

## The Business of Software Improvement

**T**he return on investment in improved software practices is well documented. In 1994, James Herbsleb reported that the average “business value” (roughly the same as ROI) for 13 organizations that took on CMM-SW-based improvement programs was about 5 to 1, with the best organizations realizing returns of 9 to 1.<sup>1</sup> In 1995, Neil C. Olsen reported similar returns for organizations that made significant investments in staffing, training, and work environments.<sup>2</sup> In 2000, Capers Jones reported that the ROI from process improvement could easily go into double digits (meaning returns greater than 10 to 1).<sup>3</sup> A recent analysis by Watts Humphrey found that the ROI for improved software practices could be in the neighborhood of 4 to 1.<sup>4</sup>



### Indirect benefits are even more significant

The ROI figures in the published literature are based on operational savings—that is, on reducing development cost per line of code written or per function point delivered. Although these savings are impressive, the greater business benefit might arise from the significant *indirect* returns that arise from improved software practices. Better software practices improve predictability of costs and schedules, reduce risk of cost and schedule overruns, provide early warning of problems, and support better management.

Many organizations that have focused on improving their software practices have re-

ported improvements in predictability similar to the results in Figure 1.<sup>5</sup> For a software products company, what is the business value of improving schedule estimation accuracy from plus or minus 100 percent to plus or minus 10 percent? What is the value of being able to make a commitment to customers six to 12 months in advance of a scheduled completion date, with high confidence of delivering on that commitment?

For a company that develops custom software, what is the business value of being able to provide a fixed price bid with high confidence that the project will not significantly overrun the bid?

For a retail sales organization, what is the value of being able to plan cutover to a new system with pinpoint accuracy? What is the value of knowing with confidence that cutover will occur 1 October, as planned, with little risk of overrunning to 15 November or 1 December?

Unlike the operational benefits that most of the industry literature has focused on, these indirect benefits open the door to additional revenue opportunities. These benefits are based not on reducing costs, but on increasing access to additional business. For top decision makers in organizations, these indirect benefits are often more compelling than the direct, operational benefits.

### Organizational challenge

Considering the strong—even compelling—case for improving software practices, it might seem surprising that some organizations have not made a commitment to use best practices. I have recently been think-

## DEPARTMENT EDITORS

**Bookshelf:** Warren Keuffel, wkeuffel@computer.org

**Construction:** Andy Hunt and Dave Thomas, Pragmatic Programmers, (Andy, Dave)@pragmaticprogrammer.com

**Country Report:** Deependra Moitra, Lucent Technologies d.moitra@computer.org

**Design:** Martin Fowler, ThoughtWorks, fowler@acm.org

**Loyal Opposition:** Robert Glass, Computing Trends, rglass@indiana.edu

**Manager:** Don Reifer, Reifer Consultants, dreifer@sprintmail.com

**Quality Time:** Jeffrey Voas, Cigital, voas@cigital.com

## STAFF

Senior Lead Editor

**Dale C. Strok**  
dstrok@computer.org

Group Managing Editor  
**Crystal Chweh**

Associate Editors

**Jenny Ferrero, Shani Murray, and Dennis Taylor**

Staff Editors

**Scott L. Andresen and Kathy Clark-Fisher**

Editorial Assistants

**Rebecca Deuel and Ty Manuel**

Magazine Assistants

**Dawn Craig**, software@computer.org

**Pauline Hosillos**

Art Director

**Toni Van Buskirk**

Cover Illustration

**Dirk Hagner**

Technical Illustrator

**Alex Torres**

Production Assistant

**Monette Velasco**

Production Artists

**Carmen Flores-Garvey and Larry Bauer**

Executive Director

**David Hennage**

Publisher

**Angela Burgess**

Assistant Publisher

**Dick Price**

Membership/Circulation Marketing Manager

**Georgann Carter**

Advertising Assistant

**Debbie Sims**

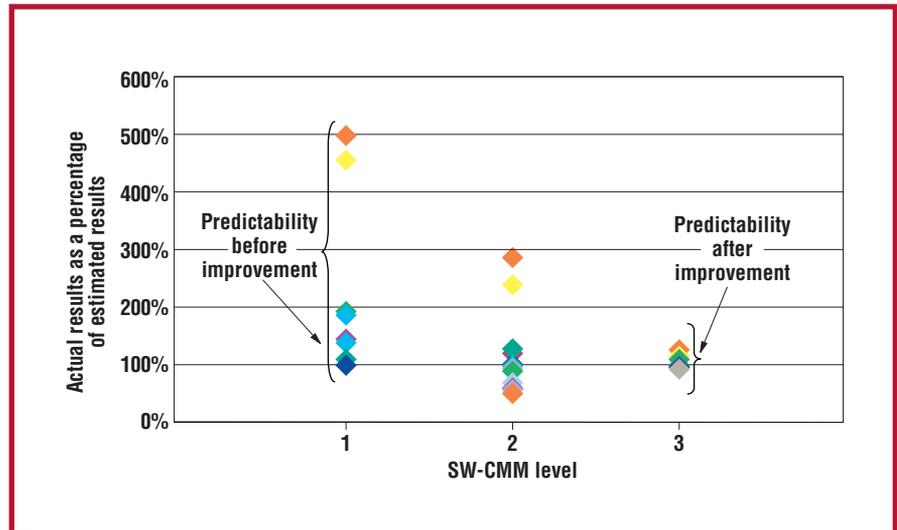
## CONTRIBUTING EDITORS

Greg Goth, Keri Schreiner, Joan Taylor, and Margaret Weatherford

**Editorial:** All submissions are subject to editing for clarity, style, and space. Unless otherwise stated, bylined articles and departments, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *IEEE Software* does not necessarily constitute endorsement by the IEEE or the IEEE Computer Society.

**To Submit:** Send 2 electronic versions (1 word-processed and 1 postscript or PDF) of articles to Magazine Assistant, *IEEE Software*, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1314; software@computer.org. Articles must be original and not exceed 5,400 words including figures and tables, which count for 200 words each.

## FROM THE EDITOR



**Figure 1. Project performance compared to estimated performance.<sup>5</sup> This example demonstrates different projects in the US Air Force.**

ing a lot about why best practices aren't used. Several factors seem to be in play.

First, there is a basic technology transfer issue. Many software development best practices have been available for decades, but only a few companies use them, and undergraduate programs have not generally taught these best practices. The scarcity of experienced users of these practices limits the rate at which current users can train new users. Although a person might reasonably assume that the average software organization's capability is halfway between the worst organization's capability and the best's, in reality, the average software organization's practices are much closer to the worst organization's practices than the best's. The result is that software developers who work in average organizations—which includes most developers—have never seen a really well-run software project, much less a really well-run software organization. The software industry faces the problem of bootstrapping best practices into common usage because of limited current usage of them.

A second factor is that recent economic circumstances have prevented software organizations from feeling any strong imperative to switch to better practices.<sup>6</sup> Throughout the 1990s, software-related companies rode a technology wave that rewarded companies just for being in the software in-

dustry. Companies didn't need to focus on operational improvements because that would have shifted too much focus away from generating revenue. For a time, improved software practices seemed to be more of a distraction than a help.

A final factor is that many organizations push responsibility for software development improvement down to the project level. In reviewing the "effort multiplier" factors in the Cocomo II estimation model,<sup>7</sup> I was struck by how few of the factors are under the control of an individual project manager. Of the 22 factors Cocomo II uses to fine-tune a project's base effort estimate, in my judgment only three are typically under the individual project manager's control: documentation, architecture and risk resolution, and development for reuse. Numerous factors are dictated by the nature of the company's business—product complexity, required reliability, platform volatility, unprecedentedness of the software, and so on. A company cannot easily change these factors without changing businesses. The remaining factors—staff capability, multisite devel-

opment, personnel continuity, process maturity, and so on—can be influenced by the organization but not by individual projects.

### What can you do?

We could hope that upper management, sales, and marketing staff would read every issue of *IEEE Software* cover to cover or educate themselves about the finer nuances of software development some other way. But this isn't likely to happen, so leading software practitioners have an ongoing responsibility: the education of nontechnical software project stakeholders. Software practitioners sometimes perceive upper management and other nontechnical staff to be blocking the use of better practices. We complain that they fail to support better practices or even undermine them. I've generally found, however, that upper management, sales, marketing, product support, and other personnel are receptive to improved software practices when I take the time to explain those practices to them. Indeed, they are acutely aware of the problems caused by current practices and are eager to

hear how they can help improve software projects.

What have you done to educate executives about better software practices? What has worked well for you? I'd love to hear your comments at [stevemcc@construx.com](mailto:stevemcc@construx.com). ☺

### References

1. J. Herbsleb et al., *Benefits of CMM Based Software Process Improvement: Initial Results*, tech. report CMU/SEI-94-TR-13, Software Eng. Inst., Carnegie Mellon Univ., Pittsburgh, 1994.
2. N.C. Olsen, "Survival of the Fastest: Improving Service Velocity," *IEEE Software*, vol. 12, no. 5, Sept./Oct. 1995, pp. 28-38.
3. C. Jones, *Software Assessments, Benchmarks, and Best Practices*, Addison Wesley, Reading, Mass., 2000.
4. W. Humphrey, *Winning with Software: An Executive Strategy*, Addison Wesley, Reading, Mass., 2001.
5. P.K. Lawlis, R.M. Flowe, and J.B. Thordahl, "A Correlational Study of the CMM and Software Development Performance," *Crosstalk*, Sept. 1995.
6. S. McConnell, *After the Gold Rush*, Microsoft Press, Redmond, Wash., 1999.
7. B. Boehm et al., *Software Cost Estimation with Cocomo II*, Addison Wesley, Reading, Mass., 2000.

## Call for Articles: Industry Experience Reports

Experience reports with lessons learned in industry are one way to share successes or failures with others who likely face similar situations. We invite you to report on a technology or process you introduced in your company, analyze its impact, and explore what you would do differently the next time.

Articles should be 2,000 to 2,400 words with each illustration counting as 200 words. We also encourage you to submit up to 10 short bullet points on lessons learned and references to related Web sites. Submissions are reviewed by members of our Industry Advisory Board and are subject to editing for style, clarity, and space.

For detailed author guidelines, see [computer.org/software/genres.htm#IndustryExperienceReport](http://computer.org/software/genres.htm#IndustryExperienceReport) or contact [software@computer.org](mailto:software@computer.org). Submissions are accepted at any time.

Associate Editor: Wolfgang B. Strigel, Software Productivity Center, [strigel@spc.ca](mailto:strigel@spc.ca).

EDITOR IN CHIEF:

Steve McConnell

10662 Los Vaqueros Circle  
Los Alamitos, CA 90720-1314  
[software@construx.com](mailto:software@construx.com)

EDITOR IN CHIEF EMERITUS:

Alan M. Davis, Univ. of Colorado  
at Colorado Springs

ASSOCIATE EDITORS IN CHIEF

**Design:** Maarten Boasson, Quaerendo Invenietis  
[boasson@quaerendo.com](mailto:boasson@quaerendo.com)

**Construction:** Terry Bollinger, Mitre Corp.  
[terry@mitre.org](mailto:terry@mitre.org)

**Requirements:** Christof Ebert, Alcatel Telecom  
[christof.ebert@alcatel.be](mailto:christof.ebert@alcatel.be)

**Management:** Ann Miller, University of Missouri, Rolla  
[miller@ece.umar.edu](mailto:miller@ece.umar.edu)

**Quality:** Jeffrey Voas, Cigital  
[voas@cigital.com](mailto:voas@cigital.com)

**Experience Reports:** Wolfgang Strigel,  
Software Productivity Center; [strigel@spc.ca](mailto:strigel@spc.ca)

EDITORIAL BOARD

Don Bagert, Texas Tech University  
Richard Fairley, Oregon Graduate Institute  
Martin Fowler, ThoughtWorks  
Robert Glass, Computing Trends  
Andy Hunt, Pragmatic Programmers  
Warren Keuffel, independent consultant  
Brian Lawrence, Coyote Valley Software  
Karen Mackey, Cisco Systems  
Deependra Moitra, Lucent Technologies, India  
Don Reifer, Reifer Consultants  
Suzanne Robertson, Atlantic Systems Guild  
Dave Thomas, Pragmatic Programmers

INDUSTRY ADVISORY BOARD

Robert Cochran, Catalyst Software (chair)  
Annie Kuntzmann-Combelles, Q-Labs  
Enrique Draier, PSINet  
Eric Horvitz, Microsoft Research  
David Hsiao, Cisco Systems  
Takaya Ishida, Mitsubishi Electric Corp.  
Dehua Ju, ASTI Shanghai  
Donna Kasperperson, Science Applications International  
Pavle Knaflic, Hermes SoftLab  
Wojtek Kozaczynski, Rational Software Corp.  
Tomoo Matsubara, Matsubara Consulting  
Masao Matsumoto, Univ. of Tsukuba  
Dorothy McKinney, Lockheed Martin Space Systems  
Nancy Mead, Software Engineering Institute  
Stephen Mellor, Project Technology  
Susan Mickel, AgileTV  
Dave Moore, Vulcan Northwest  
Melissa Murphy, Sandia National Laboratories  
Kiyoh Nakamura, Fujitsu  
Grant Rule, Software Measurement Services  
Girish Seshagiri, Advanced Information Services  
Chandra Shekaran, Microsoft  
Martyn Thomas, Praxis  
Rob Thomsett, The Thomsett Company  
John Vu, The Boeing Company  
Simon Wright, Integrated Chipware  
Tsuneo Yamaura, Hitachi Software Engineering

MAGAZINE OPERATIONS COMMITTEE

George Cybenko (chair), James H. Aylor, Thomas J. Bergin, Frank Ferrante, Forouzan Golshani, Rajesh Gupta, Steve McConnell, Ken Sakamura, M. Satyanarayanan, Nigel Shadbolt, Munindar P. Singh, Francis Sullivan, James J. Thomas

PUBLICATIONS BOARD

Rangachar Kasturi (chair), Jean Bacon, Mark Christensen, George Cybenko, Gabriella Sannitti di Baja, Lee Giles, Thomas Keefe, Dick Kemmerer, Anand Tripathi