

# My Bookshelf Part III

Adrian Kosmaczewski

2005-11-20

This is the third and last part of the article “My Bookshelf”, with the list of the books that I recommend anyone in the software engineering field to read.

This last part has my preferred books about:

Project Management and Methodologies

Computing History and Industry Trends

As usual, all book references include links to their official websites, to Amazon.com as well as their ISBN, cover and list of authors.

Index

Books about Programming Languages, Platforms & Frameworks

Books about Software Architecture and Design, and Security

Books about Project Management and Methodologies, and Computing History and Industry Trends

## 4. Project Management and Methodologies

The previous categories had more to do with technology; this section has to do with people, and the way that they interact with technology. I think that these books (at least the first two) should be mandatory readings in any computer science course.

The Mythical Man-Month: Essays on Software Engineering

by Frederick P. Brooks, Jr. (ISBN 0201835959) <http://www.amazon.com/gp/product/0201835959/>

This book is considered as one of the most fundamental, and indeed it is one of the finest books ever written on the subject of software project management; as Brooks once joked, “They call this book the Bible of Software Engineering... and that’s because everybody reads it but nobody does anything about it!”. Brooks was project manager of the ill-fated OS/360 project in the sixties, after having worked in IBM since the mid fifties, and wrote this book in 1974. It makes a comprehensive review of all the problems that plagued his project, describing them in great detail and providing solutions and best practices. It

is an enlightening, short, easily readable and enjoyable book that is referenced almost everywhere. Heck, it even has its own Wikipedia entry...

In 1995 this book was re-edited as a Special 20th Anniversary edition, augmented with two more chapters written by Brooks; one of them, No Silver Bullet is a controversial paper that has caused turmoil in the industry, and countless responses to its main statement: “there will be no more silver bullets, i.e., there will be no more technologies or practices that will create a 10-fold improvement in software engineering productivity over 10 years”.

I cannot stress this too much; you are not a real project manager if you have not read this book (and applied its principles in your everyday work). Period. The sad thing of this is that none of my past project managers had even heard about this book, and even worse, most of them repeated the same mistakes that Brooks described. Why is not this book part of the official curriculum in universities?

Peopleware: Productive Projects and Teams

by Tom Demarco & Timothy Lister (ISBN 0932633439) <http://www.amazon.com/gp/product/0932633439/>

This one is another little gem; referenced almost everywhere, Peopleware is the result of almost ten years of research of two human resources consultants in New York. They have studied, from 1976 to 1986, hundreds of different software development companies, their human resources policies, the way they treat employees, the way they distribute office space, and condensed all of this information into a small but extremely valuable book, that provides a lot of pragmatic insight on how to build a successful software development team.

And most important, they make an underrated statement against open spaces. They say (and I fully agree with them) that open spaces are the #1 cause of failure of software projects. The lack of dedicated work spaces for programmers result in late deliveries, bad quality, high turnover, schedule slippages, and countless other problems.

Sadly (again) none of the HR guys Ive worked for has ever read this book (no exceptions), as is the case of all the project managers Ive had so far. I do believe that both The Mythical Man-Month and Peopleware should be the basis of any software management university degree.

Joel on Software: And on Diverse and Occasionally Related Matters That Will Prove of Interest to Software Developers, Designers, and Managers, and to Those Who, Whether by Good Fortune or Ill Luck, Work with Them in Some Capacity

by Joel Spolsky (ISBN 1590593898) <http://www.amazon.com/gp/product/1590593898/>  
<http://www.joelonsoftware.com/BuytheBooks.html>

It all started as a blog on software engineering. Joel worked as a project manager of the Excel team back at the beginning of the nineties, then worked for an internet startup called Juno and after that, he had enough of managers who had never read neither The Mythical Man Month nor Peopleware (and for more

reasons than that, actually), and decided to star his own business in New York: Fog Creek Software.

During that time, he wrote in his blog his insight on building a software company, on how to manage software teams, how to hire and deal with people, all in all a very deep and practical knowledge; in 2004, he merged his best articles into this excellent book that I highly recommend. Joels point of view is that of a doer; no big management theories, no deep knowledge needed beforehand; practical tips, spanning from project schedule to office space set-up. Things that you will find useful whether you work for small company or a multinational.

#### Facts and Fallacies of Software Engineering

by Robert L. Glass (ISBN 0321117425) <http://www.amazon.com/gp/product/0321117425/>

This is another treasure; Glass makes a summary in this book of 65 principles, 55 facts (such as “Software estimates are rarely corrected as the project proceeds”, “Modification of reused code is particularly error-prone” or “Better methods lead to more maintenance, not less”) and 10 fallacies (“You can’t manage what you can’t measure”, “Software needs more methodologies”, “Given enough eyeballs, all bugs are shallow”), that surround the task of software engineering; he destroys common urban myths, show misconceptions and rectifies common notions that many managers take for granted. Short book that I just recommend to any project manager.

#### Software Project Survival Guide

by Steve C. McConnell (ISBN 1572316217) <http://www.construx.com/survivalguide/>  
<http://www.amazon.com/gp/product/1572316217/>

This book and the next are both targeted to developers that have been promoted to the rank of project managers; this one, written by the author of Code Complete (a classic in my to-read list) gives useful insight in whats needed to achieve small projects; it gives a complete methodology (without any fancy name) targeted directly to not-so-large software projects. The descriptions are clear and concise, the book is well structured and it gives a good overview, and a good introduction, to the problems of project management and to how to solve them.

#### Leading a Software Development Team

by Richard Whitehead (ISBN 0201675269) <http://www.amazon.com/gp/product/0201675269/>

This one is the perfect complement to Software Project Survival Guide; also targeted to developers that suddenly face management tasks, Whitehead structured this book around 40 particular questions, one per chapter, so that the reader can randomly pick the answer for a particular question, without having to read the whole book. “I’ve just been made team leader of a new project, where do I start?”, “How can I stop my project from coming in late?” and “I’ve got someone on my team who’s a real problem, what should I do?” are some of the problems that the author tries to solve, and I have found so far that the answers provided are really pertinent.

## 5. Computing History and Industry Trends

Knowing the past helps build the future; personally I love to read the stories of how some ideas that we now consider for granted were created. It makes me wonder about the future developments, and helps me find trends in my job that would not be easy to spot otherwise. And, last but not least, it makes one understand that technology evolves quicker than human relationships do.

### The Success of Open Source

by Steven Weber (ISBN 0674012925) <http://www.hup.harvard.edu/catalog/WEBSUC.html>  
<http://www.amazon.com/gp/product/0674012925>

This book is my 2005 preferred one. I must admit not having yet directly collaborated in an open source project (even if I have myself added features to some open source code for some project of mine), so this book gave me a good introduction (both historical, philosophical and economic) to the open source movement.

The author starts the book with the history of UNIX and its derivatives, stating clearly the political struggles among the different people involved in it, the economic outcomes, the licensing and copyright problems, and even the vocabulary needed to understand the different concepts. This book does not directly target technologists; actually the author himself states very clearly at the beginning that he is a professor of economic politics, not a technologist.

I think that all CIOs that continue to spend millions in proprietary software, and avoid the open source solutions mainly because they do not have support should read this book and have a brighter idea about it. There is a whole world of opportunities behind the open source hype, and many projects out there (Linux, Apache, OpenOffice, GIMP, to name a few) are providing real solutions and working software for common problems. For free, and with the support of thousands of knowledgeable developers that have the same problems as you (instead of a phone support number that is always busy).

### Dealers of Lightning : Xerox PARC and the Dawn of the Computer Age

by Michael A. Hiltzik (ISBN 0887309895) [http://www.harpercollins.com/global\\_scripts/product\\_catalog/book.html](http://www.harpercollins.com/global_scripts/product_catalog/book.html)  
<http://www.amazon.com/gp/product/0887309895>

At this precise moment, you are reading this website on a screen plenty of windows, menus and icons that you drive with a mouse attached to your computer. You most probably access this website with an Ethernet adapter, and maybe you even use a laser printer to get some hardcopies of all of this babble. Did you know that the mouse, the icons, the Ethernet and the laser printer (as well as Smalltalk and Object-Oriented Programming, Postscript, much later Aspect-Oriented Programming, and many other inventions) all come from the very same place, the Xerox PARC (Palo Alto Research Center) in Silicon Valley?

This book is a must, one of the best ever Ive ever read, all categories considered. If you want to know how all of these things were created, who are the incredible

people behind these inventions, just buy this book. It is written as a novel, describing each character and every story in detail, and I really enjoyed it a lot.

Revolution in The Valley: The Insanely Great Story of How the Mac Was Made

by Andy Hertzfeld (ISBN 0596007191) <http://www.amazon.com/gp/product/0596007191/>  
<http://www.oreilly.com/catalog/revolution/>

I am a Mac user, and this book is a great one for anyone interested in knowing how the Mac was created, between 1978 and 1984. Andy Hertzfeld joined Apple as an Apple ][ engineer, and he managed to get into the Macintosh team early on. His book is plenty of insight, details, and more than anything, an incredible amount of stories about the first Macintosh team. After reading this, you will wish you were born twenty years before in southern California...

CODE: The Hidden Language of Computer Hardware and Software

by Charles Petzold (ISBN 073560505X) <http://www.amazon.com/gp/product/073560505X>  
<http://www.charlespetzold.com/code/>

This books gives a deep overview to many different ways used in the last two centuries to encode and transmit information, from the Morse code to programming languages used in current computers. It is an easily readable, enjoyable book, with plenty of information about the history of computers.

## 6. Coming Soon

These are the IT books that I am currently reading, or that I am planning to read soon:

From The Pragmatic Programmer:

Programming Ruby

Agile Web Development with Rails

My Job Went To India (And All I Got Was This Lousy Book)

The Pragmatic Programmer: From Journeyman to Master

(actually I'm currently reading the first three of the above list)

From O'Reilly:

The Cathedral and the Bazaar

Better, Faster, Lighter Java

Beyond Java

I will post details about these books as soon as I read them :)

Conclusion

I hope that this list will be a good starting point for those interested in software development; it is of course incomplete: there must be lots of books that you

could consider fundamental and that do not appear here. That is why I would love to have your feedback about my list, and recommend me other titles as well! I would love to have new reading ideas from you.

By the way, I've just found Joel's book reviews...

Update, 2005-11-23: Joel has published another book list, this time targeted to the management training program of his company...