Just Right: Rethinking the How and Why of Technology Instruction

Instruction should go well beyond a skill focus to one that connects technology use with the actual aims of curriculum and learning outcomes

by Mary Burns

[Mary Burns is a senior technology specialist at the Center for Online Professional Education at Education Development Center. Prior to her work at EDC, she worked as a professional development provider at Southwest Educational Development Laboratory in Austin, TX; in the Department of Learning Technologies at the Instituto Tecnologico de Estudios Superiores de Monterrey in Mexico City, México; as a consultant for international educational agencies; and as a middle school and high school teacher.]

Technology training for students and teachers in K-12 settings is often akin to the difficulties encountered by Goldilocks as she attempted to make herself comfortable in the Three Bears' home. Some types of technology training are "too long" and others "too short." The challenge is how to provide teachers with technology professional development that is as "just right" as possible.

The problem, from the point of view of one who has spent a good number of years observing, conducting and participating in technology training, is not just the *how* but the *why* of technology professional development, and here the problem is threefold. First, too much time is spent on technology instruction for teachers (and students) in software that is fairly technically straightforward and conceptually simple—software such as *Word, PowerPoint, Inspiration, Publisher*, and various other "show and tell" applications.[1] Second, and consequently, too little time is given to applications that are conceptually difficult (and are therefore scarce in most K-12 settings), such as spreadsheets, databases, Geographical Information Systems (GIS), and statistical analysis software, to name a few. Finally, technology professional development still focuses on skills training, which does little to help teachers make the important connection of the software to his/her classroom curriculum.

Each critique is examined in detail.

1. Too much: Instruction in "show and tell" software

Technology in-service days for teachers are often the equivalent of sitting through a Wagner opera—five hours of highly regimented singing when two hours will do nicely. Much of this instruction is in what I call "show and tell," intuitive software that is often Wizard-driven and conceptually simple —such as *PowerPoint*, *Word*, *Inspiration* and *Publisher*. As a brief illustration, in the late 1990s, I worked in a school district in which teachers, in order to become "certified" in technology skills, had to complete two three-hour sessions in *PowerPoint*. In preparation for this article, I recently checked the district web site only to discover that *eight years later* the two-three hour sessions are still required for *PowerPoint* certification, and that

[1] Burns, M. (2003). *Beyond Show and Tell: Using Spreadsheets as Problem-Solving Tools*. In Learning and Leading with Technology. Available at: http://www.iste.org/L&L

PowerPoint certification is required for all teachers who possess a classroom computer.

This district is by no means an anomaly nor is this focus on technology training confined to US schools. Two years ago, as part of a required orientation for new faculty in a Mexico City university, I was forced to sit through 2 eight-hour days devoted to learning *BlackBoard* (until I devised an escape plan), though it became clear to me after a couple of hours that I had enough information about this course management system to actually conduct my class. Neither *PowerPoint* nor *BlackBoard* is a proverbial "rocket science" application (nor is my grey matter equivalent to that of a rocket scientist) and in both cases, the essentials could be grasped in a lot less time than 6 and 16 hours, respectively. Yet this *sitzprobe* model of technology training prevails, not just in the US (where, granted, it should inevitably diminish as new teachers enter teaching with good "technological literacy" skills), but internationally as well.

PowerPoint is quite intuitive. Its slide structure simplifies user input by constraining choice and guiding the user through the software via templates and Wizards. I've seen teachers become "advanced" *PowerPoint* users (using transitions, animations, buttons, and linking to other applications) within an hour. *Publisher* too is template and Wizard driven. *Inspiration* consists of dragging, dropping, typing, and drawing links and can be mastered in less than an hour. Basic proficiency in Word is dependent upon two factors: knowing how to type and having a message to communicate.

Unlike other types of software which will be discussed in the next section, the concepts behind the software mentioned here—*PowerPoint, Publisher, Word, Inspiration* and other software commonly used in K-12 settings, such as *KidPix*, *PrintShop*, etc.—are simple. Such applications are about communicating a message in a variety of formats. Only *Inspiration* differs in that it focuses on helping users visually organize ideas.

Yet in spite of both the conceptual simplicity and mechanical ease of such applications, many school districts require teachers to spend hours "learning" such applications. Indeed, the high level of classroom use of these applications (relative to other types of software) might arguably prove the rightness of such an approach. Nonetheless, the approach is objectionable on two grounds. First, it is inefficient in terms of teacher time. Why must teachers devote six hours to something they can "master" in two? Second, many districts teach such applications precisely because they are simple to learn, oftentimes at the expense of other more complex and rigorous types of software. Thus classroom technology use stalls at *PowerPoint, Publisher* and *Word*. Such complacency ill serves students in that many are not using the types of software that really do promote higher order thinking.

2. Too little: Cognitively complex software

Because time and funding are finite resources, a focus on teaching one type of applications ("show and tell") lessens the amount of time for other types of applications. Complex software such as spreadsheets, databases, simulation software, statistical programs, or "mind tools," (so called[2] because of their ability to

[2] Jonassen, D.H. and Carr, C.S. (1997). *Mindtools: Affording Multiple Knowledge Representations for Learning*.

promote higher order thinking) are most obvious in US classrooms by their absence. In tracking software use by 300 teachers with whom I worked over a four-year period, only about 12 percent reported spreadsheet use (mostly among math teachers and for purposes of creating graphs). When math teachers were removed from the equation, spreadsheet use fell to 2 percent. In eight years of classroom-based work with teachers, I have never witnessed database, GIS, simulation or statistical software use. And in many districts in which I have worked, professional development in such software applications is not even offered.

Spreadsheets, simply put, are conceptually "hard." They demand both abstract and concrete reasoning by the learner, modeling the mathematical logic implied by calculations. [3] Databases, like spreadsheets, are more naturally inclined toward the cultivation of higher order thinking skills. By its very taxonomical nature, relational database design can help users to systematically organize, arrange, and classify data into groups or categories according to established criteria. [4] Database design can help users to think relationally, in a detailed fashion, and in an inductive (in aggregating data) and deductive (in disaggregating information) manner. Yet, the conceptual and technical difficulty of databases renders them invisible in terms of classroom use. School districts often lack technology trainers who are proficient not just in the *mechanics* of databases, spreadsheets, GIS and statistical software, but in the classification, quantitative, spatial analysis and statistical skills demanded, respectively, by each of these software applications.

3. The wrong focus: skills training

School districts, in large measure, still focus on skills training for teacher as a means of promoting classroom technology integration. Yet this approach has severe limitations. To illustrate, I participated in a long term professional development project which focused on technology integration and differentiated instruction. Most of the teachers with whom I worked had undergone some form of district technology training. In some districts this training was quite comprehensive and cumulative, with six hours of instruction in word processing, electronic presentation, Internet and e-mail applications—approximately 25 hours of skills training in various software applications.

Yet this rather substantive training did not translate into classroom use, by the teacher, or particularly, by the student. Arguably, 25 hours of technology training should have conferred some degree of "proficiency," translating into actual classroom use. Yet, despite such training, surveys, classroom observations, and interviews with these teachers revealed a surprising lack of non-use and low levels of teacher-reported proficiency with the applications in question. When asked to survey their individual software application skills, 60 percent reported "no" or "low" levels of proficiency with the applications in which they had received instruction. More critically, only 17 percent of teachers reported regular use of technology in their classes, and of that 72 percent was with remediation or skills-type software or word processing software.[5]

---

[3] Ibid.

[4] Adams, S., & Burns, M. (1999). *Connecting Student Learning and Technology*, 32. Austin, TX: Southwest Educational Development Laboratory.

[5] Dimock, K.V.; Burns, M.; Heath, M. (2001). *Applying Technology to Restructuring Learning: How Teachers Use Computers in Technology Assisted Constructivist Learning Environments*.

I have seen this pattern repeated across schools in which I have worked, both in the US and abroad, highlighting several fundamental weaknesses in the prevalent model of computer skills training. First, in spite of the inevitable focus on the creation of an academic product (using *PowerPoint* to present a lesson, for example), in such a skills-training model, technology is the focus and curriculum an adjunct. Such training, according to teachers with whom I have spoken, casts technology and curriculum as separate entities in teachers' minds and makes technology manipulation appear more important than it actually is.

Second, the "one size fits all" approach of a good deal of skills training often means that instructors spent time focusing on the "bells and whistles" of the application or on features-driven instruction (i.e., teaching "about" the technology). Thus, the software, according these same teachers, is inadvertently presented as more complex than it really is. More critically, this form of instruction does not allow teachers to moderate technology use in order to match their own instructional needs.

Third, such sessions often have the unanticipated consequence of conflating proficiency with mastery. Teachers interviewed as part of our research reported a belief that they needed to be "experts," not just in it the operation of technology, but also in its instructional implications and in troubleshooting technical issues. In addition, the intensive length of such training sessions, in most cases 3 to 6 hours per application, inadvertently conveyed the belief that teachers too, if they were to use technology with students, would need to cede a similar portion of curriculum time to technology training.

Finally, the focus on protracted skills training unintentionally or intentionally places the locus of expertise in the person of the technology instructor, not teachers themselves, and in so doing mystifies technology. Because you need an "expert" to divine the mysteries of *FrontPage*, the technology itself remains a mystery, resulting in the creation of a technology "priest" class whose job it is to impart knowledge and through whom a teacher's technology learning must be mediated. Though this description is clearly tongue in cheek, the reality is that still in many school districts, teachers cannot even get access to a computer unless they pass through the requisite suite of software courses, even in some cases where they already know how to use such software.

The Right Stuff in the Right Amount

My proposed solutions are fairly straightforward. First, schools and school districts must spend less time on instructing teachers and students in "show and tell" applications, which do little to promote higher order thinking, and more time on the "mind tool" applications that do. But a focus on database and spreadsheets does not necessarily mean a focus on skills training. Skills training, as I've seen in my own work, often does little in terms of helping teachers make the connection between technology and actual classroom use.

Rather, technology professional development in such analytic tools such as spreadsheets, databases, and GIS, demands less involvement with the actual technology and more time with the conceptual underpinnings that drive understanding of these applications. In the case of a GIS, for example, teachers and students will need grounding in spatial analysis techniques. In terms of databases, they will need to understand the importance of accurate relationships and rules of normalization and will most likely need to plan their database on paper before

plunging into *Access* or *FileMaker*. With quantitative tools such as Excel and SPSS, a high degree of numeracy; quantitative skills; and an ability to understand types of data (nominal vs. ordinal) and proper analyses techniques are absolute requisites.

Even when teachers and students turn to actual use of the technology tools at hand (spreadsheets, simulation software), instruction should go well beyond a skill focus to one that connects technology use with the actual aims of curriculum and learning outcomes. The technology professional development process must be an iterative one (not just a one-shot workshop) and allow teachers to tailor software use to meet the aims of a particular curricular unit that focuses on student learning outcomes. Only then will technology professional development progress beyond "too much" of a focus on simple tools, and "too little" concentration on more difficult tools and the concepts that undergird them, and instead toward an approach that is mechanically and conceptually "just right."

References

1. Adams, S., & Burns, M. (1999). *Connecting Student Learning and Technology*, 32. Austin, TX: Southwest Educational Development Laboratory. Available at" http://www.southcentralrtec.org

2. Associated Press. (December 30, 2003). *Does PowerPoint make us stupid?* Available at: http://www.cnn.com/2003/TECH/ptech/12/30/byrne.powerpoint.ap/

3. Burns, M. (2003). *Beyond Show and Tell: Using Spreadsheets as Problem-Solving Tools*. In Learning and Leading with Technology. Available at: http://www.iste.org/L&L

4. Burns, M.; Dimock, K.V.; Heath, M. (2001) *Applying Technology to Restructuring and Learning: How Teachers Use Computers in Technology Assisted Constructivist Learning Environments*. Austin, TX: Southwest Educational Development Laboratory. Available at http://www.southcentralrtec.org/applying_technology.html

5. Jonassen, D.H. and Carr, C.S. (1997). *Mindtools: Affording Multiple Knowledge Representations for Learning*. Available at http://www.ed.psu.edu/insys/400/ssdb.htm.

Source: Ubiquity, Volume 6, Issue 8, March 8 - March 15, 2005)

http://www.acm.org/ubiquity