IBM Podcast

MATHENY: Welcome to this IBM podcast, A Software Factory Approach to Software Delivery. I'm Angelique Matheny with IBM. Toyota became famous with its maxim, no work without process, no process without metrics. But can we apply these same principles to the creative endeavor of team-based software development?

Well, joining us today is Rational CTO and IBM Distinguished Engineer Alan Brown to explore the benefits and potential pitfalls of a software factory approach to team-based software and systems delivery. Hi Alan, welcome to the podcast. Thanks for joining us.

BROWN: Thanks very much. I'm glad to be here.

MATHENY: You've compared software delivery to a factory, yet I've always thought of software as a creative endeavor. What do you mean by software factory?

BROWN: Well, what I'm finding that most customers is we start to work through some of the issues they face today with the current crisis, with the changes in the way software is being developed and delivered, with the oversight they're seeing from multiple sources, with the global look that they have now at their organization.

We're seeing some interesting changes that are going on in the world. Firstly, we're seeing a very different context within which software is being delivered. So, the pressures are much greater, the timescales, of course, are much shorter for software delivery, and the kinds of applications that they're being asked to deliver are very different than they were.

We're also seeing a very different organizational structure to those organizations right now. So, for example, they're very much more distributed, they use many more supplies of technologies, and they work much more with outsourced suppliers in other countries, with system integrators and other sorts of ISVs and providers of technology.

They are also trying to create these kinds of software with very traditional sorts of development methods that have been in place for 15, 20 years. So, the skills and the organization and the processes that are being applied are not well matched with these kinds of changes that they've seen over the last few years.

So, this struggle that organizations are having to deliver these new kinds of software capabilities with new technologies in a very different kind of workforce, it's very hard for them to see how they're going to apply these

techniques that they've always used in the past in this different contexts.

So, what they're looking toward is, how can they use different sorts of approaches to their organization, to the delivery of software, and to the kind of quality of systems they need to deliver than they did in the past.

So, many organizations are taking some of the ideas that come from the world of supply chain automation or the software factory automation approaches that have been applied in the past, or even the product line approach that we've seen from organizations that are trying to repeatedly solve similar kinds of problems and apply them to software.

So, what we see is the organization is looking at itself as an assembly of different sorts of components supplied by multiple organizations with a very well-defined approach to how demand management occurs and how production processes can be applied to take those assemblies and put them into the context that customers need.

So, this kind of view, this very different viewpoint that is needed in order to try and solve today's problems, is helping organizations to think very differently about what they do, how they do it and the kinds of metrics and measurements that they need in order to be successful.

So, this is a very different world, I think, than we saw a few years ago where organizations were much more homogenous, much more focused in very well-defined areas where there was a lot of consistency and control.

And this is leading to very interesting changes in the tools and techniques that they're applying. So, that's the context I'm finding, and within that context lots of interesting things are going on, I think.

MATHENY: Alan, what are some of the principles of factory design that are most relevant to software delivery?

BROWN: Well, I think what we're seeing is as we look towards this assembly approach we have to consider software as a collection of integrated pieces rather than simply one large application, one large delivery module.

And as we do that and as we think about different suppliers of those pieces, we have to have different development approaches, different design approaches, different control mechanisms for governing how the applications are created, and then also different approaches to quality and management.

So, it's affecting actually all the parts of the software

delivery process to different degrees, depending on how the organization is structured. Let me take an example if I can. If we take the idea of how we get software into production.

If when you look at the architecture of your application, it's actually a complex assembly of pieces, many of which you didn't design and build, but you're responsible for the overall delivery of the solution, then you have to take a very different approach to the risk that you're taking as you assemble the pieces, the versioning and managing of the different components and also how you work with the users of the technology to solve problems they have with the delivered software.

And that's something we are seeing as a really interesting change in organizations as they talk to us about their delivery styles. How did they address those concerns when they're used to being completely in control of the end-to-end process.

MATHENY: You mentioned the metrics earlier. Why are metrics key to the successful software factory?

BROWN: Well, it's an interesting question, because if we look at today's modern factories, if we take that metaphor perhaps a little further. If we look at those

factories, the way in which they operate is largely by a set of measures and metrics and a set of standards by which the different assembly organizations can bring their parts together, the components together, to integrate and to provide a solution.

So, it relies on two or three key aspects. The first is a standard infrastructure against which all these different suppliers can provide these components. And secondly, it requires a series of measures -- in other words, a series of metrics, a series of quality measures, a series of validations of the kinds of capabilities both functional and non-functional that are needed to allow you to bring these pieces together and deliver them as a consistent whole.

So, if we take that same metaphor and apply it to software, we have to also focus on the same ideas. What are the measures and metrics by which we can look at the assembly of pieces by which we can measure the quality of what we put into production, by which we can control the relationship we have with customers and suppliers.

And there are some obvious metrics like the quality of the delivered system in terms of perhaps defect density, perhaps defect [glide slope], perhaps the number of requests we have for updates from customers over a certain period of time.

But there are also some new metrics that I think are beginning to become more interesting. So, for example, we can look across suppliers and look at the quality of the components that are being delivered by suppliers and perhaps even begin to create some interesting supply chain competition between suppliers of parts to create the ability to have a marketplace of different providers of parts much more like a traditional factory has.

So, we already see organizations wishing to go in that direction and therefore focusing on the metrics and the standard infrastructure is to allow them to do that. So, this is why I think metrics and measures have become critical. They allow us to create that marketplace, to assemble the pieces and then to measure the quality of what we deliver into production.

MATHENY: You work a lot with the European customers. Can you give us a practical example of how the software factory improves production and software, basically what was the aha! moment when customers made the link.

BROWN: Well, that's very interesting, because there are a variety of customers in some different contexts, so let me take one or two examples and just give you a couple flavors of that. One situation that we see quite often here in Europe is that we have a lot of outsourcing going on,

much as we do all over the world right now.

So, in particular, in Europe they're using providers in India, China, Latin America and some other places. And when they're using multiple providers of technology in other parts of the world, a key question they have to ask themselves is, how do we control our relationship with that subcontractor, with that provider of technology, to ensure quality of the delivered artifact?

In the past it's been relatively straightforward to do that where there have been very fixed requirements to what's being developed and where there's a great deal of consistency on the delivery platform, the needs of the customers and so on.

As the world has changed substantially with much quicker turnaround, much quicker need for availability of solutions to new market needs, much greater pressure on the kinds of features that are required and the kinds of solutions that we deliver to market, as that's added more pressure to organizations, defining that relationship with their suppliers is beginning to break down a little and they don't have the control or the insight with their outsourced providers that they would like.

So, we have a number of organizations asking us very

specifically, how can we control and govern more effectively across that interface we have with our subcontractors? How do we measure what's going on across that interface with our subcontractors, and how can we begin to exert some control or perhaps even some pressure on the outsourced organizations to improve their quality, to improve their productivity and to lower cost?

So, we're seeing some interesting examples of trying to improve the processes by which those organizations communicate, by which they collaborate with the outsourced providers using technologies of course such as the Rational Team Concert technologies and others.

We're seeing some interesting questions around how can we create a dashboard, if you will, of the outsourced providers' capabilities and where we are right now with the delivery of components and assets from those outsourced providers. So, we're seeing some interesting activities around the use of Rational Insight to create these kinds of dashboards.

And then of course, we're also seeing some interesting demand management issues around how do we control new requests, new requirements that come into the system and then triage them so that we can ask effectively to these outsourced providers for what we need and then judge the

response that they give in terms of how much it would cost or how difficult it is to implement that change within the context of the systems that they have available to them.

So, the new and interesting ways of managing that demand through things like, of course, Requirements Composer, using DOORS and Requisite Pro and other sorts of technologies to analyze the impact of change. So, that's one set of customers.

Another set of customers actually is much more interested in just the quality aspects -- creating what you might call a quality management center or a test center around the delivery of artifacts from these different providers, essentially, creating the quality control department, if you will, from a factory.

And they want to do that in a consistent way with well-defined standards and of course, a lot of automation about how they run tests against the kinds of solutions that are delivered. So they're focused on creating that quality center, if you will, that quality management center, that allows them to manage and support the delivery of these different components into an assembly and then into production.

So we have a number of organizations working with us using

the Rational Quality Management technology and using a number of the components of our test solutions in order to understand and analyze what's being produced, and then to create the infrastructure that they need in order to take those solutions and put them into production. So we're finding those two very interesting models, as well as a few others.

MATHENY: Of course, the modern factory is tooled differently than the factories of the turn of the century. How are modern software engineering organizations tooling up to best support a factory approach?

BROWN: Well, again, if we look at this factory model, if we use this guidance, I think what happened over the last perhaps 10, 20 years with many factories was that they don't actually do much development or creation of the solution anymore, they actually are run by the R$D, the Research and Development departments, looking at what should they be producing into the marketplace. So, they're doing a lot of market analysis, a lot of product analysis and maybe a lot of prototyping.

They're then also run by the quality department that's looking at the suppliers and the kinds of components provided by the suppliers to ensure that they're of the right quality and they can be integrated effectively.

And then they're also run by project management and project oversight organizations to ensure the assembly of the pieces. And this is exactly the areas that we are seeing with the software factory. I was talking with one chief architect for a large organization just a few weeks ago, and he actually was saying to me that our goal is to be the best in delivering software in the business that we're in.

But, we're not interested in being the best in software development, because we're not experts in development, and in fact, we don't want to employ development people per se. We want to outsource development as much as we can, but we are responsible for delivery into production.

So, it's all of those areas that I just mentioned is where they want tooling, where they want automation, where they want best practices, and in fact, where they think they can differentiate themselves or others in the marketplace. Not necessarily being the best software coders, software developers, software architects.

So, I think that it's in those areas that I think the tooling is very interesting, and I think we're going to see interesting changes over the coming months and years in how Rational and the rest of the marketplace is looking at tooling in those spaces to help organizations create the

software factories.

MATHENY: Alan, I think we'll end with this question today. What are some practical first steps that customers can take to gain the benefits of a software factory approach?

BROWN: Well, as I say, this is an interesting vision and approach, and organizations are looking at this as the future of where they would like to be, their aspiration, if you like, for where they'd like to see themselves in a few years. So, in order to get there there's a few different approaches we've seen for how organizations can take the first steps.

So, one sort of style of moving in this direction is to start by looking at the organization and saying, how can I take perhaps a service based approach or a service oriented approach to creating the components that are necessary in order to become more of a software product line? What would those components be in terms of services and service delivery?

How can I perhaps extract those components or create a sort of component library that would take me there and in fact, how can I begin to use some outsourced suppliers to provide many of those components for me so I can remove the

obligation of creating and supporting those components.

So, for many organizations it started with a service oriented, an SOA approach. This was the direction they went in creating these service libraries, focusing on the components and starting to create a reuse culture within their organization that's more focused on assembly of components.

And then, of course, what the infrastructure would be to allow them to assemble those pieces, maybe through an enterprise service bus or some other sort of architecture. So, one style of moving in this direction is to start by taking the service SOA perspective.

Many organizations have taken the first steps, and now they are trying to take advantage of the investments they've made in services and SOA to start to move towards the software factory approach.

Other organizations are actually starting in quite a different area. They're starting by focusing on the business. Let's do some business analysis, some business process analysis, business process modeling, and start to optimize our business processes to see where we can then assume responsibility for key aspects of the business and where we can perhaps outsource key parts of the business

where they're not differentiating for us, where they don't really add value in the way we bring ourselves to market.

So, many organizations are taking this business process modeling, business process analysis view to begin to get a handle on where they can differentiate, where they can focus. And that may mean certain aspects of their business that they spent a lot of time on in the past, maybe some core applications that are tying up a lot of their resources...

...could be outsourced to cheaper labor markets or be replaced by packages or be controlled in some way to reduce costs and improve efficiency giving them the revenue then, giving them the saved amount of money and resources to work in other growth areas. So, that's two ways we're seeing this direction.

The other is to focus in some specific area like the factory area for testing for example or around perhaps demand management -- some very specific focused areas for the development organization where they'll see early return on investment.

For example, taking the resources they have in each line of business that are doing testing in certain areas, combining them into a single pool creating what we might call a test

factory, and then using them as a shared resource across their lines of business. So, that gives them at least a starting point for creating these centers as I mentioned that are key to how you would move towards a factory.

MATHENY: Alan, thank you so much for sharing your time today to discuss the software factory approach to software delivery. We really appreciate it.

BROWN: My pleasure, and I hope this has been very useful.

MATHENY: It's been very informative, yes. Thank you. That was IBM's Alan Brown, Distinguished Engineer and Chief Technical Strategist for IBM Rational Design and Construction Products.

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

IBM Podcast

[END OF SEGMENT]