A Support Crisis of Our Own Making? Training Versus Equipping

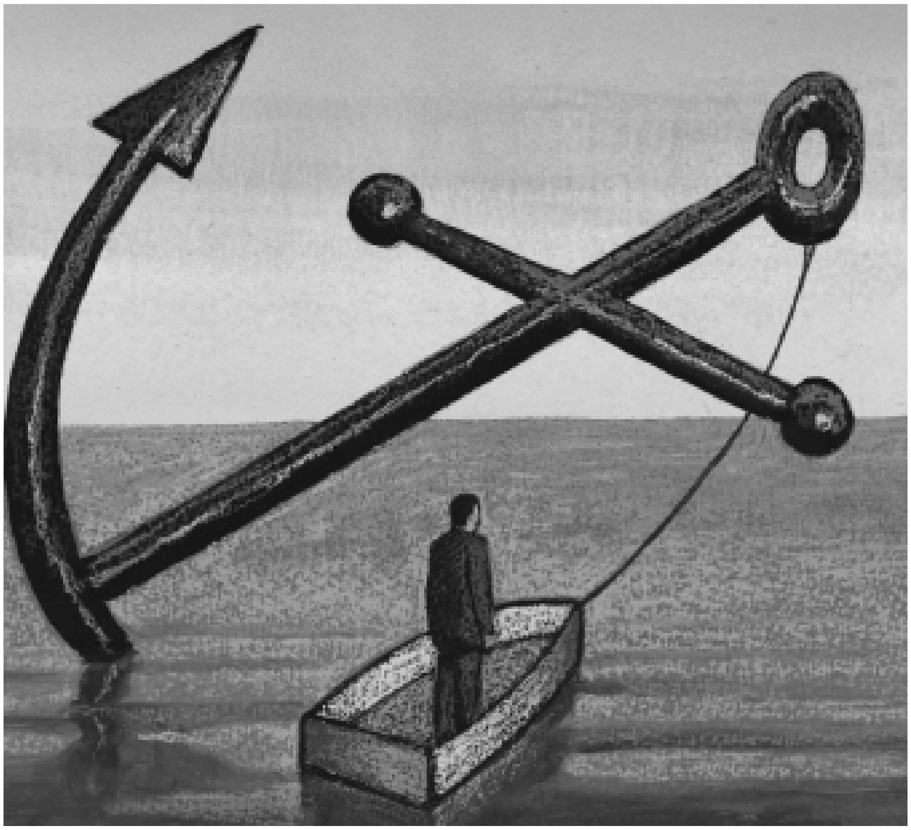
For those in technology support roles who are groaning under ever-increasing demands for training — focus less on training and more on equipping

By **Jeff Overholtzer**

At many colleges and universities, technology education programs focus on training. Training has its place, but the model of learning on which it's based is very limiting. Applied to technology, the word suggests a highly structured, mechanistic process that helps produce a specific skill, such as formatting text in a word processor or adding animation to slides in presentation software.

We need a wide focus to plan and promote a range of activities that equip faculty members to cope with a turbulently changing technology environment. An emphasis on training is doomed to failure because a task-oriented training approach ensures the user's dependence on formal training and help desk support. With the considerable array of technology tools available on the typical campus, and the rapid rate of change in software, it's impossible to train everyone.

An equipping approach, by contrast, uses certain strategies to help faculty and staff learn on their own. These strategies include building a foundation of essential computer concepts, presenting model uses of software in real-life contexts, assisting faculty and staff to understand their learning styles, and employing various strategies to help them apply their knowledge creatively to solve technology problems and learn software on their own.



Training Bogs Down

Training strategies draw heavily from the behavioral view of learning, which posits that specific behavior outcomes can be produced in a learner in response to manipulations in the learner's environment. Training effectively helps students learn skills that don't require high-order mental processes but are nonetheless important for using information technology effectively.

For instance, mouse skill is essential to effectively using modern graphical user interfaces. Mousing can be taught using rote exercises repeated until proficiency is attained. Some computer-based mouse training software provides feedback for the student: perform the function correctly — for instance, double-clicking on a small square — and the square morphs into a bouquet of flowers. This stimulus-response interactivity characterizes learning activities driven by behavioral theory.

Required Knowledge of Hardware, Software, and Networks

The program

- The program as a sequence of steps
- Distinction between operating system and application software
- Upgrades: Why are they necessary? Why so many?

Data

- What it is
- How it's stored and accessed
- Why back-ups are important
- The end user's responsibility

The desktop computer

- Volatile (such as RAM) and permanent (such as hard disk) memory
- Central processing unit
- Bus
- Input and output devices

Networks

- Local area networks: wires, switches, software
- Client-server computing: application servers, file servers, access rights
- Security: maintenance, end user's responsibility
- The Internet and the World Wide Web: connection of the campus LAN to the Internet by an ISP, the Internet as a network of networks, the Web as a *medium*, open versus licensed or restricted sources of information, library-quality resources on the Web

Perform the task correctly, receive a reward; fail to perform it correctly, and the reward is withheld.

Training in this fashion is perfectly appropriate for skills, like mousing, that are performed mechanically and without much conscious thought. The goal is to achieve automaticity. In educational psychology literature, automaticity is described as the ability to perform certain tasks with little apparent mental effort, freeing the learner to concentrate on higher-level cognitive processes.

The approach that works so well for teaching mousing is less effective when it comes to more complex activities, however. For instance, if we want a faculty member to produce a gradebook of a particular kind using a spreadsheet, we can provide a structured exercise that guides the person from step to step until an outcome is achieved. The process might look like this: launch the spreadsheet program; type in these row entries, representing students in a hypothetical class; type in these column entries, representing assignments; and so on. The faculty member is told little about the reason for typing these entries or about the underlying logic of the spreadsheet. He has little opportunity to experiment or to adapt the exercise to his specific needs. The goal is to produce a "model" spreadsheet gradebook with certain characteristics.

If the faculty member faithfully follows the prescribed steps, he will achieve the desired result. But when the course, assignments, and grade weights in his own course diverge significantly from the exercise, he'll be stuck. You know what comes next — the call to the help desk or an appeal to the nearest and most convenient computing support person.

An over-reliance on training shortchanges both learners and the computing support operation. Learners require a constant stream of workshops to update their skills and must call for help when they attempt anything new or encounter anything unexpected. Those in computing support must pour time and resources into organizing and repeating workshops, and answering calls for help. The continuing rapid change in computer technology only aggravates the tension between growing demand and finite resources for training.

The most efficient use of limited IT staff is to focus on equipping strategies: help faculty understand which tools to use, when to use them, and how to quickly adapt to new and upgraded software tools. Learning theory suggests some approaches in helping faculty and staff members learn how to learn using technology:

- Teach learners a foundation of knowledge necessary to function effectively with technology.
- Teach metacognitive skills that help learners apply knowledge to novel situations and to learn on their own.

Build a Knowledge Foundation

Knowledge structures — that is, knowledge about fundamental computer concepts — provide an important foundation for the learning of computer skills. Nickerson,[1] Gagne and Briggs,[2] and others emphasize the importance of certain intellectual skills as a means of acquiring more knowledge.

A National Research Council study[3] shows how these ideas relate to the specific domain of computing. The study asserts that an understanding of "foundational concepts" of computers, networks, and information is essential to becoming a proficient and self-sufficient user of information technology:

> Concepts explain the how and why of information technology, and they give insight into its opportunities and limitations. Concepts are the raw material for understanding new information technology as it evolves.

What are these foundational concepts? We can distinguish two basic areas of knowledge necessary for faculty in a typical desktop computing environment with network resources: general concepts on desktop computing and networks (see the sidebar "Required Knowledge of Hardware, Software, and Networks"), and information about the desktop computing operating system and the general capabilities of common software tools (see the sidebars "Required Knowledge of the Windows Operating System" and "Required Knowledge of Application Software").

These foundational concepts help users become more effective and self-sufficient in several important ways:

- Users are better equipped to make good choices about which technologies to use and when to use them. Furthermore, understanding these concepts helps users become more flexible and creative in using technology tools.
- Users can draw on the concepts for basic troubleshooting and problem solving.

These concepts should form the core of any program to equip users in higher education to make effective use of computers. The concepts can be the basis for entry-level workshops (see Table 1, next page). All users should be encouraged to take the first three levels of workshops or to obtain the equivalent knowledge on their own.

Web-based tutorials offer a viable alternative to workshops for these basic skills. The tutorials can be home-grown, drawing on collaborative resources such as <http://www.docshare.org/>, a Web-based service to facilitate sharing of educational materials among institutions of higher education. A proliferating variety of commercial Web-based tutorials such as those produced by NetG, SmartForce, and Element K offer a high level of sophistication, featuring interactive quizzes and simulations. The products also typically include tools to let administrators track the students' progress — a handy feature if online courses are a prerequisite to high-level workshops. Computer-based and Web-based tutorials are expensive, but can form a valuable part of a technology education program.

Of course, foundational knowledge of computer concepts isn't enough by itself to equip people to be effective and self-sufficient users of technology. They must also understand something about themselves, particularly how they learn and what techniques can help them solve problems and learn on their own in a computing environment.

Learning Styles

Research on learning suggests that everyone uses multiple modalities of learning; in other words, there is no single "normal" way to learn. Learning modes include learning by reading or other visual methods, by listening or

Required Knowledge of the Windows Operating System

Screen objects

- Icons representing computer resources — storage devices, files, folders, applications, and network
- Windows for documents and applications

File management

- Files and file-naming conventions: the file extension and its importance in Windows; copying, moving, deleting, and renaming files
- Folders (directories): containers for files; copying, moving, deleting, and renaming folders; folders can contain other folders; folders as a method of organizing data

Windows conventions within applications

- The Help utility and how to use it; accessing functions within software — menus, toolbars, right-clicking, keyboard commands

Housekeeping: how to install software, check available memory and disk space, run utilities, set system clock and screen resolution, choose a screensaver

Required Knowledge of Application Software

The user should understand the categories of software for performing common work-related tasks. While it's necessary for all users to master the operating system concepts described in the other sidebars, each user must determine which application software she should learn in order to do her job effectively. All users should master e-mail and browsers.

- Word processors: tools for writing, manipulating text
- Spreadsheets: tools to model simple processes or financial tables. Includes specialized forms of spreadsheets such as tax programs.
- Databases: tools to organize and access useful information
- Presentation tools for presenting information in a compelling and organized way, incorporating a variety of media, to varied audiences.

Table 1

What Users Learn in Entry-Level Workshops

Sequence/Name	Description
Pre-1: Develop a learning plan	- Determine my learning style
	- Decide what I need to learn
	- Make a learning plan, choose from a menu of learning choices, set a timeline
	- Obtain administrative and departmental support for this
1. Computer fundamentals	Learn the basics of networks, hardware, software (see the first sidebar)
2. Strategies for effective computing using Windows	Build on a basic proficiency in Windows (see the second sidebar)
3. File management	- Find out where my stuff is
	- Determine what to do with my stuff
	- Decide how to organize my stuff
	- Learn how to find my stuff
4. Application workshops on major supported software. Example: Using MS Word documents and shared directories on a LAN for peer review of writing	Expand my skills beyond a basic proficiency in the application. Focus on the application's overall capabilities and explore problem-solving strategies using the application. Learn to apply the tool to real-life problems and tasks. Participate in collaborative work.

engaging in conversation, or by doing (a kinesthetic approach). Most people use all of these modes of learning, but favor one over the others.

Self-knowledge in this area is valuable. For example, an understanding of personal learning style can help a person choose a learning method. In addition, it can help a learner derive the most from a learning experience. For instance, a person who learns best by reading could request written outlines and seek other written materials before attending a workshop.

Colleges and universities should provide a blend of instructional materials to accommodate various learning styles, such as

- Instructor-led workshops (see Table 1), which support kinesthetic learning and learning by listening.
- Peer-to-peer informal learning experiences (see the sidebar "Peer Support"), which support learning by listening and engaging in conversation.
- Self-paced materials, which support learning by reading. Examples include workbooks and other paper reference materials as well as computer- and Web-based instructional materials.

Learners can help discern their learning styles with a bit of self-reflection. Formal self-assessment instruments also are available. Interactive assessment forms can be published on the Web, providing the learners with instant feedback on their learning styles. In some cases, institutions may find it worthwhile to schedule workshops to administer the assessment and invite discussion about learning styles. Such a discussion could include creation of a personal development plan, in which each person identifies technology skills important to professional development in his or her job and creates a timeline for learning those skills.

Helping people understand their learning styles is an important step toward making them take responsibility for their own learning. It also helps them in moving from a training to an equipping approach.

Learning Techniques

The next step in equipping faculty members to learn independently is to provide strategies for learning to learn. This step fits closely with building the knowledge foundation, described earlier. The ideal learner will apply previous knowledge in creative ways to adapt to and learn new technologies as needed.

The teaching of heuristics — flexible rules of thumb that require adjustment based on circumstances — can prove key to self-sufficient learning.[4] Heuristics contrast with a cookbook approach that requires use of rigid formulas. To take the analogy a step further, the best cooks don't simply memorize recipes; rather, they understand the principles that underlie good recipes, such as the use of complementary spices or the effect of moisture content when baking.

A workshop on the Windows operating system (see the learning objectives in the second sidebar) provides an excellent opportunity to impart heuristics. Users armed with certain techniques can increase their effectiveness in using a wide variety of software applications in the Windows environment:

1. Accessing functions: menus, button bars, keyboard shortcuts, right-click with mouse
2. Manipulating objects with the mouse: drag-and-drop, resizing, click, double-click, right-click, right-click and drag
3. Cutting and pasting: with keyboard or mouse

4. Selecting multiple objects: contiguous and noncontiguous
5. Using the Help function: when and how

Such techniques help leverage the knowledge gained in the concepts workshops. For example, the right-clicking heuristic can be a powerful one. In applications that fully support the Windows conventions, you can point to any object in an application window — button bar, title bar, data in the content area, even a blank region of the window — and right-clicking opens a menu that presents choices associated with the clicked object. Right-clicking offers an opportunity for spontaneous learning and a shortcut to the functions related to a particular object in a window.

Here's how a user might use the right-click technique to solve problems in a spreadsheet application:

- I want to rename a worksheet in my electronic spreadsheet program.
- The worksheet is represented by a tab at the bottom of the window.
- I know that the Windows technique to manipulate a screen object is to right-click on the object, calling up a menu of choices affecting that object. I use this technique all the time when I want to change the text font in my Windows-based word processor.
- I surmise that if I right-click on the screen object (the tab) representing the worksheet, I will see a list of choices affecting it. I attempt this procedure.
- A menu pops up; one of the choices is "rename."
- I select the rename option and type in the new worksheet name.

This example also illustrates the principle of transfer. The user is able to apply a Windows technique first used in a word processor to a new context — a spreadsheet application. Helping users make such discoveries for themselves should be a goal in face-to-face workshops.

One way to promote transfer of computing knowledge to new contexts is to model the process for students. For instance, an instructor can show that the idea of a word-processing template can also apply to spreadsheets and presentations.

Prompting also can help students transfer knowledge to new contexts. When users in a workshop face a new task, they can be given general prompts — "Can you think of something you did earlier that might be relevant?" — or more specific prompts — "How did you create a template in Word?" Gradually, users gain confidence in their ability to apply metacognitive techniques and learn on their own, and require fewer prompts.

Clearly, the workshop in question bears little resemblance to the common practice of having students march in lock-step through procedures carefully prescribed by the instructor. A workshop in which students learn to learn for themselves is informal and interactive. Questions are encouraged, but students seldom receive explicit answers. The process is one of guided exploration, with the instructor as the guide.

Additional activities and processes that can be useful in this type of workshop include

- Encouraging group work, in which participants brainstorm with members of their team about how to accomplish certain procedures with software. This technique also models a process that can be very effective for spontaneous learning on the job — talking with a colleague.
- Identifying problems by type.
- Using analogies.
- Linking the problem to previous learning experiences.

Emphasizing equipping strategies affects the kinds of workshops offered and the teaching style in those workshops. At Washington and Lee University (see Table 1 and the sidebar "Peer Support"), we've eliminated many entry-level workshops in office productivity applications. In part, this decision is based on our assumption that the foundational workshops — computing and network basics, and desktop computing and office applications — will help users achieve basic proficiency on their own. In effect, we're encouraging users to follow through on what we're teaching by showing them how to learn for themselves, then expecting them to do it. Workshop offerings now

Peer Support

Informal groups of peers can support each other in their use of technology by meeting periodically for demonstrations and discussion. Examples of peer support groups include Access database users and faculty users of the Web in teaching. We use the following guidelines at Washington and Lee University:

- Brief: sessions should last no more than one hour.
- Informal: provide snacks and drinks.
- Focused: the session should have a clear, task-oriented focus; for example, using GroupWise to view and manage multiple calendars within a group or department.
- Applicable: the session should present use of a technology that benefits a wide range of users.
- Demonstration and discussion: A staff or faculty member (*not* someone from IT operations) demonstrates a specific use of a technology to solve a work-related problem or improve a process, taking no more than 30 minutes and followed by a question-and-answer period.
- IT staff included: Although the session is led by the staff or faculty member doing the demonstration, IT staff should be present to answer technical questions.
- Handout: A handout should be provided summarizing the steps involved in the particular technology implementation.

focus on specific work needs, such as how to use shared documents on a LAN to facilitate peer review of writing.

Challenges

Technology education using equipping strategies can be more challenging than conventional training. Instructors who lead equipping sessions often have to develop original materials and outlines. Many of the commercial workbooks and other materials for instructor-led workshops use the behavioral training approach described at the beginning of this article and aren't useful for workshops in which the goal is for users to learn on their own.

In addition, teaching a workshop that takes an equipping approach requires creativity and self-control. It's easier to give explicit instructions than to get learners to think for themselves. Many technologists in the university setting have a help-desk mentality that equates success with solving as many problems as possible for users. Tremendous self-control is required to answer a question with another question — and put the user on the path to self-sufficiency.

Users may actively resist an instructor who insists that they take responsibility for learning. They may complain that the instructor is shirking his duties. The technology instructor or coordinator can head off such concerns by laying out a technology education vision for administrators, department heads, and other key leaders on campus. Administrators will support the vision when they understand that the equipping approach

- can save money because fewer workshops are required;
- will result, in the long run, in more satisfied and self-sufficient users;
- can help ensure the best and most efficient use of computing resources by the user community; and
- will help create a true learning community, in which users know what they need to learn, have a plan for learning, and are empowered to learn on their own.

Furthermore, it seems reasonable that employees who take the time to understand their learning styles and to map out a plan for learning will be more proactive and effective in all areas of their work, not just technology.

Endnotes

1. R. S. Nickerson, "The Teaching of Thinking and Problem Solving," *Thinking and Problem Solving*, R. J. Sternberg, ed. (San Diego: Academic Press, 1994), 409–441.
2. R. M. Gagne and L. J. Briggs, *Principles of Instructional Design* (New York: Holt, 1979).
3. Committee on Information Technology Literacy, "Being Fluent with Information Technology," National Academy of Sciences, 1999, <http://stils .nap.edu/html/beingfluent/>, accessed March 2001.
4. M. Feasel, "Can You Teach Your People to Think Smarter?" *Across the Board,* March 1996, 16–27.

Jeff Overholtzer (joverholtzer@wlu.edu) is a technical communications specialist at Washington and Lee University in Lexington, Virginia.