SELECTING SOFTWARE FOR YOUR TRAINING DEPARTMENT IS ONLY PART OF THE EQUATION. HERE ARE SEVEN STEPS FOR SUCCESSFUL SOFTWARE IMPLEMENTATION.

Illustration by Susan Shap

IN THIS ARTICLE

HRD Management, Information Systems

JOHN SAW AN OPPORTUNITY to improve his training department. As director of training at a large midwest service company, his goal was to install a training management system to automate the tedious tasks necessary to run his training department.

It all started so well, too. He reviewed three popular training management systems and chose one based on price and features. He felt great when the package arrived in the mail the next week. What could possibly go wrong?

John quickly installed the package and was ready to go. *Oops.* His trainers weren't. They complained: "What do we do with the data in the old dBase system?" "Why do we have to change now? It's the peak of our training this quarter." What do we do if the system crashes?" Did you know that the new system requires a word processor that we don't have? We can't even print our reports." "How do we access the features that are in the brochure?" "Do we have to read these big manuals first?"

Oops is right. Unfortunately, John spent less time planning the implementation than he did deciding which software package to purchase. In fact, the two should have been done together.

Training management software is a strategic business application, necessary for the efficient operation of a corporate training department. It requires planning, planning, and more planning before the system is operational. Too often, trainers judge the cost of the software itself as the biggest expense of installation. While significant, it is only part of the total expense. The implementation is often larger.

What could John have done differently? What should he have done to make the implementation successful? Let's take a look at the basics in implementing any strategic business application.

Seven basic steps

An easy way to accomplish complex tasks is to break the tasks down into smaller parts. John could have understood the "Seven Basic Steps for Successful Software Implementation."

1. Determine the business need. As with any major project, the focus should be on problem resolution. The same applies to implementing training management software. First, determine exactly what the problem is. Why are you installing a new training management system? Who will be the beneficiaries? Who will be the users? Often the needs are different for various departments.

Don't forget future requirements. Do not settle for solving only today's problems. Plan not only for the future growth in classes and enrollments, but also for incorporating training in the strategic planning process.

There are other challenges, too. They include:

- How much time do you have to implement the system?
- How much budget, staff, equipment, and management commitment do you have?

- Who are the other players involved in the solution?
- What part does quality play?

Never forget what you are trying to accomplish. It is not unusual for the goal to change—what doesn't anymore? But, whether it changes or not, keep the goal clearly in focus. Question what role the tasks assigned to this project have in accomplishing the goal. Be aware that others will throw all kinds of tasks your way at the expense of your project. It will be confusing enough without these distractions.

2. Select the software and vendor. There will be trade-offs in any application that you choose—even if the application is developed internally. There is no perfect application for your exact needs. Study the choices available. Understand the functionality of each product, and test each function to understand how you would use the application. Don't be afraid to buy more product than you need initially. Not only will you soon grow into it, but you won't soon be repeating this process.

The products have depth—that is, there is functionality behind the features. Make sure that the functions of each feature are complete. It is easy to add features to an application; it is more difficult to add the functionality behind the features.

Just as important as selecting the software, is selecting the vendor behind the product. Make sure the vendor can provide the support you need and don't hesitate to use this expertise. Be wary of any software vendor that does not offer a full range of professional services, either through a local representative or one that the vendor provides. This means having more than a support person on the other end of the phone. Choose a vendor willing and able to come to your site to help with implementations, to train your employees in how to use the products, and to provide the telephone support services that you need.

Determine Cost Constraints and Real Cost

So much attention is usually focused on the initial cost of the software license that the total cost of converting to a new training management system is often overlooked. Everyone wants to know, "What does it cost?" before they ask, "Can it do the job?"

There can be a significant cost to implementing software. Installation, conversion, and training can cost many times more than the software license.

Make sure that the software that you are implementing will be a long-term solution for your training department. Get the best price that you can for software license, but be leery of trying to save a little money on the software by buying a product that will cost more in headaches later. Probably no software product will do everything that you need. Understand the trade-offs and the underlying cost of not providing services.

How much does it cost when you under-book a class? How much do cancellations cost? Are you able to charge cancellation fees? How much does it cost when students enroll in classes for which they have not met the prerequisites?

Newer training management software includes a feature for building relational data bases. This kind of design is preferred. Choose a solution that offers not only a relational database, but also normalized data structures. Have your MIS department do a technical evaluation of the product.

You are buying more than diskettes and manuals; you are buying a strategic business solution. Make sure the vendor offers not just a solid product, but also a team able to help you make the solution work.

■ *Know what your disaster recovery plans are* ■

Make sure when developing your timelines that you consider the ability of the vendors to provide their products and services. Special hardware and software take time to order and install. Not only does unavailable hardware cause delays, but so can professional services such as conversion, installation, and training. Confirm in advance that your software vendor can schedule the services when you need them.

Ask the questions that you normally ask anyone providing key business solutions: How long have they been in business? Is this software the main part of their business or is it just an add-on product? How knowledgeable are their employees? How confident are they in the solution? Are they profitable? And check references. (See accompanying checklists for selecting a software application and vendor.)

3. Establish the environment. Pay attention to the underlying technology. These support systems will play an important role in the success of your system. Operating systems, networks, database management systems (DBMS), and communication systems are each complicated technology. Trying to connect all of these systems is difficult even before the application technology is introduced.

Also part of the environment is a backup system. You have all heard war stories about what can happen when information is not backed up. You may even have one or two of your own. Recovering data is not a pleasant task, so a good rule to follow is: Never enter more data than you can afford to spend the time to recreate. In other words, back up your data often.

Backup is normally the responsibility of the MIS department. However, do not assume that they will take care of it for you. Know how often the backup occurs, what is backed up, who is responsible for verifying the backup, and who is responsible for recovering any lost data. Develop the recovery procedures. Who is authorized to decide that data should be recovered?

Be sure to test your recovery procedures, and the first time that you test the procedures should not be the first time that you need to use them.

Know what your organization's disaster recovery plans are. Does your application qualify for disaster recovery? Many strategic systems have procedures in place for off-site processing if a disaster shuts down a central processing center. Check with your MIS staff to see if you qualify. (You may not want to qualify. There are usually more procedures that must be followed every time you make a change.)

Networks are responsible for storing and retrieving programs and data for multiple users and multiple applications. A network includes:

◗ File servers. Network file servers physically store data shared by many people. File servers also provide some security access protection and other services such as printer sharing.

◗ Database servers. Database servers (such as Oracle, SQL Server, and Sybase) are used when record volumes get larger, when quick response is needed, or when data is dispersed across wide areas. These servers require specialized skills to administer and sometimes specialized hardware, but can be justified for the security and performance they provide.

◗ Workstations. It is better to buy more hardware than you need than to buy less. However, there is no need to overbuy. Check the minimum and recommended requirements of the vendors. Make the recommended your minimum. It is well worth the small difference in price for a strategic system. Factor in the requirements of other applications. Go for quality, especially if you are buying a quality software application. This is a long-term investment. It will pay off.

◗ Communications. Computers talk to each other; they pass information back and forth. Just as a person who speaks only English has trouble talking with someone who speaks only French, computers must also have a common language. There are two terms that you will hear when discussing computer communications:

> ■ ***Don't be afraid to buy more product than you need initially*** ■

Protocol layer. Protocol layer defines the network language. Examples include TCP/IP, IPX, and SPX.

Bandwith. Bandwith determines how much data can travel from one computer to another in a given time. Just as superhighways can carry more traffic than country roads, wide bandwidth can carry more data traffic.

4. Integrate your data. Replacing an older system sometimes requires running both the old system and the new system concurrently for a while. This is an excellent way to fine-tune the new system, to address any peculiarities, and to work out any differences. Usually the new system is used for tests and the old system contains the "live" data.

Data conversion always takes longer than anticipated. Make sure that your MIS department understands the underlying technology of the old system and can format its data into a file that the new system can import. Understand also that while the data is being converted, neither system can be used. Be realistic about how long this process takes.

You may also want more data from other applications to the new system. Human resources data, accounting, billing, even payroll data are often integrated with corporate training data. However, this may be difficult because of ownership and confidentiality issues, differences in data base systems platforms (Windows versus Unix), and application problems.

If data cannot be shared, then some mechanism for updating data periodically must be determined. Work on a solution with your MIS staff, the owners of the other data, and your vendor.

5. Change processes. Implementing the new software applications will change the processes in your department. There may be new tasks to perform, obsolete ones to remove, or just a different way to do the tasks. Here are some brief examples.

◗ Identify your reporting needs. Entering *data* is one process. Extracting *information* from the data is another.

◗ Decide what parts of the new application are most important. As tempting and exciting as a new application can be, you may not need to use all features of the system immediately.

◗ Standardize your data. Have consistent ways to define course titles, students, trainers, classrooms,

Determine Time Requirements

How much time do you have allocated to complete this project? That time includes any calendar requirements (end of quarter, before next training session starts, and so forth), plus the implementation team's personal time constraints.

Balance the time between software selection and software implementation. Do not fall into the trap of postponing the selection process until the last minute, expecting to have a fully implemented system by the project end date.

Consider the time constraints of others, too. Choose a time to implement that has the least impact on your training department. The impact of training management software at the wrong time can doom even the best products. For example, do not implement a new system at the start of a new training semester.

You will probably rely heavily on your MIS department during the implementation. Understand their busy times also. They often have their own end-of-quarter and end-of-year workloads.

During the actual conversion process, there will be time when neither the old system nor the new system will be available. Plan this outage for a convenient time for the user of the system. Work with your MIS department to minimize the outage.

SOFTWARE VENDOR CHECKLISTS

Vendor

□ How long has the vendor been in business? ______ years

□ How many years have they been profitable? ______ years

□ Have you run a Dun and Bradstreet on the company? □ Yes ______ Date

□ Who owns the company? ______

□ What is their customer base? ______

□ What is the newest version of their product? ______

□ What are their support hours? ______

□ What are their support terms? ______

□ What are their support costs? ______

□ Do they provide toll-free support? □ Yes □ No

□ What expertise do they have available to you? ______
On-site consulting? □ Yes □ No
Telephone consulting? □ Yes □ No
Online consulting? □ Yes □ No

□ What are the consulting costs? ______

□ What training do they provide? ______
How many courses? ______
How many classes per year? ______
Do they train on your site? □ Yes □ No

□ Do they offer client/server technology? □ Yes □ No

□ Do they understand your business requirements? □ Yes □ No

Software

□ Is the product: □ Feature rich □ Function rich

□ Does the product meet your core business requirements? □ Yes □ No
Does the product solve your extended business requirements? □ Yes □ No

□ Does it have a relational database design? □ Yes □ No
How many tables? ______
Average width of tables? ______
How many indexes? ______ bytes? ______
Any redundant data? Why? □ Yes □ No
(Too much redundant data indicates poor product design.)

□ Any repeating data groups? □ Yes □ No
(Another indication of poor product design.)

List functionalities ______

equipment, and so forth. Identify the exceptions.

- Verify imported data and make sure it matches new standards. Otherwise, you will be generating the old mistakes into the new application.
- Enforce your standards.

6. Installation. Product installation is usually the easiest part of the implementation process (that is, except for those who think that installation and implementation are the same thing), but there can be problems during installation. Diskettes can be defective. You may not have enough disk space to load the new application. There may be conflicts between applications.

Equipment must also be installed. Sometimes the installation of the hardware must be coordinated with the installation of the software. Windows programs often require more memory, a mouse, a better monitor, or maybe even a CD-ROM drive.

When replacing an older system, an absolute must is to back up the old system before installation. This is true even if the new system is an upgrade from the old system. (Going from Version 1.0 to Version 2.0, for example.)

Make sure that you have enough hard disk space for both the old system and the new system until you are ready to turn off the old system.

Install any customizations to the system—for example, custom reports. Also, install any complementary software such as word processors, calendars, or spreadsheets that may be used.

7. Training. Save training to the end—just before testing the old and new system in parallel. Users should get the most intense training. They need to know not only how to use the new system, but also which options are installed and which can be defined by the user.

Train management on the benefits and paybacks of the solution. Show them examples of the information that they will be interested in.

Train the support staff not only on their responsibilities, but also on the importance of the system so that they will understand the significance of their roles. Make sure they understand backup, recovery, security, and performance.

8. Prepare for upgrades. I can hear you say, "What do you mean step eight? You said there were only seven steps!"

And so there are. But as soon as you have completed all seven steps, prepare for upgrades. New versions of software can have the same impact as installing a new application. Any upgrade to any component needs to be planned—including the opening system, network, and DBMS. Continually review your goals. Know how your application changes as your business changes. Business changes can shorten the life cycle of your application. Software, like hardware, can become obsolete. Be prepared. ■

Bob Dust *is president of Gyrus Systems, 620 Moorefield Park Drive, Suite 200, Richmond, VA 23236. Phone: 804/320-1414; fax: 804/320-2323.*