

CHAPTER 2**If You're in a Hole, the First Thing to Do Is Stop Digging**

Isn't it interesting how some problems only continue to get worse over time? It is as if the problems have some mind of their own and grow like weeds in a well-fertilized lawn. Some problems seem to happen in spite of all of our efforts to prevent them, like that poor chap Murphy immortalized into his law "Whatever can go wrong will go wrong." However, although we can't avoid all our problems in life, at least we can avoid making them worse.

Case in point: the "Big Dig" boondoggle of a civil engineering project that took place in Boston starting in the mid-1980s. A project with the seemingly simple goal of reducing traffic congestion by moving high-traffic corridors under the overcrowded streets of the city of Boston ballooned into a multidecade, multibillion-dollar blunder that to this day has yet to be completed. With billions already invested in this project, there seems to be no other option but to invest billions more. Alas, it seems that our IT systems are often Big Dig boondoggles of their own. Somehow, we must emerge from our black hole of spending—and the best way to do that is to stop digging the hole deeper.

IT DECISION MAKING'S FATAL FLAW

Businesses, like people, are complex organisms. The larger the business, the more complex it becomes, by virtue of its numerous moving parts. However, what makes businesses complex are not just the number of employees, systems, and processes they are composed of but also the sheer number of decisions that their people make on a daily basis. In fact, instead of examining the parts of a business, an alternative way to look at a business is to take a close look at the way that people make decisions and the impact those decisions have on an organization's long-term success.

People throughout an organization make decisions, with different

amounts of impact on the company's success. Big ones, such as decisions to acquire other firms or enter new markets, can in turn spawn hundreds of other, smaller decisions. Other decisions, such as whether to take a lunch break, are less critical but still impact the business, especially if the people taking the break are responsible for a vital task. However, it is the totality of *all* the decisions that people make, regardless of where in the organization they emerge, that determines a business's overall success. In particular, decisions made by different parts of the organization are often in conflict with each other, leading to wasted effort and, of course, much-dreaded inflexibility. Obviously, to counter the plague of inflexibility, we must first understand the decision-making process in the organization, especially as it applies to IT.

A decision is a real, tangible thing—that's why we say we *make* a decision rather than *think* a decision. Decisions have real outcomes and tangible consequences, and people have a wide range of motivations for making decisions, many of which have nothing to do with long-term business goals. Given that decisions are tangible things, it's clear that the more decisions people make, the more potential for problems there are.

People always make decisions in light of the facts that are apparent at the time as well as their own motivations. The more facts that they know or the more facts have proven themselves over time to be correct, the easier making a decision becomes. The more that people's motivations are in line with the business, the greater the likelihood that their decisions will result in positive benefits for the business. However, in light of few facts and poor alignment with business goals, decision making in IT has proven to be problematic, at best.

Why Incremental Decision Making Leads to Exponential Problems

IT executives approach spending with the best of intentions and with the desire to solve the problems that senior management and market conditions mandate. And this is where the problem begins. As technology continues to mature at a rapid and unrelenting pace, any technology purchasing decision made today will necessarily be obsolete by tomorrow. Given the pressure to solve big problems with limited facts and inflexible technologies, it is often easier to make smaller, incremental decisions than it is to rethink a corporation's overall IT goal and structure. However, some managers feel that it is simpler to make decisions without thinking through all the resulting consequences.

Furthermore, the IT decision-making process is fraught with emotion in

ways that imperil the well-being of the corporation. No one likes to be wrong, so when facts change and a decision proves to be mistaken, many IT managers will simply redouble their efforts to prove that their decision was right rather than take a step back, change their minds, and make a new decision based on the new facts. The motivation to save face versus the motivation to do “what’s right” therefore hurts IT significantly. And the promises of technology vendors that claim to solve recurring IT problems sway many IT managers, or at least extend the tenure of their ill-advised decisions. This inertia toward recurrent decision making that continues in a poor direction becomes an IT “religion” that often pits rational, business-led thought against emotional, egotistic resistance.

But IT decision making’s fatal flaw is one that takes into consideration only today’s requirements. Such shortsighted decisions necessarily result in inflexible, brittle systems that become loaded with emotional baggage and inertia. Just as in developing countries, where independent individual decisions to connect one building’s electrical supply to another results in a chaotic, fragile mess, shortsighted decisions plague today’s IT organizations. Even though the decisions may solve today’s problems, they result in a mess that eventually requires the total replacement of the existing system.

Where Has All the Planning Gone?

The truth of the matter is that the senior management in most companies doesn’t have the stomach to make investments that go beyond problems that affect one or two quarters. There’s simply too much risk involved in engaging in multiyear, total-company alignment initiatives, especially given the reality of constant change. And so, although companies desire long-term flexibility, they in fact reward short-term decision making that addresses only today’s requirements.

Companies reward short-term, incremental decision making in a number of ways. First, they want to quickly turn their IT investments into realizable business gains and move on to the next project as promptly as possible. When these projects relate to products or services a company offers, we call this a desire for fast *time-to-market*. When these projects solve internal problems or address general business requirements, we say that businesses desire quick *return on investment*. And so organizations reward developers and IT staff when they achieve those goals as rapidly as possible, even if the approach they take compromises long-term flexibility.

Companies also motivate their IT organizations to spend as little as possible to solve the given problem at hand. Cashed in business code-word phrases such as lowest *total cost of ownership* (TCO), the desire to spend

very little often comes at a compromise to long-term viability of the systems that they are implementing. It's much cheaper to repair a broken window with tape or fix a car with bailing wire and chewing gum than it is to replace the broken parts. Although those low-cost solutions might indeed solve the problem at hand, they will come back to bite a company in the rear when inevitable changes stress those short-term solutions to the breaking point.

And yet, even though IT managers might be aware of the shortcomings of a narrow focus, they do nothing to mitigate the potential damage, because nobody is rewarding them for doing so. Very few businesses reward IT management for spending more than absolutely necessary to solve a problem in a more robust way or to take longer than necessary to build a more robust solution that might be able to meet unforeseen needs. The prevailing wisdom for most IT organizations is to move as quickly and spend as little as possible today, and to push potential problems into the future for someone else to solve. (Sounds like how we got into the Y2K mess, doesn't it?)

Breaking the Cycle of Shortsightedness

However, we're not suggesting you ignore the business goals of fast time to market, speedy return on investment, and overall low total cost of ownership. Those are all noble and mandatory goals for businesses. It's not the questions you're asking, rather it's the time frame in which you're seeking the answers. Are companies interested in fast time-to-market for just one project or for *all* their projects going forward? Do businesses want the lowest total cost for just that one particular implementation or to reduce the ongoing cost of *all* their future implementations? Are companies most interested in realizing a quick business return for one particular investment or for all investments going forward? Clearly, the optimal answer realizes benefits both in the here and now as well as in the not-too-distant future.

Understanding how to make the right decisions to realize the most benefits both now and in the future is the trick that's required to implement the next wave of IT solutions and is an essential prerequisite of Service Orientation. In this book we hope to convince you that the technology required to implement the vision of flexible business that can respond to unpredictable change doesn't matter as much as the decision-making process you should use to implement technology as a whole. The proper decision-making process will determine the very success of your business and will bring you that much closer to realizing the vision of a flexible and agile business.

THE IT "RATS' NEST"

Sigmund Freud once conjectured that at our core, we're all animals, and our primitive desires motivate us in all of our activities. Obtaining food, securing a family, and procuring shelter are certainly such core desires. As a result, it's no surprise that people like the idea of building comfortable nests for themselves. For instance, the notion of building a "nest egg" and returning home to the family nest (unless your children have left, in which case, you are an "empty nester") are all comforting thoughts to people.

It is no surprise, then, that we find nests rife in IT systems in the organization; and these nests are not a desirable kind by any means. In the course of building the sort of systems that satisfy the basic desires of IT, we have developed a "rats' nest" of systems, processes, and relationships among them that threaten to strangle all the flexibility out of business. What do we mean by a rats' nest, and how did we get here in the first place?

Interconnectedness of Systems

As we discussed earlier, most IT decision making is of the short-term, incremental, and shortsighted kind. Early on in a company's life, IT management tends to make simple, straightforward decisions to solve current problems with specific answers, but over time, incremental solutions to new problems increasingly require some level of dependency on previous solutions. And so, although little complexity or interdependency exists in the first set of systems a company might implement, over time these incremental decisions result in a spaghetti of connections, processes, and technical dependencies that only grow *more* complex.

In fact, computer scientists often speak of complexity problems like the one we just described as a problem that grows in *geometric*, or "*n*-squared," complexity over time. Some problems double in complexity as you double the number of elements that contribute to them. For example, remembering a list of 20 people's names is twice as difficult as remembering a list of 10 names. Such linear problems are reasonably manageable; you need to have only incrementally greater resources to deal with such problems that grow incrementally.

However, other problems grow with the *square* of the number of elements that contribute to them. For example, more than four times as many handshakes are required in a group of eight people to shake each other's hands as there are in a group of four. So, in this case, doubling the size of the problem *quadruples* the complexity at hand. For example, if you wanted to connect two computers, you would have a system that looked something

like that shown in Exhibit 2.1(a). So far, so good. Add a third computer, and you have Exhibit 2.1(b). Not too complicated, right? Okay, add a fourth, and you get Exhibit 2.1(c). Now we're getting a bit messy. Note how we need six arrows to connect four computers? Well, let's cut to the chase. What if you had dozens or even hundreds of different things to connect? Exhibit 2.1(d) shows a vastly simplified look.

That's what we mean by the rats' nest: If you don't have some sort of organizing principle, then even though each incremental change might be simple and straightforward, over time the sum total of all such changes gets out of hand. Clearly, these problems become not only more complex, but more expensive and hopeless over time. Businesses eventually won't be able to survive with problems that continue to grow at geometric rates.

And yet the problem of interconnected systems and processes is exactly the type of problem that grows with geometric complexity. The more systems people build and the more processes they develop, the more they depend on previous solutions and processes. As they continue to solve more problems incrementally, the number of dependencies increases geometrically, to the point where companies now spend most of their time managing the interconnections and dependencies than managing the systems and processes themselves.

We call this situation a rats' nest because rats aren't known for their cleanliness or orderliness. And so a great analogy to describe the sort of interconnected mess of systems and dependencies is the sort of messy nest that a rat would make. Birds' nests are great for soup. Rats' nests are good for nothing (except maybe baby rats). Just so with IT. We must not only deal with the decision-making process that results in rats' nests of systems, but we must also come up with a way to untangle the complex spaghetti of dependencies so that we don't have to resort to throwing away the whole mess and starting from scratch. Furthermore, we also need to figure out how to avoid creating rats' nests in the future.

Rats' Nests and Return on Investment

The reason companies build rats' nests in the first place is quite straightforward: Each individual decision is the least expensive, most expedient choice they can make at the time. And after all, isn't everybody supposed to choose the least expensive, most expedient option whenever they're faced with a decision? The surprising fact is that adding up several least expensive, most expedient options doesn't yield either the least expensive or most expedient overall long-term solution. On the contrary, rats' nests are among the most expensive, time-consuming parts of IT today. So, why do we keep making decisions and investments the way we do?

If You're in a Hole, the First Thing to Do Is Stop Digging

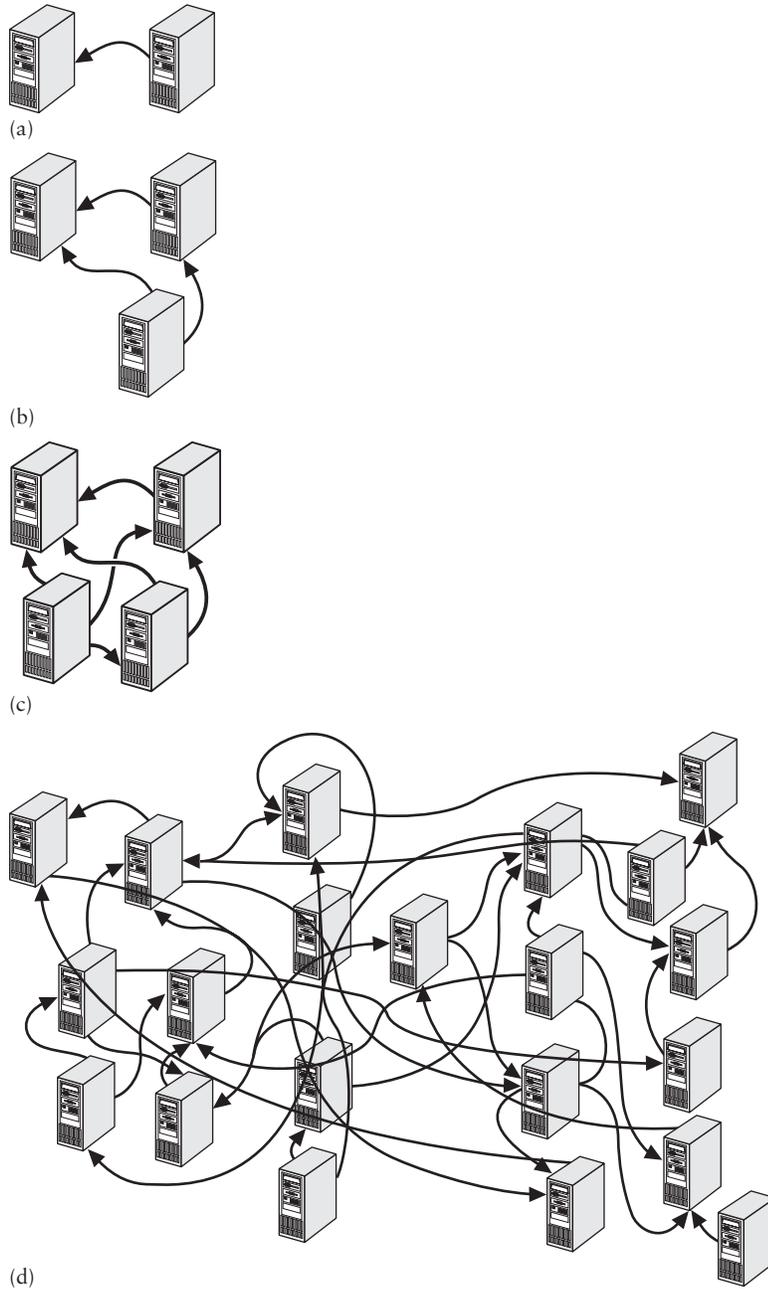


FIGURE 2.1 How to Build a Rats' Nest

The answer has to do with the way companies justify their investments. Despite the relentless and unpredictable nature of change, many companies seek comfort in some sort of quantification of their investments and gains from those investments. The desire to quantify the unknown through return on investment calculations is very understandable. If companies can get a grasp on how much they're spending on something, and if they can quantify what sort of return they are getting, either through top-line revenue growth or bottom-line cost reduction, then they believe they can make sound business decisions that are guaranteed to increase their success.

Jargon Watch: Return on Investment

Return on investment (ROI) is a fairly simple business concept for quantifying the business value of some investment. Technically, ROI is a measure of a company's ability to use its assets to generate additional value for shareholders. To calculate ROI, divide the net profit from an investment by the net worth of the assets invested, and express the answer as a percentage.

An ROI calculation exercise consists of determining how much a company must invest to realize some quantifiable business gain. However, software, hardware, and solutions vendors bandy about the term *ROI* as if it were some simple-to-identify number that you could easily glean from a spreadsheet or a Web site. In fact, there are really two kinds of ROI: the kind you can calculate *after the fact* using actual numbers and the kind that you can postulate *before* you make the investment. The problem is that neither of these ROI numbers matches reality when it comes to IT.

Many companies fail to accurately, quantitatively determine how much a given IT investment returns to the business, either through the addition of top-line revenue or the elimination of bottom-line cost. In fact, the business often fails to justify the effort to quantify the return, and so after-the-fact ROI calculations are actually a rarity. Furthermore, before-the-fact ROI calculations are too often pie-in-the-sky theoretical values that companies scarcely, if ever, achieve in reality. There are simply too many factors and unknown quantities that affect a business's ability to realize a promised ROI, so most before-the-fact ROI calculations are wishful thinking at best. Therefore, companies must rethink their approach for calculating the ROI for IT investments in light of ongoing change and the ability of systems to respond to that change economically.

In fact, businesses regularly think in terms of ROI. Companies determine whether to hire people based on their return to the business, build factories based on their contribution to the top line, and outsource business processes that can reduce their operational expenses. Thus, it's no surprise that companies want to apply the same rationale to IT systems. The problem is that ROI calculations typically focus on the short term—this month or this quarter—and only occasionally as much as a year. In addition, although the number and complexity of factories and marketing campaigns are fairly limited and well understood, the number of IT systems and business processes number in the hundreds, thousands, or perhaps even millions. In many cases, it simply isn't possible to calculate the required total investment in a given IT system that takes years to build, let alone quantify the business return of that investment.

However, the problem is not just the inability to pin numbers on complexity. More often than not, management gets in the way of realizing any ROI from an investment. Cultural resistance to change, emotional religious wars, shortsighted planning, and unintentional sabotage by misguided middle management often impede the ability to complete an investment according to a well-thought-out plan. And if a company can't even complete a project according to its wishes, how could it ever hope to get a planned return?

An even bigger problem is that the calculated ROI for a project is meaningful only if an investment is independent from other, ongoing investments an organization is making. What happens when two concurrent projects, each with its own planned ROI, are in fact dependent on each other in ways that are not well understood? How will those dependencies increase cost, and hence investment, more than anticipated? And even worse, how will interdependencies negatively impact the realization of business return? How can companies design to pin an ROI on multiyear projects that involve so much complexity and interdependency that the organization is lucky even to complete the project, let alone have it deliver real value?

The IT rats' nest not only harms companies' ability to respond to unpredictable change, but makes the task of ROI calculation an impossibility. The challenge of determining the "I," or investment part of ROI, the impossibility of determining the "R," or return, and the inability to calculate long-term ROI in the face of geometric complexity makes ROI meaningless in an environment where businesses rarely understand the full scope of the interdependency of their systems.

WHY ARE THE NERDS SITTING AT THEIR OWN TABLE?

We have a confession to make. We were among those nerds you remember from high school. You know the type: finished calculus in tenth grade,

won the state math contest, took courses at the local college while in high school, and not only won but *ran* their school's science fairs. Now, before you roll your eyes (oops! too late!), let us make our point. Back then, nerds like us were ostracized, relegated to chess club limbo, while everybody else went to parties, got in trouble, and basically acted like regular high school kids.

Now that we're all adults, that old social division from high school still haunts us in the form of "technical" and "nontechnical" people. Look inside any company or on any job board, and this division is readily apparent. A techie's resume is chock-full of acronyms, and careers tend to progress by tackling increasingly more complex projects. However, a nontechnical person's resume is supposed to focus on money saved or earned: "reduced cost by 20%" or "increased revenue by 15%" are the nontechie's equivalent of "developed J2EE-based server solution using WebLogic." The nontechnical person frequently runs the company or manages the largest budget, while the techies are relegated to the part of the business that incurs the costs and spends time on the technical details. Yup, the nontechies are still the captains of the football team while the nerds are still in the business equivalent of the chess club.

For many organizations, this technical versus nontechnical dichotomy is actually getting wider. Ask a typical executive what Linux or J2EE or ERP or (insert technical term here) means, and the best answer you're likely to get, if you don't just get a blank stare, is some high-level simplification. The

Jargon Watch: Techie

The term *techie* is a colloquial term for *technologist*, who is a person with a deep understanding of technology and who applies that understanding in work. Needless to say, there are many less complimentary terms for techies, including *nerd* and *geek*, but we'll generally try to avoid these terms in this book since, after all, we're techies ourselves.

The opposite of techie for the purposes of this book might be *businessperson*. There are fewer derogatory terms for such folks, although *suit* comes to mind as a popular one, as in "the meeting was filled with geeks and suits." To be fair to all the suits out there, we'll avoid using this term as well.

business press isn't a big help either. You would think that *The Wall Street Journal*, for example, would talk about IT issues from a perspective that a nontechie would find compelling, but you'd be hard-pressed to find anything but a superficial discussion of the latest IT trends and some examples of how certain businesses benefited in this most business-oriented of business periodicals. After all, the paper wouldn't want to get too technical for fear of losing its audience. How then are businesspeople going to understand the power, complexity, and risks facing IT, if they can't understand the techies, and the business press gives them nothing but superficialities? The answer is, many won't, at least not until it's too late.

Merging Worlds of Business and IT

Techies usually get a sense of empowerment from the knowledge that they simply know how things work—at least, how technical things work. Now, if you're a techie, this is no excuse to gloat; the techie world is no more enlightened about business than the business world is about technology. For many techies, the business world is a necessary evil that's most useful for generating paychecks. Many techies wouldn't admit it out loud, but their jobs would be so much damn *easier* if only it weren't for all the users! (A *user*, after all, is a techie's term for "everyone who's not a techie.") When technologists do delve into business issues, it's often on a simplistic level, for example, when data-centric folks think about customers as sets of data fields rather than as living, breathing humans with unique concerns and motivations. A techie revels in the complexity of object-oriented systems and database optimizations but can't tell a general ledger from a General Mills' cereal box.

Now that we've alienated both our business and technical readers, let's get to the critical point: *It doesn't have to be this way*. Although clearly, people must specialize in their field of interest, regardless of whether it is technical or nontechnical, there is a large gap in the business world when it comes to business-savvy technologists (or technical businesspeople, if you will) and technologies that are firmly rooted in the nontechnical side of the business. After all, there's no getting around the fact that technology is becoming increasingly pervasive in the business world, and likewise, there's no avoiding the fact that IT must respond better to the changing forces of business.

There's no reason why businesspeople can't have a reasonably good understanding of technology and why technologists can't likewise have a pretty good grasp of business. In fact, many people actually do have such an understanding, and they are the ones who are often the most successful

at companies that leverage IT for critical business value. Many of today's software applications and systems would have been considered too complex for business users only a few decades ago. A generation ago, highly trained professionals who wore lab coats managed computer systems in the department known at the time as *data processing*. Hardly any techies got close enough to the systems to get any value from them, let alone business users. Had someone in executive management proposed letting business users control their own information, the lab coat-wearing techies would simply have laughed. Today's computers, however, while thousands of times more powerful than the computers of the data processing days, are significantly more simple and easier to understand, and thus well within reach of businesspeople. As such, we can afford now to think of putting the power of computing in the hands of the nontechies.

Such technology-savvy business users are often called *knowledge workers* or *information workers*, pointing out the fact that dealing with information is a key part of their day-to-day work. These people not only use their computer and other high-tech gizmos on a daily basis, but they're actually *comfortable* doing so. They are advanced users of the applications most valuable to them, and they have a reasonable understanding of the distributed technology behind the scenes. Even though they may not know all of the technical details of how those systems work, they are experts at wringing the last bit of productivity and efficiency from them.

Likewise, there is a rising class of business-savvy technologists as well. Many of these people began their careers as "hard-core" techies, but then moved into a more business-oriented role. They often have titles of *analyst* or *product manager* or the like, and they are able to leverage their deep technical understanding of IT to better address the business issues at hand in their work. Chief information officers almost always fall into this category, and an increasing number of chief executive officers do as well.

If these knowledge workers and business-savvy techies can do it, well, why can't you? Or more to the point, why *don't* you? After all, these cross-over professionals often get paid more and have better long-term career prospects than either low-tech businesspeople or business-adverse techies. With all the ways to improve your skills, through formal training, working with a particular mentor, and extracurricular learning through books, Web sites, and the like, there is little excuse for nontechie business folks not to improve their knowledge. The more that techies get involved with the day-to-day operations of the business, and the more that nontechies explore the ways that IT can make them more effective, the more that the worlds of business and IT will merge.

Why Crossover Professionals Are So Important

The reason why we're taking the time to discuss the issue of business-savvy techies and tech-savvy businesspeople is because the long-term trend in the business world is toward a future where there's no separation between business and technology in the typical enterprise. In this vision, all technology is simply a business resource to meet business needs, just like office furniture or human resources or finance. Furthermore, the day-to-day work of businesspeople will become steeped in technology—a trend we're seeing today, with all the phones, CrackBerries, and the like that today's business executives have to carry with them at all times. It will simply become impossible for businesspeople to run their daily lives without deep immersion in technology, and it will be impossible for the technical part of a business to operate without a strong business mandate.

Consider the hypothetical scenario in which everybody in your organization is equally proficient in business and technology, so that there is no need to separate the two types of efforts into different groups or departments. Would your company be as likely to go down the IT rats' nest trap we discussed earlier? Remember, the reason people go down the path to the rats' nest is because at every step, they select the cheapest, most expedient option and don't consider long-term gains over short-term benefits. But what if all the executives in your organization understood the value and importance of combining long-term IT planning with short-term goals? What if business users had enough technical acumen to understand the subtleties behind such technology trade-offs? And what if techies knew enough about the business to know when they are playing a losing game of connect-the-system? There's no guarantee, but the chances are significant that your company would never have built a rats' nest in the first place.

Now, that thought exercise is clearly unrealistic on its face because today's business leaders are not proficient in technology, and many technologists simply have no interest in running the business. What's more realistic is to expect *some* executives at *some* companies to really get it—and to build a deep understanding of technological realities into the way they do business. Those technology/business *crossovers* who can build such an understanding will be more successful in their careers than the dinosaurs they leave behind. And companies that can make this change across their organizations will be more competitive, more agile, and, in the long run, more successful than their competitors who are unable to make this shift toward business/technology savvy.

For those of us techies who were stuck in chess club limbo, this broad trend in twenty-first-century business is the revenge-of-the-nerds scenario

we so hoped for in our high school years. The popular kids may have gotten all the attention back in high school, but if they weren't able to pick up sufficient technology savvy, then their careers will plateau—and where will their popularity be then? A knowledge of technology will make business folks more popular than ever. For us nerds who build business expertise on top of a technical background, the world is our oyster!