Is Better Always Better?

BY MICHAEL NORTON

The theory of evolution dictates that a species must adapt or become extinct. What is not conveyed is how they must adapt. A competitive advantage doesn't always guarantee adaptation. The entire environment, including the symbiosis between various factors, must be taken into consideration. Take for example giraffes. They have long legs as well as long necks. In the evolutionary process giraffes with short legs and very long necks probably got run down by lions more frequently than their long-legged, shorter-necked brethren.

The evolution of the computer has followed similar principles. It is quite obvious that the best technology does not always thrive — or survive, for that matter. Is PC technologically superior to the Mac? Is Ethernet technologically superior to Token Ring? Is Windows a better operating system than OS/2? This isn't an argument for or against any product; fill in your own answers. The phenomena is so pervasive it is beginning to become a maxim based on an experience I'm sure many of us shared: the appearance of the first video recorders. Looking back, we ask ourselves, "Was VHS format really better than Beta?"

Corporate America today is much less susceptible to the "better technology" sales pitch. In the first place, the reasons one product is superior to its competitors are becoming increasingly technical, and are therefore better suited for discussions among engineers rather than businessmen. Such minutia tends to get lost on consumers who have neither the inclination nor patience for it. Secondly, IS has a track record of not delivering on its promise of improved productivity through technology.

Corporate America today is much less susceptible to the "better technology" sales pitch. In the first place, the reasons one product is superior to its competitors are becoming increasingly technical, and are therefore better suited for discussions among engineers rather than businessmen.

Or does it? One would be hard pressed today to find any corporation in any competitive industry that eschews computers and still uses the old fashioned paper method of maintaining inventory, employee records, and customer accounts. Why? Because computing does bring a competitive edge. The average worker today produces more than the average worker of two decades ago. This is in no small part due to computer phenomena such as robotics and office automation. The sales rhetoric, though certainly exaggerated, was true, a fact not entirely lost on the underlying business psyche. Businesses do, indeed, keep buying; but they are showing an increasing reluctance to adopt the latest and greatest just because it is the latest and greatest. Netscape and Microsoft both learned this lesson the hard way last year when their browser upgrades were met with disappointing sales and a howl from IS managers to slow down.

Moreover, it is not only the CEOs who are putting the brakes on technological advance; CIOs are joining their boardroom compatriots. I suspect that CIOs are actually leading the way. My reasoning is that computer professionals are a conservative lot, if not by birth, certainly by experience. What systems person doesn't shudder at the thought of installing anything? And while the technician in us may drool thinking about the technical wizardry represented by a new piece of hardware or software, the implementation of this new product in our currently working system more often than not dissuades us from pursuing the glories of hi-tech. Moreover, another corporate dynamic makes IS professionals more likely to reject innovative technologies. Purchases must be justified, and to sell the product, IS managers often find themselves in the position of having to exaggerate positives and minimize negatives. This leads to a scenario where any implementation is doomed to be disappointing — and we all know where the fingers are going to point when this happens.

Psychological dynamics aren't the only dredges of the technological innovation. In the last few years the term "legacy" has arisen. Corporations have untold billions of dollars invested in systems. In the unlikely event that it would make financial sense to reinvent those systems, the diversion of time, energy, and manpower to such an effort would make it unfeasible. Indeed, much of the current innovation in software is in the area of making this legacy data accessible to other applications and across platforms.

The multiplicity of platforms, of course, has long been viewed as being a bane to development, and consequentially to innovation. Not only is there a direct effect as developers have to create generic products to work with multiple platforms (ignoring the specific benefits one platform can offer), there is perhaps an even more substantial indirect effect as IS resources are diverted to the problems of networking the various platforms. Indeed, it seems that many of us spend more time figuring out interaction between systems than within systems.

How did that happen? Innovation, partly. As IS needs grow, new systems are introduced that are part of a new generation of IS products. More importantly, however, is the fact that most corporations grow by gobbling up other corporations, including IS resources, which must then be integrated with current systems. This adaptation leads to a hodgepodge of disparate systems scattered throughout the enterprise — mainframes, minis, PCs, UNIX, MVS, VSE, DOS, Windows, OS/2 — connected by a variety of network facilities and protocols — NetBIOS, ISPX, TCP/IP, Ethernet, Token Ring. Certainly, if anyone could design a system this would not be the result.

Which is precisely my point: There is less "design" in the modern enterprise than we care to admit. We rebel against the heterogeneous nature of our systems, if for no other reason than we are firm believers in deductive reasoning and consequently the "top down" stratagem for building systems. Unfortunately, that simply isn't the way the real world works. Perfectly homogeneous systems are as rare as albinos.

And thus the perennial cry for "standards." The most glowing example of the possibilities of "standards," of course, is the Internet with its publicly designed specifications. And in the Internet we also see the inevitable impossibility of any lasting standard; just witness the current profusion of HTML "replacements" — Java, Dynamic HTML, XML. Sooner or later proprietariness rears its head. And it must, for innovation is inherently proprietary. There is no incentive for a company to invest in research and development unless the investment yields advantages over the standard that makes the product more attractive than the competition.

Of course, giraffes with the longest necks don't necessarily survive. There are forces against innovation as well as for innovation. It is precisely this precarious balance between the new and the old, between innovation and standardization, that determines the level to which technological advancement is adopted, embraced, and eventually standardized.

Michael Norton is the network administrator at SofTouch Systems, Oklahoma City, Okla., which provides both mainframe and PC software solutions. He has written mainframe manuals in addition to articles for a number of publications. Michael can be contacted at norton@softouch.com.