After Amateur Hour

Author: Beardsley, Timothy M.

Source: BioScience, 62(5) : 447

Published By: American Institute of Biological Sciences

URL: https://doi.org/10.1525/bio.2012.62.5.1

BioScience®
A Forum for Integrating the Life Sciences
American Institute of Biological Sciences

After Amateur Hour

Researchers and resource managers must make ever more use of computer programs to handle data, modeling, and analysis. Nowhere is this truer than in ecosystem-based management (EBM), an approach advocated by many *BioScience* authors working in conservation, especially in coastal zones. Commercial software developers have not rushed to develop the tools needed, so it is fortunate that hundreds of graduate students and researchers funded to do other work have donated their time to "skunkworks" efforts to write suitable code and have then distributed it for free.

Not so much, say Corrie Curtice and her colleagues at Duke University's Marine Geospatial Ecology Laboratory and at NatureServe. They describe their investigation of EBM code development in their article that begins on p. 508. They heard, over the course of five years' work with users of EBM software, complaints that the tools "were often difficult to use, lacked documentation, contained numerous bugs, and were poorly supported and maintained." Interviews with tool developers (analyzed, incidentally, with the help of commercial software) confirmed the suspicion that they were motivated by the desire to see their work widely used rather than the desire to build revenues. The developers lacked the appetite to administer and promote a business (or if they were government employees, could not).

Accordingly, they mostly rejected charging substantial license fees. Some said they were conscious that their funder had not specifically authorized or requested the development of a software tool. The resulting lack of steady revenues deterred them from hiring experienced professional software developers or from researching the market. Many had not properly documented their product.

In consequence, promising concepts often failed when supporting them became too big a job. Some became obsolete after their originators moved on to other projects—a familiar story that explains why many research data are effectively lost to science.

Yet even in fields, such as EBM, that lack a large market, it is possible for a project to thrive, Curtice and her coauthors discovered. The success stories occurred when projects found dogged champions willing to involve influential advocates in boosting awareness of and growing the product. In particular, the champions searched for stable funding.

Curtice and her colleagues have important recommendations for both funders and developers of software—and not just EBM software. Those recommendations probably apply to other specialized scientific applications as well. Ensuring the widespread availability and currency of a software tool is heavier lifting than the original writer might guess: A neat idea is not enough. And promotion and maintenance of a product demand skills that he or she may have no interest in learning. Funders as well as altruistic code writers will do well to reflect on these findings—and to adapt accordingly if they want to ride a wave into the future.

TIMOTHY M. BEARDSLEY
Editor in Chief

doi:10.1525/bio.2012.62.5.1