

IBM Podcast

[MUSIC]

WEN: Welcome to this IBM podcast, the IBM WebSphere SOA appliances for optimizing your SOA. In this third of the series of podcasts, we speak again with Steve Craggs, Lustratus Research Ltd., on how SOA appliances help optimize TCO in your SOA project. I'm Ben Wen with IBM. Hello, again, Steve, and welcome back to the podcast series, and thank you all for joining us.

CRAGGS: Great to be here again, Ben.

WEN: Good. And I know that one of the things that we've been talking about here in our podcast series is the Craggs SOA adoption paradoxes. And maybe just for a quick overview in a minute, if you could just go through the paradoxes as you've seen them over the years of implementing and working with implementers of SOA.

CRAGGS: Sure, Ben, yes. Okay. So three paradoxes that we've come up with talking to lots of people have done SOA adoption at various levels. And they really all come about as you start getting serious with SOA and start going beyond sort of departmental use into an interdepartmental or a value chain type use.

The first is scale, and here we're talking about that scale of being able to spread across that value chain of getting the value of using your business services all over the value chain, in your partners, in different departments.

SOA needs scale to deliver the value that you're putting it in for, really. But on the other hand, SOA hates scale, because it brings about issues of design skills and infrastructure needs and manageability challenges. So scale is definitely a bit of an issue.

The second one is security. And you know, to get that SOA adoption going, SOA adoption needs freedom and needs people to be able to use the business services that are being created, interoperate with each other's services. That's where value comes. That's with agility comes.

But on the other hand, SOA security hates the freedom, because now you're talking about opening up different parts of the different applications and different services to a wider audience where you may not know exactly who it is who's using it and what their security level is and all those sorts of things. So that's the second one, security.

And then the third is cost. And the fact is SOA can reduce your costs and can definitely bring down the cost of

implementing and maintaining IT solutions. But if you're not careful, SOA can actually increase costs. And I think it's this last one that we're probably going to spend most of the time on on this podcast.

WEN: Well, certainly can be scary from a business manager's perspective, looking towards some of these technologies around SOA to help drive business agility and provide business benefit and lower TCO. So that's exactly what I'd like to be able to understand a little bit more about what you see and what you're recommending for folks. So what are some of the key concern points around the cost of SOA that our listeners should be aware of?

CRAGGS: Well, I mean, first of all, it's just the pure cost of actually implementing. You think of the skill [INAUDIBLE] for start, and you're going to have to have different brokers or whatever you want to call them, different connectivity points across the different departments and partners who are going to be using these services.

So you're going to have to have middleware there to perhaps switch between different data formats and to have to worry about things like authorization and authentication. You know, a lot of different challenges that have to be met there and they've all got to be designed and programmed,

and that's going to need a very much distributed skills.

And it's always costly to train people with new skills. You can go down some sort of a center of excellence route, of course, but even that has costs attached to it. And the issue is that the benefits all come when the SOA is widely spread and running smoothly.

And in the early part of the cycle, that's where you're having to put in a lot of these costs to train people and to consider the security and management issues of this SOA adoption. And these are all things that are going to drag down the overall benefits you're going to be able to achieve.

WEN: Certainly things to think about. And I'm sure there are ways that you've seen folks mitigate some of those concerns up front.

CRAGGS: Yes, definitely. I mean, one of the things that always begets me is that there are some new technologists who love technology, and there's always a natural tendency for any department or any organization to say, well, don't worry, we can program that. We can, you know, we'll learn all about this stuff, we'll do it. Unfortunately from a money point of view, that may not be the best answer.

And one of the answers that I've seen work very well with companies is to say, well, can we get some level of uniformity? Can we get some level of standards of, almost a package just like you'd have a workstation image that is used whenever somebody comes into your organization and you need to get them a new workstation, they all get PowerPoints, they all get Microsoft Word or whatever it happens to be and it's the same picture.

Is there a way that you can actually present this middleware need in a uniform way and have some sort of standard implementation that you can just roll out with minimal skills to any location or any organization that wants to be part of your SOA?

And the answer, or at least one of the answers I've seen used is the idea of having a specialized appliance, because the appliance, via nature, is preloaded with software. It's all installed. It's all there, and usually has a fairly simple configuration capability which is all that you really need to do to start using it.

And this idea of being able to plug an appliance in and turn it on with a little bit of configuration perhaps done remotely. And this all of a sudden really takes a chunk out of those costs in terms of the costs of having to

actually, you know, distribute that middleware and set it all up.

WEN: So a lot of interesting things that you're saying there. If I can reflect back, it sounds like, then, an appliance, because it's prepackaged and preoptimized for a particular set of use cases, in this case an SOA appliance, as we talked about here in our title, can mitigate some of the concerns around the complexity, around the skills, around the training, because it simplifies a lot of the underlying complexities and therefore, potential cost if an implementer is not careful, try and mitigate those.

And because the appliance has that sort of prebuilt, retargeted configuration driven instead of program or developer driven architecture, that that could help take out some of the concern and allow customers not only to lower those overall costs, but also if I could hazard something here, is also increased the rate at which an enterprise can implement the next iteration of SOA.

CRAGGS: Absolutely, absolutely. And that's really what this is all about. This is all about SOA adoption and helping people to get benefits more quickly. You know, when we look at SOA adoption, what we see is if you were to draw a graph of what the customer's expecting to get back

in terms of agility and visibility and cost reduction and whatever else...

And we actually think that a lot of customers [dials off] on that upward swing that they're looking for, and then they come upon these very different pressures which drag them back, things like the skill design pressure point that we mentioned, things like this middleware issue, got to have middleware everywhere, things like security and performance and management and all these things. And in order to get the best possible cost benefits and reduce the TCO, you have to try and address these things.

Now, the uniformity of the appliance approach is certainly a strong thing in its favor. But actually, I think there's even more to it than that. And when you actually put an SOA in place, and it does create the management issues, not management as in organization management but operational management...

And the issues it tends to create are, it's no longer like operating in a monolithic environment where I've got my application here and if an end user has a problem, I can go in and see what the application is doing, because in an SOA, an application consists of a whole different set of business services that may be spread across departments and beyond. So this issue of trying to manage the environment

becomes a real problem.

Now, one of the things you can do if you go down this appliance route is not just get the uniformity you talked about, but also get some level of concentration of resources. So you can imagine a department using SOA, rolling out to other departments.

And we've seen this many times, I have to say, where they've now got a broker in one department, a broker in another department, and they've got a number of servers now who are doing SOA traffic handling and looking after the XML parsing that comes about when you do SOA, all sorts of things like that. And the reason that they're doing that is because they don't want to put that all on one server because the server will be driven into the ground.

So what is attractive about the appliance option in this place is to say, well, hang on a minute. If I actually have a dedicated machine that is going to do all of that stuff, and is going to do all of the mapping between data formats and the checking of the traffic and all of those sorts of issues, then I may be able to, for instance, replace ten instances of a broker with one instance of an appliance.

Now, if I can do that, it will have a major impact on the

operational support costs. You know, the idea is it's a heck of a lot easier for me to manage one standard appliance or...let's take it up a bit, let's say five standard uniform appliances as opposed to managing 25 or 50 installations of middleware in different servers.

WEN: That makes a tremendous amount of sense. So it's the operational characteristics that otherwise would be spread across an enterprise, everything we talked last time about security policy, you're mentioning here around brokering and message handling and XML parsing some things that otherwise sounds like if someone's not careful and dominate in operational costs but also sort of simplifying as we were talking about before.

These are some of the areas that it sounds like that an implementer can go and help [compress] not only time to availability but also compress the operational costs, compress the complexity and simplify the overall architecture by using an appliance that's dedicated design for handling SOA.

CRAGGS: Yes. I mean, I think there's one thing I would like to add on this, which is in case any of the listeners think I'm saying, right, so don't use software, just use appliances, that's not actually what I'm saying.

What I'm saying is I'm saying there are certainly situations where it's going to make more sense from a cost point of view to have functionality carried out by an appliance. You know, you just mentioned some of the things, but some of the, I would call it the grunt work that has to be done.

Things like going through massive XML streams and parsing them and doing transformations and that sort of, these are things which are pretty compute intensive but really, you don't want them sapping resource from other key business applications. So, offloading them makes a lot of sense.

But there will be situations where you will have parts of your SOA where you don't want to offload it to an appliance. You may well be looking at it saying, well, actually here, I would like a different level of flexibility, or perhaps in this level here, the load characteristics don't warrant the use of an appliance. And there will be a decision to take about whether you use appliances to some part of the middleware and whether you just use software.

But what I'm saying is we're finding a lot of the customers we talked to are finding benefit from looking at these two options with these different characteristics and deploying them where they make the most sense.

WEN: That sound like a very pragmatic adoption for SOA. I think what's interesting around whole concept of appliances in general is that one of the reasons that we've seen, I've seen the ability to migrate certain functionality into some type of dedicated appliance, this is across not just SOA but across other areas of computing...

...is that there's a certain amount of regularity, and, you know, ability to see the common pattern and put that in by this hardware component that says that you know what, everybody understand these standards well enough that we can now optimize a solution.

And be able to say, here's the delivered component that has exactly what needs to be done because we've all agreed on these standards and now, as implementer, you can have an optimized component. And I with you, you're saying that there are areas where there's flexibility or different volume concerns or there's experimenting happening, that they need to have the full flexibility of the software, the awesome software component that exists out there.

So with that, if you think it's a good characterization, thank you for giving us more of your insight around SOA adoption in the third part of our series, we're talking

about overall cost and managing the overall cost.

So this is Ben Wen from IBM with Steve Craggs from Lustratus Research. And we've got one more in this series of these podcasts, to discuss the overarching summary and view of handling these three paradoxes of SOA adoption, utilizing appliances. Thank you for listening, and hope to hear from you.

IBM Podcast

[MUSIC] [END OF SEGMENT]