Book Reviews

ELIZABETH ZWICKY, WITH SAM STOVER

12 Essential Skills for Software Architects

Dave Hendricksen

Addison-Wesley, 2011. 242 pp.
ISBN 978-0-321-71729-0

This is a book about non-technical skills for technical types who want to get into the higher reaches of designing software. It's not a book about managing, and it's not a book about architecting software, either. It's about what are often called "soft skills" (as if programming were inexorably scientific and talking to people nicely was easy). It lays out, carefully and clearly, the non-technical skills you need to succeed in getting things done in groups of people you are probably not in charge of: how to get along with people, how to talk to management, how to think about and talk about risk and failure.

I agree with the author almost all the time, and I think the book is a useful guide for people who may find themselves blocked for mysterious reasons. If you know you're right but you're not winning, this book will tell you why and will give you a good idea of what the winning strategy would be. It is as non-political as it is possible to be; that is, it advises that you think about strategies, personalities, and implications, but it does not advise you to be devious or manipulative.

What it won't do is get you all the way to implementing these skills. It's one thing to know that being right is not the most important thing; it's another thing entirely to manage to implement that in a meeting. If these skills don't come naturally to you, you're going to need to do some follow-up reading to pick up implementation strategies, and then you're going to need practice and probably assistance to get good at using them. It's not impossible to do, but it's not easy, either.

Privacy and Big Data

Terence Craig and Mary E. Ludloff

O'Reilly Media, 2011. 79 pp.
ISBN 978-1-449-30500-0

Designing Data Visualizations

Noah Iliinsky and Julie Steele

O'Reilly Media, 2011. 93 pp.
ISBN 978-1-449-31228-2

Big Data Glossary

Pete Warden

O'Reilly Media, 2011. 43 pp.
ISBN 978-1-449-31459-0

Building Data Science Teams: The Skills, Tools, and Perspectives Behind Great Data Science Groups

DJ Patil

O'Reilly Media, 2011. 25 pp.
ISBN 978-1-449-31623-5

Big Data Now: Current Perspectives from O'Reilly Radar

O'Reilly Media, 2011. 125 pp.
ISBN 987-1-449-31518-4

Here's a whole heap of short books, published by O'Reilly, about "big data." I will follow the example of the authors and leave the definition of "big data" vague; it's definitely more data than will fit on your laptop, probably more data than you have at home, and maybe more data than you feel comfortable having in one place for somebody to poke through. Aside from the common theme, the books vary greatly.

Privacy and Big Data tells you (most of) why you should be worried about big data, what the rules around it are, and where those rules come from. Its primary audience is not big data processors but other people who are thinking about how these giant data collections affect them. I found it competent, but as somebody whose employer's business is based on big data, I felt that there were some important omissions. The authors reliably fail to distinguish between intentional actions by companies and accidents, presenting (for instance) Google's collection of wireless data while doing Street View photos as if it were an intentional policy. This is important not only because it paints big data holders as worse than they actually are, but also because it implies that you only have to worry about companies that intend to invade your privacy. In fact, as in most things, well-meaning ineptitude, accidents, and oversights do as much damage to consumers as intentional violations. In addition, there is little discussion of the loopholes and inconsistencies of data privacy law; almost all data privacy rules have an exception for data collection to provide security measures and detect fraud. Those exceptions are very important and useful, but they also lead to piles of tempting data that would otherwise be uncollectable left lying around.

Designing Data Visualizations is a nice overview of the issues. It isn't the only data visualization book you'll ever need, but if you're looking for a good, short lead-in to the processes and issues, it's a nice place to start, with a practical bent and a helpful bibliography to take you to the next level. If you're staring at your Excel charts with a sinking feeling and no idea what to do to fix them, this will show you where to go.

Big Data Glossary is another introduction. It is not, as you might have expected, a glossary. Instead, it's an introduction to big data tools and languages; sort of a tour guide to the land of big data, detailing the main hotels and attractions, important phrases in the language, and how to get there. Again, this is a nice starting point; it's not going to get your data ready for human consumption, but it will at least enable you to understand the pieces of the problem and contemplate which ones you want more information on.

Building Data Science Teams is free, which is good; it's short; and it details one very specific approach. When it says "data science teams," it means it; silly of me, I guess, but I was hoping for something about teams that deal with big data in general, and the author is really only interested in teams where you put "people who think about big data" on one team and people who're related to particular subjects elsewhere. And he's very biased towards advanced degrees in the people who think about big data. It's a valid approach, and he has some interesting ideas about running teams and selecting people, but as a detailed description of a single approach, it's only going to be useful as a whole if you happen to have the right problem and the right environment.

Big Data Now is also free, and it's an anthology of short pieces already published. They contain their original advertising, which is a slightly odd effect, since they're often talking about conferences which are now over. As you can see from this collection of reviews, "big data" as a topic doesn't make for a terribly cohesive document. But there were several pieces here I found particularly useful or illuminating: Mike Loukides on "Data hand tools" (yes, I work at the home of Hadoop, but sometimes knowing how to run grep correctly over a terabyte of data is still faster than moving that data onto a Hadoop cluster), Pete Warden on "Why you can't really anonymize your data," and Alistair Croll on "There's no such thing as big data."

Programming Pig

Alan Gates
O'Reilly Media, 2011. 193 pp.
ISBN 978-1-449-30264-1

(Disclaimer: I am employed by Yahoo!, the primary location of Pig development and the copyright holders for this book. My corporate overlords have not, however, expressed an opinion on this book to me.) If you need to program in Pig, you are going to want a copy of this book around. If you're not sure whether you need to program in Pig, it's a scripting language used with Hadoop; and if that doesn't help, either you don't deal with big data or you need a copy of the *Big Data Glossary*.

Pig is an odd little language (and it is definitely a little language; 193 pages is enough to explore the entire language, the common extensions to it, and the ways to write your own extensions). It has a glancing resemblance to SQL, from which it borrows some syntax, but in spirit it's more like R: mind-bendingly fitted to its problem domain, to the extent of introducing new data types. When I utter sentences like, "Remember, that's not a bag, it's a tuple," I begin to wonder what non-obscene nouns have not yet been used to describe data structures, and whether I will some day in all technical seriousness borrow a cup of data from my neighbor.

Pig's strong point is its ability to express a lot of data transformations simply and in such a way that the software can do a reasonable job of optimizing them, sometimes with your assistance. Its corresponding weak point is its specialization. It's very easy to learn enough Pig to express simple queries, but then there tends to be a wall of non-comprehension where it seems like things must be expressible but you're not quite sure how. This is a good book for getting you past that, to the point where you can tell a bag from a tuple, write nested filters, and use Pig to manage complex data flows (the ability to handle multiple inputs and outputs is one of the great advantages of Pig over straightforward Hadoop streaming).

Pluralism in Software Engineering: Turing Award Winner Peter Naur Explains

Edgar G. Daylight

Lonely Scholar Scientific Books, 2011. 119 pp.
ISBN 978-94-9138-600-8

This is a transcription of an interview with Peter Naur, best known to most of us as the Naur in Backus-Naur notation. The interview covers the rest of his career, never mentioning the notation. It breaks into roughly three topics: a section on the early history of computing, one on issues about computing and formalism, and one on Naur's theories of neurology.

I expect that not very many people will find all three equally gripping. I understand that there are people who are fascinated by the early history of computing, but I am not one of them; sadly, I would rather hear the interpersonal gossip that Naur carefully (and very appropriately) avoids discussing. And while his theories about neurology are interesting, I have a well-earned distrust of very smart people who hypothesize outside the fields they are expert in.

From that you can deduce that I liked the section on formalism, particularly Naur's assertions that as far as he can tell, it's not terribly useful for programming—programming is an idiosyncratic process—and most of the people who describe neat, beautiful ways they arrive at solutions to problems are, if not lying, creatively rearranging the truth to make better reading.

If this sounds gripping to you, I recommend searching it out; if you're on the fence, it's probably not going to win you over. The interview format is always a little clunky, and it is not perfectly implemented here.

MongoDB and Python

Niall O'Higgins

O'Reilly Media, 2011, 66 pp.
ISBN 978-1-449-31037-0

Since I started messing around with Hadoop, I've been exposed to a couple of other similar technologies, and one of them was MongoDB. I'll admit (more than) half the reason I picked this book up is the Python part, and I figured I'd see what MongoDB can do. Now that I see how easy it is to use MongoDB with Python, I really need to learn it from the ground up. That aside, this book packs a lot into its 66 pages and certainly wasn't a waste of my time.

There are just four chapters, and the first gets you started with a brief intro to MongoDB and some comparisons with both traditional and NoSQL databases. Installing, running, and setting up a Python environment round out Chapter 1 in typical O'Reilly fashion. Chapter 2 gives you the basics of interacting with a MongoDB: connecting, getting a database handle, insertion, queries, etc. MongoDB is a document-oriented database, and this chapter does a decent job of explaining how that is different from a relational database. Two things I really liked about this chapter were (1) comparisons to SQL concepts and (2) consistently putting things into Pythonic terms. I might not know what a JSON document is (well, I didn't then), but I know what a Python dictionary is. This is a database book written for people who know databases—which I am not. Keeping things in Python style made it much easier for me to follow what was going on.

Chapter 3 goes a little deeper into different ways to use MongoDB more efficiently: The concept of embedding documents, while mentioned previously, is explained in depth, and there are some suggestions for making your queries more efficient with proper indexing, among others. The really cool feature though, which I was totally not expecting, was the section on geospatial indexing, which is supported out of the box. MongoDB uses a public domain algorithm (geohashing), which "translates geographic proximity into lexical proximity." This is a pretty cool capability and it's not hard to use—examples given in Python (grin).

Chapter 4 discusses integrating MongoDB with three Web frameworks: Pylons 1.x, Pyramid, and Django. Pylons and Pyramid are somewhat similar, and it seems that Pyramid is the more active of the two. Django differs from them in that it is "one well-integrated package" with its own set of templates, interfaces, etc. Regardless of what your needs are, chances are good that one of these will fit.

Overall, a solid book with lots of examples and code. Installing and setting up MongoDB, Pylons, Pyramid, and Django are all covered well and should make getting started pretty simple and fast for anyone. As a Python guy with a little database experience, I still found the book very accessible and am already coming up with ideas on what I can use MongoDB for. IMHO, MongoDB does not replace or compete with Hadoop but, as a document-oriented database, provides answers to different questions.

—Sam Stover

Privacy and Big Data

Terence Craig and Mary E. Ludloff

O'Reilly and Associates, 2011, 79 pp.
ISBN 978-1-449-30500-0

Right out of the gate, the authors give the disclaimer that they are executives from a "growing startup in the big data and analytics industry," which caught me off guard. I was expecting this to be a Top Technical Tips for preserving online privacy, and what I got was something different—very interesting and educational, but different. It's a short book—five chapters and fewer than 80 pages, but chock full of interesting facts and tons, I mean *tons*, of references. Not that a book should be judged by end/foot notes, but there are a lot: 218 by my count. This makes for a lot of research potential, should you want to go chasing down the facts they present.

Chapter 1 starts with ARPANET and takes you to today; from zero personal data online to the mess we all live in now. It's an interesting chapter. While at first glance you might think it doesn't bring anything that we aren't all aware of, one point that really jumped out at me was that it's not just about advertising. There are lots of other ways to use our personal information out there, and they're probably scarier than presenting you with a targeted ad (which no one likes anyway).

Chapter 2 deals with privacy in the digital age, with an emphasis on US vs. EU stances. It's probably the driest of the chapters but, again, was just full of stuff that I didn't know. The authors are of the opinion that in the US, privacy is a commodity that we can use to barter for stuff/conveniences we want for "free." In the EU, privacy is a basic human right "that transcends commoditization." Pretty heady stuff, but it does make you think. Well, it made me think.

Chapters 3 and 4 deal with The Regulators and The Players, respectively: lots of discussion on who is doing the regulating, how they are doing it, pros and cons of the different approaches, as well as how the Players deal with (or ignore) the Regulators. It gives plenty of examples, some that you might not be aware of, some you probably have seen on the news.

Chapter 5 wraps everything up and lays out the most important point of all. Whether you view your privacy as a right or a commodity, the fact still stands that once you release information into the Internet, it will (probably) never go away. No matter what legislation comes about, no matter what rights you think you may have, "you can't unring the bell." It might not be scary, or it might be, depending on how paranoid you are, but either way, this book is a good read. It's not going to walk you through protecting yourself, although there is some discussion on ways to make it harder for your privacy to be invaded. But I think after you read this, you might just start looking at how much of you exists on the Internet, and what you can do about it. Plus, it's pretty non-technical, so if you're already paranoid but know people who aren't, you may have just found a great present for them.

—Sam Stover