

## Best Practices: Who Says?

**Warren Harrison**

**W**e often hear the term “best practices” used in software development contexts. Since I’ve been *IEEE Software*’s editor in chief, I’ve seen the phrase in many submissions, a few of which have even proudly proclaimed in the title that they were presenting a best practice. Like countless others in the development community, I’d fallen into the habit of using the term indiscriminately for any activity that looked even remotely useful.



However, over the past several years, I’ve taken a serious interest in computer forensics, including looking for better ways to extract criminal evidence from suspects’ hard drives. In a lot of ways, the problems are analogous to software engineering issues—a mixture of process and technology. Because of this similarity, I was surprised to find many of my law enforcement colleagues horrified by the idea of labeling something a “best practice.” It turns out that in a criminal trial (at least in the US), the failure to follow an established best practice in an investigation could result in an acquittal. I learned it’s far safer to avoid the term when discussing techniques investigators might use.

### What does it matter?

In software development, using (or failing to use) a best practice could have numerous ramifications. For instance, if your software fails, customers sue, and your organization

didn’t follow best practices, is that negligence? If your organization did follow best practices, is that a defense? Should customers select Company A’s products over Company B’s because Company A follows best practices? What exactly *is* a best practice?

The term “best practice” has several definitions, but the American Society for Quality’s definition probably comes closest to what we in software development mean when we use the term. The ASQ defines “best practice” as *a superior method or innovative practice that contributes to the improved performance of an organization, usually recognized as “best” by other peer organizations.*

### Problems with best practices

First of all, I’d like to state that I’m not fundamentally opposed to using the term “best practice” or its significance in establishing due care or negligence. I’m not even against using it to distinguish between competing products; in fact, I’d be very much for it. However, it’s that last phrase in the ASQ definition that troubles me: *usually recognized as “best” by other peer organizations.* Although I think this phrase is important (in fact, essential) in establishing something as a best practice, it raises serious issues about how we determine what’s a best practice to begin with:

■ *Exactly who are these “peer organizations”?* Are they peer organizations simply because they develop software? Or do they have to be in the same general industry? What about size? Do Motorola’s peers in-

clude a 10-person start-up? Do the peers of an organization that produces Web content include NASA as well as Podunk Savings and Loan?

- *How many peer organizations have to recognize the practice as best?* Must it be a certain number or percentage? If it's a percentage, how do we know how many peer organizations exist?
- *Who's responsible for collecting data about which peer organizations recognize the practice and which don't?* Consultants? Researchers? Academics? Magazine editors? I was at a recent conference in which the keynote speaker (an academic) proposed that a special board of notable researchers be formed to determine and publish a list of best practices. In principle, this sounds like a pretty good idea. But few researchers are purely observers—most are (or at least try to be) change agents. If change agents sit on the board, which is likely, they'll no doubt innocently record observations about organizations they've influenced or tried to influence. This seems to guarantee a skewed observation.
- *How does a peer organization recognize a best practice?* Does simply performing an activity make it a best practice? If other peer organizations don't perform the activity, does that imply that *not* performing it should be a best practice? For example, some companies use inspections and some don't. If performing an activity is the only requirement, we could end up with two best practices: using inspections and not using inspections.
- *What about organizations in which some units use a particular practice and others don't?* Must an organization always use a particular practice for it to be recognized? My experience leads me to believe that large gaps often occur between different business units' practices (and often even in projects within a given business unit). Certainly an organization's use of almost any software de-

velopment practice will differ depending on whom you ask.

- *And what about conflicting best practices?* CMM versus agile? Middleware versus native OS support? J2ME versus WAP? Objects versus datastores? Surely not everyone agrees on every good idea ever introduced in the software development community.

### Good ideas and best practices

The fact is that we really don't have a mechanism to formally establish best practices in the software development community. Of course, plenty of practices exist that most, if not everyone in the community, think are good ideas. But when does a good idea become a best practice? Frankly, I don't know. And, if "best practices" were just another slick advertising term that folks like to throw around, I wouldn't care. However, I fear that one of these days, adherence to best practices will mean the difference between negligence and responsible development in a lawsuit. We'd better have an answer by then.

As editor in chief of a fairly influential magazine dealing with software de-

velopment, I consider this a serious question. If readers see an article, in this or any other publication, that claims an activity is a best practice, it's reasonable for them to think the activity is *recognized as "best" by other peer organizations*.

Because of this, I've begun restricting use of the term "best practice" in articles that we publish in *IEEE Software*. In most cases, "good practice" or "effective practice" serve as acceptable substitutes. I hope other editors consider similar measures for their publications. At the same time, I would welcome attempts to formalize its meaning for software development. Until we get there, be wary about using the term indiscriminately. It might come back to bite you or your company.

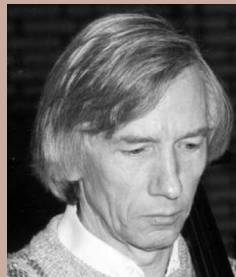
### What do you think?

I'd like to hear your thoughts on best practices. What does it take to convince you that something is a best practice? Do you think we have any best practices (using the ASQ definition) in the community today? Please write me at [warren.harrison@computer.org](mailto:warren.harrison@computer.org).

## Goodbye ... Hello

The Hawaiian greeting "aloha" is an acknowledgment that means hello or goodbye. I'd like to take this opportunity to say aloha to some old friends as well as some new ones.

Maarten Boasson, associate editor in chief for design, and Jeffrey Voas, associate editor in chief for quality, have completed their terms of service with *IEEE Software's* Editorial Board. They've both been instrumental in making *IEEE Software* what it is, and we all owe them a hearty thank



Maarten Boasson



Jeffrey Voas

you. But, in keeping with the aloha theme, it really isn't just "goodbye"; it's also "hello" because they're continuing with us as members of our Industrial Advisory Board.

### DEPARTMENT EDITORS

**Bookshelf:** Warren Keuffel, wkeuffel@computer.org  
**Construction:** Andy Hunt and Dave Thomas, {andy, dave}@pragmaticprogrammer.com  
**Design:** Martin Fowler, fowler@acm.org  
**Loyal Opposition:** Robert Glass, rglass@indiana.edu  
**Manager:** Don Reifer, d.reifer@ieee.org  
**Quality Time:** Nancy Eickelmann, nancy.eickelmann@motorola.com, and Jane Hayes, hayes@cs.uky.edu  
**Requirements:** Suzanne Robertson, suzanne@systemsguild.com

### STAFF

Senior Lead Editor  
**Dale C. Strok**  
 dstrok@computer.org

Group Managing Editor  
**Crystal Shif**  
 Senior Editors  
**Shani Murray and Dennis Taylor**  
 Assistant Editors  
**Rebecca Deuel and Denise Kano**  
 Editorial Assistant  
**Joan Shim**  
 Magazine Assistant  
**Pauline Hosillos**, software@computer.org  
 Art Director  
**Toni Van Buskirk**  
 Cover Illustration      Technical Illustrator  
**Dirk Hagner**              **Alex Torres**  
 Production Assistant      Production Artist  
**Monette Velasco**        **Carmen Flores-Garvey**  
 Executive Director  
**David Hennage**  
 Publisher              Assistant Publisher  
**Angela Burgess**        **Dick Price**  
 Membership/Circulation Marketing Manager  
**Georgann Carter**  
 Business Development Manager  
**Sandra Brown**  
 Senior Production Coordinator  
**Marian Anderson**

### CONTRIBUTING EDITORS

Candace English, Robert Glass, Anne Lear, Rita Scanlan, Keri Schreiner, Joan Taylor

**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:** Access the IEEE Computer Society's Web-based system, Manuscript Central, at <http://cs-ieee.manuscriptcentral.com/index.html>. Be sure to select the right manuscript type when submitting. Articles must be original and not exceed 5,400 words including figures and tables, which count for 200 words each.

I'd also like to say aloha to two new Editorial Board members: Philippe Kruchten, who joins us as associate editor in chief for design, and Stan Rifkin, who will serve as associate editor in chief for quality.

Philippe Kruchten is a professor of software engineering at the University of British Columbia in Vancouver, Canada. He has 28 years of industrial experience developing large-scale software-intensive telecommunications, aerospace, defense, and transportation systems. He spent 17 years at Rational Software, now part of IBM, where he led the development of the Rational Unified Process, a Web-based, generic software development process. He wrote three books on the RUP and created a model for representing software architecture based on multiple coordinated views, which led to an IEEE standard. He coauthored the Object Management Group's Software Process Engineering Metamodel, an industry standard for process modeling. He also represented Rational on the industry advisory board of the Software Engineering Body of Knowledge project. As a member of the International Federation for Information Processing Working Group 2.10 on software architecture, he leads the steering committee for the Working IEEE/IFIP Conferences on Software Architecture.

Philippe received his diploma in mechanical engineering from École Centrale de Lyon, France; his doctorate in information systems from École Nationale Supérieure des Télécommunications in Paris; and his certificate in intercultural studies from the University of British Columbia. Find out more about Philippe at <http://philippe.kruchten.com> or at [www.computer.org/software/experts](http://www.computer.org/software/experts).



**Philippe Kruchten**



**Stan Rifkin**

Stan Rifkin is known throughout the software engineering community. He is a principal at and the founder of Master Systems, a software consulting company.

Stan helps clients implement engineering and management process improvements. He also helped establish and participated in several software engineering process groups and software process improvement network groups.

Stan cowrote (with Priscilla Fowler) *Software Engineering Process Group Guide*, considered a seminal work on how to establish and sustain a SEPG and related software engineering process improvements. The guide has had major influence on the proliferation of grassroots SPINs around the world. He also cowrote *Measurement in Practice*, a study of the US's best metrics practices, and recently wrote "Why New Software Processes Are Not Adopted" for *Advanced Computers*. Rifkin previously served on the editorial board of *Empirical Software Engineering*.

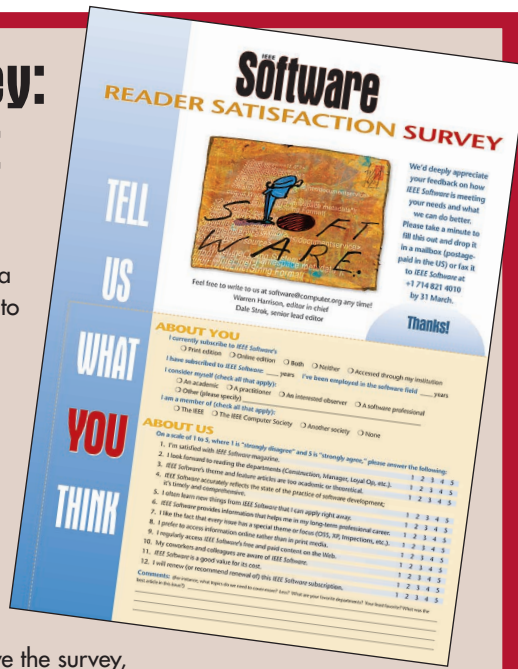
He earned his BS in business administration (quantitative methods) from California State University at Northridge and his MS in computer science from the University of California at Los Angeles. He also did PhD-oriented postgraduate study in applied science and engineering at UCLA and the University of Geneva, Switzerland. He is currently a doctoral fellow in executive leadership at George Washington University. You can find out more about Stan at [www.master-systems.com/Stan.ivnu](http://www.master-systems.com/Stan.ivnu) or at [www.computer.org/software/experts](http://www.computer.org/software/experts).

To all of these individuals ... aloha! 🍻

# Reader Survey: Tell Us What You Think!

IEEE Software is published as a service to its readers. We'd like to know how well we're doing. If you receive the print version of the magazine, you'll notice that this issue includes a postage-paid, self-addressed survey. Please take the time to fill it out and send it back to us. We can tailor the magazine to your interests only if you give us feedback. If you subscribe to only the online version and don't have the survey, email us at [software@computer.org](mailto:software@computer.org), and we'll make sure you get a copy. Look for this and other surveys in the future at [www.computer.org/software](http://www.computer.org/software).

—Warren Harrison, editor in chief



EDITOR IN CHIEF:  
Warren Harrison

10662 Los Vaqueros Circle  
Los Alamitos, CA 90720-1314  
[warren.harrison@computer.org](mailto:warren.harrison@computer.org)

EDITOR IN CHIEF EMERITUS:  
Steve McConnell, Construx Software  
[stevemcc@construx.com](mailto:stevemcc@construx.com)

**ASSOCIATE EDITORS IN CHIEF**

**Education and Training:** Don Bagert, Rose-Hulman Inst. of Technology; [don.bagert@rose-hulman.edu](mailto:don.bagert@rose-hulman.edu)

**Design:** Philippe Kruchten, University of British Columbia; [kruchten@ieeeg.org](mailto:kruchten@ieeeg.org)

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

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

**Quality:** Stan Rifkin, Master Systems; [sr@master-systems.com](mailto:sr@master-systems.com)

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

**EDITORIAL BOARD**

- Nancy Eickelmann, Motorola Labs
- Richard Fairley, Oregon Graduate Institute
- Martin Fowler, ThoughtWorks
- Jane Hayes, University of Kentucky
- Andy Hunt, Pragmatic Programmers
- Warren Keuffel, independent consultant
- Karen Mackey, Cisco Systems
- Deependra Moitra, Infosys Technologies, India
- Don Reifer, Reifer Consultants
- Suzanne Robertson, Atlantic Systems Guild
- Richard H. Thayer, Calif. State Univ. Sacramento
- Dave Thomas, Pragmatic Programmers

**INDUSTRY ADVISORY BOARD**

- Robert Cochran, Catalyst Software (chair)
- Dave Aucsmith, Microsoft
- Maarten Boasson, Quaerendo Invenietis
- Annie Kuntzmann-Combelles, Q-Labs
- David Dorenbos, Motorola Labs
- Enrique Draier, MAPA LatinAmerica
- David Hsiao, Cisco Systems
- Takaya Ishida, Mitsubishi Electric Corp.
- Dehua Ju, ASTI Shanghai
- Donna Kaspersen, Science Applications International
- Pavle Knaflic, Hermes SoftLab
- Wojtek Kozaczynski, Microsoft
- Tomoo Matsubara, Matsubara Consulting
- Masao Matsumoto, Univ. of Tsukuba
- Dorothy McKinney, Lockheed Martin Space Systems
- Stephen Mellor, Project Technology
- Susan Mickel, Lockheed Martin
- Dave Moore, Vulcan Northwest
- Melissa Murphy, Sandia National Laboratories
- 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
- Jeffrey Voas, Cigital

**MAGAZINE OPERATIONS COMMITTEE**

- Jean Bacon (chair), Thomas J. Bergin, Pradip Bose, Doris L. Carver, George Cybenko, John C. Dill, Frank E. Ferrante, Robert E. Filman, Forouzan Golshani, Rajesh Gupta, Warren Harrison, Mahadev Satyanarayanan, Nigel Shadbolt, Francis Sullivan

**PUBLICATIONS BOARD**

- Rangachar Kasturi (chair), Jean Bacon, Laxmi Bhuyan, Mark Christensen, Thomas Keefe, Deependra Moitra, Steven L. Tanimoto, Anand Tripathi

**New!**

## Software Engineering Overview:

### Preparing for the IEEE Computer Society CSDP Examination

Expand your knowledge and advance your career with a new online training course created exclusively for software professionals.

This interactive course offers 10 modules that cover all 11 knowledge areas of the CSDP exam. Key concepts and principles of software engineering are presented with clear and concise explanations. Plus the course provides a list of references to aid further study for each area.

**Register Now**

For more information or to enroll for  
*Software Engineering Overview*, visit  
[www.computer.org/certification/distancelearning](http://www.computer.org/certification/distancelearning)

