JOURNAL OF OBJECT TECHNOLOGY

Online at http://www.jot.fm. Published by ETH Zurich, Chair of Software Engineering **©JOT, 2005**

Vol. 4, No. 1, January-February 2005

Books

My Best Books of the Year 2004

An overview by Charles Ashbacher, Charles Ashbacher Technologies, Hiawatha, IA, USA

Once again, another year has passed and it has again been a difficult one for people who work in IT. Jobs continue to disappear altogether or are outsourced, although that trend is beginning to slow and in some cases is even reversing. Many of the companies who rushed to outsource are finding that their actual savings are less than anticipated and in a few cases nonexistent. Some companies who have cut their staffs to the bone and beyond are finding that too many “optional” projects have been cut and are starting to begin the process of expansion. Although provable return on investment or ROI is still the mantra of the IT business, there is a palpable loosening of the frozen budgets and staff sizes.

However, as is always the case in IT, the jobs that exist now or are being created are not the same ones that left, even it was only a few years ago. To be hired for a new position in IT generally requires a broader set of skills and so, the only way to compete for those jobs is to acquire those skills. The percentage of person-hours in the development of a project devoted to the writing of new code continues to drop when compared to all other facets of the project. The books that I consider the best of the last year reflects these changing aspects of project development.

The book that I consider the best one that I read all year is **Fearless Change: Patterns For Introducing New Ideas**, by Mary Lynn Manns and Linda Rising and published by Addison-Wesley. People in IT have always known how rapid change is in the field and that is of course true. However, in many earlier cases, the changes have been new programming languages or new versions of the same language. While these changes are difficult to deal with, experienced people can handle them fairly easily. New languages tend to be similar to the old ones, for example C# is really not all that different from Java, so programmers simply learn the minimum they need to function.

The more recent changes are far more substantial and difficult to absorb, hence the value of this book. Where earlier, collaboration between programmers could be handled by a discussion in a meeting room, collaboration now is often between people halfway around the world and there are major cultural and language differences. Software developers must now know more about sound business practices in general and their

Cite this book review as follows: Charles Ashbacher: “My Best Books of the Year 2004”, in *Journal of Object Technology*, vol. 4, no. 1, January-February 2005, pages 183-186. http://www.jot.fm/books/best2004

business in particular. Manns and Rising describe fundamental methods that can be used to introduce change into an organization without sending it into a collective funk. Since this is the first step in the solution of most problems, I give this book the nod as the best book of the year.

The second book on my list is **Refactoring to Patterns** by Joshua Kerievsky and published by Addison-Wesley. In many ways, it is a natural successor to the previous, separate works on refactoring and patterns. Since both topics are now major forces in the software development community, it makes sense that there are many instances where the improvement in the code makes it more generic and maintainable. In other words, more like a software pattern.

It is a collection of examples presented in a style similar to the classic book by Martin Fowler, **Refactoring: Improving the Design of Existing Code**, also published by Addison-Wesley. The style is:

- A statement of the refactoring, generally using a UML diagram.
- A section on motivation, an explanation of why it is done.
- The mechanics of how to carry out the refactoring.
- At least one example of the refactoring. Using Java code, he starts with the initial code and steps through the changes until the refactoring is complete.

Next on my list is **UML for Mere Mortals**, by Robert A. Maksimchuk and Eric J. Naiburg, and published by Addison-Wesley. Written more for managers, who are of course just as mortal as the developers, it is not a compact, diagram-laden style of book. While it certainly will not be useful as a reference for the UML, for the person who wants to learn how the UML can be used in all phases of the software development cycle, there is none better.

The principles of agile modeling and how to use it to build very large projects are covered in **Agile Software Development in the Large** by Jutta Eckstein and published by Dorset House. While I am a fan of the agile process of software development, I have always had enormous doubts that the agile process could be applied to the largest projects. Eckstein did a great deal to convince me that the book is worth reading on page 14. The two sentences, “Although agile processes are treated as a panacea in a lot of circumstances, they often have to be adapted to the specific needs of the project. A defined agile process has to be regarded as a starting line, which later has to be adapted”, got me interested. The rest of the text served to convince me that her approach to agile development could indeed be scaled up to the largest projects.

Countless books have pounded on the principle that the key factor in software development success is the quality of the people working on it. Therefore, for many projects, the key step is the first one, hiring the very best people. If they are on the high end of the talent scale, then their specific set of skills at the time of the hire can be of secondary consideration. The book **Hiring the Best Knowledge Workers, Techies & Nerds: The Secrets & Science of Hiring Technical People**, by Johanna Rothman and published by Dorset House is invaluable in showing you how to do this critical task right. Every step, from developing the strategy to be used in hiring to the first day on the job for the new hire is covered.

With computer security becoming more critical with each passing day, I was pleased to open a copy of **Introduction to Computer Security** by Matt Bishop and published by Addison-Wesley. Last year, I taught a course in the basics of computer security and wrote my own material, as I could not find anything I considered appropriate. After reading this book, it is most unlikely that I will need to do that again. The basics are covered and if you are considering offering such a course, I recommend that you scan this book when you are making your textbook decision. It will also serve you well if you want to learn the material on your own.

Finally, in the area of reference books, Addison-Wesley has published the second edition of **The Unified Modeling Language Reference Manual** by James Rumbaugh, Ivar Jacobson and Grady Booch. Version 2.0 of the UML is now a factor in software development, so if you use the UML in your projects, this book is a mandatory element of your reference library. A CD containing a searchable version of the book is also included, with the topics connected via clickable links. This really makes it easier to use the book, as the dictionary portion is over 550 pages of definitions and tracking down all of the parts of a definition would involve a lot of page turning.

The past year was another one where the prospects for IT workers was bleak. However, if you have the right skills and attitude, you can find work. The use of computers is still expanding, so there is a great deal of work that still needs to be done. This list contains a set of books that will go a long way in helping you develop the right skills. Cultivating the right attitude is of course up to you.

Books mentioned in this article:

Fearless Change: Patterns For Introducing New Ideas, Mary Lynn Manns and Linda Rising, Addison-Weley, Boston, Massachusetts, 2005. 273 pp., $24.99 (hardbound). ISBN 0-201-74157-1.

Refactoring To Patterns, Joshua Kerievsky, Addison-Wesley, Boston, Massachusetts, 2005. 367 pp., $49.99 (hardbound). ISBN 0-321-21335-1.

Refactoring: Improving the Design of Existing Code, Martin Fowler, Addison-Wesley, Boston, Massachusetts, 1999. 431 pp., $39.95 (hardbound). ISBN 0-201-48567-2.

UML For Mere Mortals, Robert A. Maksimchuk and Eric J. Naiburg, Addison-Wesley, Boston, Massachusetts, 2005. 248 pp., $39.99 (paper). ISBN 0-321-24624-1.

Agile Software Development in the Large, Jutta Eckstein, Dorset House Publishing, New York, NY, 2004. 248 pp., $39.95 (paper). ISBN 0-932633-57-9.

Hiring the Best Knowledge Workers, Techies & Nerds: The Secrets & Science of Hiring Technical People, Johanna Rothman, Dorset House Publishing, New York, NY, 2004. 336 pp. ISBN 0-932633-59-5.

Introduction to Computer Security, Matt Bishop, Addison-Wesley, Boston, Massachusetts, 2005. 747 pp., $59.99 (hardbound). ISBN 0-321-24744-2.

The Unified Modeling Language Reference Manual, Second Edition, James Rumbaugh, Ivar Jacobson, Grady Booch, Addison-Wesley, Boston, Massachusetts, 2004. 752 pp. $64.99 (hardbound). ISBN 0-321-24562-8.