NSU NOVA SOUTHEASTERN UNIVERSITY Abraham S. Fischler School of Education

Innovate: Journal of Online Education

Volume 3
Issue 1 *October/November 2006*

Article 2

11-1-2006

# Open Source/Open Course Learning: Lessons for Educators from Free and Open Source Software

Robert Stephenson

# Open Source/Open Course Learning: Lessons for Educators from Free and Open Source Software

**Open Source/Open Course Learning:**
**Lessons for Educators from Free and Open Source Software**
*by Robert Stephenson*

Free/Open Source Software (FOSS) has transformed the software industry. As noted by other authors in this issue, academic information technology (IT) is already realizing many benefits by adopting open software; such benefits include reduced cost, absence of user restrictions and vendor lock-in, and consistency with traditional academic values of openness and sharing. The greatest benefit of the FOSS movement for educators, however, is not cheaper or better software but the model it provides of a social, cultural, and legal framework capable of harnessing IT to improve learning.

At this point, some may object: "Universities have been using IT for a half-century and hardly need a new model." But formal education has used IT principally to support administration and research and has been slow to adapt it to improve its core business of teaching and learning. Traditional learning is still too passive, too parochial, too hierarchical, and too artificial. By harnessing IT effectively, educators can make instruction more graphic, dynamic, and active than it is now. They can introduce students to real-world experts and real-world problems and create communities of practice that promote learning. Others may object that a huge amount of online content is already available at no charge, so open source learning is old news. But price is the least important issue in open source learning, as a review of the factors critical to the success of FOSS will make clear.

In the model I outline below, the characteristics of FOSS that have contributed to its rapid rise and success serve as the inspiration for grassroots, open-source learning communities—or more succinctly, *open course* communities—that would be capable of transforming education just as FOSS communities have transformed the software industry. To be sure, the participants and domains of FOSS communities are not the same as those within the open course communities I envision, and consequently the former provide a pattern for education to emulate rather than a precise blueprint. Nevertheless, in light of the similarities that do exist between these two worlds, I believe that the FOSS-inspired approach can revitalize educational practice.

**How and Why Open Source Succeeds**

Richard Stallman (1996), founder of the Free Software Movement, defined FOSS as follows:

"Free software" is a matter of liberty, not price. To understand the concept, you should think of "free" as in "free speech," not as in "free beer."

Free software . . . refers to four kinds of freedom, for the users of the software:

- The freedom to run the program, for any purpose . . .
- The freedom to study how the program works, and adapt it to your needs . . .
- The freedom to redistribute copies so you can help your neighbor . . .
- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits . . . (¶ 2)

The license definition does not preclude a modest charge for the software, but the freedom to redistribute

prevents monopoly rent and thereby limits the price that may be charged. With the near-zero cost of Internet distribution, the market price for downloaded material is often less than the overhead of a financial transaction and therefore effectively becomes zero.

The success of FOSS, however, is not directly due to any of the above freedoms but rather to the community that comes along with FOSS's egalitarian philosophy. FOSS projects succeed—when they do—because they are embedded in a supportive community of practice, not just because they make their source code available (Weber 2004). FOSS communities organize and govern themselves, continually evaluate and improve their products, and grow in size and influence—generally in a bottom-up fashion with little or no external funding or institutional support.

Mozilla's Firefox browser, for example, owes its stunning success to its community: the teams of (largely) volunteer programmers and testers, the volunteers who write documentation or staff the help forums, and especially those who tell friends about it and help answer their questions. This support community plays a key role in driving adoption because by reducing the difficulty of mastering how to download and use the browser, it reduces adoption cost. Commercial software companies offer support, too, but it is usually less helpful (and often more expensive) because they cannot match the sheer manpower and enthusiasm behind many open source projects. Too often commercial support consists of a low-level employee reading from a script: "First, try rebooting the computer. Next, uninstall and reinstall the software . . ." The crucial difference is that successful FOSS communities grow spontaneously and become self-sustaining and self-supporting, recruiting new users who become members of the community themselves.

**The Concept of Open Course Communities**

Like their FOSS counterparts, open course communities are defined by their freedoms:

- The freedom to use material for teaching and self-study.
- The freedom to adapt material to meet students' needs.
- The freedom to redistribute material to help others.
- The freedom to release improvements so that the entire community benefits.

The freedom to adapt and improve is even more important for learning materials than for software. Adapting is more than just a matter of internationalization and localization. Learning occurs more readily in a culturally familiar context, so a Homer Simpson example that connects with community college students in Atlanta may be ineffective or inappropriate for medical students in Kuala Lumpur. Using learning assets in a new setting often requires that they be adapted. As Robby Robson puts it, "Context is the friend of learning and the enemy of reuse" (2003, "Context," ¶ 2).

Affording these freedoms is not sufficient, however, for open content to be embraced by most teachers. Fifteen years of institutional and philanthropic support has produced many digital libraries and large stores of free and open content, yet their impact on mainstream education remains slight. Why?

The answer is that zero price does not mean zero adoption cost. Adopting a new learning asset takes time and is in itself a learning process. Most teachers have more work than time and are wary of adopting new technology without knowing how long it will take to master. Their cost (expressed in terms of hours) is: Adoption Cost = Mastery Time (difficulty, experience) + Price per Instructor/Hourly Wage.

Mastery time is an increasing function of the difficulty of the materials and a decreasing function of the instructor's experience with them. The price of the materials can be nil, but mastery time is always greater than zero so it is usually the dominant factor in the cost of adoption. This observation is as true for software as for courseware, and FOSS's most important lesson is that a strong community can reduce this cost.

**Open Course Communities: Key Ingredients of Success**

*Support and Collaboration*

A strong support community is crucial in helping mainstream faculty members adopt online learning materials. The community can help teachers find materials that are right for their courses and students, and it can help show them how the materials might be used. It can evaluate the materials, creating a feedback loop for improvement. Community members can contribute back improvements, new learning assets, lesson plans, and assessment questions so that all benefit.

Experience to date has shown that content alone—however excellent it may be—is not enough to impact formal learning significantly. Like the vacuum cleaner salesman's boast that "this machine will do most of your work by itself," it is false advertising. If educational content is to be effective for the mainstream, it needs to be supported by a strong community of teachers, teacher trainers, developers, technologists, and students. Learning is always the result of a conversation, however indirect that conversation may be. What is needed is not just open content but open course, defined as follows: Open Course = Open Content + Community.

Students benefit from this community not only indirectly but also from direct participation. Involving students in the evaluation and improvement of their class materials fosters more active and critical learning. Focusing on content alone perpetuates the idea that teaching is no more than feeding students safe, predigested bits of knowledge for their enlightenment. Although such an approach may be useful for building foundation knowledge, stopping learning at that stage creates a crippling dependency. More advanced learning is messy and open-ended, and in such contexts an open course approach is more effective pedagogically when it welcomes student participation.

An open course community is a knowledge ecosystem where teacher, courseware developer, librarian, and student roles are linked together by collaboration, sharing, and feedback. Like any ecosystem, it is characterized by circular flows of information, resources, feedback, and credit; the knowledge ecosystem develops its own, alternative economy based on exchange and reputation. Since the academy is already structured according to disciplinary boundaries, most open course communities are likely to be discipline-specific, but they will entail a much higher degree of collaboration—both horizontally across institutions and vertically between students and faculty—than is traditional in educational practice.

*Assessment and Evaluation*

Any successful complex system is the product of some sort of evolutionary improvement process. Engineering has well-developed standards of quality that software developers can use to choose between two competing programs. This is not the case in education where learning assessment is seldom standardized or tied to agreed-upon competencies and where different institutions and teachers are often fiercely loyal to their own ways of doing things. For open source learning to succeed, however, it still needs to address the issue of objective assessment. Just as many FOSS developers are moving to test-based development (Beck 2003), so should open course developers. This would require that

- each learning object have explicit learning objective(s);
- each objective have an associated battery of mastery questions, exercises, and problems;
- teachers administer these assessment materials to their students; and
- aggregated results be made available to the community.

This approach will enable the community to weed out ineffective assessment items and compare assets that share learning objectives based on the assessment results. This is, of course, no absolutely best way to teach a given subject because students are not all the same in any particular context; yet it is important to determine whether, for a particular group of students, this or that learning object is more or less effective in

teaching a given subject. As Thomas Angelo points out, "assessment techniques are of little use unless and until local academic cultures value self-examination, reflection, and continuous improvement" (Angelo 1999, "Changing Our Mental Models: Assessment as Culture Transformation," ¶ 2).

A side benefit of this approach is the development of a large bank of vetted assessment materials classified by learning objective, a valuable open educational asset in its own right. For example, the question bank would not only allow for informed decisions about what learning object to use in a given situation but would also make it possible to compare the outcomes of teaching a topic with and without using learning objects. The evaluation I advocate is open, voluntary, decentralized, and bottom-up, and thus it avoids the controversy that has plagued standardized testing.

*IT Infrastructure for Collaboration*

These beneficial network effects I have described presuppose that many participants in an open source learning community contribute substantially to the community by developing learning assets, creating mastery questions, administering tests and submitting results, answering questions, helping faculty colleagues, submitting bugs and suggestions, and in a dozen other ways. Such contributions will occur only if they are easy and become a social norm.

Facilitating the human interactions that are important for learning and the growth of the community may require new technical tools. Open course communities need a technical infrastructure that makes collaborating with distant colleagues, finding the content needed for tomorrow's class, incorporating content into a lecture or online module, creating and administering student assessment, and returning anonymous data to the community easy. For students, the infrastructure needs to make using the online materials, taking assessments, interacting with the content developers, and generally contributing to the community easy. The role played by IT here is to enable collaboration, reduce friction, and facilitate data collection. It should also, like good plumbing, stay in the background and not become an issue.

*Promoting Participation*

Active collaboration is vital to the health and growth of the open course community, but there are obstacles to member participation. Most teachers are too busy with their workload, too uncomfortable with technology, and too unaccustomed to collaborating. To overcome these obstacles, the following elements are necessary:

- Barriers to participation and contribution need to be made as low as possible (through technology, careful planning, and thorough user testing).
- Active members of the community need to reach out to colleagues and encourage them to join the community.
- Educators need clear incentives to participate. These could be institutional rewards like release time or a stipend or increased status in the eyes of their peers. The most effective incentive, however, is a convincing demonstration that by actively participating, they will teach more effectively and save time in the long run.
- Educators need to understand clearly that their contribution is needed and in what ways they may give it. There should be enough ways to contribute to accommodate every talent and taste.

In addition, the license may include a giveback clause that makes helping an obligation (Exhibit 1). Specifically, for every hour that an educator's class spends using open course content, the teacher and students collectively owe an hour's contribution to the community. The open source learning communities are not in the business of policing compliance, however, and giveback is imposed only by persuasion and reputation.

Some will choose not to contribute while others will prefer to continue teaching in a more traditional way and

eschew open course learning resources altogether. Those who do participate in the community, however, will contribute to the improvement of the open content and help recruit new colleagues. In the process, they will strengthen the community and all will benefit.

**The Status and Future of Open Course Communities**

Briefly, the main arguments made in this commentary are as follows:

- FOSS demonstrates that network effects occur at the edge of the network: in most cases bottom-up is more powerful than top-down.
- FOSS's lesson is that an active community of practice is the key to success. An open course collaboration is a knowledge ecosystem with an economy based primarily on exchange and reputation. When such a community involves all stakeholders, it not only provides the most value to its members but also grows the fastest.
- Including students in this community of practice strengthens their education.
- An open course community needs the ability to modify its resources since this is the only way they can be improved or adapted for new contexts.
- Community resources will evolve only if they include assessment as an integral component and the results of this assessment are used to drive improvement.
- Stakeholders need lots of simple, easy ways to make helpful contributions to the community so that it becomes a social norm. Ways to promote this include incentives, a reputation system, and a license that requires contribution. Technology is needed to make these contributions as frictionless as possible.

No such community exists yet, although a few, such as the grassroots Harvey Project (Stephenson 2000), come close. Some top-down projects, such as the Global Education and Learning Community and the Social Authoring Project of the National Registry of Online Courses have similar goals.

Ad hoc solutions for most of the technical issues exist today. For example, MERLOT makes it possible to find and review large amounts of online content, and the eLearning XHTML Editor helps teachers create their own online courses using open content. OpenCourse.Org and Utah State University's EduCommons support collaboration around building, adapting, and evaluating open content. Commercial tools like Question*mark* and Respondus and the Sakai Project's open source tool SAMigo build and administer standards-based tests. What are still lacking are tools that are more integrated and more transparent for mainstream educators to use. These improved tools can be built by FOSS developers within the academic community. This is exactly the sort of "scratch your own itch" solution that FOSS has mastered in its approach to software development.

As these issues are solved, open course communities will begin to grow in size and improve in effectiveness. Eventually they will transform education, no matter how modest their beginnings.

[Editor's note: This article was adapted from a presentation at the Open Education Conference at Utah State University in Logan, UT, September 2005.]

**References**

Angelo, T. 1999. Doing assessment as if learning matters most. *AAHE Bulletin* 59 (9): 3-6.

http://education.gsu.edu/ctl/outcomes/Doing%20Assessment%20As%20If%20Learning%20Matters%20Most.htm (accessed September 30, 2006).

Beck, K. 2003. *Test-driven development by example*. Boston: Addison-Wesley.

Robson, R. 2003. Reusable learning | design. http://www.reusablelearning.org/index.asp?id=30 (accessed September 30, 2006).

Stephenson, R. 2000. The Harvey Project: Open course development and rich content. In *Case studies on information technology in higher education*, ed. L. Petrides, 185-194. Hershey, PA: Idea Group Publishing Co.

Stallman, R. 1996. The free software definition. http://www.gnu.org/philosophy/free-sw.html (accessed September 30, 2006).

Weber, S. 2004. *The success of open source*. Cambridge, MA: Harvard University Press.