The Register®
Biting the hand that feeds IT

Original URL: http://www.theregister.co.uk/2008/09/29/software_dev_changes/

Developing software in the global village

Have things really changed in the past few decades?

By **Jon Collins, Freeform Dynamics**

Posted in Workshop, 29th September 2008 13:02 GMT

Reg Reader Workshop Anyone would think the kids of today had invented globalization.

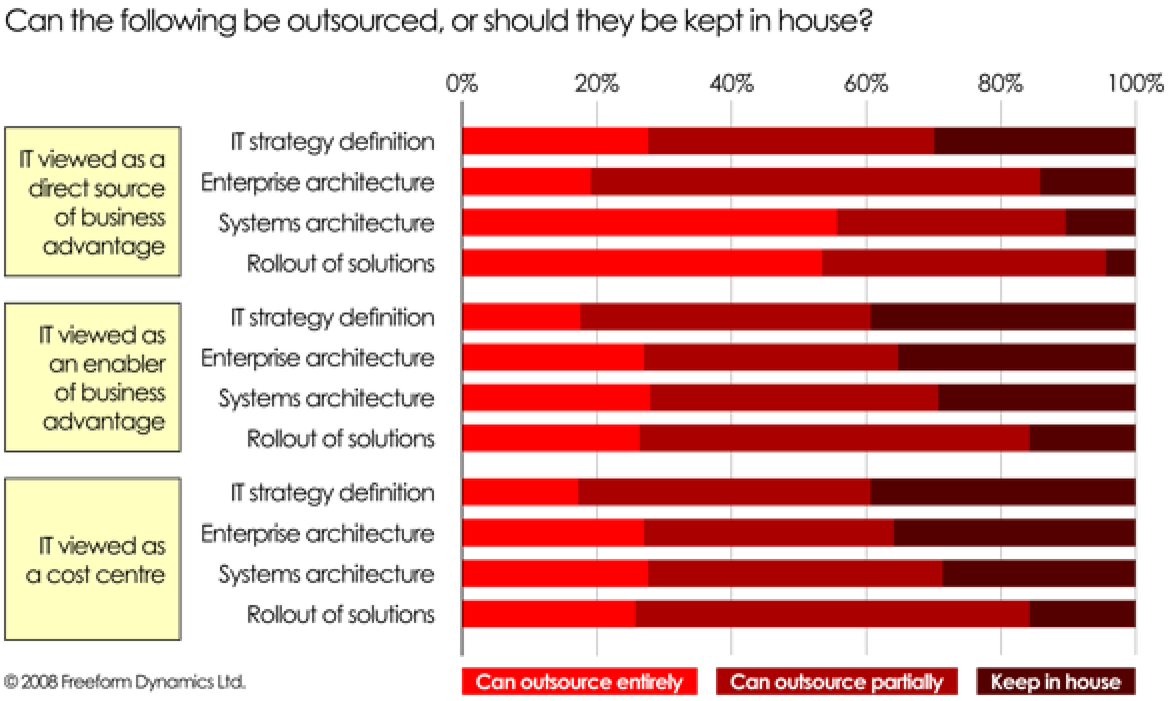
All this talk of largely text-based communications mechanisms used in social networking and blogging gives the impression that the mechanisms we relied upon in the earlier days of computing – bulletin boards, Usenet News and so on – were in some way different.

Similarly, one could argue that software development activities are no more distributed than they were a decade or two ago – the larger multinationals and systems integrators have long been using the most appropriate resources for the job wherever they happened to be, bringing in third parties and subcontractors as needed.

This picture may be rose-tinted, but nostalgia aside, there has for many years been such a thing as a geographically-dispersed development team, drawing on a variety of resources.

So, what gives – are things really any different today? There's a few factors that may have had an impact, notably how the twin topics of outsourcing and offshoring have affected software development organizations. As we've emerged coughing and spluttering from the outsourcing wave only to have the offshore wave crash on our heads, the only thing clear is the ever-broadening number of options, for organizations large and small.

When we did some research (http://www.freeformdynamics.com/fullarticle.asp?aid=318) into outsourcing in particular a few months ago, what became clear was that not all development activities were suitable candidates. It stands to reason that organizations should want to keep the more strategic activities in house and outsource the more tactical stuff. What's fascinating is how this view on outsourcing depends significantly on whether IT is viewed as a source of business advantage, or as a cost centre – we can see this from the chart below. And similar research tells us that the picture is the same for local vs off-shore project resourcing.


Can the following be outsourced, or should they be kept in house?

© 2008 Freeform Dynamics Ltd.

But isn’t this counter-intuitive? Surely, goes the common wisdom, the purpose of outsourcing or offshoring is to take costs out of the system? This may have been the case a few years ago, but today we’re seeing plenty of counter-examples – use of specialist organizations that happen to be based in specific locations for example, or geographically distributed teams in order to reap the time zone benefits, all are reasons why to take advantage of outsourced resources.

But the road to hell is paved with failed outsourcing or offshoring examples. From a software development perspective there are all kinds of things that can go wrong - not just contractual issues. The route is rife with process failures and communications breakdowns. When it comes to offshoring we know of organizations that have had great experiences and those who haven't – and we suspect that this is as much down to the checks and balances in place as any inherent difficulties with the distributed model. “Never again,” said one project manager to me earlier this year. “You just can’t get the quality.” “Absolutely not true,” said another. “It’s all down to recruitment.” Are they both right, or was one just luckier than the other?

So what do you reckon? Is it possible to get the perfect balance of resources, be they off- or on-shore, in-house or external, and get the whole lot working better than a bunch of people in a room? Where are the real benefits to be had, and what are the absolute no-nos? Is it really just a case of applying the age-old principles of best practice, whatever the situation? Have you any classic examples, or indeed horror stories to share? Let us know – we’ll feed them into the *Reg* research machine and see what picture emerges. ®

© Copyright 1998–2008