

# The ABCs of ERP

## By Christopher Koch

### 1.1.1.1 *What is ERP?*

Enterprise resource planning software, or ERP, doesn't live up to its acronym. Forget about planning—it doesn't do much of that—and forget about resource, a throwaway term. But remember the enterprise part. This is ERP's true ambition. It attempts to integrate all departments and functions across a company onto a single computer system that can serve all those different departments' particular needs.

That is a tall order, building a single software program that serves the needs of people in finance as well as it does the people in human resources and in the warehouse. Each of those departments typically has its own computer system optimized for the particular ways that the department does its work. But ERP combines them all together into a single, integrated software program that runs off a single database so that the various departments can more easily share information and communicate with each other. That integrated approach can have a tremendous payback if companies install the software correctly.

Take a customer order, for example. Typically, when a customer places an order, that order begins a mostly paper-based journey from in-basket to in-basket around the company, often being keyed and rekeyed into different departments' computer systems along the way. All that lounging around in in-baskets causes delays and lost orders, and all the keying into different computer systems invites errors. Meanwhile, no one in the company truly knows what the status of the order is at any given point because there is no way for the finance department, for example, to get into the warehouse's computer system to see whether the item has been shipped. "You'll have to call the warehouse" is the familiar refrain heard by frustrated customers.

ERP vanquishes the old standalone computer systems in finance, HR, manufacturing and the warehouse, and replaces them with a single unified software program divided into software modules that roughly approximate the old standalone systems. Finance, manufacturing and the warehouse all still get their own software, except now the software is linked together so that someone in finance can look into the warehouse software to see if an order has been shipped. Most vendors' ERP software is flexible enough that you can install some modules without buying the whole package. Many companies, for example, will just install an ERP finance or HR module and leave the rest of the functions for another day.

### 1.1.1.2 *How can ERP improve a company's business performance?*

ERP's best hope for demonstrating value is as a sort of battering ram for improving the way your company takes a customer order and processes it into an invoice and revenue—otherwise known as the order fulfillment process. That is why ERP is often referred to as back-office software. It doesn't handle the up-front selling process (although most ERP vendors have recently developed CRM software to do this); rather, ERP takes a customer order and provides a software road map for automating the different steps along the path to fulfilling it. When a customer service representative enters a customer order into an ERP system, he has all the information necessary to complete the order (the customer's credit rating and order history from the finance module, the company's inventory levels from the warehouse module and the shipping dock's trucking schedule from the logistics module, for example).

People in these different departments all see the same information and can update it. When one department finishes with the order it is automatically routed via the ERP system to the next department. To find out where the order is at any point, you need only log in to the ERP system and track it down. With luck, the order process moves like a bolt of lightning through the organization, and customers get their orders faster and with fewer errors than before. ERP can apply that same magic to the other major business processes, such as employee benefits or financial reporting.

That, at least, is the dream of ERP. The reality is much harsher.

Let's go back to those inboxes for a minute. That process may not have been efficient, but it was simple. Finance did its job, the warehouse did its job, and if anything went wrong outside of the department's walls, it was somebody else's problem. Not anymore. With ERP, the customer service representatives are no longer just typists entering someone's name into a computer and hitting the return key. The ERP screen makes them businesspeople. It flickers with the customer's credit rating from the finance department and the product inventory levels from the warehouse. Will the customer pay on time? Will we be able to ship the order on time? These are decisions that customer service representatives have never had to make before, and the answers affect the customer and every other department in the company. But it's not just the customer service representatives who have to wake up. People in the warehouse who used to keep inventory in their heads or on scraps of paper now need to put that information online. If they don't, customer service reps will see low inventory levels on their screens and tell customers that their requested item is not in stock. Accountability, responsibility and communication have never been tested like this before.

People don't like to change, and ERP asks them to change how they do their jobs. That is why the value of ERP is so hard to pin down. The software is less important than the changes companies make in the ways they do business. If you use ERP to improve the ways your people take orders, manufacture goods, ship them and bill for them, you will see value from the software. If you simply install the software without changing the ways people do their jobs, you may not see any value at all—indeed, the new software could slow you down by simply replacing the old software that everyone knew with new software that no one does.

#### 1.1.1.3 *How long will an ERP project take?*

Companies that install ERP do not have an easy time of it. Don't be fooled when ERP vendors tell you about a three or six month average implementation time. Those short (that's right, six months is short) implementations all have a catch of one kind or another: The company was small, or the implementation was limited to a small area of the company, or the company used only the financial pieces of the ERP system (in which case the ERP system is nothing more than a very expensive accounting system). To do ERP right, the ways you do business will need to change and the ways people do their jobs will need to change too. And that kind of change doesn't come without pain. Unless, of course, your ways of doing business are working extremely well (orders all shipped on time, productivity higher than all your competitors, customers completely satisfied), in which case there is no reason to even consider ERP.

The important thing is not to focus on how long it will take—real transformational ERP efforts usually run between one and three years, on average—but rather to understand why you need it and how you will use it to improve your business.

#### 1.1.1.4 *What will ERP fix in my business?*

There are five major reasons why companies undertake ERP.

**Integrate financial information**—As the CEO tries to understand the company's overall performance, he may find many different versions of the truth. Finance has its own set of revenue numbers, sales has another version, and the different business units may each have their own version of how much they contributed to revenues. ERP creates a single version of the truth that cannot be questioned because everyone is using the same system.

**Integrate customer order information**—ERP systems can become the place where the customer order lives from the time a customer service representative receives it until the loading dock ships the merchandise and finance sends an invoice. By having this information in one software system, rather than scattered among many different systems that can't communicate with one another, companies can keep track of orders more easily, and coordinate manufacturing, inventory and shipping among many different locations at the same time.

**Standardize and speed up manufacturing processes**—Manufacturing companies—especially those with an appetite for mergers and acquisitions—often find that multiple business units across the company make the same widget using different methods and computer systems. ERP systems come with standard methods for automating some of the steps of a manufacturing process. Standardizing those processes and using a single, integrated computer system can save time, increase productivity and reduce head count.

**Reduce inventory**—ERP helps the manufacturing process flow more smoothly, and it improves visibility of the order fulfillment process inside the company. That can lead to reduced inventories of the stuff used to make products (work-in-progress inventory), and it can help users better plan deliveries to customers, reducing the finished good inventory at the warehouses and shipping docks. To really improve the flow of your supply chain, you need supply chain software, but ERP helps too.

**Standardize HR information**—Especially in companies with multiple business units, HR may not have a unified, simple method for tracking employees' time and communicating with them about benefits and services. ERP can fix that. In the race to fix these problems, companies often lose sight of the fact that ERP packages are nothing more than generic representations of the ways a typical company does business. While most packages are exhaustively comprehensive, each industry has its quirks that make it unique. Most ERP systems were designed to be used by discrete manufacturing companies (that make physical things that can be counted), which immediately left all the process manufacturers (oil, chemical and utility companies that measure their products by flow rather than individual units) out in the cold. Each of these industries has struggled with the different ERP vendors to modify core ERP programs to their needs.

#### 1.1.1.5 *Will ERP fit the ways I do business?*

It's critical for companies to figure out if their ways of doing business will fit within a standard ERP package before the checks are signed and the implementation begins. The most common reason that companies walk away from multimillion-dollar ERP projects is that they discover the software does not support one of their important business processes. At that point there are two things they can do: They can change the business process to accommodate the software, which will mean deep changes in long-established ways of doing business (that often provide competitive advantage) and shake up important people's roles and responsibilities (something that few companies have the stomach for). Or they can modify the software to fit the process, which will slow down the project, introduce dangerous bugs into the system and make upgrading the software

to the ERP vendor's next release excruciatingly difficult because the customizations will need to be torn apart and rewritten to fit with the new version.

Needless to say, the move to ERP is a project of breathtaking scope, and the price tags on the front end are enough to make the most placid CFO a little twitchy. In addition to budgeting for software costs, financial executives should plan to write checks to cover consulting, process rework, integration testing and a long laundry list of other expenses before the benefits of ERP start to manifest themselves. Underestimating the price of teaching users their new job processes can lead to a rude shock down the line, and so can failure to consider data warehouse integration requirements and the cost of extra software to duplicate the old report formats. A few oversights in the budgeting and planning stage can send ERP costs spiraling out of control faster than oversights in planning almost any other information system undertaking.

#### 1.1.1.6 *What does ERP really cost?*

Meta Group recently did a study looking at the total cost of ownership (TCO) of ERP, including hardware, software, professional services and internal staff costs. The TCO numbers include getting the software installed and the two years afterward, which is when the real costs of maintaining, upgrading and optimizing the system for your business are felt. Among the 63 companies surveyed—including small, medium and large companies in a range of industries—the average TCO was \$15 million (the highest was \$300 million and lowest was \$400,000). While it's hard to draw a solid number from that kind of range of companies and ERP efforts, Meta came up with one statistic that proves that ERP is expensive no matter what kind of company is using it. The TCO for a "heads-down" user over that period was a staggering \$53,320.

#### 1.1.1.7 *When will I get payback from ERP—and how much will it be?*

Don't expect to revolutionize your business with ERP. It is a navel-gazing exercise that focuses on optimizing the way things are done internally rather than with customers, suppliers or partners. Yet the navel gazing has a pretty good payback if you're willing to wait for it—a Meta Group study of 63 companies found that it took eight months after the new system was in (31 months total) to see any benefits. But the median annual savings from the new ERP system were \$1.6 million.

#### 1.1.1.8 *What are the hidden costs of ERP?*

Although different companies will find different land mines in the budgeting process, those who have implemented ERP packages agree that certain costs are more commonly overlooked or underestimated than others. Armed with insights from across the business, ERP pros vote the following areas as most likely to result in budget overrun.

### 1. Training

Training is the near-unanimous choice of experienced ERP implementers as the most underestimated budget item. Training expenses are high because workers almost invariably have to learn a new set of processes, not just a new software interface. Worse, outside training companies may not be able to help you. They are focused on telling people how to use software, not on educating people about the particular ways you do business. Prepare to develop a curriculum yourself that identifies and explains the different business processes that will be affected by the ERP system.

One enterprising CIO hired staff from a local business school to help him develop and teach the ERP business-training course to employees. Remember that with ERP, finance people will be using the same software as warehouse people and they will both be entering information that affects the other. To do this accurately, they have to have a much broader understanding of how others in the company do their jobs than they did before ERP came along. Ultimately, it will be up to your IT and businesspeople to provide that training. So take whatever you have budgeted for ERP training and double or triple it up front. It will be the best ERP investment you ever make.

### 2. Integration and testing

Testing the links between ERP packages and other corporate software links that have to be built on a case-by-case basis is another often-underestimated cost. A typical manufacturing company may have add-on applications from the major—e-commerce and supply chain—to the minor—sales tax computation and bar coding. All require integration links to ERP. If you can buy add-ons from the ERP vendor that are pre-integrated, you're better off. If you need to build the links yourself, expect things to get ugly. As with training, testing ERP integration has to be done from a process-oriented perspective. Veterans recommend that instead of plugging in dummy data and moving it from one application to the next, run a real purchase order through the system, from order entry through shipping and receipt of payment—the whole order-to-cash banana—preferably with the participation of the employees who will eventually do those jobs.

### 3. Customization

Add-ons are only the beginning of the integration costs of ERP. Much more costly, and something to be avoided if at all possible, is actual customization of the core ERP software itself. This happens when the ERP software can't handle one of your business processes and you decide to mess with the software to make it do what you want. You're playing with fire. The customizations can affect every module of the ERP system because they are all so tightly linked together. Upgrading the ERP package—no walk in the park under the best of circumstances—becomes a nightmare because you'll have to do the customization all over again in the new version. Maybe it will work, maybe it won't. No matter what, the vendor will not be there to support you. You will have to hire extra staffers to do the customization work, and keep them on for good to maintain it.

#### 4. Data conversion

It costs money to move corporate information, such as customer and supplier records, product design data and the like, from old systems to new ERP homes. Although few CIOs will admit it, most data in most legacy systems is of little use. Companies often deny their data is dirty until they actually have to move it to the new client/server setups that popular ERP packages require. Consequently, those companies are more likely to underestimate the cost of the move. But even clean data may demand some overhaul to match process modifications necessitated—or inspired—by the ERP implementation.

#### 5. Data analysis

Often, the data from the ERP system must be combined with data from external systems for analysis purposes. Users with heavy analysis needs should include the cost of a data warehouse in the ERP budget—and they should expect to do quite a bit of work to make it run smoothly. Users are in a pickle here: Refreshing all the ERP data every day in a big corporate data warehouse is difficult, and ERP systems do a poor job of indicating which information has changed from day to day, making selective warehouse updates tough. One expensive solution is custom programming. The upshot is that the wise will check all their data analysis needs before signing off on the budget.

#### 6. Consultants ad infinitum

When users fail to plan for disengagement, consulting fees run wild. To avoid this, companies should identify objectives for which its consulting partners must aim when training internal staff. Include metrics in the consultants' contract: for example, a specific number of the user company's staff should be able to pass a project-management leadership test—similar to what Big Five consultants have to pass to lead an ERP engagement.

#### 7. Replacing your best and brightest

It is accepted wisdom that ERP success depends on staffing the project with the best and brightest from the business and IS divisions. The software is too complex and the business changes too dramatic to trust the project to just anyone. The bad news is a company must be prepared to replace many of those people when the project is over. Though the ERP market is not as hot as it once was, consultancies and other companies that have lost their best people will be hounding yours with higher salaries and bonus offers than you can afford—or that your HR policies permit. Huddle with HR early on to develop a retention bonus program and create new salary strata for ERP veterans. If you let them go, you'll wind up hiring them—or someone like them—back as consultants for twice what you paid them in salaries.

#### 8. Implementation teams can never stop

Most companies intend to treat their ERP implementation as they would any other software project. Once the software is installed, they figure the team will be scuttled and everyone will go back to his or her day job. But after ERP, you can't go home again. The implementers are too valuable. Because they have worked intimately with ERP, they know more about the sales process than the salespeople and more about the manufacturing process than the manufacturing people. Companies can't afford to send their project people back into the business because there's so much to do after the ERP software is installed. Just writing reports to pull information out of the new ERP system will keep the project team busy for a year at least. And it is in analysis—and, one hopes, insight—that companies make their money back on an ERP implementation. Unfortunately, few IS departments plan for the frenzy of post-ERP installation activity, and fewer still build it into their budgets when they start their ERP projects. Many are forced to beg for more money and staff immediately after the go-live date, long before the ERP project has demonstrated any benefit.

#### 9. Waiting for ROI

One of the most misleading legacies of traditional software project management is that the company expects to gain value from the application as soon as it is installed, while the project team expects a break and maybe a pat on the back. Neither expectation applies to ERP. Most of the systems don't reveal their value until after companies have had them running for some time and can concentrate on

making improvements in the business processes that are affected by the system. And the project team is not going to be rewarded until their efforts pay off.

## 10. Post-ERP depression

ERP systems often wreak havoc in the companies that install them. In a recent Deloitte Consulting survey of 64 Fortune 500 companies, one in four admitted that they suffered a drop in performance when their ERP system went live. The true percentage is undoubtedly much higher. The most common reason for the performance problems is that everything looks and works differently from the way it did before. When people can't do their jobs in the familiar way and haven't yet mastered the new way, they panic, and the business goes into spasms.

### 1.1.1.9 *Why do ERP projects fail so often?*

At its simplest level, ERP is a set of best practices for performing different duties in your company, including finance, manufacturing and the warehouse. To get the most from the software, you have to get people inside your company to adopt the work methods outlined in the software. If the people in the different departments that will use ERP don't agree that the work methods embedded in the software are better than the ones they currently use, they will resist using the software or will want IT to change the software to match the ways they currently do things. This is where ERP projects break down. Political fights break out over how—or even whether—the software will be installed. IT gets bogged down in long, expensive customization efforts to modify the ERP software to fit with powerful business barons' wishes. Customizations make the software more unstable and harder to maintain when it finally does come to life. The horror stories you hear in the press about ERP can usually be traced to the changes the company made in the core ERP software to fit its own work methods. Because ERP covers so much of what a business does, a failure in the software can bring a company to a halt, literally.

But IT can fix the bugs pretty quickly in most cases, and besides, few big companies can avoid customizing ERP in some fashion—every business is different and is bound to have unique work methods that a vendor cannot account for when developing its software. The mistake companies make is assuming that changing people's habits will be easier than customizing the software. It's not. Getting people inside your company to use the software to improve the ways they do their jobs is by far the harder challenge. If your company is resistant to change, then your ERP project is more likely to fail.

### 1.1.1.10 *How do I configure ERP software?*

Even if a company installs ERP software for the so-called right reasons and everyone can agree on the optimal definition of a customer, the inherent difficulties of implementing something as complex as ERP is like, well, teaching an elephant to do the hootchy-kootchy. The packages are built from database tables, thousands of them, that IS programmers and end users must set to match their business processes; each table has a decision "switch" that leads the software down one decision path or another. By presenting only one way for the company to do each task—say, run the payroll or close the books—a company's individual operating units and far-flung divisions are integrated under one system. But figuring out precisely how to set all the switches in the tables requires a deep understanding of the existing processes being used to operate the business. As the table settings are decided, these business processes are reengineered, ERP's way. Most ERP systems are not shipped as a shell system in which customers must determine at the minutia level how all the functional procedures should be set, making thousands of decisions that affect how their system behaves in line with their own business activities. Most ERP systems are preconfigured, allowing just hundreds—rather than thousands—of procedural settings to be made by the customer.

### 1.1.1.11 *How do companies organize their ERP projects?*

Based on our observations, there are three commonly used ways of installing ERP.

**The Big Bang**—In this, the most ambitious and difficult of approaches to ERP implementation, companies cast off all their legacy systems at once and install a single ERP system across the entire company. Though this method dominated early ERP implementations, few companies dare to attempt it anymore because it calls for the entire company to mobilize and change at once. Most of the ERP implementation horror stories from the late '90s warn us about companies that used this strategy. Getting everyone to cooperate and accept a new software system at the same time is a tremendous effort, largely because the new system will not have any advocates. No one within the company has any experience using it, so no one is sure whether it will work. Also, ERP inevitably involves compromises. Many departments have computer systems that have been honed to match the ways they work. In most cases, ERP offers neither the range of functionality nor the comfort of familiarity that a custom legacy system can offer. In many cases, the speed of the new system may suffer because it is serving the entire company rather than a single department. ERP implementation requires a direct mandate from the CEO.

**Franchising strategy**—This approach suits large or diverse companies that do not share many common processes across business units. Independent ERP systems are installed in each unit, while linking common processes, such as financial bookkeeping, across the enterprise. This has emerged as the most common way of implementing ERP. In most cases, the business units each have their own "instances" of ERP—that is, a separate system and database. The systems link together only to share the information necessary for the corporation to get a performance big picture across all the business units (business unit revenues, for example), or for processes that don't vary much from business unit to business unit (perhaps HR benefits). Usually, these implementations begin with a demonstration or pilot installation in a particularly open-minded and patient business unit where the core business of the corporation will not be disrupted if something goes wrong. Once the project team gets the system up and running and works out all the bugs, the team begins selling other units on ERP, using the first implementation as a kind of in-house customer reference. Plan for this strategy to take a long time.

**Slam dunk**—ERP dictates the process design in this method, where the focus is on just a few key processes, such as those contained in an ERP system's financial module. The slam dunk is generally for smaller companies expecting to grow into ERP. The goal here is to get ERP up and running quickly and to ditch the fancy reengineering in favor of the ERP system's "canned" processes. Few companies that have approached ERP this way can claim much payback from the new system. Most use it as an infrastructure to support more diligent installation efforts down the road. Yet many discover that a slammed-in ERP system is little better than a legacy system because it doesn't force employees to change any of their old habits. In fact, doing the hard work of process reengineering after the system is in can be more challenging than if there had been no system at all because at that point few people in the company will have felt much benefit.

#### 1.1.1.12 *How does ERP fit with e-commerce?*

ERP vendors were not prepared for the onslaught of e-commerce. ERP is complex and not intended for public consumption. It assumes that the only people handling order information will be your employees, who are highly trained and comfortable with the tech jargon embedded in the software. But now customers and suppliers are demanding access to the same information your employees get through the ERP system—things like order status, inventory levels and invoice reconciliation—except they want to get all this information simply, without all the ERP software jargon, through your website.

E-commerce means IT departments need to build two new channels of access in to ERP systems—one for customers (otherwise known as business-to-consumer) and one for suppliers and partners (business-to-business). These two audiences want two different types of information from your ERP system. Consumers want order status and billing information, and suppliers and partners want just about everything else.

Traditional ERP vendors are having a hard time building the links between the Web and their software, though they certainly all realize that they must do it and have been hard at work at it for years. The bottom line, however, is that companies with e-commerce ambitions face a lot of hard integration work to make their ERP systems available over the Web. For those companies that were smart—or lucky—enough to have bought their ERP systems from a vendor experienced in developing e-commerce wares, adding easily integrated applications from that same vendor can be a money-saving option. For those companies whose ERP systems came from vendors that are less experienced with e-commerce development, the best—and possibly only—option might be to have a combination of internal staff and consultants hack through a custom integration.

But no matter what the details are, solving the difficult problem of integrating ERP and e-commerce requires careful planning, which is key to getting integration off on the right track.

One of the most difficult aspects of ERP and e-commerce integration is that the Internet never stops. ERP applications are big and complex and require maintenance. The choice is stark if ERP is linked directly to the Web—take down your ERP system for maintenance and you take down your website. Most e-commerce veterans will build flexibility into the ERP and e-commerce links so that they can keep the new e-commerce applications running on the Web while they shut down ERP for upgrades and fixes.

The difficulty of getting ERP and e-commerce applications to work together—not to mention the other applications that demand ERP information such as supply chain and CRM software—has led companies to consider software known alternately as middleware and EAI software. These applications act as software translators that take information from ERP and convert it into a format that e-commerce and other applications can understand. Middleware has improved dramatically in recent years, and though it is difficult to sell and prove ROI on the software with business leaders—it is invisible to computer users—it can help solve many of the biggest integration woes that plague IT these days.