



W

hen the *Harvard Business Review* (HBR) published “IT Doesn’t Matter” in May 2003, the point was to start an argument, or, as they say in the more genteel world of academia, a debate. The provocative title of the article and its timing — at the tail end of a long slump in technology spending — ensured that a dustup would ensue. The resulting debate has been impassioned and often revealing, and is still going on.

For those who may have missed it or might welcome a reminder, the central point of the essay, written by Nicholas G. Carr, then editor at large of HBR and now a consultant and author, was that there is nothing all that special about information technology (IT). He declared that information technology is inevitably going the way of the railroads, the telegraph, and electricity, which all became, in economic terms, just ordinary factors of production, or “commodity inputs.” “From a strategic standpoint, they became invisible; they no longer mattered,” Mr. Carr wrote. “That is exactly what is happening to information technology today.”

Photograph by Opto

Does Nick Carr Matter?

A controversial new book on the strategic value of information technology is flawed — but right.

Steve Lohr

(lohr@nytimes.com), who covers technology for the *New York Times*, is the author of a history of computer programming, *Go To: The Story of the Math Majors, Bridge Players, Engineers, Chess Wizards, Maverick Scientists and Iconoclasts — The Programmers Who Created the Software Revolution* (Basic Books, 2002).

The reaction was swift. Within weeks, Mr. Carr was branded a heretic by many technologists, consultants, and — especially — computer industry executives. Intel's Craig Barrett, Microsoft's Steve Ballmer, IBM's Sam Palmisano, and others felt compelled to weigh in with varying degrees of fervor to reassure corporate customers. Their message: Don't listen to this guy. Keep the faith in IT's power to deliver productivity gains, cost savings, and competitive advantage. And the reaction continued. HBR got so many responses that it set aside a portion of its Web site to accommodate them, and Mr. Carr kept the controversy bubbling on his own Web site. He became a traveling celebrity of sorts, defending his stance in forums across the country, from the Harvard Club in New York City to the Moscone Convention Center in San Francisco, where he traded verbal jabs with Sun Microsystems' Scott McNealy. The article became fodder for countless columns in newspapers, business magazines, and trade journals.

In the interest of full disclosure, I should note that I contributed to the phenomenon. I did not know Mr. Carr before his article was pub-

lished, but HBR had sent me an advance copy of the manifesto, which I quoted in a long Sunday business piece for the *New York Times* on the maturing of the IT industry. To the best of my knowledge, it was the first mention of Mr. Carr's article in the press. Two weeks later, I cited Mr. Carr again in a piece headlined "Has Technology Lost Its 'Special' Status?"

When "IT Doesn't Matter" was published in HBR, I thought Mr. Carr had delivered an important, thought-provoking reconsideration of the role of IT in the economy and inside companies. Now that his analysis has been expanded to book length, I still do. This time, his ideas are packaged with a less incendiary title, *Does IT Matter? Information Technology and the Corrosion of Competitive Advantage* (Harvard Business School Press, 2004). But his message is unchanged, though more fleshed out and nuanced.

Mr. Carr's thinking, in my view, is flawed — at times seriously flawed — but not necessarily in ways that undermine his essential thesis. So let's first examine what his fundamental point is, and what it is not.

The title of the original HBR article was misleading. Mr. Carr is not arguing that information technology doesn't matter. Of course it does. Among other things, IT improves productivity by reducing communications, search, and transaction costs, and by automating all sorts of tasks previously done by humans. But Mr. Carr asserts that as IT matures, spreads, and becomes more standardized, the strategic advantage any individual firm can gain from technology diminishes. Paradoxically, the more the economy gains from technology, the narrower the window of opportunity for the competitive advantage of individual companies. This was the pattern for railroads, electricity, and highways, which all became utilities. In the IT world, Mr. Carr sees evidence of mature standardization all around him. The strategic implication, according to Mr. Carr, is clear. "Today, most IT-based competitive advantages simply vanish too quickly to be meaningful," he writes.

Thus, IT strategy for most companies should become a game of defense. The shrewd executive, Mr. Carr writes, will in most cases keep his or her company focused on the trailing, rather than the leading, edge of technology. He offers four guidelines for IT strategy: "Spend less; follow, don't lead; innovate when risks are low; and focus more on vulnerabilities than opportunities."

In Mr. Carr's view, there are two kinds of technologies: "proprietary technologies" and "infrastructural technologies." The first yields competitive gain, whereas the second is just plumbing, at least from a strategic standpoint. Technologies shift from proprietary to infrastructure as they mature. When a tech-

nology is young, companies can gain a big strategic advantage, and Mr. Carr deftly describes how companies like Macy's, Woolworth, and Sears, Roebuck exploited the new economics of retailing made possible by rapid, long-distance shipments by rail, and how a new breed of national high-volume manufacturers like American Tobacco, Pillsbury, Procter & Gamble, Kodak, and Heinz sprang up by gaining advantage from modern transportation, the telegraph, and electricity.

Once a technology moves into the infrastructure category, however, corporate opportunity wanes. In IT these days, Mr. Carr sees just about everything being folded into the infrastructure, including the Internet, Linux, Web services, and Windows. Mr. Carr is particularly insightful on the subject of enterprise software, such as SAP's enterprise resource planning offerings and Siebel's customer relationship management programs. As he does throughout the book, he succinctly draws the analogy between the present and an earlier technology. In this case, enterprise software is depicted as the modern version of machine tools.

Before the 20th century, machine tools were bespoke gadgets made by each factory for its own requirements. But then machine-tool vendors emerged. Their economies of scale brought lower costs and standardization to the machine-tool industry. Innovation continued, but it was the vendors who developed and distributed those innovations for all manufacturers — and thus no competitive advantage accrued to any individual manufacturer. Mr. Carr sees a similar “vendorization” in enterprise software, where core business processes like

supply chain management and customer relationship management are handled by standard software packages. The result is a straitjacket of standardization, leaving little room for a company to distinguish itself. Small wonder, Mr. Carr writes, that in the late 1990s enterprise systems came to be called “companies-in-a-box.”

Even the companies that seem to be IT-based success stories — notably Dell Computer and Wal-Mart — are not, Mr. Carr tells us. Yes, Wal-Mart was a leader in using advanced computing and private networks to link sales, inventory, and supply information. But Wal-Mart's real edge today, Mr. Carr says, is the scale of its operation, which enables it to strong-arm suppliers and zealously pursue efficiencies everywhere in its operations. And Dell, he contends, has an edge over rivals because of its direct marketing and build-to-order strategy. “It's true that IT has buttressed Dell's advantage, but it is by no means the source of that advantage,” Mr. Carr writes.

More generally, Mr. Carr observes, strategic advantage derives not from technology itself but “from broad and tightly integrated combinations of processes, capabilities, and, yes, technologies.” Translation: How you use technology, not the technology itself, is the crucial variable. “Indeed,” Mr. Carr writes in his preface, “as the strategic value of the technology fades, the skill with which it is used on a day-to-day basis may well become even more important to a company's success.”

It has the ring of innocuous truism, but wait a moment: Does that statement really apply to a utilitylike infrastructure technology? Does the skill with which we use electricity,

commuter rail service, or the telephone have anything to do with corporate success or failure? No one seeks insights from research firms, like Gartner, or advice from consultants, now including Mr. Carr, on how to use real infrastructure technologies. This suggests that information technology may be a bit different after all.

The main difference between computing and the industrial technologies Mr. Carr cites is that the stored-program computer is a “universal” tool, which can be programmed to do all manner of tasks. The general-purpose nature of computing — especially software, a medium without material constraints — makes it more like biology than like railroads or electricity. It has the ability to evolve and take new forms. Speech recognition, natural language processing, and self-healing systems are just three of the next evolutionary steps on the computing horizon.

Mr. Carr might dismiss such comments as romanticized nonsense — and he certainly could be right. Yet understanding the nature of the technology is crucial to determining whether computing is truly graying or, more likely, whether some parts of the industry are maturing while new forms emerge further up the computing food chain. Are we seeing old age — or merely the end of one stage in a continuing cycle of renewal?

Mr. Carr notes that the technology bubble of the 1990s resembled the booms and busts of railway and telegraph investment, which marked the passing of youthful exuberance in those industries. In the computer industry, however, there already had been two previous

Google is only one example of where Mr. Carr's neat thesis leads him astray.

boom-and-bust cycles — in the late 1960s, when mainframe time-sharing services appeared to be the computing utilities of their day, and in the mid-1980s, when legions of personal computer companies were founded and soon perished. Again, the pattern seems to be cyclical and evolutionary, as innovations accumulate and eventually cross a threshold, opening doors to broader market opportunities.

Let's take one potential example, Web services. The nerdy term refers to the software protocols that could allow a new stage of automation as data and applications become able to communicate with each other over the Internet. More broadly, Web services are seen as the building blocks of a new "services-based architecture" for computing. Mr. Carr briskly brushes Web services into his "vendorization" bucket. He writes, "Here, too, however, the technical innovations are coming from vendors, not users." The vendors — IBM, Microsoft, Sun Microsystems, and others — are working jointly only on the alphabet soup of software protocols: XML, UDDI, WSDL, and so on.

Yet when technologists talk of a services-based architecture, they are

speaking of a new computing platform that they see as the next big evolutionary step in decentralizing the means and tools of innovation — much as the minicomputer was a new platform that decentralized computing away from the mainframe, and then the personal computer put power in many more users' hands. Computer scientists regard the Web as a "dumb" medium in a sense. It is, to be sure, a truly remarkable low-cost communications tool for search, discovery, and transactions, but the Web is mostly raw infrastructure because it is not very programmable. Web services hold the promise of making the Internet a programmable computing platform, which is where differentiation and potentially strategic advantage lie.

I cite this as only one example of where Mr. Carr's desire to fit everything neatly into his thesis leads him astray. There are others. He mentions Linux, and its adoption by Internet pacesetters such as Google and Amazon, as proof that commodity technology is plenty good enough for almost any need. Linux, the open source operating system, does allow those companies to build vast computing power

plants on low-cost hardware from the PC industry. But the other great appeal of Linux — and open source software in general — is that it also frees those companies from the vendors. The rocket scientists at Google and Amazon can tweak the software and change it without seeking permission from Microsoft or Sun Microsystems or anyone else. Today, Google is both a brand name and verb. But technological differentiation has been the bedrock of its comparative advantage. It is the better mousetrap in Internet search. As an example, Google undermines, rather than supports, Mr. Carr's point.

His thesis is often the same kind of straitjacket of standardization that packaged software, as he says, is for companies. Mr. Carr approvingly cites studies showing a random relationship between total IT spending and corporate profits. But these merely demonstrate that aggregate technology spending is neither the only nor the crucial variable in determining corporate profitability. That is hardly surprising. Again, it is how companies use the technology — integrating the tools with people and

IT Resources

Works mentioned in this review

Nicholas G. Carr, *Does IT Matter? Information Technology and the Corrosion of Competitive Advantage* (Harvard Business School Press, 2004), 208 pages, \$26.95

Does IT Matter? An HBR Debate. Letters to the *Harvard Business Review*:

http://harvardbusinessonline.hbsp.harvard.edu/b01/en/files/misc/Web_Letters.pdf

IT Doesn't Matter, responses, articles, and resources related to Nicholas Carr:

www.nicholasgarr.com/articles/matter.html

Steve Lohr, "Is the Technology Business Still a Growth Industry?" Originally published in the *New York Times*:

www.iht.com/articles/96640.html

Remarks by Bill Gates, chairman and chief software architect, Microsoft Corporation, CEO Summit 2003, Redmond, Wash.:

www.microsoft.com/billgates/speeches/2003/05-21ceosummit2003.asp

processes — that counts the most. And Mr. Carr can be quite selective in citing the work of others. He points to research from Paul Strassmann, an industry consultant, that supports his case while gliding over the fact that Mr. Strassmann was a prominent critic of Mr. Carr's original HBR article.

Still, these can all be seen as quibbles. They do not necessarily shake the accuracy of Mr. Carr's central point — that the period of sustainable advantage a company can derive from technology is diminishing. But is that really surprising? Everything, it seems, moves faster than it did 10, 20, or 30 years ago, including technology. To say that the advantages technology gives a business are more fleeting than they once were is not to say those

advantages aren't worth pursuing. Dawn Lepore, vice chairman in charge of technology at Charles Schwab, estimates that a lead in new IT-based financial products lasts from one to 18 months. "You still get competitive advantage from IT, but there is no silver bullet," she observes.

Mr. Carr's book is a thoughtful, if at times overstated, critique of faith-based investment in technology, and it makes a real contribution to the field of technology strategy. But Mr. Carr understates the strategic importance of defense. The old adage in baseball is that defense and pitching win championships; in basketball it is defense and rebounding. In business, if you don't make the defensive technology investments to keep up with the productivity and

efficiency gains of your industry peers, you simply lose.

The drift toward more standardized technology that Mr. Carr describes also points to a different kind of pursuit of strategic advantage. It may not be IT-based, but it is certainly dependent on technology. This is what Irving Wladawsky-Berger, a strategy executive at IBM, calls the "post-technology era." The technology still matters, but the steady advances in chips, storage, and software mean that the focus is less on the technology itself than on what people and companies can do with it.

The trend is already evident in companies and in universities. The elite business schools and computer science programs are increasingly emphasizing multidisciplinary approaches, educating students not only to be fluent in technology, but also in how to apply it. In companies, the same is true. The value is not in the bits and bytes, but up a few levels in the minds of the skilled businesspeople using the tools. Large chunks of the technology may be commoditizing, but how you use it isn't. That is where competitive advantage resides. +

Reprint No. 04213