;login:

THE MAGAZINE OF USENIX & SAGE
April 2002 • Volume 27 • Number 2

USENIX & SAGE
The Advanced Computing Systems Association &
The System Administrators Guild

the bookworm

by Peter H. Salus

Peter H. Salus is a member of the ACM, the Early English Text Society, and the Trollope Society, and is a life member of the American Oriental Society. He is Editorial Director at Matrix.net. He owns neither a dog nor a cat.

peter@matrix.net

BOOKS REVIEWED IN THIS COLUMN

TELECOMMUNICATIONS ESSENTIALS
LILLIAN GOLENIEWSKI
Boston: Addison-Wesley, 2002. Pp. 582.
ISBN 0-201-76032-0.

WIRELESS INTERNET
MARK BEAULIEU
Boston: Addison-Wesley, 2002. Pp. 632.
ISBN 0-201-73354-4.

RUBY IN A NUTSHELL
YUKIHIRO MATSUMOTO
Sebastopol, CA: O'Reilly & Associates, 2002.
Pp. 204. ISBN 0-596-00214-9.

WEB CONTENT MANAGEMENT
RUSSELL NAKANO
Boston: Addison-Wesley, 2002. Pp. 238.
ISBN 0-201-65782-1.

MAC OS X: THE MISSING MANUAL
DAVID POGUE
Sebastopol, CA: O'Reilly & Associates, 2002.
Pp. 583. ISBN 0-596-00082-0.

TROUBLESHOOTING TECHNIQUES OF THE SUCCESSFUL TECHNOLOGIST
STEVE LITT
NP: Troubleshooters.Com, 2001. Pp. 313.

In the February column, I lauded Rich Morin's publication effort, DOSSIER (Documenting Open Source Software for Industry, Education, and Research). At that time, I'd looked at the *Mail and Sendmail* volume. I've now looked at *Text* (which covers both RedHat and FreeBSD commands, as well as the standbys: groff (troff), eqn, pic, and tbl) and *Kernel* (which covers the FreeBSD kernel). Keep 'em comin', Rich!

Collect 'em all from: *www.ptf.com/ptf/dossier/*.

Telecoms

Goleniewski's book was a disappointment to me. It is nicely written but superficial to a point that I began looking for sections about stuff I knew, only to be disappointed. Its glibness can result in distortions, as in the brief paragraphs on the history of the Internet. There are no references nor is there a bibliography. Each chapter ends with a reference to a Web site. When I went there, it demanded registration information. I consider this a blatant marketing ploy (Ms Goleniewski runs a "seminar and e-learning" business) and didn't bother. Despite its title, the volume contains too few "essentials" for me to recommend it to anyone below the level of corporate vice president.

Unwired

Beaulieu's volume on wireless, on the other hand, was both interesting and valuable. Beaulieu discusses the concept of the wireless Internet, the language used in the discussion, and the basic concepts involved. His exposition of WAN, LAN, and PAN standards is quite good. He also talks about e-commerce and m-commerce (mobile commerce) servers. Read together with Wheeler, Milojicic, & Douglis, *Mobility* (1999) and Solomon's *Mobile IP* (1997), this yields a really good picture of where we're likely to go.

Ruby

I read *Ruby in a Nutshell* with great interest. I thought that Ruby was an interesting attempt at creating a language that, like the Second City's WFMFMT had "the best of everything worthwhile." Matsumoto has pulled together the features of a number of other scripting languages. Unfortunately, I think that the history of programming languages has shown us that being good is not the same as being successful. So, I think that Ruby most likely won't make it, because it's not enough better to make users of Java, Perl, and/or Python want to switch. (Ray Schneider differs with me here, and I have requested a full review of O'Reilly's's Ruby book from him to give readers a more "balanced" view.)

Web Stuff

Nakano's volume is highly nontechnical. But I think that it will prove useful to those who are trying to manage their Web sites at the same time the content is waxing at a furious pace. My major problem with the book was that so many items in the "executive summary" sections sounded like platitudes. (This may be my fault, for expecting content in something containing "executive"; it may be a real benefit, as Nakano recognized, that executives read only summaries; I'm not sure.) However, I think that the section in Chapter 3, on versioning, needs references to SCCS, RCS, and their friends.

book reviews

MacOS X

About 15 years ago, in Phoenix, I complained to Steve Jobs that I had found Mac "manuals" fairly useless where trying to actually locate information was concerned. Steve told me I wasn't supposed to "look things up" but to "just mouse around."

Well, if you are using a Mac and running OS X, you didn't find a book in the box. Pogue has written a big, fat manual that a user will find indispensable. I have a devoted colleague, running a cube, who nearly tore the book from my hands when it arrived.

I think it's really a fine job.

Troubleshooting

My guess is that no matter what aspect of computing you're involved in, you end up spending time troubleshooting. Litt has produced (in some sense) a really fine volume to serve as a guide.

Litt points out that most technologists are employed to solve technical problems and/or design new technology. Solving problems is "pure troubleshooting." Right. We all know that. But Litt goes on, outlines a respectable methodology, and provides a system which should enable the intelligent troubleshooter to work efficiently.

I enjoyed reading the book. Its single drawback is that it is 8.5 x 11 inches and stapled. Maybe there's an opportunity for a "real" publisher here. It's $42.50 and available from *http://troubleshooters.com/bookstore*.

A Concluding Note

When I look at the number of books I get about the Web or XML or Java or Perl or . . ., it makes me wonder just why most publishers are afflicted with rampant me-too-ism. In a more sensible industry, neither Morin nor Litt wouldbe in the publishing or distribution industries. And there might be fewer books on how to design Web pages or use Office 2000.

AGILE SOFTWARE DEVELOPMENT

Alistair Cockburn

Boston: Addison-Wesley, 2001. Pp. 304.
ISBN 0-201-49834-0

Reviewed by Raymond M. Schneider
ray@hackfoo.net

Get ready, we are going to start hearing more and more about agility when it comes to software development processes. *Agile Software Development* is one of two books anchoring an up-and-coming series.

Have you ever looked at your development process and thought it was bloated? Have you ever noticed people in projects suffering from argh-minutes? What is an argh-minute? Cockburn introduces us to the term "as the unit of measure for frustrating communication sessions." *Agile Software Development* brings the reader up to speed about adapting agility in development methodologies.

Agile Software Development is seemingly broken down into two distinct parts. The first three chapters acquaint the reader with the terminology and concepts that are being discussed. Levels of listening, failure and success modes of individuals, and convection currents of information are all core ideas developed in the first part of *Agile Software Development*. There are three levels of listening: following, detaching, and fluent. These levels are referred to by Cockburn throughout *Agile Software Development* in sections where readers' focus may differ depending on their listening level.

Methodologies, sweet spots, self-adaptation, and Crystal methods are discussed in the second half of *Agile Software Development*. Methodology is defined and explained in depth in Chapter 4. Extreme programming is examined, and ways to adjust it to the needs of a project are introduced. "Sweet spots" are characteristics of agile methods that work best. Cockburn gives examples of five different sweet spots that affect an agile process. How to become self-adapting is another area in which Cockburn offers ideas about how a group might adapt processes to their projects. A starter methodology is given and ways to adapt it are discussed. The Crystal family of methods is offered by Cockburn in the final chapter. These methods exemplify how he has solved problems in methodology design in the past.

Most chapters contain a concluding section called, "What Should I Do Tomorrow?" This section is used by Cockburn as a place to make suggestions to the reader about how they might notice the subjects covered in that chapter in their group, in the work place.

Agile Software Development is excellent and proved to be fun to read. Cockburn has done a wonderful job introducing the reader to adaptation of agile methods of software development.