

A System for Reviewing Software in Scientific Journals

Kevin Ahern

Contributing Editor, *Science*
Department of Biochemistry/Biophysics
Oregon State University
Corvallis, Oregon

In 1987, when I began serving as editor of *Biotechnology Software* (later renamed *Biotechnology Software & Internet Journal*), I faced several hurdles. First, there was little purely scientific software. Second, I had no staff to support the publication. Third, I had no budget to hire staff. Little did I know what was ahead. The first year was, to understate it, a challenge. I ended up being editor of the journal for 12 years, expanded readership 60-fold, and learned a considerable amount about the mechanics of reviewing scientific software.

Today, scientific software can legitimately be called an industry. A comprehensive directory of scientific software published in 1996 (1) identified over 2000 products in all the major disciplines. Numerous scientific publications, including *Science*, now review software as part of their regular coverage. In this article, I will describe how to set up a system for performing software reviews.

Deciding to Review Software

As the computer has come to occupy an increasingly prominent part of our professional and personal lives, so has media coverage of computer-related products increased. When should a publication review software? Most editors probably know the answer to that question. If a journal's "beat" includes coverage of relevant tools for researchers, reviews should follow whenever specific software tools are available to solve problems in the field. The first 2 actions after a decision to add software reviews to a publication are locating reviewers and securing software. It is not a chicken-and-egg problem. The software comes first because it provides bait for locating reviewers. Resist the temptation to review general software products, like word processors, because these reviews will be of little use to scientific readers, unless there is a slant on using

such products in ways that are peculiar to a discipline. Generally, only software specific for the field(s) covered by the journal should be reviewed.

Editors who are not software-savvy should consult with someone in the field who is to identify relevant products. Obtain appropriate addresses and fax numbers and write short, simple letters requesting a copy of the software for review. The letter should express interest in reviewing the product for publication and include the editor's telephone number. Software distributors such as SciTech, which handle hundreds of products from nearly as many manufacturers, don't see software reviews as money-makers and will ignore requests. Instead, direct letters to the manufacturers or developers of the products. Faxing works much better than e-mail for most requests,

The software comes first because it provides bait for locating reviewers.

particularly if the company is not familiar with the journal.

If journal circulation is small, be coy about size. I didn't give out numbers until my journal's circulation exceeded 5000 subscribers. Emphasize to developers how the journal readers are a great target audience. Doing this, I had little difficulty in obtaining free software even when my publication had fewer than 200 subscribers. Don't guarantee that a review will appear in a publication, and don't agree to an exact publication date. Most companies will send free software for review despite those restrictions.

Be sure that the software is fully functional. It should not be time-limited or disabled in any way. The manufacturer will sometimes balk at the request, but if you persist it will usually come through. Manufacturers may press to start a review with beta-test (prerelease) software so as to have the software review appear at about the time the final version is released. It is all right for a reviewer to start work on a beta-

test product, but be sure to make clear that the reviewer must obtain the official version of the released software before the article can be published. That will ensure that the reviewer is reporting on exactly what the reader will use. Remember also that fully functional software is the primary inducement for reviewers to write for a publication. Most of them won't be happy to do reviews simply to have an opportunity to decide what they'd like to buy or to see a beta-test version, because they can usually download such free demos from the Web. If you don't have someone in mind to start a review, allow at least a month to find reviewers. I keep a continuing search in place for reviewers, and this helps maintain a ready supply. The willingness of reviewers and companies to cooperate is, not surprisingly, directly related to the prestige and circulation size of the publication.

Reviewers can be identified in 4 ways: by contacts with colleagues, by placing small advertisements in the printed version of the journal, by placing a message on the journal Web page, and by posting notes on newsgroups relevant to the field. The last option is particularly useful for locating reviewers quickly. Consult with a knowledgeable colleague for help, if necessary. If you are careful in dealing with reviewers, there will be few problems. Most academic reviewers will be associated with a department that has a Web page. Check them out on line, if you can, before sending software. After you've worked with a reviewer, it is more convenient and less expensive to have manufacturers send software directly to him or her. I've lost only one software product to an unscrupulous person in over 12 years, but on 2 occasions I had to threaten to sue reviewers to fulfill their promises to complete articles. In both problem cases I received excellent reviews via overnight express delivery within days.

My experiences with new reviewers have been quite positive: Only about 1 in 5 has not worked out. I've had complete success working with reviewers recommended by colleagues who were already writing for me.

Reviewers needn't be software experts. The best reviewers are computer-knowledgeable and thorough, but they are usually inexperienced with the particular package they are reviewing. That is an important point. A reviewer who is not very familiar with a product will use it as a new user. During the discovery phase, such a reviewer will discover and report "potholes" in a product that experienced users will often drive around without noticing.

I generally ask to see people's CVs, with special emphasis on computer expertise and the systems they use, before seriously considering them. The best reviewers use both Mac and Windows systems, although there are exceptions. I generally expect reviewers to have a PhD, but on occasion I use others whom I know and trust to do a thorough job. I do not, as a rule, ask to see writing samples before giving them software to review. It is much easier to edit poor writing than it is to find thorough reviewers of products. If there is no budget to pay reviewers, be sure to inform them of this up front. Reviewers should be told that the software they receive is not for sale and that they should not attempt to make money by selling it. Tax issues can arise from the selling of such software.

The primary problems with software reviews involve meeting deadlines (surprise!). Before sending software to a reviewer, agree on a definite deadline for article submission. Keep new reviewers on short, tight deadlines. Communicate with reviewers before the deadline to avoid surprises. Anyone who jeopardizes deadlines should be dropped. Allow at least a month for completion of reviews. It is a good idea to ask manufacturers of products to examine reviews for factual errors before publication. Be explicit that only factual errors will be changed, and most will cooperate to avoid later corrections.

One editorial concern arises with this system when the software that a reviewer works with is his or her sole compensation. Many reviewers will not want to review products that they do not think will be useful

to them. Worse, if a reviewer gets a product and then doesn't like it, you may have difficulty extracting the review. Without effort on the part of the editor to bring reviews of weaker products forward, poor software will be underreported. The simplest way to deal with the problem is the most expensive one: Pay reviewers for their work and make reviewing software their job. If that is not an option, do some horse-trading. Make availability of desirable products conditional on reviews of less-desirable products.

Objectivity

Scientists use instruments to measure quantities precisely and impartially. Editors often want the same in product evaluations. That is a worthy goal, but it is important to be realistic. Many parts of a software review do not lend themselves to quantitation, despite widespread efforts to assign numerical ratings to software products. If you don't evaluate books on a scale of 1 to 10, don't evaluate software by this scale either.

Some aspects are easy to be objective about. For example, consider including a feature list, particularly when performing multiproduct reviews. This works because scientific software products distinguish themselves more by the variety of features they contain than by how well each one works. Evaluating how well each feature works is, of course, a subjective part of a review. With the exception of a function giving incorrect results (extraordinarily rare), reviewers base ratings on how well they liked it. Do opinions vary among reviewers, readers, and developers? Yes, yes, and yes. Most Macintosh users will cringe at software design problems that many Windows users will not even notice. Users of one platform often are not familiar with conventions on the other. Good reviewers report on interface problems only as they are related to the particular platform for which the software was designed.

It is tempting to rate products objectively by comparing their speeds. That is not as straightforward as it may seem. There are dozens of considerations in obtaining

impartial, comparative speed data for software of which a casual user or reviewer may not be aware. Major review magazines use expensive computer laboratories established solely for this purpose to reduce problems. As a precaution, be sure to have each reviewer describe as precisely as possible the computing environment under which the review was performed, and ask a second party to verify the results in the same environment.

When publishing speed ratings, consider this: The tolerance of most users for waiting for a computer operation is about 10 to 15 seconds. Operations that take longer lead to annoyance. Operations that take less time are tolerated. Speed comparisons should therefore focus on functions that take 10 seconds or more, because differences in speed of operation in this range will be most noticed by users. Also, with respect to speed, what is fast today is slow tomorrow. If a reviewer uses a state-of-the-art computer at the beginning of a review, the speeds reported will probably be less favorable than state-of-the-art speeds at the time of publication. For that reason, some publishers have opted to put software reviews on line to reduce the time lag before publication. Others have eliminated speed reports altogether.

In summary, good software enables editors to snag good reviewers and jump-start a program of software reviews in a publication. Setting up a system for reviewing software is time-consuming at first, but it pays many dividends. The review system can become self-supporting, with both software developers and reviewers approaching the journal about getting into print.

Reference

1. Ahern K, editor. The biotechnology software directory—a buyer's guide. Larchmont (NY): Mary Ann Liebert; 1996.