Training 101

Computer Anxiety

You can spend thousands of dollars on a new computer system to increase productivity. But if employees can't use it, or don't want to, all you've increased is the red ink in your ledger books.

Here are some recommendations for getting the most out of automation. The first set deals with the basics of any training course—the attributes that make successful trainers and learners—but tailors them to a high-tech training environment. The second gives advice on technological change, and ways of overcoming employee reluctance to learning a new system.

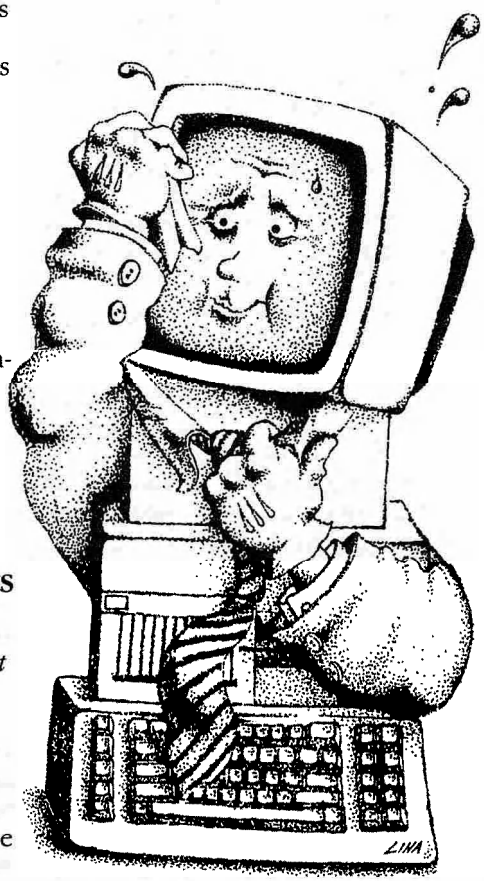
The Essential Ingredients

By Susan B. Rakes, an instructor at SAS Institute, Box 200075, 12675 Research Boulevard, Austin, TX 78720-0075.

The recipe for serving up a successful training course calls for three ingredients: good learners, a good trainer, and a good learning environment. But how do you know when you've got them?

For high-tech training, the right mix seems particularly important. High-tech training environments can be intimidating, even to experienced computer users. Along with the usual hurdles, you've also got to try to alleviate computer anxiety, so that students can learn the skills and transfer them to the workplace.

In *Understanding and Facilitating Adult Learning,* Stephen Brookfield discusses what it takes to put together a successful training course. He lists attributes of good students and teachers for general adult-education classes, but his recipe can be adapted for high-tech training environments. Many of these 20 recommendations are also useful for other kinds of training sessions.



1. Know whether learners are motivated to learn. I have often taught technical computer-software classes to students who have been told that they would be the office experts in the software. They are always either excited about learning the new software, or upset about the redirection of their daily office routines. Students who are upset about their new roles seldom learn more than the basics. They often slow down anyone else who works on shared projects during computer lab time.

2. Use a learning format that allows for individual differences in ability and style. The classes I teach are a combination of lecture—where each student has a paper copy of my overhead-projector transparencies—and computer-terminal lab work. Student hear the information, see it on paper, and then use the techniques on the computer. The course notebook leaves ample room for note taking during the lecture. A technique I've picked up from another instructor is to use a window of colored transparency to surround and highlight a particular line of code in a program or point being made on the overhead screen.

3. Make sure new learning builds on learners' current knowledge and attitudes. My most successful students are those who have used the software before attending the class. They have already learned the terminology and can add to that base of knowledge in the classroom. Other successful students have previously attended the class, viewed videotape versions of the material, or worked through computer-based training courses.

If some students have not been exposed to the material, I try to interview them briefly during the introductions. If I find that they have some background in data-processing, I assure them that what we are going to cover is similar to other programs or languages. That puts them in a more receptive state. Students who have never used computers have to overcome a very real anxiety, especially those who have been out of school for some time.

4. Reinforce learning. Most of the classes I teach run for three days. Each morning I begin with a review of the most important concepts and syntax covered the previous day. The course material is designed in a building-block fashion: concepts introduced in each chapter appear again in later chapter examples.

5. Allow students opportunities to practice what they are learning. At least twice a day we leave the lecture room and reconvene in the computer-workshop room. The students are assigned a set of exercises. I stress that some people will not be able to finish the problems during lab time, and offer the use of the terminals before class each morning,

and at the end of the day.

6. Help learners to be active participants. During the lectures, I encourage students to ask questions as they come to mind. I often open the floor to anyone who would like to answer a question. During the workshop periods, all students use the computer to solve problems. When there are two people to each terminal, the dialogue between them often sparks faster and more imaginative answers than a single student might find when working alone.

7. Organize lessons into manageable units. My three-day course is divided into periods of lecture followed by periods of computer workshop. Most lecture periods last less than an hour. Occasionally, we cover two chapters before adjourning to the workshop room. That can take a long time, so I usually allow a five-minute break between chapters, in order to keep the students (and teacher) alert.

8. Offer guidance in developing solutions. During computer-workshop labs, I walk around the room looking over the shoulders of students to see how they are doing. It is difficult for me to go on with my lecture without stopping to point out their errors. I answer questions when asked, and offer advice when students appear stymied, but most students can eventually solve their own problems. That is an important part of the lab experience.

9. Show students how new skills and knowledge relate to what they already know. In the world of computers, very little is unique. Most programmers agree that every computer language does the same things as every other language, but with different commands. Students who are new to computers may not know that. I mention to novices the broad applications available for the programming techniques learned.

10. Make sure the material to be learned is meaningful to learners. That relates back to the first point. If a student is positively motivated, rather than thrown into class against his or her will, he or she will see ways the new information can be applied to work tasks. Many students bring copies of their programs from work to see how the skills they are learning relate to their jobs. Often I get comments such as, "Now I know what this program is doing."

11. Be precise and clear. As a teacher of technical computer-software classes, I must remind myself what it was like when I was first exposed to the material I now teach regularly. Being able to put myself in the position of the novice student has helped me explain and re-word the information so it will be understood by everyone. Public-speaking skills require practice. If I don't teach for a few weeks, it takes a few hours for my normal cadence and flow to return.

12. Be animated, generate excitement, and use humor. That means both physical animation and voice variety. During the lecture periods of my classes, I often shift into what I call "play-acting" to rouse the students. I become the computer and talk to the programs as if they were asking for impossible tasks. I often talk with my arms and use pointers and colored transparencies to highlight information. My wry sense of humor is sometimes missed by students who are fading in and out of concentration, but when others laugh, the out-of-touch students come back.

13. Show concern for learners. I start each class with a brief introduction of myself. Then I have students introduce themselves. I find that once a student has broken the ice by talking out loud, he or she is more willing to ask questions and contribute information during class. It also shows students that I know they are there, and that I care about them, not just about the course materials.

14. Know thy subject. If your knowledge is only as deep as the course notes you're teaching from, it won't take the students long to plumb the depths. I study and practice the techniques before teaching, and admit to the class when I don't know an answer, rather than try to bluff my way out. It takes a long time to realize that students don't expect you to know all the answers off the top of your head, and that it's OK to research the answer after a question comes up.

15. Relate theory to practice, and relate learners' field to other fields. Because I usually lecture for about an hour, and then allow students to practice in a computer workshop, they get immediate reinforcement of the new information.

16. Appear confident. Study, practice, study, practice, study, practice, and use a good deodorant.

17. Be open to different approaches. When teaching computer-software classes, I always tell the students that suggested solutions can be found in the back of the book, but that many other approaches can accomplish each task. Each student usually comes up with a different program, most of them correct. If someone discovers a particularly unusual or effective technique, I share it with the class.

18. Present an authentic personality to the class. It took years for me to figure this one out. I had my social persona, my office persona, and my teacher persona. When I rolled them all together and found the real me, all aspects of my human interactions improved. Now I have fun in the classroom, and I get things accomplished at the same time.

19. Be willing to go beyond class objectives. Each class I teach has a set of objectives. As I review them, I ask students to add to the list any personal objectives they've brought to the class. If what they want is not within the scope of the course, I schedule time outside of class hours to cover that information.

20. Create a good atmosphere for learning. I work for a company that knows the value of a comfortable learning environment. There's always coffee, tea, cocoa, soft drinks, juice, breakfast breads, crackers, and peanut butter available for the students. I make sure the students know the class is under their control. If someone wants to leave the room, it's OK. If everyone leaves the room, it's break time! Students are allowed to bring food or drink into the classroom or computer-workshop room. They are invited to arrive before class time to use the terminals. They are offered telephones to keep in touch with office and home, and a first-aid kit with aspirins and band-aids.

For out-of-town students, my training coordinators offer maps, restaurant lists, and directions. I feel like a hostess for my guests and their families. If a student is relaxed and comfortable, learning is much easier.

When Technology Meets People

By Lloyd P. Steier, a visiting associate professor at 3–23 Faculty of Business Building, University of Alberta, Edmonton, Canada T6G 2RG.

Computers, office automation, and new communication technologies are changing organizations more than anything we have seen in the last 200 years. New information systems pop up at a phenomenal rate, as organizations adapt to their changing environment with ongoing efforts to improve efficiency and effectiveness.

But new office technologies have not achieved their much-acclaimed promise in the workplace. Despite huge increases in high-tech spending, white-collar productivity has remained virtually flat. One reason, according to an article in the February 1988 issue of the *Training & Development Journal* ("In Practice," page 8), is that many organizations "approach computer-software purchases equipped with little more than a blank purchase-order and blind faith in the power of technology itself to improve business operations."

Despite huge increases in high-tech spending, white-collar productivity has remained flat

In the face of such rapid change, many managers are preoccupied with the technical issues surrounding the new systems. Meanwhile, they neglect the equally important human resources element. But the way people relate to technology often determines how useful it will be.

HRD specialists who want to help companies achieve those all-important connections between people and computers should keep in mind eight key points.

1. Technological change is inevitable

A commitment to the new information technologies is a commitment to ongoing organizational change. The technological capability to produce new and innovative communication devices and applications will continue to expand.

Most organizations recognize that they must adopt new technologies in order to remain competitive, but many take a short-term view, believing that "one-shot" acquisition and training will suffice. Managers must take a longer-term view of change. Implementing new technologies and training employees to take advantage of them should be an ongoing process, not a single event. Appropriate policies need to be established to reflect the ongoing nature of technology acquisition and training.

2. Resistance to change is natural

Don't assume that people will automatically adopt a new technology, merely because it is made available in the workplace.

In his 1980 book, *Mindstorms*, Seymour Papert uses the often-cited example of the standard "QWERTY" typewriter keyboard to illustrate that we have a tendency to preserve "practices that have no rational basis beyond their historical roots." While computer technology now makes it possible to tremendously increase typing efficiency by rearranging the standard keyboard, such a change has not received serious consideration, primarily because of the mammoth "human" implementation problems.

Yet, many organizations often proceed with far more ambitious changes without even addressing the problem of resistance to change. Information technology and the many facets of its effective utilization require managers to address the problem of resistance to change.

3. Change must make sense to everyone involved

Individuals must understand why a new way of doing things is preferable to the one being replaced.

For example, an ambitious attempt to implement a new Local Area Network was met with much resistance to change from people involved with word processing. Users of the system being replaced believed that its software and print quality was superior to those of the new system.

Management intended to phase out the old system because of its limited networking capability.

Furthermore, top managers viewed the new system as being in a prototype phase, with the software and printer-quality problems as temporary. But they did not explain that to the people working with word-processing. Management viewed the new network as being on the "cutting edge," actually providing enhanced career opportunities for employees, but employees felt like unwilling guinea pigs, developing skills for which there was little demand!

The problem was compounded because most employees viewed the skills they needed for using the old system as being important to their advancement. Not all employees had yet acquired those skills, and they continued to be coveted within the organization. In such a situation, a little communication would obviously go a long way in alleviating the resistance to the new technology.

Management must establish effective dialogues with all interest groups, particularly those directly affected by an innovation. The dialogue must allow participants to make sense out of the intended change.

4. Organizational settings vary

Some organizations are receptive to change and adapt easily to it. Others resist anything new. The interests of particular managers, the educational level and experience of employees, demographic factors such as age, and employees' contact with other elements in their environment may affect an organization's receptivity and capacity to change.

Even within one organization, different work groups may vary in their willingness to change. One department may show enthusiastic bottom-up support, while a similar work group exhibits outright resistance.

Unless it is absolutely critical to the success of the organization, management should not assume that implementation of an innovation can, or should, be uniform throughout the company. Implementation strategies must allow for a broad range of differences in receptivity and capacity. Each work group should be encouraged to adapt to the change in a way that makes sense to its members. Excessive top-down control may result in only a symbolic following of guidelines, and may even be counter-productive.

A primary task of management is to determine where particular work groups are in relation to use of the new technology, and to provide each with an appropriate level of support. Such an approach requires a "simultaneous loose-tight system of control" similar to that recommended by Peters and Waterman in *In Search of Excellence.* In such a model, some elements are tightly controlled while others are given more flexibility. Successfully managing that apparent paradox requires managers to identify and use strategies appropriate to the organizational setting and different elements of the innovation.

5. Change takes time

Managers often expect too much, too soon. As a result, they may prematurely judge an innovation as unsuccessful.

For example, senior management at one organization spent a lot of money doing a cost/benefit analysis of a prototype system that was not yet fully implemented. The evaluation was carried out according to previously established, rigid timelines, which did not take into account the unforeseen implementation problems that had arisen.

Organizational settings must be able to make some sort of sense of and respond to innovations in their own ways. Often there is no quick-fix solution or substitute for an investment of time. Management must have a realistic time perspective when implementing changes and judging their success.

6. Change results in role overload

Several departments in one large organization were already overloaded. A new technology required fundamental changes in the way people did their everyday tasks, but management didn't take into account the extra demands. People were expected to learn new skills and ways of doing things, while at the same time maintaining their previous level of performance. The increased workload contributed to worker anxiety and frustration, which fostered a negative attitude toward the change and made successful implementation more difficult.

Management must recognize the potential for role overload when implementing change, and provide appropriate support such as additional resources, support staff, or a reduced workload.

7. Change requires ongoing technical support

In a division of one company, a Local Area Network was installed by an outside consultant. After it was up and running, it was turned over to the organization. During the first three and a half months of implementation, 26 major problems were encountered—everything from a loose cable to a complex problem with the internal hardware.

To deal with the problems, the department had to call in outside help about twice a week. Sometimes another department in a different building could give the needed support, but most of the time, the outside installer had to be called in. Unlike Maytag repairmen, the outside consultant's staff was usually committed to work in other organizations and not readily available. That left the new system down for

unnecessarily long periods of time, hampering implementation efforts as users became more frustrated with the new technology.

To avoid similar situations, managers must try to provide an adequate level of technical support for any intended innovation.

8. All change involves learning

Learning is said to occur when practice or experience leads to change in behavior. It is an incremental process. But a learning process that appears to work smoothly in one organizational setting often cannot be effortlessly transferred to another setting, even a similar one. Skills that may appear simple and effortless may be the result of a long learning process. Some things can only be learned through practice and evaluation.

Managers and HRD specialists have to keep in mind the complex learning processes that accompany technological innovation and implementation. In particular, they must address questions about what employees need to know about the new technology. They must design appropriate steps toward gaining that knowledge and skill, and provide a supportive environment in which the learning can take place. And, they must develop an ongoing system to monitor the achievement of what they have set out to do.

A commitment to change

We cannot predict how the rapid technological changes we are experiencing will ultimately affect our organizations. The only certainty is that technological change is inevitable. A commitment to technology—and often we have little choice about whether we want to make that commitment—is a commitment to ongoing change. Managers must develop strategies that take into account the reality of technological change and the human factors that can determine its successful implementation.

"Training 101" is edited by Cathy Petrini. Send your short articles for consideration to Training 101, Training & Development Journal, *1630 Duke Street, Box 1443, Alexandria, VA 22313.*