

Technology Training – Got that Covered?

You bought the training with the software. Maybe the software company threw it in. You believe you have what you need to get the system into production. The software sales guys want you to think that, also. However, as a savvy buyer, you need to be aware of two principles:

1. Tech training is for technology guys.

A common mistake IT professionals make when acquiring a software package is believing that the accompanying training fits all users. However, most standard training is for systems implementers – those who need the technical understanding to establish codes and map data. The intent of this training is to equip the technologist to program, operate and troubleshoot the system.

The hallmarks of this training are obvious. It incorporates extensive documentation, often housed in multiple five-inch binders. It has to – it must be comprehensive enough to address rat holes and odd error messages, so the new programmer understands the logic of the author. This training might be weeks in duration, so participants grasp how the system works end-to-end. If you are angry at your end-users, send them to this training.

2. End-user training is for your business people.

In contrast, end-user training is built to equip the lay person to use the system to accomplish an outcome. This training is narrowly focused, written for specific jobs. It is hours or days in duration, and relies heavily on job aids and online help. It is contextually driven: “Ms. Accounting Manager, here is how to close your books using this tool.”

This training is tight and tailored. There are as many shades of it as there are job roles because the focus is on doing the job. And well-constructed end-user training is process driven, much like an in basket exercise: “Here are the invoices, Participants. Use the system to run accounts payable, check for and correct errors, create a final report, and send it to the appropriate recipients.”

Even packaged end-user training must be customized.

New training is just like a new system. A “vanilla” installation still requires input before it can operate. Likewise, the field codes, approvals, error messages, and reports need to be “loaded” into the “vanilla” training for the learning experience to be relevant and effective.

You might make a case that accounts payable is accounts payable is accounts payable. But anyone who has changed jobs a few times will tell you work processes vary from company to company. Ultimately, the goal of end-user training is to get users productive as soon as possible. The more specific we make the training to each individual’s job, the more quickly they will learn and want to use the system.

What about intuitive systems?

We live in exciting times. Programmers are developing packages that are visually gorgeous and genuinely intuitive. All hail the era of great design!

Still, intuitive systems require training and suffer from a lack of it. For example, while most businesses use Microsoft Outlook for email and scheduling, very few employees use it for anything more. The Journal capability for customer relationship management? The Task capability for project management? Exactly.

The iPhone is another example. If you have one, think about how you learned its hidden capabilities. Did you know how to monitor traffic, or cut and paste? It’s a gorgeous, elegant design that requires months of inefficient learning. Can you afford your team taking months to figure out the cool uses of your new system?

In short, you need systems training for the systems team, end-user training for the end-user, and some kind of training even for intuitive systems.

Questions worth asking before you buy:

- **What kind of training do you provide?**
- **Who is the intended audience?** Beware of any response resembling “anyone can take this.”
- **May I see the learning objectives?** This will reveal the intended audience and instructional quality.
- **What kind of training customization do you provide?** Is it part of the contract?
- **What training support materials do you provide?** Insist on context-specific help.
- **Will you provide electronic source materials?** This is like providing access to source code. Someone will have to maintain these materials!
- **How often do you update training?** Hint: With every software update!
- **What are the electronic materials created in?** It might tie to your learning management system.
- **Who teaches the instructor-led sessions?** Is this designed as a train-the-trainer? Are the instructors programmers, subject-matter-experts, or professional trainers?