Why Ecosystem-Based Management May Fail without Changes to Tool Development and Financing

Author(s): Corrie Curtice, Daniel C. Dunn, Jason J. Roberts, Sarah D. Carr and Patrick N. Halpin

Reviewed work(s):
Source: *BioScience*, Vol. 62, No. 5 (May 2012), pp. 508-515
Published by: University of California Press on behalf of the American Institute of Biological Sciences
Stable URL: http://www.jstor.org/stable/10.1525/bio.2012.62.5.13
Accessed: 09/05/2012 11:25

Why Ecosystem-Based Management May Fail without Changes to Tool Development and Financing

CORRIE CURTICE, DANIEL C. DUNN, JASON J. ROBERTS, SARAH D. CARR, AND PATRICK N. HALPIN

Resource managers rely on tools to enact ecosystem-based management (EBM) principles and frequently express frustration at the difficulty of use and unreliability of available tools. EBM tool developers lack the consistent, long-term funding needed to develop high-quality tools. Through interviews, we determined several reasons for this funding problem including: (a) most EBM tools are developed by academics rather than software professionals and (b) most tools are offered at no cost. These factors create a double-edged sword for managers who cannot afford high license fees or to waste time with low-quality, unmaintained products. Without a fundamental shift in tool funding and development, many potentially useful tools will remain poorly implemented and underused. Without a significant increase in the number of high-quality EBM tools, governmental mandates to implement EBM will remain unfulfilled. This problem can be addressed if both developers and funders change the ways in which they seek and grant financial support.

Keywords: ecosystem-based management, tool development, software tools, funding, financing

Coastal and ocean resource management is uniquely challenging because of the complex interconnections and dynamic natures of these ecosystems. Many past attempts to manage coastal and ocean resources have failed because they did not fully represent these ecosystem interconnections (Christensen et al. 1996, Botsford 1997, Pew Oceans Commission 2003, Rouyer 2008). In contrast to historic management schemes that sought to manage a single species, sector, activity, or concern, ecosystem-based management (EBM) is an integrative approach that considers ecosystems as a whole, including humans and human activities (Slocombe 1993, McLeod et al. 2005). EBM has therefore been widely posited as a management philosophy that is more likely to support a healthy and productive ecosystem because it better accounts for ecosystem interconnections and the ever-increasing number of anthropogenic stressors on the environment. The challenge is now how best to implement EBM principles in the marine environment (i.e., marine ecosystem-based management) (Pew Oceans Commission 2003, US Commission on Ocean Policy 2004, McLeod et al. 2005, WHCEQ 2010).

Successful implementation of EBM requires managers to perform a large array of tasks, including gathering, managing, analyzing, visualizing, and summarizing information; modeling and simulating ecosystem processes; making and monitoring complex decisions; and coordinating the work of the process's many participants and stakeholders (figure 1). In marine ecosystems, this involves activities such as conducting scientific research to increase understanding of marine ecosystem processes (McLeod et al. 2005), determining sustainable harvest levels for fish and other marine resources (Gamble and Link 2009, Garrison et al. 2010), modeling and simulating watershed processes (Wang 2001), and selecting optimal sites for conservation (e.g., Margules and Pressey 2000) or restoration (e.g., Possingham et al. 2000). Most of these activities can be accomplished faster and more effectively using specially designed software tools.

Software tools are often also necessary for incorporating the best-available science into and engaging stakeholders in EBM processes. For example, EBM tools can help collect local knowledge on resource use, such as preferred areas for fishing or diving; provide models of ecosystems or key ecosystem processes; generate scenarios illustrating the consequences of different management decisions on natural resources and the economy; help visualize the impact of development on a coastal community and coastal ecosystems; help select areas for conservation, restoration, or development that meet ecological and stakeholder criteria; and collect stakeholder feedback on management alternatives. Examples of tools with these diverse functions include polling tools that take advantage of smartphones and the Internet to include remote participants and broader audiences in stakeholder meetings and decisionmaking processes (Snyder 2001) and numerical optimization algorithms, such

BioScience 62: 508–515. ISSN 0006-3568, electronic ISSN 1525-3244. doi:10.1525/bio.2012.62.5.13

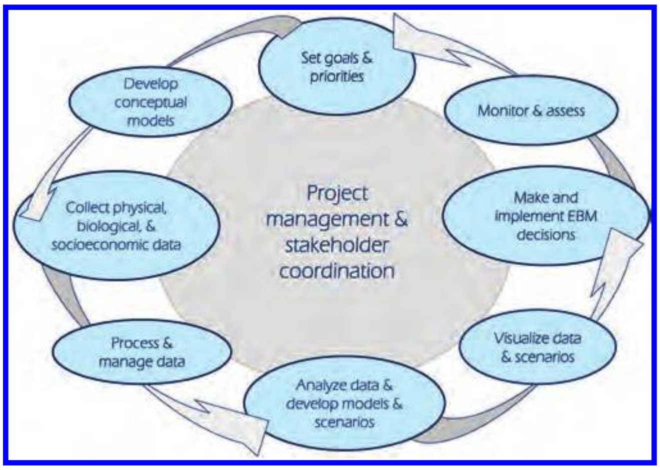

Figure 1. Typical life cycle of activities undertaken by resource managers implementing ecosystem-based management principles. Using specially designed software tools can make accomplishing each of these tasks faster and more efficient.

as C-Plan (Pressey et al. 2008) and Marxan (Ball et al. 2009), that can be used to select protected areas that minimize socioeconomic impact and maximize biodiversity relative to the reserves selected by traditional ad hoc methods (Leslie et al. 2003, Klein et al. 2008). Additional examples of tools and how they can be utilized in EBM can be found at *www.ebmtoolsdatabase.org*, and case studies showing how specific tools have been used by managers can be found at *www.ebmtoolsdatabase.org/projects*. Successful implementation of EBM often depends, in part, on the appropriate selection, acquisition, and operation of tools by resource managers and on the quality and accessibility of the tools themselves.

Over the last five years, while participating in the Coastal–Marine Ecosystem-Based Management Tools Network (EBM Tools Network; *www.ebmtools.org*), a consortium of EBM tool users and developers, we heard persistent complaints from users that the tools were often difficult to use, lacked documentation, contained numerous bugs, and were poorly supported and maintained. EBM software products did not measure up to the quality standards that software users had become accustomed to with commercial software. For their part, developers frequently attributed their inability to produce higher-quality, better-maintained EBM software to a lack of sufficient long-term funding. As developers of our own EBM software tools (e.g., Halpin et al. 2006, Roberts et al. 2010) that were funded through small episodic research grants, we sympathized with this complaint.

We (the Duke coauthors: CC, DCD, JJR, PNH) have also administered a grant program—the Marine Ecosystem-Based Management Tool Innovation Fund (MEBM-TIF), which was intended to stimulate the development and dissemination of software tools for marine EBM. Our responsibilities included soliciting proposals, selecting the grantees, disbursing money to the grantees, and monitoring the projects for successful completion. The program received an overwhelming level of interest from the tool-development community: 154 applications seeking over $20 million in funding. We had a $1 million budget and funded 12 projects. The MEBM-TIF experience appeared to support the development community's claim that obtaining funding was a problem, but was the lack of sufficient funds the sole cause of low-quality EBM tools, or were there other contributing factors? How are developers and funders contributing to the lack of quality, and what could they do differently to promote higher-quality software?

To investigate these questions, we interviewed 24 tool developers (Curtice et al. 2010) to test several theories, including that the funding mechanisms typically used for commercial software development are not viable sources for conservation tools, that researchers are intentionally pursuing grants rather than charging fees to fund tool development, and that the organization type or the funding mechanism affects software quality. In this article, we present an overview of our interviews, highlights of our results from the full report, and recommendations for both tool developers and funders.

In the interest of full disclosure, we remind the reader that we are EBM tool developers ourselves and acknowledge that we might benefit if our recommendations were broadly implemented. However, having served as both developers and funders, we believe that we are in a unique position to comment on the community's situation and have tried to minimize personal bias in this study. To that end, we excluded the Duke Marine Geospatial Ecology Laboratory from the pool of interviewees, and although we included NatureServe in the interview process, the NatureServe coauthor (SDC) has never been associated with their EBM tool project. Although we focused this study on the EBM tool community, we also believe that our questions and conclusions are likely to be applicable to any type of software development by nonprofit organizations.

Interviews with tool developers

Drawing from the relationships that we built while participating in the EBM Tools Network and while running the MEBM-TIF grants program, we identified a group of tool developers to interview that represented a range of organization types (e.g., government, academic, nonprofit), funding mechanisms (box 1), user-community sizes (i.e., the number of tool users), and tool longevity. The interviewees were located around the world. All of these individuals received an introductory e-mail describing our study and asking for their participation. The e-mail informed them that their responses would be recorded in a report and possibly a manuscript, that they had the option to keep their responses anonymous, that there was no compensation for participation, and that participation was entirely voluntary. Each interview lasted about one hour and consisted of approximately 18 questions (in some cases, we did not ask questions that were not relevant to the project, and in other cases, we probed further with additional ad hoc questions),

Box 1. Sources of funding for software tool development.

The sources of funding commonly used by academics and others to fund tool development can be grouped into nine categories, including the venture-capital mechanism used by commercial software developers described in the text. Below, we describe the basic characteristics of the other eight mechanisms.

Grants can be from government agencies like the National Air and Space Administration or the National Science Foundation, from nongovernmental organizations (NGOs) such as the World Wildlife Fund, or from foundations such as The David and Lucille Packard Foundation. Grant money is episodic, because it is a one-time event with a limited term and no follow-on funds.

Internal government money is used by a government agency, such as the National Oceanic and Atmospheric Administration's Coastal Services Center. Agencies get annual budgets that are allotted to various projects. There is an internal process for choosing what projects to fund, so a project needs a champion and a good reason to be continued each year.

Fee-for-license models charge a fee for the use of a product. Fees can be based on the number of users or installations; they can be one-time fees or annually renewable fees.

In a *fee-for-service* or "contract" model, services are sold to generate revenue. Services range from basic training courses and access to technical support to more in-depth software customizations that meet the specific and unique needs of individual users.

Donations are usually associated with NGOs in the form of memberships but can also be in the form of corporate sponsorship for a specific product. Some tool developers also solicit donations directly from users to support the development effort (this may be referred to as *shareware*).

Endowments are funds donated to an institution that are invested. Interest earned from the investments can be used to fund research projects. This creates a perpetual source of income for the institution, although the amount of income will vary with the economy and depends on the investments made.

In a *skunkworks* project, funding allocated for scientific research or similar work is used to develop a software product, even though no money was originally allocated for that purpose. Many EBM tools start as skunkworks projects and often come about when a researcher realizes that code developed for a specific scientific analysis could benefit others and attempts to turn that code into a fully fledged reusable product.

Finally, despite not being an actual source of money, *open-source* projects are gaining momentum. R, Open Office, and Linux are all popular examples of open-source products. In the EBM world, OSGeo—the Open Source Geospatial Foundation—is a group of projects developed and made freely available under an open-source license. OSGeo is a nonprofit foundation created to support the collaborative development of open geospatial technologies and data.

and most were audio recorded (with permission). We asked open-ended questions about how the respondents sought funding for the development of their tool, how well that funding worked to get the tool implemented, whether they believed that the tool was ready for widespread use, and whether they believed that the tool was financially sustainable (see the supplemental appendix, available online at *http://dx.doi.org/10.1525/bio.2012.62.5.13*, for the full interview questionnaire). Open-ended questions are useful for gathering information, since they allow respondents to provide more in-depth answers than closed-ended questions do. In addition, we ascertained to what degree the developer was able to complete the full software-development life cycle (e.g., Larman 2003 or Royce 1970, but there are hundreds of texts available and many methodologies describing how to develop robust software programs) by asking specific questions about which stages they were able to complete with their funding (e.g., prototype development, documentation, software updates, technical support; see question 10 of the supplemental appendix for a full list of the criteria).

Because we used a survey that solicited unstructured information, we had to analyze our data and generate results using qualitative research methods. Qualitative research is a well-established field used across various disciplines (Glaser and Strauss 1967, Grbich 2007). To analyze our survey results, we used the qualitative data-analysis software package NVivo (Version 8; QSR International, Cambridge, Massachusetts). We inductively discovered themes (termed *codes* in NVivo) from the data (rather than designating themes before gathering the data). Through the NVivo codes that were applied to sections of each interview response, we then identified patterns and connections among funding mechanisms, tool success, organization type, and other factors. Occurrences of certain codes were counted to quantify different interviewees who expressed the same message.

Findings from the survey

Below, we summarize our findings in four sections: (1) why the EBM tools market is served by academics and not commercial software companies, (2) the strong sense of mission held by many EBM tool developers, (3) the heavy reliance on grants to support tool development and the consequences of choosing that funding strategy, and (4) differences in how financial sustainability is defined inside and outside of the EBM community.

Commercial versus nonprofit software development. Commercially produced software generates self-sustaining revenue through the sale of licenses or services. One question we wanted to address through our interviews was why the EBM market is not served by commercial companies. Commercial software ventures exist only if the potential market is thought to be large enough to allow companies to recoup the costs of researching, designing, implementing, releasing, marketing, and supporting the first version of the product, plus turn a profit. These companies are often started with the backing of venture-capital firms hoping to receive a large return on their investment through the success of the company, either by its becoming a publicly traded entity or by its purchase by a larger company. To achieve any return on investment, the license fees for a tool must be sufficient to offset the costs of bringing the product to market and produce a sizable profit. Therefore, the minimum sustainable license fees are inversely proportional to the number of expected licenses.

Because the potential market for EBM software products is relatively small, products developed under the traditional commercial paradigm would generally require license fees that EBM practitioners and resource managers would be unable to afford. This lack of a viable market has resulted in the EBM field being built and run on tools developed by researchers in academia rather than by commercial products developed by for-profit firms. Almost all highly successful EBM tools developed by academics are free, and the field remains dependent on academia to maintain and add to this collection of tools and to keep them affordable. The profits and income from selling software licenses, which could help maintain and support a tool, are not often considered a measure of success; success for academic tool developers is measured by the number of publications and citations the tools generate in peer-reviewed literature and, to a lesser extent, the degree to which the tools help solve real-world problems. That secondary metric, what we termed the *mission belief*, is a significant factor in deterring tool developers in the academic arena from charging license fees.

Why tools developed by academia are free. The aversion of most EBM developers to charging license fees goes beyond a focus on publications and citations. During our interviews, we heard many developers express a strong belief in their "mission"—that is, an altruistic desire to help solve real-world environmental problems as quickly and effectively as possible. For these developers, the top priority is having their product used by the people who may solve those problems. Much of the world's coastline is in developing nations that cannot afford expensive software, and keeping tools free ensures that managers working in those parts of the world have access to the same capabilities as managers in developed countries. Developers see license fees as an impediment to progress and quickly discard them on practical or even moral grounds. The developers verbalized this belief repeatedly during our interviews. For instance, one developer from a nonprofit organization said, about transitioning from a fee-for-license model to a fee-for-service model, "The license fee was hurting our 'mission success': We built the tool to help conservation, and if not many people are using it, how much of a dent are we making?" A developer from another nonprofit answered, "Charging a license fee is not our philosophy. Especially being a nonprofit doing conservation work." A developer from a government agency said, "Selling my product would go wholly against the grain of the science that I do, since the tool would be out of the dollar range of developing countries explicitly. I would rather give it to the people who need it."

In addition, researchers in academia, nonprofits, and government agencies expressed a strong lack of desire to deal with the business administration that is required to implement and support any type of fee-based model. Even if an EBM researcher were willing to take on that job, there is often little or no institutional support for it. Universities do not typically have the capacity to collect money from sales of software or to defend it from piracy (being stolen and released through unauthorized channels). Similarly, government agencies that develop EBM tools are not in a position to implement a fee-based business model.

Shortfalls of the grant-based funding model. Lacking the market share needed for commercial businesses to develop tools, most EBM tools are funded episodically with grants, often as "skunkworks" projects that covertly divert funds from short-term grants intended to support basic research, not software product development (box 1). The prevalence of skunkworks efforts may be due to the scarcity of funders willing to directly fund software development (a common complaint of our interviewees) or to researchers' often considering hiring professional software tool development expertise a lesser priority than covering core salaries and benefits for themselves and the existing members of their lab. Researchers may also have trouble estimating the effort required to turn prototype-quality code written for a specific analysis into a high-quality redistributable product and unwittingly launch into development projects that eventually become huge, unfunded time sinks. In any case, the skunkworks pattern has several consequences that all stem from the fact that, under such circumstances, the software tool is not listed on the grant as a deliverable specifically authorized and expected by the funder.

The consequences begin even before tool development starts, since the researcher is usually unable to hire professional software developers. Professional developers command high salaries in the commercial world, and without a steady stream of revenue, researchers have trouble keeping them on staff. One survey respondent summed up the situation: "We are severely limited in [the] salaries that we are able to offer to developers. [It] has been a significant problem. At universities, salaries are typically lower than [those] outside. Another nonprofit had a talented developer quit due to being underpaid; [we have] trouble recruiting due to low salary offers." Hiring outside developers on short-term contracts is even more expensive in hourly terms, and it is often impossible to bill the expense to the grant, because software development was not specified in the grant's budget. Therefore, graduate students and postdoctoral researchers are often tasked with the software design and programming. Their lack of programming expertise and training in software-engineering methodologies often leads to excessive development times and low-quality software. Staff turnover is a continual problem, because students graduate and postdoctoral researchers move on to their next assignments.

The situation is exacerbated by the short duration of most grants. When a grant ends, the tool it funded is often minimally maintained for a period of time until additional injections of money allow for further development, either explicitly through grants that cover software development or more commonly as additional skunkworks projects. Experienced software developers are further deterred from engaging in the project because of these funding gaps, which they perceive as a lack of financial stability. Longer gaps between funding events increase the likelihood that the tool will become obsolete, which means that the time and funds that have already been invested may have been wasted.

In addition, researchers usually lack formal training and experience with managing software projects. Consequently, most tool-development projects do not follow well-defined software-engineering process methodologies, such as agile software development (Larman 2003) or the waterfall model (Royce 1970), that are often used in the commercial world. These methodologies ensure that all the important tasks of a successful software project are addressed, not just the coding. These tasks include speaking with potential and existing users to determine functional requirements, verifying that the software executes correctly, writing documentation, preparing training materials, providing support to users, and fixing bugs reported after the software's release. The short-term nature of most grants and the general lack of funds for tool development unquestionably contribute to abbreviated and fragmented development cycles. The reliance on grants and the lack of a viable tool market have resulted in a serious resource limitation for EBM tool developers relative to the resources of traditional for-profit software developers. For example, in the MEBM-TIF program that we managed, the average award was $80,000. By comparison, in 2010, the average amount invested by venture-capital companies in a new commercial-software development project was $4.7 million (NVCA 2011). The amount of funding available to EBM tool developers is less than 2% of that invested in new commercial software efforts.

Financial sustainability. We expected to uncover a link between funding mechanism or organization type and the level of success of a tool (see Curtice et al. 2010 for more details on how the level of success was determined). Surprisingly, we found none. Instead, what we found was a trend that the more successful tools had more successful long-term champions—someone who intentionally and continuously devoted time and resources to finding funding, to finding graduate students to continue work on the tool, and to push it forward and support it on his or her own time and money when he or she was unable to raise funds. As one champion of a successful tool put it, "If I hadn't worked with it continuously all these years, [the tool] would have died long ago. [It] needs someone to carry the flag." Many researchers revealed that the financing they did find was often "opportunistic"; they were in the right place at the right time, had the right personal connections to hear about a funding opportunity, and cultivated the right relationships to be awarded grants. Publishing in peer-reviewed literature was a measure of success and helped open doors to new research funding. We also found that charging a license fee for a product was not a guarantee of financial sustainability. None of the four tools that were charging a license fee at the time of the interviews were sustainable on the sole basis of that stream of revenue; they all had received additional grants, donations, endowments, or venture-capital money. Furthermore, three of the four tool providers that charged license fees did not believe that their tool was financially sustainable at the time of our interview.

We defined *financially sustainable* as meeting the following criteria: (a) the product had gone through more than one development and release cycle; (b) there was an expected source of funding for the foreseeable future; and (c) the product met certain requirements that we considered the minimum necessary for successful deployment by users—namely, that complete user documentation existed and that technical support was available through e-mail or telephone. Several respondents indicated that they felt their product was financially stable but implied the use of different criteria for this judgment: One answered, "Yes, [our project is financially sustainable] because it will probably continue to be a low-cost operation. Small and focused will be sustainable." Another said that their product was "sustainable in the sense [that] we've continued to work on it for two years and continue to work on it even without direct funding." This difference in attitude toward sustainability is paramount. If a researcher believes that keeping a project small and updating and supporting a tool on his or her own time is sustainable, it is unlikely that alternative solutions will be actively sought. Although this might sound similar to product champions who continue to self-fund and push their product forward between funding episodes, there is a subtle but important difference. Product champions know they must continue to force the issue, seek additional funds, and pursue relationships and personal connections with funders in order to move their product up to the next level, and they do not believe that this situation meets a definition of *sustainable.* They do not simply want to keep their tool alive; they want to expand its capabilities, public awareness of the tool, and the user audience. "Small and focused" is not realistically sustainable at the scale of use that we expect marine EBM tools will start experiencing as more projects are begun in response to federal and state mandates for EBM of the marine environment.

Conclusions and recommendations for developers and funders

Successfully implementing EBM practices increasingly requires the use of software tools. Most EBM software tools originate in academic research labs, government agencies, and nonprofit organizations using short-term research grant money that is not intended to and cannot support the diverse and long-term activities required to maximize the

Box 2. Recommendations for developers.

We recommend that developers do the following:

Seek multiple revenue streams to ensure financial stability. Consider a scaled fee-for-license or -services model to maximize adoptions without sacrificing revenue. Do not undervalue the expert knowledge and the services that you can provide; where it is applicable, charge for services that make your tool more valuable to users.

Plan for and allocate funds and personnel for the development of documentation and training, for marketing the tool, for supporting users, and for maintaining the code (fixing bugs) after the product has been released. Educate funders about the benefits of funding these activities.

Retain professional software engineers and other skilled specialists.

Use engineering practices that promote sustainability. Educate yourself about different software-development methodologies (e.g., agile unified process, extreme programming) and choose one that best fits your organization. When transitioning from a prototype tool developed by amateur developers (e.g., graduate students) to a tool intended for a wide distribution, consider having professionals rewrite the code from scratch. This will likely be more cost effective in the long run than trying to build on the existing code. Consider a modular product plan that promotes interoperability by packaging key functionality in compact modules that can be easily reused in other projects.

Explore open-source licensing models as a means for creating a community-development environment that will contribute to the long-term sustainability and support of the tool.

Commit yourself to funding, maintaining, supporting, and championing your tool over its foreseeable lifetime, which is usually at least several years.

Box 3. Recommendations for funders.

Break the episodic funding cycle. Consider endowing established tools or open-source development communities. Consider longer-term funding.

Perform more due diligence and market research before funding a project to avoid funding duplicate efforts, unless you see a specific benefit to duplication (e.g., competition, risk mitigation).

Explicitly fund the development of tools, not just the research that makes them possible.

Explicitly fund the additional activities needed to deliver quality tools to the users who need them, including the development of documentation and training, marketing the tool, supporting users, and maintaining the code (fixing bugs).

Favor proposals that describe how the tool will be sustained in the long term, and address how the activities listed above will be funded, staffed, and completed.

Favor proposals that identify specific customers or users and describe how this audience will be incorporated into the requirements-gathering and product-validation phases.

Favor proposals that plan to employ sufficiently talented software engineers and other required personnel. Be prepared and willing to pay competitive compensation to attract those people away from commercial development opportunities to ecosystem-based management software development. Be wary of proposals that plan to use amateur developers, such as students and postdoctoral researchers who lack university or professional training in software development. Be wary of underbudgeted proposals.

Favor proposals that promote developing generic and interoperable core-functionality modules that can be repackaged in different applications.

Encourage partnerships between academic, government, or nongovernmental-organization developers and the commercial-development community.

Favor proposals with a demonstrated product champion.

Favor proposals that plan to use proven software-development methodologies.

Encourage academic, nongovernmental, and government institutions to make it easier for their employees to use fee-for-service or fee-for-license funding mechanisms.

quality and impact of such tools. EBM tool developers often cannot afford or attract professional software engineers and must use graduate students or other less-experienced developers who lack the knowledge to produce high-quality, supportable, and extensible code.

Despite all the challenges associated with obtaining long-term funding for the development of EBM tools, it can be done. There are many successful tools available. Supporting and maintaining these tools is the primary challenge of the developer. Resources are needed to fix bugs, support users, and update the software's compatibility with other products. If resources are not available for these purposes, users will not be able to accomplish their objectives for using the tool, and the tool will become unused and obsolete. This wastes the money originally invested in developing the tool and deprives EBM practitioners and managers of the full benefits the tool could provide. Together, changes adopted by both developers and funders will increase the rates of success and sustainability of EBM software tools and will contribute to better understanding and management of our coastal and marine resources.

In the present article, we put forth comprehensive recommendations for developers and funders to promote the creation of more sustainable and more widely adopted EBM tools (boxes 2 and 3). In summary, developers should not undervalue their knowledge and the services they provide. We encourage them to seek multiple sources of funding, including fee-based sources, to help assure the long-term financial stability of the product. Furthermore, development budgets should account for the need to hire professional software developers, to produce documentation and customer training courses, and to support and maintain the product for several years after its initial release. Funders should acknowledge the fundamental need for software tool development to aid in the implementation of EBM processes and should support the employment of experienced programmers to improve product quality. Proposals in which documentation, support, and maintenance of the tool are addressed and in which the ways in which the tool will be sustained in the long term are discussed should be favored. To help break the episodic funding cycle, funders should consider creating tool endowments, requiring participation in open-source development communities, and proactively funding software projects for longer terms.

Acknowledgments

The authors wish to thank the David and Lucile Packard Foundation for funding this study. We are also grateful to all the people who agreed to be interviewed, taking the time to speak openly with us about their experiences developing software tools (the full list of interviewees may be found in the supplemental material). The Coastal–Marine Ecosystem-Based Management (EBM) Tools Network (*www.ebmtools.org*) and the EBM Tool Developers' Collaborative were both helpful in putting together the potential list of interview subjects and in providing feedback on the report. The opinions, findings and conclusions expressed in this report are solely and exclusively those of the authors and not those of Duke University, NatureServe, or the David and Lucile Packard Foundation.

References cited

Ball IR, Possingham HP, Watts ME. 2009. Marxan and relatives: Software for spatial conservation prioritization. Pages 185–195 in Moilanen A, Wilson KA, Possingham HP, eds. Spatial Conservation Prioritization: Quantitative Methods and Computational Tools. Oxford University Press.

Botsford LW, Castilla JC, Peterson CH. 1997. The Management of Fisheries and Marine Ecosystems. Science 277: 509–515.

Christensen NL, et al. 1996. The report of the Ecological Society of America committee on the scientific basis for ecosystem management. Ecological Applications 6: 665–691.

Curtice C, Dunn D[C], Roberts J, Halpin PN. 2010. A Study on Sustainable Financing for the Development and Maintenance of Ecosystem-Based Management Tools: Report by the Marine Geospatial Ecology Lab at Duke University for the David and Lucille Packard Foundation, March 2010. (8 February 2012; *http://ebmtools.openconcept.ca/sites/ebmtools/files/SustainableFinancingFinalReport.pdf*)

Gamble RJ, Link JS. 2009. Analyzing the tradeoffs among ecological and fishing effects on an example fish community: A multispecies (fisheries) production model. Ecological Modelling 220: 2570–2582.

Garrison LP, Link JS, Kilduff DP, Cieri MD, Muffley B, Vaughan DS, Sharov A, Mahmoudi B, Latour RJ. 2010. An expansion of the MSVPA approach for quantifying predator–prey interactions in exploited fish communities. ICES Journal of Marine Science 67: 856–870.

Glaser BG, Strauss AL. 1967. The Discovery of Grounded Theory: Strategies for Qualitative Research. Aldine de Gruyter.

Grbich C. 2007. Qualitative Data Analysis: An Introduction. Sage.

Halpin PN, Read AJ, Best BD, Hyrenbach KD, Fujioka E, Coyne MS, Crowder LB, Freeman SA, Spoerri C. 2006. OBIS-SEAMAP: Developing a biogeographic research data commons for the ecological studies of marine mammals, seabirds, and sea turtles. Marine Ecology Progress Series 316: 239–246.

Klein CJ, Steinback C, Scholz AJ, Possingham HP. 2008. Effectiveness of marine reserve networks in representing biodiversity and minimizing impact to fishermen: A comparison of two approaches used in California. Conservation Letters 1: 44–51.

Larman C. 2003. Agile and Iterative Development: A Manager's Guide. Addison–Wesley.

Leslie H, Ruckelshaus M, Ball IR, Andelman S, Possingham HP. 2003. Using siting algorithms in the design of marine reserve networks. Ecological Applications 13: S185–S198.

Margules CR, Pressey RL. 2000. Systematic conservation planning. Nature 405: 243–253.

McLeod KL, Lubchenco J, Palumbi SR, Rosenberg AA. 2005. Scientific Consensus Statement on Marine Ecosystem-Based Management. Communication Partnership for Science and the Sea.

[NVCA] National Venture Capital Association. Total U.S. Investments by Year Q1 1995–Q4 2010. NVCA. (8 February 2012; *www.nvca.org/index.php?option=com_docman&task=doc_download&gid=690&Itemid=93*)

Pew Oceans Commission. 2003. America's living oceans: Charting a course for sea change. Pew Oceans Commission.

Possingham H, Ball I[R], Andelman S. 2000. Mathematical Methods for Identifying Representative Reserve Networks. Pages 291–305 in Ferson S, Burgman M, eds. Quantitative Methods for Conservation Biology. Springer.

Pressey RL, Watts ME, Barrett TW, Ridges MJ. 2009. The C-Plan conservation planning system: Origins, applications, and possible futures. Pages 211–234 in Moilanen A, Wilson KA, Possingham HP, eds. Spatial Conservation Prioritization: Quantitative Methods and Computational Tools. Oxford University Press.

Roberts JJ, Best BD, Dunn DC, Treml EA, Halpin PN. 2010. Marine Geospatial Ecology Tools: An integrated framework for ecological geoprocessing with ArcGIS, Python, R, MATLAB, and C++. Environmental Modelling and Software 25: 1197–1207.

Rouyer T, Fromentin J-M, Ménard F, Cazelles B, Briand K, Pianet R, Planque B, Stenseth NC. 2008. Complex interplays among population dynamics, environmental forcing, and exploitation in fisheries. Proceedings of the National Academy of Sciences 105: 5420–5425.

Royce WW. 1970. Managing the development of large software systems. Pages 328–338 in Proceedings, IEEE WESCON August 1970. Institute of Electrical and Electronics Engineers.

Slocombe DS. 1993. Implementing ecosystem-based management. BioScience 43: 612–622.

Snyder K. 2001. Decision support tools for community planning. Public Management 83: 4–8.

US Commission on Ocean Policy. 2004. An ocean blueprint for the 21st century: Final report of the U.S. Commission on Ocean Policy. (8 February 2012; *www.oceancommission.gov/documents/full_color_rpt/welcome.html*)

Wang X. 2001. Integrating water-quality management and land-use planning in a watershed context. Journal of Environmental Management 61: 25–36.

[WHCEQ] White House Council on Environmental Quality. 2010. Final Recommendations of the Interagency Ocean Policy Task Force July 19, 2010. WHCEQ. (8 February 2012; *www.whitehouse.gov/files/documents/OPTF_FinalRecs.pdf*)

Corrie Curtice (corrie.curtice@duke.edu), Daniel C. Dunn, and Jason J. Roberts are all researchers with the Marine Geospatial Ecology Lab at the Nicholas School of the Environment at Duke University, in Durham, North Carolina. Sarah D. Carr is the coordinator of the Coastal–Marine Ecosystem-Based Management Tools Network and is based at the nonprofit conservation organization NatureServe, in Arlington, Virginia. Patrick N. Halpin is an associate professor of marine geospatial ecology and director of the Geospatial Ecology Program at the Nicholas School of the Environment, at Duke University, in Durham, North Carolina.