Propagating Software-Based Educational Innovations

Edward F. Gehringer
North Carolina State University
Depts. of Computer Science
Raleigh, NC 27695-7256
+1 919-515-2066

efg@ncsu.edu

SUMMARY

Much software has been created for teaching various aspects of IT. Most of it, however, is sparsely used. Educators frequently do without any of it, or "reinvent the wheel." This paper discusses approaches to improving the visibility and adoptability of educational software.

First, others must be aware of the software. There is no substitute for face-to-face contact. Conferences, e.g., CCSCs, SIGCSE, ITiCSE, SIGITE, and ACM regional meetings, can play a major role. Paper presentations are quite useful, but because of strictures against self-plagiarism, it is difficult to write multiple papers on the same tool. Work can also be presented in a poster or a demo, or by organizing a panel or a BoF on a related topic,. Publishers can also help, by publicizing your work on Web sites associated with their textbooks. E-mail forums, such as SIGCSE-members, are also useful. Do not ignore repositories of course material, e.g., MERLOT and CITIDEL, but active publicity is still needed to attract a critical mass of users.

Once aware of the innovation, faculty must be reminded, e.g., by e-mail, at the appropriate time in their course. If the software deals with doing homework, the most critical time is the month before the start of an academic term. Try to minimize the need for risk-taking, especially for software without a long track record: If it can be used in a short assignment (or a demo), instructors will be more inclined to "get their feet wet" and expand usage later.

The full paper will consider the experience of such projects as Web-CAT and JHavéPop, as well as several algorithm visualizations, and the author's own Conoscenza and Expertiza projects. It will include a survey of instructors who have expressed interest in, but not adopted, software innovations, and analyze how they might have been persuaded to follow through.

Categories and Subject Descriptors

K.3.2 [**Computers and Education**]: Computer and information science education, Curriculum. K.3.1 [**Computer Uses in Education**]: Collaborative learning.

General Terms

Measurement, Documentation.

Keywords

Cooperative learning, wiki, active learning

INTRODUCTION

As educators teaching about technology, we are well aware that technology can improve our teaching. Many of us are involved in developing such technology. The National Science Foundation makes about 100 awards a year in its Course, Curriculum, and Laboratory Improvement program. Virtually all of the computer-science awards involve some sort of software development, as well as many of the awards in other fields.

Yet just as publication of an article does not insure it will be read, much less cited, development of software does not insure that it will be used, much less disseminated to other institutions. For example, in 2006, The author had a student go through the proceedings from the Eclipse Technology Exchanges […]in an effort to find Eclipse plugins that could be used in an o-o design course. Of the 50-odd tools described, only two of them turned out to be useful. Many required a specialized environment in which to run. Some required extensive hand-tuning to come up with reasonable results. Some no longer worked in the current version of Eclipse. It was quite surprising that in a situation where we sought out tools to reuse, we could find so few.

Even assuming a good match between the software developer's aims and the instructor's needs, barriers remain. First, the instructor must know about the software. Second, the software must fit into the instructor's class without requiring "too much" change in style or substance. Third, the documentation needs to be sufficient for the instructor to figure out how to use it with ordinary effort. Finally, the developer needs to provide enough support to make the instructor comfortable with the software, and to fix any problems that may arise. In subsequent sections, we will consider each of these hurdles.

THE HISTORY

The author is the developer of a project called Expertiza. Initially it was a peer-review application oriented toward producing reusable learning objects by a divide-and-conquer approach [GECW 07] (each student or team chooses a small part of a large project, and competes with

other students/teams to produce the best version of that part). It has grown into a more general platform for collaborative learning, supporting file, Web-page, and wiki submissions and several different kinds of reviews.

Starting in 2005, outside instructors have been invited to use the system in their classes. They have been made aware of the system primarily by talks given by the author (15 talks since 2005 at professional conferences, 3 panel presentations and 3 poster presentations, as well as six talks on various college campuses). During this period, nearly 300 faculty have signed up to be informed of progress on the project, and approximately three dozen have indicated their intention to use the system in their classes. To date, however, only 12 other instructors have used it, and most of those have used it for only one semester. It is instructive to investigate why so much interest has led to so little use. Toward that end, 25 "intenders" among the instructors were surveyed in June 2008. Eleven responded, for a rate of 44%. In addition, four instructors who did not fill out the survey sent e-mail comments; including these yields an overall response rate of 60%. Their answers give some insight into why this project—or any project offering innovative approaches to teaching—tends to meet resistance.

BARRIERS TO ADOPTION

Instructor Reticence and Inertia

The first hurdle is instructor reticence. It takes effort to introduce an innovation into one's teaching. An instructor will be willing to undertake that effort only if (s)he expects to reap some benefit from the innovation. As one of our survey respondents put it,

> "No matter how much good documentation, there is an attitudinal issue about why any instructor would take on the additional workload of incorporating a new system into their already daunting task of teaching the number of students they have … A similar thing happens when there is an implied message that the administration would like to see "online courses" -- most faculty think that hybrid/online is here to stay but many of them see it as an unfunded mandate for them to change their mode of instruction."

The experiment might even backfire if the students are confused, or resistant to the innovation. While the Expertiza project has published papers on positive results [OOPSLA ES, ASEE wiki], these might be described as anecdotal. Thus, adopters tend to be concentrated among those instructors who already believe in collaborative learning. Most college and university faculty are quite cognizant of research in their academic field, but rather ignorant of research into teaching methodologies. Thus, the collaborative-learning believers tend to be a small group.

Student Resistance

Any software application has a "learning curve." Even if the application makes it easier to learn the course material, from the student's point of view, there are now two things to be learned: The application, and the subject that the application covers. The issue may become whether learning the application *plus* the material is harder than learning just the material, without the application. This is less of an issue for computer-science students than it is for others. In 2005, we found, via a focus group, that a user interface that posed no problems for the author's computer-science students was seen as a major hurdle by students in a zoology class. These problems might diminish as our classes become increasingly populated by digital natives; however, students' expectations of UI functionality are increasing.

Student resistance can lower morale and thus get in the way of learning. It can also hurt in another way. As one of our respondents points out,

> "'New' approaches are new for the student as well as the instructor, and unless the instructor really pushes the system really hard, there's going to be student resistance which might show up in final reviews."

Ours is not the only electronic peer-review system to encounter student renitance. Keeney-Kennicut et al. [2008] report the same phenomenon using Calibrated Peer review [..], with a majority of students in the first semester saying that the system should be discontinued. But as they gained experience with the system, student perceptions improved.

Institutional Considerations

Instructors frequently feel constrained by the perceived wishes of their institution. This may include adopting a particular learning-management system (LMS), such as Blackboard or Moodle. If so, they may favor resources that are integrated with the LMS and disdain those that do not fit into the mold. As one instructor put it,

> "The number 1 reason why I didn't use the system is that I teach older adult students and it seemed to be a burden for them to learn Vista/Blackboard along with a separate technology."

At smaller colleges and high schools, anything that requires more technology may be a stumbling block. One of our instructors in 2007 was at a small college whose network capacity was inadequate to handle demand. Only the students who had off-campus ISPs were able to use the system successfully. Since most of his students were residential, he had to drop out of the program.

Suitability for Course and Teaching Style

Any software application presupposes a particular way of covering the material. Some instructors simply cover the material in a different way. Our Expertiza application uses peer review. Instructors whose homework consists of problem sets cannot effectively use it. Another instructor told us that he runs design competitions for his final project, and students do not want to submit their ideas for peer review, since other students would then be able to "steal" them.

One might, of course argue, as the accrediting agencies have, that assigning only problem sets is not an effective way to teach most subjects. Regardless, the fact remains that type of work assigned is an additional reason for instructors not to adopt a tool.

Usability and Documentation

As long as the developer uses a system solely in his/her own classes, only student documentation is needed. But in many systems, the instructor needs to perform many tasks the students don't. Early development tends to concentrate on the student UI and student documentation. The instructor knows the system intimately, having designed it, so little effort is expended on the instructor UI and documentation. This makes it very difficult for other instructors to get acclimated. Moreover, the instructor UI and documentation are much larger tasks than the student interface. Until recently, we encouraged instructors to let *us* enter the assignments. But this diminished the other instructors' sense of ownership and ability to make updates quickly. One told us that he dropped the system because he had to ask us to perform trivial operations, such as adding a new user.

Support

Software support is a concern beginning on Day 1, and lasting as long as the system is in use. In the early going, most questions relate to bugs, which must be fixed quickly. As the system grows more complex, questions concern functionality, or misperceptions of functionality.

As a system grows larger, it is difficult to pay for support on a research grant, and it is probably not something for which college credit can routinely be given. Therefore, documentation and help facilities become much more important. Videos are rapidly growing in popularity as a documentation medium. They can be recorded in commercial applications such as Camtasia, or with open-source screen recorders such as Camstudio (http://camstudio.org).

WHAT'S MOST IMPORTANT?

In our survey, we posed ten reasons to our adopters on why they did not try, or did not continue with, Expertiza:

- I had trouble learning to use the system.
- I would have felt more comfortable if I had seen more documentation on what the system could do.
- I tried to use the system, but had trouble with the user interface.
- I would have felt more comfortable if I had seen more documentation on how to use the system.
- I was unable to get enough help from the Expertiza staff.
- I was concerned that my class might have trouble using the system.
- I had trouble with bugs in the software.
- I was requiring my students to use other software applications, and did not want to introduce too much technology in one semester.
- I had trouble thinking of a suitable assignment to use with the system.
- I intended to use the system, but just ran out of time.

Table 1. Results from Instructor Survey

	Strongly agree	Agree	Neutral	Disagree	Strongly disagree	Mean (5=SA, 1=SD)
I had trouble learning to use the system.	1	1	5	2	1	2.9
I would have felt more comfortable if I had seen more documentation on what the system could do.	1	5	0	2	2	3.1
I tried to use the system, but had trouble with the user interface.	1	3	3	0	2	3.1
I would have felt more comfortable if I had seen more documentation on how to use the system.	5	4	1	0	0	4.4
I was unable to get enough help from the Expertiza staff.	0	0	2	4	3	1.9
I was concerned that my class might have trouble using the system.	4	2	1	0	1	4.0
I had trouble with bugs in the software.	0	1	6	0	0	3.1
I was requiring my students to use other software applications, and did not want to introduce too much technology in one semester.	1	3	1	3	2	2.8
I had trouble thinking of a suitable assignment to use with the system.	1	1	2	2	5	2.2
I intended to use the system, but just ran out of time.	0	5	1	2	0	3.7

Responses were obtained using a 5-value Likert scale (strongly agree, agree, neutral, disagree, strongly disagree). Respondents had an opportunity to add a prose comment on any question.

Easily the most important reason (see Table 1) was the need for more documentation, with all but one instructor agreeing, and half the instructors strongly agreeing. Next was doubts about student reaction, especially among the non-CS/IT instructors. The only other important reason was "just running out of time."

At the other end of the scale, the instructors said they had no difficulty getting help from our staff, or thinking up assignments for which peer review could be used to build resources useful to others. The latter was a surprise, since it is often not easy to think of reusable learning objects that a class is capable of creating; this was discussed in a 2006 paper [fie 06]. It may indicate that these instructors are thinking only of how to use peer review, not using it to create learning objects to enhance other classes' learning.

W. Keeney-Kennicutt, A. B. Gunersel, and N. Simpson (2008), "Overcoming student resistance to a teaching innovation," *International J. for the Scholarship of Teaching and Learning* 2:1 (Jan. 2008).

1997 Aaron had a Web-based db on Mac

Did students who see videos do better than those who didn't?

He met Larry Martin who had done it earlier w/flat files

2003 spun off

Ed dept grant $1M to expand from physics to other til ~ 2001 John talked to tech xfer for a long x

Hard to pay salespeople at U

a lot of people paid for the sem they left

bought computrs theuy were using'

226K students for spr 1241 schools 3409 teachers

dgjertsen@webassign.net

About 200 insts. using now; started 4 or 5 yrs. ago

kwolanin@dyknow.com