JOURNAL OF OBJECT TECHNOLOGY

Online at http://www.jot.fm. Published by ETH Zurich, Chair of Software Engineering ©**JOT, 2004**

Vol. 3, No. 1, January-February 2004

Books

My Best Books of the Year 2003

An overview by Charles Ashbacher

In the past year, the major international news has of course been the war in Iraq. At this time, things do not look good, and there seems to be no scenario where American involvement can end soon. Inside the United States, another major conflict is brewing, one whose long term impact on American society may be greater than the war in Iraq. That conflict is outsourcing, where jobs in I.T. are moving out of the United States so fast that some observers are now raising national security concerns about how much development is being sent to countries other than the United States. Whether you favor it or not, the simple fact is that it is here to stay. The economic advantages of having high quality developers producing software for one tenth the cost of a U.S. programmer are too great to ever be prevented by legislation or labor organization.

If you are a U. S. developer and are facing the newfound challenges of global competition, the first thing to understand is that you do have advantages and much of your future is still under your control. In the U.S., your biggest advantage is your location. There are substantial quality control and security issues in outsourcing software, so if you can make yourself competitive in other ways, it is possible to win the competition. Keep in mind that many applications still cannot be exported outside the Unites States and Canada.

To keep yourself competitive, you must constantly be learning, and that demonstrates your second advantage. The educational system of the United States is still the envy of the world, as demonstrated by the large number of people who move here to study. The educational opportunities are there, you must have the drive and determination to take advantage of them. If you are not a U. S. developer, then your primary concerns are similar to those in the U. S., namely to keep your skills at a high level. All of the books on development that I have identified as the best of the year are those that I consider required reading for anyone wishing to maintain their personal, professional edge.

The most interesting book that I have read in the past year is **Malware: Fighting Malicious Code**, by Ed Skoudis and Lenny Zeltser and published by Prentice-Hall. It is both frightening and comforting to read about the enormous number of ways in which our systems are vulnerable. The ease with which some attacks can be launched against

Cite this book review as follows: Charles Ashbacher: “My Best Books of the Year 2003”, in *Journal of Object Technology*, vol. 3, no. 1, January-February 2004, pages 177-180.
http://www.jot.fm/books/best2003

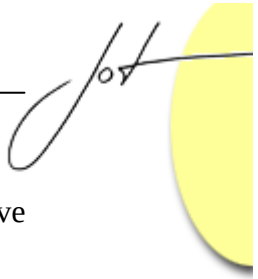
computer systems certainly raises concerns, but the solutions are often obvious. Most of the really damaging attacks were effective only against systems where the available and appropriate protection(s) had not yet been installed. From this, it is clear that some form of efficient patch management process is a sorely needed application. It is also comforting to understand that security workers are well-aware of the common modes of attack and have created appropriate defenses. If I were the manager of any group of computer users, I would mandate that they read a chapter of this book a week until done, even if it had to occur on company time.

The second edition of **Death March**, by Edward Yourdon and published by Prentice Hall is another of the "survivor" books. A software development project that stretches people beyond the point of exhaustion without concomitant rewards is the classic description of a death march. Yourdon emphasizes the social, political and psychological aspects of such projects, how they get started, how a "normal" project gets morphed into a death march and how to identify and survive them. He quite rightly places much of the blame for death march projects on developers, who optimistically believe that they can accomplish anything, if they want it bad enough. In many ways, as I read the book I started wondering if large numbers of death march projects are a necessary filter that helps drive the enormous technological strides that have been made. The situation of being pushed to the limit and sometimes succeeding but often failing is an act of technological Darwinism, and ferocious competition is where the most dramatic evolutionary change takes place.

Addison-Wesley has published the third edition of Martin Fowler's now classic work**, UML Distilled: A Brief Guide to the Shandard Object Modeling Language**. What is surprising is that although the UML has grown since the previous editions were published, Fowler has resisted the temptation to add pages. In fact, the third edition is ten pages smaller than the second. However, there is no compromise in quality, as he once again captures what is the most significant subset of what is becoming a monstrosity.

Another Martin Fowler book that I consider essential is **Patterns of Enterprise Application Development**, published by Addison-Wesley. In simple terms, an enterprise application is a big project with a large number of semi-distinct parts that must work together. Put another way, the programs that run most of the world economy. The patterns in the book are generally "in the small" describing how one carries out details rather than answering the large architectural questions. Nevertheless, they are essential to making these hulks work, and anyone building an enterprise application can save themselves a lot of effort by looking up the solutions before creating their own.

Significant software development is all about taking and managing risks, and in the book, **Waltzing With Bears: Managing Risk On Software Projects**, by Tom DeMarco and Timothy Lister and published by Dorset House, the risks are catalogued and quantified. This is about making calculated gambles and keeping yourself and the project under control as it moves from the original idea to completion, and their mantra is, "If there is no risk in your next project, then don't do it." They also describe their free risk assessment tool, RISKOLOGY, which will help you organize and describe the risks



unique to your project. Buy a copy and give it to your manager as a gift, even if you have to give it to yourself.

As an educator, I was particularly impressed by the book, **Programming Challenges: The Programming Contest Training Manual**, by Steven S. Skiena and Miguel A. Revilla. It is a list of programming problems that can be used to prepare students for programming contests. The solutions can be sent to a robot judge at http://www.programming-challenges.com for validation of the coded solution. In looking over the problems, I identified several that I will use as assignments the next time I teach a programming class. The problems are placed in several categories, including sorting, combinatorics, number theory, graph traversal, grid operations and geometry.

In the area of reference books, Addison-Wesley has published **The Unicode Standard 4.0** by The Unicode Consortium. The name is self-explanatory and it is on my reference shelf right next to my dictionaries and other quick reference manuals. It is heavy enough to serve as exercise equipment, which emphasizes the thorough coverage of the standards. I consider it an essential component of my reference library and no academic library should be without it.

While most developers will not have a use for it, those who code mathematical formulas will find **Mathematical Constants**, by Steven R. Finch and published by Cambridge University Press to be invaluable. I was originally trained as a mathematician decades ago and have worked in many areas of the field. And yet, I was amazed at the number of constants there are. The coverage of each is brief, yet thorough and it is the first reference I consult when there is a problem coding mathematical formulas.

In closing, I would like to mention an event in technology. I consider the single greatest technological event of the last year to be the recent construction of one of the five fastest computers in the world, done by networking off the shelf computers produced by Apple. While the cost was still over five million US$, this does herald a new age in computing. It shows us that our abilities to have computers communicate for collaboration has grown to the extent that we can now consider many computers to be one. This indicates that distributed computing on demand is a feasible proposition, opening up many possibilities for collaborative production. I would be willing to bet that one or more of my best books for next year will be about programming clusters of computers to perform as a single entity.

Books mentioned in this article:

Malware: Fighting Malicious Code, by Ed Skoudis and Lenny Zeltser, Prentice Hall, Upper Saddle River, New Jersey, 2003. 647 pp., $44.99(paper). ISBN 0-13-101405-6.

Death March Second Edition, by Edward Yourdon, Prentice Hall, Upper Saddle River, New Jersey, 2003. 230 pp., $29.99(paper). ISBN 0-13143635-X.

Patterns of Enterprise Application Development, by Martin Fowler, Addison-Wesley, Boston, MA, 2003. 533 pp. $49.99(hardback). ISBN 0-321-12742-0.

UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition, by Martin Fowler, Addison-Wesley, Boston, MA, 2004. 175 pp., $34.99(paper). ISBN 0-321-19368-7.

Waltzing With Bears: Managing Risk On Software Projects, by Tom DeMarco and Timothy Lister, Dorset House, New York, New York, 2003. 208 pp., $33.95(paper). ISBN 0932633609.

Programming Challenges: The Programming Contest Training Manual, by Steven S. Skiena and Miguel A. Revilla, Springer-Verlag, New York, New York, 2003. 359 pp., $32.95. ISBN 0-387-00163-8.

The Unicode Standard 4.0, by the Unicode Consortium, Addison-Wesley, Boston, MA, 2003. 1462 pp., $74.99(bardbound). ISBN 0-321-18578-1.

Mathematical Constants, by Steven R. Finch, Cambridge University Press, New York, NY, 2003. 602 pp., $95.00(hardbound). ISBN 0-521-81805-2.