You know, in your experience, what is the cycle of this? You know, should we be looking at sort of revamping the way we do things every three years, the way we throw out computers?

Grady Booch: The thing about software is that it accretes in its mass and its value. So throwing things out is almost impossible to do. Google has a growing Legacy problem, Facebook and Twitter have a growing Legacy problem. So, I think the economic forces of software are such that we can't really use the same model as our hardware systems, because there is not that same fungibility.

What I have found in practice and EBay is a good example of this. They're one of the few companies that I’ve seen that does this really well. It's a notion of not blowing things up every so often, as you were suggested, but rather continuous re-factoring. It's a matter of taking the systems that I have now and finding ways that I can harvest from them that I can find ways to bring them together in ways I couldn't have anticipated before, to find opportunities for economies of scale in them and to add energy to make them simpler over time.

So this notion of continuous re-factoring is one that I think is the key that one has to deal with in mature software systems. Now, that's not to say that one shouldn't do some wild innovation and large companies need to do this. You need to have some, you know, Skunk Works kind of folks who are unconstrained by reality, so they can be off on the fringe trying these things. This is also why I see this bifurcation in most enterprise systems, where you see a core architecture that's well trusted, well understood, well architected.

But, you architect it in such a way, that you can do things on the fringe very rapidly in almost disposable ways. But, you have mechanisms, through architectural governance that let you say, “Well, you know, we don't want to keep throwing these things away, let’s harvest from them, learn them and embrace back into this core over time, so that we provide a platform we can grow on and provide a significant advantage for (unintelligible) as well as our competitors can.

So, do keep - two takeaways from this. Continuous architectural re-factoring, I think, is the key to what you're doing and treating architecture as an artifact as a means of governing that over time, means you have to deal with both organizations, as well as processes and policies to make that happen.

Back to the quality thing. I did Google it to see who was involved here and two of the principals, I know them well, are Capers Jones and Bill Curtis. They're both really good guys. So, you know, I sort of know what their work is. My guess is that it is going to be a continuation along that end. I don't know what their intent is so I can't comment on that. But in parallel, I'd urge you to take a look at a group called the International Association of Software Architects, I-A-S-A- something, something.org, go Google it and you'll find a similar organization, similar in the sense that they're worried about building a body of knowledge in a practice for architects. And I'm encouraged to see these things grow. Great question.

((Crosstalk))

Man: Great. Thank you.

Grady Booch: So (Robin), I should probably turn it over to you to wrap this up here and I guess...

(Robin): Yes...

Grady Booch: I will give my thanks to everyone for coming and joining. Thanks very much.

Angelique Matheny: Yes, thanks to everyone who joined the conversation and thank you, Grady for taking time out to talk with us and being here for this Rational Talk to You conversation. We'll make sure that those resources get posted on the Rational Talk to You site, that's www.ibm.com/rational/talks. Our previous teleconferences will be there, as well, and we'll make this available for replay in about a week or so in a MP3 format, so be sure and check with us and we'll include all those links to the resources.

We would also like to thank you, our audience, for your interest in IBM. We hope to see you back for another one of our events in the near future. Thank you very much. Talk to you soon.

Coordinator: Thank you for participating. Today's conference has concluded, please disconnect.

Grady Booch: (Robin), I’ll stay here with you for a minute.

END