JOURNAL OF OBJECT TECHNOLOGY

Online at www.jot.fm. Published by ETH Zurich, Chair of Software Engineering ©**JOT, 2002**

Vol. 2, No. 1, January-February 2003

Books

My Best Books of the Year 2002

An overview by Charles Ashbacher

As the year 2002 moves nearer to completion, we can look back on a series of tumultuous events, both in and out of the IT arena. There is an ongoing war against terrorism, several hot wars that continue, and the threat of war against Iraq, which could involve the use of biological and chemical weapons. In the business arena, there is an ongoing sluggish American economy, with massive fraud, continued uncertainty and yet steady growth in certain areas. In this tight job market, your key to success is having an extensive set of skills, which means that now, more than ever, it is necessary to continue to examine and learn new technologies. In this column, I will describe those books that I consider most helpful to you in your quest.

Of all the books that had my fingerprints placed on an internal page in the last year, there was no book that elicited a WOW! reaction from me. Therefore, this list of books will not contain any that I consider a reading requirement. As always, the criteria for selection is that it is a book that I encountered and I make no claim to having performed an exhaustive search of all possibilities.

Computer programmers are not known as being terribly communicative, except about geek stuff such as the latest Star Wars movie. This is unfortunate, as the building of quality software requires that everyone concerned practice the most precise forms of communication possible. From the customer to the most reclusive programmer, there must be no unnecessary ambiguity in what they understand concerning the goals. That is of course not the case, and in her book, **Communication Gaps and How to Solve Them**, Naomi Karten describes many ways that ambiguity and uncertainty can enter and be removed from our messages. This is a book that will improve all processes at the basic level, by raising the quality of communication between humans. Her solutions require no new software or hardware, just the time and effort to make sure that all messages are exactly what they appear and are meant to be.

The book **The Manager Pool: Patterns for Radical Leadership** by Don Sherwood Olsen and Carol L. Stimmel is included because of the radical nature of the ideas rather than any belief on my part that it will dramatically change the methods of constructing software. Most new ideas are met with justified skepticism and then if the idea proves to be good enough, it eventually is taken seriously and adopted. The main premise of this

Cite this book review as follows: Charles Ashbacher: *My Best Books of the Year 2002*, in Journal of Object Technology, vol. 2, no. 1, January-February 2003, pages 123-125. http://www.jot.fm/books/best2002

book is that the developer team is formed first and then the group chooses their manager from a pool of those available. While I doubt that it would work as a general practice, it could be applied to some selected cases, and there is enough doubt in my doubts to warrant considering this form of managerial selection as a possible solution to some of the problems that arise in software development.

Bernd Oestereich has once again created an excellent text on the practical usage of UML in his second edition of the book, **Developing Software with UML: Object-Oriented Analysis and Design in Practice**.Unlike some of the other books that describe only the UML, he applies it in detail to a project, so you can see the UML in action rather than just in theory.

It is absurd to think that it has taken terrorist actions to place security to the front and center of the minds of a majority of computer professionals. There has always been a small group of people who have been trying to get security taken more seriously, but to date they have been vastly outnumbered by the number of people trying to damage systems. Over the last year, I have encountered many books on computer security, and the best of that group was **Building Secure Software: How to Avoid Security Problems the Right Way** by John Viega and Gary McGraw. What made this book stand out was the level of technical detail and the explanations of how code can be altered to improve the security features. It made such an impression on me that I was motivated to write a proposal for a special topics course in computer security. That course was recently approved and at this point, this is the text I will use.

When discussing hot issues among groups of programmers, the primary one mentioned is usually about the style of a program. However, there is one that is potentially more politically explosive, and that is the role of peer examination of software. Many, if not most software shops have solved the problem of code examination by creating separate testing groups that are often also physically isolated from the developers. In his book, **Peer Reviews in Software: A Practical Guide**, Karl E. Wiegers notes that properly done, peer reviews of software can be more cost effective than any other form of bug hunting. He then describes many of the problems and pitfalls as well as how to avoid them. In my experience, peer reviews often lead to enormous problems, but he did convince me that it is possible to conduct them without doing more bad than good.

In the area of reference books, three have been placed on my “special shelf” of books that are within arms reach of my main workstation. **Internet Security Dictionary** by Vir V. Phoha is a dictionary of computer security terms that includes a CD ROM with a searchable version of the dictionary. As I began the serious exploration of the principles of computer security, I found it an excellent and immediate resource for all of the terms that I encountered. The other two reference books are **The Java Developers Almanac 1.4 Volumes 1 and 2** by Patrick Chan. As the Java language continues to expand in size and scope, such references are invaluable. I have taken the previous version with me every time I taught a Java class so that I was ready when a student asked a question of the form “What class(method) does ……?” These two are just as good as the previous edition and they now travel with me to all my Java classes.

Books cited in this article:

Communication Gaps and How to Close Them by Naomi Karten, Dorset House Publishing, 2002. ISBN 0-932633-53-6.

The Manager Pool: Patterns for Radical Leadership by Don Sherwood Olsen and Carol L. Stimmel, Addison-Wesley, 2002. ISBN 0-201-72583-5.

Developing Software with UML: Object-Oriented Analysis and Design in Practice Second Edition by Bernd Oestereich, Addison-Wesley, 2002. ISBN 0-201-75603-X.

Building Secure Software: How to Avoid Security Problems the Right Way by John Viega and Gary McGraw, Addison-Wesley, 2002. ISBN 0-201-72152-X.

Peer Reviews in Software: A Practical Guide, by Karl E. Wiegers, Addison-Wesley, 2002. ISBN 0-201-73485-0.

Internet Security Dictionary by Vir V. Phoha, Springer-Verlag, 2002. ISBN 0-387-95261-6.

The Java Developers Almanac 1.4 Volume 1: Examples and Quick Reference by Patrick Chan, Addison-Wesley, 2002. ISBN 0-201-75280-8.

The Java Developers Almanac 1.4 Volume 2: Examples and Quick Reference by Patrick Chan, Addison-Wesley, 2002. ISBN 0-201-76810-0.