

Monday, June 30, 2008

## Feature Article

### The Case against Open Source

The Case against Open Source is really a case against the "Free and Cheap" mantra that some Open Source Advocates chant when they solicit action (and wrong-headed decisions) from school district leaders. This is a case against folks (who know little about instruction and even less about the needs of teachers); who, despite their lack of knowledge, pitch this "save-money fantasy (delusional) strategy" to school district executives (who should know better than to listen, but don't). Sidebar Open Source Advocates who pitch the "Free and Cheap" strategy could even be well-meaning educators who love technology and want to see technology solutions available to students at any cost. (See the irony here?)

These folks could even be school district employees, Techies, who have used Windows®; Open Source software (the most prevalent kind); but, most of these folks are "outsiders" that&hellip; Don't know how teachers work Don't understand what teacher want or need

Don't sympathize with, empathize with, or relate to the plight of teachers

Don't comprehend the compatibility issues that are associated with running software within a school district "technology ecosystem"

Don't realize that Technology Integration is a failed concept&hellip; unless an entire program is funded to an adequate level, with additional funding contingency funding

Don't have a clue about how software costs add a minor (almost trivial) expense in the overall success of a technology program

Don't mind blaming teachers for the failure of Technology Programs to show a Return on Investment (ROI)

Forgot to do the basic research that would show exactly what Technology Tools, applied in exactly what way, result in exactly what measurable students curricular outcomes

Never pitch the one Open Source Solution that meets many (but not all) of a school district's needs, i.e., Novell®; Enterprise because Novell®; Enterprise is neither "Free nor Cheap."

Fail to recognize that Apple®; computers provide better solutions to their arguments for "Free and Cheap Open Source" than Open Source software does Classroom Toolkit documented the case against Open Source in a series of articles. These articles represent the rationale for requiring school districts to "do their homework" before dumping (dissipating, diffusing, wasting) huge amounts of money (which they claim that they don't have) on failed Technology Integration programs.

Technology Integration must be focused, solidified and shored up before Open Source software, or any other technology strategy can become effective.

This article clarifies the Classroom Toolkit position on what it would take to make an Open Source project successful.

Hint: it would take plenty of money, and would be neither "free" nor "cheap." "Straw Man" Argument: Only a

Distraction&hellip; with no Heart There actually isn't an argument against Open Source software in our schools because Commercial software, Freeware, Bannerware, Pirated software and Open Source software are only tools.

For example: While it might be patriotic to buy a hammer or screwdriver that were "Made in the USA," the origin of the tools (or the raw materials that they are made from) does not matter to the nail or the screw. Neither does the origin of the tool matter to the "home fixer-upper" that needs to build or repair a home furnishing that is broken before company arrives.

This analogy holds exactly for school districts that implement a software tools initiative. Sidebar Software tools are only tangentially related to Technology Integration. Note the meaning of the entire phrase: Integrating transparent technology use and skills into the Curriculum

Integrating the use of technology by teachers into instructional delivery and instructional management

Integrating the access to rich materials, including the Internet, into student research

Integrating thinking skills, problem solving and decision-making skills into student assignments so that students analyze, evaluate and discriminate authentic sources from the plethora of sources available in libraries and through the Internet

Integrating communication skills including speaking and writing into person-to-person and person-to-group interaction, whether that interaction is local or world-wide

However, the failure of the Technology Integration movement and the failure of an Open Source effort of marketing to teachers reflect issues that more extensive than simple "hammer and screwdriver-type home repairs." If either Technology Integration or adopting Open Source software were singular issues that assured that students and teachers received benefits, both ideologies would be further along on the trek toward making inroads into our schools&hellip; and progress would have been made.

Rather, the Technology Integration movement is more like attempting to build apartment buildings and high-rises by only

providing hammers and screwdrivers to the carpenters and electricians

Many more tools are needed, and these tools (training and professional development; Back-End programming; compatibility, scalability, interoperability) are very expensive. Programs that implement quality technology tools do not come "cheap." Teachers a.k.a. "Scapegoats" for Technology Integration School districts perform an "under-budget-and-blame-teachers" slight-of-hand ritual, every time they pretend that they are implementing Technology Integration projects. The "fall-guy teachers" are blamed for laziness (slacking off and not working hard enough), or blamed for incompetence (working in a slip-shod manner, just being there to collect a paycheck), or blamed for failing to use technology to produce measurable student outcomes (i.e., in increasing test scores).

**Sidebar** The classic "blame-game" dialog goes like this: We gave you a computer and installed a computer lab (shared by 48 classrooms), why haven't our students' test scores risen high enough to satisfy our critics. When will we recoup the major investment that we made in the computers and infrastructure? **The Role of Software in the Budget Formula** Software plays only a minor role in the Total Cost of Ownership (TCO) of a Technology Integration program. Infrastructure, equipment, technical support and refresh rates (how often equipment is replaced --should be every three years) are one; but not the most expensive "cost center."

A second cost center, training and professional development, should be at least 30% of project cost.

For example: If a project costs \$30 million USD, then training and professional development costs should run about \$10 million USD. But, school districts seldom allocate more than single-digit percentages to training and professional development; funds for stipends, trainers, follow-up support, release time, software and equipment for teachers to use at home, etc. School districts skimp and under fund in this area, and the results (as observed nation wide) is technology that is un used, under used and under utilized.

The third cost center, and perhaps the most expensive of all is the funding that must be allocated for the "Back-End" technical support, software programming and development, customization; compatibility, scalability and automation. This is the "behind the scenes" work, produced by highly-paid computer professionals (project managers, software engineers, database developers, database managers, server specialists, analysts, programmers, Web designers, Web Masters, etc.)

Without the Back-End support, technology (whether Commercial or Open Source) remains too difficult to use, lacks compatibility, lacks interoperability with other programs and lacks scalability. **Sidebar** Compatibility: Data files from one program should connect with, load, be read by other programs

Interoperability: Data that is captured in one program should be available to every other program to prevent duplicated effort (teacher labor) and to ensure that data input errors are minimized

Scalability: Technology integration requires that computers be added anywhere in a school district network. But, costs matter if you add 1 computer or 1,000 computers to a network. Every computer must function.

However, as the complexity of the network increases, costs skyrocket. Network administrators and managers know this. But, most software-only Open Source advocates don't understand IP Addressing, Super-Netting, V-LANS, DHCP Servers, or device priority; such as when IP Telephones operate on the same network as video, and data; where telephone conversations must be given a higher transmission priority than video, and video must have a higher-transmission priority than computer data. (Note: when computer data is resent, the network just appears slow. If telephone message packets have to be resent, the call is garbled.) The size of the Back-End programming, customization, support and automation components of a Technology Integration package dwarfs the size of the training and professional development costs. **How a Real Budget (i.e., Having a Chance for Positive Measurable Outcomes) Plays Out** Even if school districts skimp on the Back-End costs, these should approach 30% of the total Technology Integration program budget. For example: For the \$30 million project, \$10 million for hardware, software and infrastructure, \$10 million for training and professional development and \$10 million for Back-End programming, interoperability, automation, development and support.

But, school districts spend little or nothing on the Back-End programming, automation and support.

And, it is this lack of Back-End integration that diminishes the actual value of outcomes from any expenditure on Technology Integration, bringing the ROI down to the lowest common denominator; i.e., whatever sheer personal (and uncompensated) efforts that teachers put out; mostly on their own time. Another way of framing the issue, if teachers put in "above and beyond the call of duty type " efforts to compensate for the minimal funding of Technology Integration programs, then the Technology Integration initiative can be salvaged. **The "Zero Funding Option" for Real Technology Integration** If IT Directors (and other folks in the know) understood how teachers work, and how learning takes place, these folks would explain the full (actual) costs of a successful Technology Integration program to school district executives and meddling politicians.

Then, most enlightened school district executives would propose a "Zero Funds" Budget. In-the-know education leaders would decide to allocate "zero" expenditures for a "not ready for prime time" Technology Integration effort because they would reason that the school system could not afford to waste money by launching a "partially funded, minimal chance or return, only for show" project. Enlightened school would allocate Zero funds because they would understand what a "real chance of success" (a real Technology Program) would cost.

These enlightened leaders would also understand that IT folks failed to dialog with teachers and that It folks failed to demonstrate the direct connection of technology expenditures to measurable improvements in student achievement.

(You can only control what you can measure.) **Sidebar** If you lack adequate funds for the successful implementation of a

program; the sane, rational and competent approach is to cancel the program, rather than launch a "sham, pretend that we are addressing the issue" initiative. Technology Project Intelligence Myopia The problem is that people know what they can see, and everybody sees the computer.

If a computer system could be purchased for \$400 USD, and Open Source software were free, the cost of purchasing, installing the computer is: Computer: \$400

Installation: \$60

Three Year Warranty: \$125

Network Drop (to connect the computer to a Switch): \$130

Cost of Port on the Network Switch: \$125

Analysis: The cost for the computer connection to the server about doubles the base cost of the computer.

But, computers with "Free and Cheap" software will continue to sit idle unless teachers are trained. And, the professional development that teachers require is not software training, but specialized curriculum training, geared to the subject and age level of the students that they are teaching. Another miscue of the Technology Integration movement was to deliver software training when subject matter professional development (with the transparent use of technology) was required.

Sidebar Curriculum training using transparent technology skills is the Technology training that few school districts engage in because delivering useful training of this type is too expensive. A cadre of (extra) teachers (from each content area) would need to be assigned to assist (mentor, support) every teacher. A program like this might require 20 times the number of trainers that the district now employs. (And, trainers are looked upon as cost-centers, i.e., overhead. (Note: teachers are profit centers, because school districts are paid for the warm student bodies that sit in classrooms under the watchful eye of teachers. How to calculate how many trainers are required? Four or five grade for the elementary grade levels (Do you count Kindergarten?), trainers for each middle school content area subject, trainers for each high school subject, etc.

And, trainers must be teachers that have proven their expertise by actually demonstrating positive student improvements in learning by the use of the Technology Integration that that they are presenting. (School districts that deliver software-only training without a focus upon exactly how every teacher can apply the curriculum lessons deserve to be punished.)

A transfer to the Training Corps is the last place that mediocre and under skilled teachers should be assigned. (And, avoid promoting these folks to principal, too.)

The guideline of allocating 30% for professional development breaks down with small Technology Integration budgets. Until expenditures reach "critical mass," i.e., large amounts of money for hardware and infrastructure, training and professional development costs will actually be fixed costs. (This means a larger percentage of the Technology Integration budget must be allocated for smaller project. This also means that large numbers of real teachers will need to be hired as trainers, curriculum developers and learning specialists so that teachers can be trained by folks that actually know and have successfully implemented technology integration projects where real student achievement has been measured. And, this training cannot be a "one session- no follow-up" "one-script-fits-all" wonder. Training sessions must contain follow-up, and be tailor-made for each teacher.

This training and professional development must also include: Release time for teachers

Substitute pay (to cover the release time)

Corresponding training and professional development for campus administrators

Facilities for training students, by their teachers, or by training specialists

Parent training and support (evenings and weekends) Back-End System, Programming, Connectivity and Compatibility:

The "Sin of Omission" But by far the most expensive cost of a Technology Integration program that really delivers is the crucial (mandatory, keystone, "It-will-never-be-effective-without-this,"

"Don't-do-this-and-Technology-Integration-efforts-wither-on-the-vine") Back-End programming.

But, this part of the Technology Integration piece almost never gets done.

Why?

Answer: Because&hellip;The professionals with the technical knowledge to do this work earn more than most school district superintendents, and are worth every penny of their salary (Note: some of these professionals are billed for their services at from \$150.00 to \$300.00 USD per hour!)

There are not enough of these professionals residing in the US, and the US Government will not grant more H-1B Visas so that we can get the talented professionals we need from India, Russia and China

The ramp up time where those talented professionals learn the "Educational Process" (Educational Intelligence, Business Intelligence) so that they can integrate and automate Back-End processes is lengthy (think months and years)

The ramp-up and hiring of these talented professionals would "bid up" the salaries and make programs like this even more expensive

The only database product that can scale to the level required for this level of complexity is Oracle&#8482; (Note: SAP&#8482; also has this capacity, but SAP implantation projects take one or more years of planning and require huge start up budgets (Think tens to hundreds of millions.)

Project Managers capable of managing the required Back-End projects earn (and are a bargain at) twice or three times what an average school district superintendent earns

For school district budget managers, the costs of building out a Technology Integration program&hellip;training and

professional development, infrastructure, and Back-End support system are inordinate, excessive and budget-breakers. To fund a "Real" (i.e., guaranteed to pay off) Technology Integration program would require that all school district funds be funneled into the Technology Integration arena. Then, school district would need to hire 25% more teachers to provide the training and professional development, 400% more technical support staff members, and a large cohort of highly-paid computer professionals (programmers, database developers, project managers, Web designers, graphic artists, etc.) would need to be added to the district's staff. Sidebar Note: Just hiring this large technical staff is inadequate because these folks require time to learn the basics of "Teaching Intelligence" before they can create the Back-End infrastructure and conduits required to integrate all the systems that are connected to the school district's network. In addition, all of these folks require high-end computing equipment, development servers and office space. Training facilities would need to be built, and at least 25% more office space would be required.

Bottom line, school district will fund hardware and infrastructure projects, refresh computers at a painful rate, and blame teachers for the lack of students' progress from using the equipment.

Bottom line for teachers: Speak out and let everyone know that Technology Integration is a failed, fabled, fantasy unless major awoken and a major funding commitment "gets real." Real needs of Teachers Teachers need transparent computing. This means computer systems that are easy to use, and this means computer systems that fit with how teachers work.

These computing needs must mesh with the over-worked lifestyle and the economic realities that teachers face. These include: Teachers have spouses

Spouses work for business and industry

Teachers do not have money for more than one computer in the household

Teachers do not have the skills to develop virtual or dual boot computers, not the time to waste in learning these technical skills

Most of the planning and lesson development that teachers perform is conducted at home

Most of the technology materials, test banks, auxiliary software provided by textbook publishers is Window and Mac compatible

Therefore:

Until Open Source software is compatible with the Industry Standard operating system, the Windows operating system is the logical choice for school computer systems.

"Tip of the Iceberg" Needs for School District Technology But teachers' needs are just the "Tip of the Iceberg" when it comes to compatibility requirements for Open Source software.

The choice of an operating system must be based upon the software programs that must be used do the jobs that are required. The choice of operating systems always depends upon the applications that must be run to perform mission critical work. Then, hardware and peripherals with the required capacity to run the applications and Operating System are chosen. Only the uninformed, the folks with a moneymaking conflict-of interest and Open Source Zealots fail to follow this "industry standard" decision-making practice. Applications that School Districts must Support

The applications that a school district operating system must support include: Business Office Software (Payroll, Accounts Receivable, Accounts Payable, Purchasing, Reporting to State and Federal Agencies

Food Service Menu Planning, Free and Reduced Lunch Eligibility and Reporting, Point of Sale (POS) systems

Transportation Systems, Route Planning, Student Population Density Mapping, School Boundary Analysis and Reporting

Library Automation - Card Catalog, Book Loan and Tracking, Bar Code Generation

Grade Reporting

Reading Motivation and Student Performance Tracking

Test Score Analysis

Administrative and Counselor Access to Dashboards for all Student Achievement

Student Online Portfolio Software - With Administrative Review

Textbook Inventory, Tracking, Management and Cost Recovery Software

Test Bank Software

Master Schedule Software, and Wireless Student Schedule Queries using Handhelds

Special Education Individual Education Plan (IEP) Generation and Tracking software

The Operating System must be compatible with these and countless other applications. For example: First Grade through High School Mind Mapping

Geometry Software

Science Probeware Software

Scientific Calculators Software

Primary Grade Reading Testing and Test Scoring

Reading Motivation

Math Improvement Tracking

Remedial Reading Software

Grade Recovery Software

Computer Aided Instruction Software

The argument that the Open Source Desktop has superior specifications is mute.

And, teachers don't need a free office application program. They need seamless compatibility with mission critical applications, one-click ease of use and interoperability so that data does not have to be re-entered from one application to another.

Sidebar There is a movement to provide this interoperability. This is the Schools Interoperability Framework (SIF). Microsoft, Apple, big-name and educational publishers are involved. The only Open Source vendor to be SIF certified is Novell. Other Open Source folks don't appear to understand this school district need for interoperability, and the issue is ignored. One possibly the reason that Open Source Advocates ignore the only Open Source Product that is ready for prime time, Novell, is that Novell is neither "Free or Cheap." But, Novell has done the homework that other Open Source vendors need to do, i.e., build a Back-End system. First Step Fundamentals But, building an infrastructure, training teachers and developing a Back-End support system only makes sense if there is absolute, irrefutable evidence and solid proof of a direct connection between Technology Integration and positive, measurable student instructional outcomes.

And, these connections to instruction must be super-substantial. The instructional outcome connections to Technology Integration must show that not only can Technology Integration obtain results, but those results must be better than the results that less-costly methods produce.

These connections cannot be vague and tenuous; or open to alternate explanations.

Of course, after billions in spending on hardware and infrastructure, asking for this connection between instruction and technology is past due.

If only school folks had attended to "First-Step Fundamentals" and had discovered the direct connection between technology and student achievement; instead of spending 15 years "chirping the tune" that "Technology assists learning, but we can't prove the fact."

Sidebar It has been this failure to prove the exact role of technology in supporting student achievement that is responsible for so many "easy target Ed Tech" funding cuts. Open Source Strategies for the Real (Teaching) World To gain traction in US schools, Open Source Advocates must learn how teachers work and learn how teachers think. This requires that Open Source Advocates; Focus upon compatibility with school district mission critical software Move to full SIF Compliance

Make the software "one-click easy"

Standardize the software interface, institute a version control and upgrade system that matches the school district yearly (summer) hard-drive-imaging and upgrade cycle

Provide the conduits to full data interchange between application and conduits for back end automation Implications for Open Source The primary call to action for Open Source Advocates is to "Cease and desist" in the "misinformation" campaign that confuses "already budget-distressed and stressed" school district executives by offering them false hope of technology savings.

The current "Free and Cheap" persuasive dialog is similar to sending travelers on a cruise; then leaving them to discover that the ship didn't take on fuel, that the ship is dead in the water; and, if they don't want to die upon the deep, they better figure out a way to start rowing.

Promoting a "Free and Cheap," "easy-way-out" strategy to over stressed, under involved, uninitiated; "Only give me a summary, not the specifics, because I can't get my mind around details" mindset is unethical. Of course, the Open Source Advocates are well-meaning and sincere; but "majorly uninformed." Open Source Advocate: "The Must Do" What Open Source Advocates need to do to gain mind-share and market share in our schools is to go to schools, discover what teachers want, make sure that the technology component of the software is transparent (i.e., invisible) and make sure that each Open Source software product runs on Windows, Apple and Linux.

The amount of work that our teachers are (asked, requested, badgered) into doing is incredible. Teachers have zero frustration tolerance for tinkering with software before the software will function and perform real work. And, the skills that teachers need to perform an effective job of teaching parallel what employers say about high school and college graduates; technology skills come in near the bottom of what teachers need to be effective teachers.

Until Open Source Advocates develop Open Source software applications that are "one-click-easy" SIF Compatible, and interoperable with the industry standard Windows Operating System, these Advocates need to do their homework. Summary and Call to Action Open Source Advocates should not feel insulted or defensive when they learn that Open Source is not ready for Prime Time in our schools.

Open Source software is not ready because Open Source Developers have not gone into our schools and studied what teachers and students need. They can't be expected to know what is required in the educational arena if the executives and managers in our schools don't know, haven't figured it out, and don't ask teachers, either.

It may be difficult for Open Source Advocates to accept that their clamor for "Free and Cheap" software for our schools is detrimental.

But, what they need to be advocating is... Research to show exactly what technology results in exactly what measurable student achievement

Back-End Integration and Database Development

Adequate budgeting that includes:

30% for training and professional development

30% for Back-End programming

30% for technical support

Seamless automation

Compatibility with School system Mission Critical Software

Compatibility with Microsoft Office

The use of Open Source software for Microsoft Windows

Note: Most of the Open Source software that is in use in the world is Open Source software that runs under the Windows Operating System.

So, Open Source Zealots must put aside their disdain for Microsoft; and build a loving, caring and concern for the needs of teachers.

And the best way to do that is to stop filling school district executives with false hope that the magic of Technology Integration can be solved with one quick and easy fix, i.e., by adopting of "Free and Cheap" Open Source software.

Posted by Classroom Toolkit Newsletter in Featured Article at 08:00

Technology in schools is, in the main, a vastly expensive and misguided irrelevance.

Where people have thought through the objectives of schools IT and set measurable outcomes, the implementation could just as well be Microsoft, Open Source or goat cheese soufflé. There is no attribute of Microsoft (or goat cheese soufflé) that sets it apart from or above Open Source. There are some very unfortunate instances where IT materials have been tied to a restricted operating system, with predictably disastrous results.

But in those cases where schools IT does work, it can be a phenomenally effective (and resource-efficient) teaching and learning medium.

Anonymous on Feb 7 2010, 13:09