

SOA BREAKFAST DISCUSSION TRANSCRIPT  
5<sup>th</sup> September 2007

MODERATOR: Peter McTaggart, CTO, Renewtek

PANEL: John Brand, Research Director, Hydrasight  
Trevor Thomas, COO, Renewtek  
Tony Carrato, Worldwide Chief Operations Architect for SOA, IBM

**Peter McTaggart**

- Peter has 14 years experience in the IT industry in both management and consulting roles.
- Peter's current role is as Chief Technology Officer at Renewtek where he is responsible for developing technology capabilities to meet the needs of Renewtek's customers as well as managing the product development and innovation within Renewtek.
- His recent roles include Manager, Applications and Architecture at Hawker de Havilland (a Boeing company) where he implemented an SOA and enterprise service bus that integrated enterprise applications in a complex process manufacturing environment.
- He has also held technology positions at a number of application and middleware software vendors.

**John Brand, Research Director, Hydrasight**

- John has 20 years' experience in IT and is widely regarded as one of Australia's top IT analysts.
- John spent five years as a senior vice president and industry analyst with global IT research firm, META Group.
- John was a META Group lead analyst and subject matter expert in the areas of: enterprise content management, enterprise collaboration, digital asset management, and digital rights management.
- He is regularly invited to speak at industry events and is often quoted in the press around the globe.
- Prior to joining the META Group, John spent four years as Online Services Manager at leading Australian Stock broking firm, as well as four years with new media marketing firm, Empire Ridge.

### **Trevor Thomas, COO, Renewtek**

- Trevor has over 24 years of Information Technology experience.
- Trevor is currently COO of Renewtek, responsible for managing delivery for Renewtek clients.
- He is also a principal consultant with Renewtek and engages clients around SOA architectural and governance matters.
- Prior to Renewtek, Trevor spent over 10 years with GSJB Where where he held a variety of roles in development, architecture, ultimately becoming Head of Software & Technology Architecture.
- In that role, Trevor was responsible for aligning IT delivery with the business strategy.

### **Tony Carrato, IBM**

- Tony has over 30 years of Information Technology and Communications experience.
- Tony is an Executive IT Architect and a member of the IBM worldwide Enterprise Integration Solutions/SOA Advanced Technology team.
- In that role, he is the Worldwide Chief Operations Architect for SOA, where he has technical responsibility for developing and managing the IBM SOA architecture team.
- Prior to joining the SOA team, he was the Architecture Profession Leader for IBM's Global Business Services (GBS) in Australia, as well as the co – Technical Practice leader for Australia.
- In that role, he obtained a GBS Professional Development Framework (PDF) level 6 (the highest level) certification as an IT Architect.

## **Panel Discussion**

### **Peter McTaggart:**

There is a tremendous amount of information in the market place about SOA, about the benefits that it can deliver. Business flexibility, agility, and adaptability to address some of the challenges you have heard earlier. Every day it seems there are hundreds of articles published about SOA technologies, the latest and greatest approaches, best practices, how to demonstrate the returns in ROI, and the hidden secrets of SOA. You name it, there's probably an analyst, a white paper, a website, that's dedicated to some aspect of SOA; spewing forth article after article of information, and opinion after opinion. You're all here today, so I am assuming you're interested as well.

So is all of this self serving vendor hype, around a set of technologies? Or does it actually offer some new and fundamental way that business and IT can engage to develop agility, flexibility and responsiveness in organisations? And how do you go about doing this?

I'd like to just park those questions and hopefully over the next thirty minutes or so we can draw out some of the answers. I don't think there's any single answer to those questions, but we have an esteemed panel here who have a lot of experience and can provide insight into some of the answers.

Tony, what's different about SOA this time? Is it just EAI dressed up, and renamed and given a different coat of paint?

### **Tony Carrato:**

I think that there are some things that are not different. SOA reminds us the A in SOA stands for architecture – not that there aren't people that think SOA stand for Sleeping on Airplanes (which is what I do!!). It reinforces the notion that architected solutions are the how you get there. The big thing we see though, is that absence of discipline in that Architecture, people do in fact run back to EAI, and forget that the key issue is business flexibility. So, if you don't think about business flexibility and work hard to get it, you won't get it, that's what SOA offers, and I think that's the big difference.

### **Peter McTaggart:**

Right, so it's the disciplined architecture approach not simply the application of a technology to solve a technology problem, perhaps.

Trevor, you've been involved in a number of SOA recommendations from architecture and consulting background. Where are organisations being challenged in adopting and implementing SOA?

### **Trevor Thomas:**

One of the key areas, particularly in some recent consulting engagements that we've seen, is that organisations are certainly thinking about SOA. But when we actually examine where they are at in terms of their current readiness for adopting that sort of technology. There are some significant areas for improvement that some organisations really need to consider, in our view; first, or as part of the SOA journey.

Possibly a key area for example is understanding where their core processes are executing, usually on their legacy environments. In other words don't ignore legacy environments when having a conversation about SOA. I think that's one of the key challenges some organisations we've been dealing with are grappling.

### **Peter McTaggart:**

O.K. John you've recently been doing some research into the market place and what people are doing with it. What's happening around the place at the moment?

**John Brand:**

Well it's interesting, and for those of you who haven't already done the survey we coincidentally happen to be running an SOA survey at the time this event was being held. So if you want to participate in that survey, and you haven't already it's [www.hydrasite.com/surveys/soa](http://www.hydrasite.com/surveys/soa). The results that we're seeing at the moment are that about 46% of organisations that have responded are using SOA already. About 35% are not, and about 20% don't know whether they're using it or not (which is kind of interesting).

The other side of it though, is that the main driver or most common driver for it is integration. That is the main reason that people are deploying SOA. The other side of that is, that it is not displacing any other form of integration technology. So nothing else is coming out of the organisation as a result of deploying SOA. I wouldn't even call it technology; it's another approach to technology that people are engaged with to provide some level of benefit around integration typically.

**Peter McTaggart:**

Right, so this goes back to the first question, is it just EAI, and is ESB just the new EAI? If someone picks up an ESB, are they doing SOA? What else do they need to do, actually to be said to be doing SOA proper?

**John Brand:**

Many business people by the way think SOA stands for Serious Overuse of Acronyms!! ESB, EAI....

**Peter McTaggart:**

Maybe I should just clarify; ESB, Enterprise Service Bus (forgive me I'm a technologist!). Tony...

**Tony Carrato:**

One of the issues, and we do see this a lot, is people go buy some fine product labeled ESB, and put it in; or take a product that is already there, and rename it ESB. But they don't take anything out, and I've never seen a case where adding yet another style of integration, reduces complexity or cost. So, one of the things when we talk to people, and I do this literally all over the world, about SOA, is I go to the issues of "is the senior management in the IT organisation aligned with the business and actually trying to get some business outcome", as opposed to an interesting information technology adventure. If the answer to that is no, then I try to set a new record heading for the door.

The second thing is - is that senior management prepared to in fact bring discipline to their organisation? Because otherwise somebody will, I promise you, find reasons to hang onto and every bit of that old technology will be essential. There will be no way to get rid of it. And you know what? The costs will go up, the complexity will go up and your risk will go up.

**John Brand:**

And in many cases that's the result anyway. You're adding an SOA layer of technology on what you already had. So you're increasing the complexity in your environment by going down the SOA route in many cases.

**Peter McTaggart:**

But isn't one of the benefits SOA touts removing vendor lock in, inter-operability between systems and technology. So surely putting one in, doesn't that make it easier to make all these things talk together?

**Tony Carrato:**

Only if you're prepared to do that. Again people will find all the reasons there are, not to get rid of anything. Trevor, I assume you must be seeing that?

**Trevor Thomas:**

Certainly, I think when we start having conversations about what the current environment some organisations have, it often becomes apparent some organisations have a lot of product

running on a lot of technology and not a clear transition plan. And simply as the panel has been saying, adding another layer of technology without actually considering what's your transition plan for the existing set of products is a challenge. The question is: "Do you have one?" And to be frank, often the answer is "kind of" or "no". What that leads to is: we now have an ESB, or we've introduced some SOA enabling technologies; but what is that doing to the suite of products that are currently on your mainframe? Where are you taking those? As part of that whole equation, rather than just simply saying well we now have an ESB so "off we go, let's create services."

**Tony Carrato:**

Let me throw in another acronym Peter. JBOWS; Just a Bunch of Web Services. They're a bunch of people that will get out there and whip out your development tool of choice, wrap up existing systems and go "Wow, I now have an SOA". When actually all you've got is one more complexity. So, I will come back to something we're seeing, which is governance. And this is about the people that are running the shop, being prepared to exercise, what I think is very much their accountability to bring return to these investments.

**John Brand:**

That's an interesting comment because that's certainly what we see. From our perspective we see organisations who embark on an SOA initiative and it becomes a large architecture based project. And they don't often go that well - simply because they are attacking the wrong problem. They are trying to develop standards for their organisation, then the IT organisation has to comply with those standards. So it's about compliance, and when you're focused on compliance, everything tends to stop. You know, compliance is a great way to make nothing happen.

Then you have to look at the next level of maturity, which is governance. Where, you know what you're going to make happen, and why you're going to make it happen. A bunch of web services can be a really useful thing for some organisations, in some applications. So there's nothing wrong with going out and deploying a bunch of web services. The research actually shows a lot of organisations are still doing just http: as a way of deploying a SOA in the initial stages. I have various definitions of what SOA actually means. Once you've mastered the art of governance then you start looking at performance. How do we really drive performance out of it by applying the right technology solution to the right job? And understand that we're going to get some return from that investment long term.

**Tony Carrato:**

We haven't talked a lot about maturity Peter. If any one wants an independent way of understanding maturity, go to [www.opengroup.org](http://www.opengroup.org). Some of you may well be familiar with open group as a body that talks, among other things, about enterprise architecture. There's a project called Open Service Integration Maturity Model, OSIMM. That provides a way of measuring maturity in a vendor neutral fashion that will give you a cross of seven dimensions a way to look at where you stand. I think John talked a little about the maturity issue as well.

**Peter McTaggart:**

O.K., we'll just park that, and get back to those issues a bit later.

I want to go back to something Trevor was saying about not getting rid of things, and the legacy systems. What do you do with your mainframe, your back end systems that have been there for 10, 20, 30 years. From personal experience I have seen systems that have been there for 30 years. How do you bring them into the SOA world? Do you just wrap them in a Service Bus and try and expose services off them and then split them apart? What are options there?

**Trevor Thomas:**

Certainly some organisations do that, they see it as a skimming exercise. We're simply going to skim some core operations off our legacy environment, with varying degrees of success. I think certainly in terms of some of the organisations that have been turning to us for assistance, they're very early in their SOA journey. So often some of the first conversations are around do you have the right infrastructure? Speaking frankly, do you have the right

plumbing in place to do this for starters? It's all very well to say we've got this shiny new ESB, but have you got the right adapters into the mainframe environment. Do you have the right operational capability around that infrastructure? Surprisingly (or maybe not) quite a few organisations haven't really got the base line right. Simply saying we've got a bus now, so we should be OK. But actually we haven't got the right supporting infrastructure to actually support the conductivity to the mainframe; probably, in other areas, the right process to support that as well.

**Peter McTaggart:**

Tony, you might be able to jump in here. Is that a bottom up approach, taking your existing systems, just looking at what they can do and service enabling? Or should you be going back to the enterprise architects, the SOA architects in the business, and saying "what are your services?" And can we perhaps extract some value out of these back end systems to build our SOA?

**Tony Carrato:**

Well we argue there's a three pronged approach.

One is bottom up – what can your existing systems do? Don't ignore that they aren't going to go away quickly, they may or may not go away at all, depends whether they're useful.

The second approach is a business domain oriented approach that typically starts from understanding your business, understanding the processes and the data that moves through those processes. And that calls out the point if you aren't reinforcing your business analyst communities in your organisation, do. Those people are back to being really important, the folks that live and breathe business processing data.

The third prong is where one that we talk about being goal oriented. If your business has goals; like; increase revenue 5% from fee income. Then you have to ask what in your IT organisation supports your ability to do that. So typically in our model, it's a methodology called Service Oriented Modeling and Architecture. There's a bottom up analysis, a business as is top down. But don't forget the goals, if you're trying for flexibility and you don't examine your goals; once again you'll just build systems that can't get you there.

**John Brand:**

I think that's a good point from our perspective, the research that we've done. The orientation for most SOA initiatives are still around, it should be called more component interface oriented architecture. The service aspect of this is largely missing. The definition of what a service is, for most organisations, when you're talking about SOA is very loosely defined. And when you're talking about business services it's almost never defined. Down to that next layer of technology and how you deploy them. We definitely say that is the trend that we're seeing, is somewhat of a recognition that there needs to be a greater focus on the service aspect of SOA.

**Peter McTaggart:**

So speaking of the business and the way they talk, one of the key benefits and features touted by going down the SOA path is, changing that dynamic between how IT and the business operate. And being able to have IT aligned with business strategy, and its goals and objectives. In that sense, is SOA just a recognition of what should have always been there? IT actually serves a business purpose and not a technology shop for people to play around in.

**John Brand:**

The most common starting point that we've seen for organisations is services like, create new account, or get latest price. Those kind of services; that's generally what they've been doing for years with those kinds of services anyway.

**Peter McTaggart:**

Given that, how do you talk to the business about SOA? You know, if I go up to the Business Manager and say how's your SOA going? What sort of response should I get, or would I get?

**Trevor Thomas:**

Well I think we were talking earlier, before the session started, around walking into a business and saying “what you need is an SOA”, is probably not necessarily the right approach. As Tony was talking, it's about business information and processes. And engaging with an organisation about how to improve their business architecture versus what you need is another technology. This has certainly got to be the starting point, I would think.

**John Brand:**

That's certainly one of the benefits that we've seen for those organisations that have gone down the SOA route today, has actually had very little to do with the technology. It's actually been to change the conversation with the business, even though the business doesn't understand what SOA is, you're not focusing on the technology layer. You're starting to focus more on the information and process and business by default. That's where you should be starting with. So it does change the nature of the conversation even though it may not be immediately clear to those you're talking to, what it is that you're talking about.

**Peter McTaggart:**

Do you still need to educate the business? What's the level of education that the business owners need to know about services?

**Tony Carrato:**

Here's one way to tell. When you ask someone about business agility in the business, and the answer you get back in terms of modification to an IT system, which is really, really common that tells you right there that you're having the wrong kind of discussion. Because what you've done is we've trained the business people to think in terms of that 'RPG running on the I-series', or the 'Oracle system running on the Sun', or whatever. We're not thinking in terms of what other actual processes or data.

A fascinating thing to ask most owners is “do you actually understand your process?” And if you define *understand* as “it's documented in something”, the answer is very commonly (and surprisingly) “no”. That really makes it quite an interesting conversation. You have to typically start by, can we extract from a bunch of peoples' heads what the business roles are. While they're probably obvious to anybody in the line of business, they're probably not documented, in many cases. Now in a regulated business they have to be. But finding them is often interesting.

**Trevor Thomas:**

That often leads back to this conversation about legacy systems, because for many organisations, frankly their core processes are embedded in the legacy environment. It's in the heads of the technologists as to how they actually execute some of their key processes.

**Peter McTaggart:**

Currently a lot of these businesses are functionally silo based businesses. The IT guys are walking up to the business people and talking about breaking down the silos and implementing an enterprise SOA. But, is that the right conversation to be having with the business people, what sort of business people, what level, should you be having that conversation with? I imagine for your business function owners, it's perhaps not relevant, or quite a threatening conversation to be having about implementing their end-to-end business processes, and they're just a small part of that.

**Tony Carrato:**

The advice I usually give is: start with someone who owns a pain point. If you try and start across the entire enterprise, you're going far too broad. And it's unusual to have that be terribly successful. However, most businesses have multiple business owners. Those people typically have business issues like, I'm not flexible enough.

The other interesting question is getting someone to define what flexibility actually means to them. I tend to use, and we'll go back to John's point about most of this really being the integration, a working definition of flexibility is given a business challenge you have more than

one available response. If the only available response IT can make a change to the core system, I would argue you aren't flexible.

So that's why people actually attack integration, it is that they're trying to shorten that IT cycle. But, if you find someone who owns a business problem; that's probably the right spot to have the discussion.

**Trevor Thomas:**

I was going to add too that sometimes IT can, and I've seen this done, rush into "if we build it they will come" sort of model; without actually being clear about, in terms of business domain areas, where does this really lie, is this really needed in that area? Without actually knowing what the lines of business are, where the main re-use can really be applied and generate the most benefit.

**Peter McTaggart:**

I guess that brings up the point about ownership of SOA. You're talking about starting in a business silo, and doing an SOA project for a particular P&L owner. But as these services, and the re-use of services, which is another benefit we're looking at to get out of SOA; in terms of cost reductions and productivity and time in the market, responsiveness. Who should own it? It's obviously IT would love to own their technology, but they're not really the owners of the business problems. It needs to be some joint ownership between business and IT. What sort of structures have you seen work, and put in place to provide a level of ownership to SOA.

**Trevor Thomas:**

I think it's critical, in terms of the start of this process to engage with the P&L owner, to be clear about what is the best value for them. And what is the best model for them? As I said, IT can make a set of assumptions of what they believe they should be, based on the existing system versus where the business sees it.

**John Brand:**

We see some really interesting results in research, and there are a wide range of expectations of who owns and who should be influenced by SOA. The closer you get to IT the more IT thinks that they are the master of SOA and the tool vendors of SOA who should be the main influences of SOA. When you start going up into IT executive levels, they're more looking at the software infrastructure vendors as being the ones who should be responsible for SOA, and delivering SOA, in the organisation.

When you get to the line of business manager, and executives in the organisation outside of IT, their expectations is it's the major application vendors who are going to provide SOA solutions. So there's a very wide range of expectations in organisations as to what SOA is, who's going to deliver it, how it's going to be managed. Again, if it's not well architected from an IT delivery perspective and an operational perspective, you'll once again have a bunch of technology solutions that are thrown over the wall to IT to try and manage this mess. The complexity certainly isn't decreased if you don't attack it the right way in the first place.

**Peter McTaggart:**

I guess one real test of ownership is "who gets to pay for it". And who gets to pay for it when someone's re-using a service that's been created by some other project in the business?" Tony, do you want to say what you've seen.

**Tony Carrato:**

Most of what I see is not really very different than any other IT funding model, which is typically 'first consumer pays', is very, very commonly the case. Some organisations are adopting more of a shared funding to the infrastructure, because at some point there's a cost there - depending on what you're doing, can be fairly large. You've got three models that I've seen for service ownership - leaving aside the definition of a service for just a moment.

One is the "I paid for it, I had it built, no one else will use it, thank-you very much - stay off of my turf!" You don't see that one very often, fortunately. Another model is "I'll build it, I'm

actually perfectly happy for people to consume my service, but I'm not prepared to consume anyone else's". That's a statement of trust. I trust what I control, and I'm willing to let you use it because I control it, but I'm not about to use what you control because who knows you might change it. That's much more common.

The third model is one where we talk about build to contract, where someone has actually stepped up. There has to be some central body and that's usually IT, and that's probably where it ought to be. It says what a service will do for a whole bunch of values, by the way the one that is most forgotten is performance. They tend to work wonderfully when the original department is using it, it becomes more popular and suddenly it falls on the floor. And the reason is no one ever documents what their performance limitations are. But at that point you still have to manage the change control and so in fashion. There are a lot of strategies; usually the instigator of the change gets taxed in some way.

**Peter McTaggart:**

These centers of excellence, or competency centers, or whatever you might want to call them, that are the governing body of the SOA really have a big role to play in change. I see that from an operational point of view, introducing change into a highly re-used environment is a task fraught with danger. Do you want to talk about this Trevor?

**Trevor Thomas:**

I think some of the behaviors occur there, along those lines, I don't want to pay for this change because the cost of that involves a whole lot of regression testing and integration testing of all these other touch points, across environment. Often we actually don't know what all the other touch points are across the environment, or component servers that are being re-used it's got a number of threads out throughout our infrastructure and if someone happens to touch that one, and change it, even though I may need that change, what's it going to do with the environment? What is it going to effect in terms of the systems? More importantly what business processes are impacted by that change? It is absolutely critical to be sure about your management policy up front. How you are going to fund those changes, and making sure you have the right tools and processes to manage that change. As John said earlier, the danger is you just add to the complexity, you don't actually have the right frameworks in place to manage it.

**John Brand:**

There is another aspect to the performance, and that is, you can over engineer everything on the basis that you may need to support highly available scalable solutions in the future. We've done some research recently on application delivery, what we found was that although most people prefer is the cyclic style. It did enable to deliver applications more effectively, based on changing scalability requirements. It was much easier to deliver applications that were not necessarily written in a way that those original performance measures were going to be delivered. The issue of 'if we design it right the first time, we'll make it more effective'. This is often where you get caught by over engineering solutions, so you need to make that decision about how much do we invest, over what period of time on what services, do we evolve those services over time, architect them right the first time? The challenge organisations have is to get that right level of architecture and deploy just enough architecture just in time to support these over a long period of time. I see the architecture side has been one of the barriers over time as much as one of the enablers.

**Trevor Thomas:**

In terms of performance, an anecdote for the real world, is a situation where a set of services have been employed that were accessing a legacy environment, and admin guys were coming to the table and saying "suddenly we're using significantly more MIPS than we planned, you SOA guys told us costs would go down, what's happening?". Key things were planning for that, and educating around it that there was going to be an initial spike in use on the mainframe. There were the issues I mentioned earlier about the nature of the underlying connectivity in the infrastructure that was supporting it. It wasn't actually up to spec, because the plan hadn't been thought through about the increase in volumes that would come from service enabling some key mainframe components.

Importantly, as Tony mentioned earlier, there wasn't a clear plan. "Look if you're going to do this you're going to have to retire some key stuff". One of the reasons for introducing this architecture and transitioning the mainframe would have to be the conversation about "what are you going to retire fundamentally?" Just the notion of 'put the service layer in front, start using the mainframe, watch our costs go down', was somewhat naive.

**John Brand:**

Yes, very naïve. It's certainly the experience in banking was that originally over the counter transactions in banking would generate about 2.3 transactions per interaction. When the internet banking came on the scene; that went from about 2.3 to 23 interactions per transaction. The legacy system which was the only banking application to support internet banking at that stage and was never designed to support that massive increase in transactional loads. So enabling a service creates a different kind of demand for those services as well. It's not just about supplying a demand that we know we have today; it's about how that demand will change over time once we deliver these services to other applications or other organisations. That will change the nature of those services. That's a different angle on the agility, not just the agility of the business, it's how the business will actually change as a result of deploying these services.

**Tony Carrato:**

I agree with what Trevor and John have said. I always find it a little bit amusing when someone says "let me layer on several new layers of software put data through a few more transforms then I'll use a few less machinery sources". If anyone does that there's probably some kind of reward for how that works!! It's surprising how few people actually look at that. I'll take data in format A, put it in XML, shove it through some pipe, turn it into some other format, a couple of times, so we're starting to look at some technology solutions for that kind of thing. For example if you put it in XML, keep it that way.

I did a job for a customer in Hong Kong whose internet was falling over under load. What they were doing was moving everything into XML, named "this is the name of the banking customer" as your tag, that's a lot of bytes. It's surprising simple stuff like that, get someone to look at those things and apply a bit of rationality. How many times do you transform and in a finite set of resources.

**Peter McTaggart:**

There are a few things that came out there. I'd like to go back and dive into a couple of those in a little bit more detail. Firstly in the story you recounted Trevor, was a failure of governance for this process. Governance is certainly something that the vendors are pushing really hard. Saying if you're doing SOA you must have a good governance process. It's being spoken about in SOA circles, but a lot of organisations are not doing it well. What's stopping them from doing governance well?

**John Brand:**

I can certainly talk from a research perspective. It's because their version of governance is actually compliance. It is about stopping things happening, unless they conform to a certain standard. Not, what value does this deliver and how can we deliver something more effectively. It's not looking at a range of options; its saying these are our standards and we're not going to deviate from that.

**Peter McTaggart:**

Is there a more fundamental reason? Large organizations, complex environments, lots of people, lots of stake holders, is it a corporate political thing which is really impacting on governance here as well?

**Tony Carrato:**

The thing about governance is saying to people "no you don't get to do that". People don't like to say that. Then when they do say it, the question is, are you saying it for some arbitrary reason? Have you defined what the criteria are, against what you will choose to enforce? If you look at COBIT for example, which is for a lot of IT organisations at the IT governance, as opposed to the operational controls, where ITIL tends to play. COBIT is often what people will

look to. COBIT doesn't even have the notion of what a service model is. So, if you sought guidance there, you would struggle.

The Open Group is mapping COBIT against SOA to try and call out what COBIT processes do and do not apply. Remember it means you have to tell people no. If you're not prepared to do this and if you're not prepared to put the work in to understand why you would tell them no, your governance activity probably won't work.

**John Brand:**

Previous research suggests that the most potent combination for organisations is COBIT and six-sigma. COBIT as an effective mechanism for IT, and six-sigma as an effective mechanism for the business side. They actually complement each other particularly well because the business has this idea of improvement. If we can't do it let's find a way of doing it more effectively or more reliably and that is more the six-sigma. COBIT is the way of controlling without over controlling.

**Tony Carrato:**

What we find is you're injecting a lot of new technology into an organisation when you take on something like SOA. There's a fair amount of work in learning it. If you don't put the work in learning it you tend to make bad decisions. I review lots of projects, a common thing to find are people busily coding things that are actually in a product they've already bought. I don't understand the value add then. Unfortunately you have to put the effort in to learning this stuff.

**Peter McTaggart:**

That raises another point, skills development training and capability in organisations that want to do SOA or even in the market in general. Where do you go and find an SOA expert? How much do they cost? There are a lot of self proclaimed experts, but who's really put these into practice, and where do you find them? I think a lot of organisations are asking those questions.

**John Brand:**

You don't always know what is available and where it's available from. We see that a lot of organisations engaging in documentation projects or saying they don't have enough documentation. Documentation can certainly capture words and diagrams but it doesn't capture understanding. That's what is missing in most organisation is the understanding – where do you get this from, where is it best sourced from? Should I take it from vendor A, versus Vendor B, when they have a similar offering? I don't think there's enough of that understanding out there.

**Trevor Thomas:**

I don't think there's a simple answer. The other challenge introduced by this is that organisations really do generally struggle with continuity of IP across projects. There tends to be a bit of "well, we'll bring in a bunch of guys, get some contractors in, one or two permanent, and off we go, we'll do this piece of work, we're done now, let's move on. We have that enterprise architecture in the corner there that do stuff, but let's get on with the project and deliver it."

There's often a lack of true thinking and planning about the need to retain and manage IP and have the right tools and documentation in place that can actually retain it and evolve it.

**Tony Carrato:**

Let me give a little tip of the hat to the Government of Victoria, there's an organisation around called Enterprise Java Australia, which was actually launched from funding from the Victorian Government, it started off life as Enterprise Java Victoria. If you want to know if someone's competent to do this stuff, ask them what they're doing to continue maintenance of their skills. What you learned 20 years ago, some of it's very useful, but some of it's not. In our state, the government actually stepped up and did something, that's something we can be proud of. Not perfect but they are out there.

**Peter McTaggart:**

Well, before we wrap up, what I really wanted to ask is what the minimum height requirement to get on the SOA ride? How big does the organisation need to be to really adopt an SOA and what characteristics does it need to have to be able to pick it up?

**John Brand:**

I can certainly say from an analyst's perspective, the beauty of SOA is anybody can play and it depends on your approach to SOA. We've talked about the Web Services aspect, it is a beautiful starting point, you can do something as simple or as complex as you like. That has been the ultimate challenge too, as you have so many ways of approaching it. You have to be very clear about what you're doing, when you're doing it, how you're doing it. But it isn't a cost to get into the game at all; you can do it essentially for free.

**Peter McTaggart:**

I think there's one last question before we open things up for questions and answers, if we were sitting here in five years time, what would we be talking about? What's changed in the SOA playing field?

**Tony Carrato:**

Hopefully the first thing that will change is that we'll be past the discussion thinking about your IT as a set of business activities that are connected which is a reasonable way to think about it which is really what is meant by SOA.

**Trevor Thomas:**

I can see how this sort of style of technology or style of architecture helps organisations transition their legacy world. I think we'll see more and more of that.

**John Brand:**

If you were asking me with an analyst with a large global analyst firm I would probably come out with some outrageous prediction with some large numbers and we'd all be happy I'm sure!! I think from our research here, the way people are approaching SOA, which I've said is largely from an interface perspective today, our prediction is it is still not going to mature until 2010. I think we will start to see at that point organisations more focussing on the modelling aspects of SOA and start looking a little bit more at business process automation, business process modelling. Which is what they're talking about today, but no one is actually doing that today. I think you'll start to see the earlier adopters move in that direction around that kind of time frame in the next 3-5 years.

**Peter McTaggart:**

Great, we look forward to it. Thank you for listening to us and having us engage in this interesting debate and conversation. We'll wrap up the discussion now.