

KNOWN UNKNOWNNS

Graham Oakes finds both clarity and chaos in requirements analysis.



Graham Oakes: SOA methodologies assume services can be identified through analysis

If you're like me, you probably have a pretty good idea of many of your system requirements. You have a general feel for your needs, and you figure there are still a few things that will come and surprise you somewhere down the line. It's just a case of doing some more analysis and all will become clear.

But what if it isn't like that? What if no amount of analysis will bring out the details?

We've all come across situations where it was easier to try out a few ideas and see what works. Sometimes this is simply down to our own lack of experience. Sometimes it may be a sign that more complex dynamics are at play.

The Cynefin framework (www.research.ibm.com/journal/sj/423/kurtz.pdf) says something useful about these dynamics. Proponents of agile development have been talking about it for a while, but it's now making its way into the business press and I've even heard it discussed at a recent project management conference.

Cynefin identifies four domains – each characterised by different cause-and-effect relationships, which makes the framework amenable to different decision-making styles:

- **Known.** In this domain, there is a clear relationship between cause and effect. You identify the symptoms and classify the situation into a known category. You then chose the solution appropriate to that type of problem.
- **Knowable.** The relationship between cause and effect is complicated, but can be determined through analysis. You can eventually untangle the root causes and hence act on them.
- **Complex.** The relationship between cause and effect can only be seen in retrospect. Only when you've found a solution will you know what was causing the problem. In this domain, you actively explore the situation in order to tease out emergent patterns.
- **Chaotic.** There is no clear relationship between cause and effect, even in retrospect. Again, you need to act, but the objective is not so much to tease out patterns as to reduce turbulence. This is likely to require constant readjustment in response to the changing situation.

The first two domains are characterised by the adage 'think before you act'. In the latter two, you turn this around and 'act in order to learn'.

Most SOA methodologies assume that services can be identified through analysis. It may be difficult and it certainly requires skills and experience, but the problem is fundamentally analysable.

There are indeed places where this is true, but I suspect many interesting places – the places where significant competitive advantages are to be found – lie in complex or chaotic domains. If SOA is causing us to keep away from those places, then it is doing us all a disservice.

● *SOA, Web Services & Enterprise Integration Evaluation Centre Expert Dr Graham Oakes is the principal of content management, product development and customer service strategies consultancy Graham Oakes Ltd. Email: graham@grahamoakes.co.uk. Website: www.grahamoakes.co.uk.*