daolf.com

📕 The 25 best programming books of all-time. A data-backed answer

Pierre de Wulf

26-33 minutes

This article is a follow up of the one I did about the the most recommended startup books of all-time.

If you've read this one recently. I guess you can jump straight to the results.

There are countless lists on the internet claiming to be **the** list of must-read programming books and it seemed that all those lists always recommended that same books minus two or three odd choices.

Finding good ressources for learning programming is always tricky. Every-one has its own opinion about what book is the best to learn, and as we say in french, "Color and tastes should not be argued about".

However I though it would be interesting to trust the wisdom of the crown and to find the books that appeared the most in those "Best Programming Book" lists.

If you want to jump right on the results go take a look below at

the full results. If you want to learn about the methodology, bear with me.

Disclaimer: I spent countless hours on this article so I've decided to put Amazon affiliation links to see if those kinds of detailed articles could be a viable source of revenue, … or not 🤦.

Methodology:

I've simply asked Google for a few queries like "Best Programming Books" and its variations of. I have then scrapped all those pages (using ScrapingBee, a web scraping API I'm working on).

I've deduplicated the links and ended up with nearly 150 links. Using the title of the pages I was also able to quickly discards:

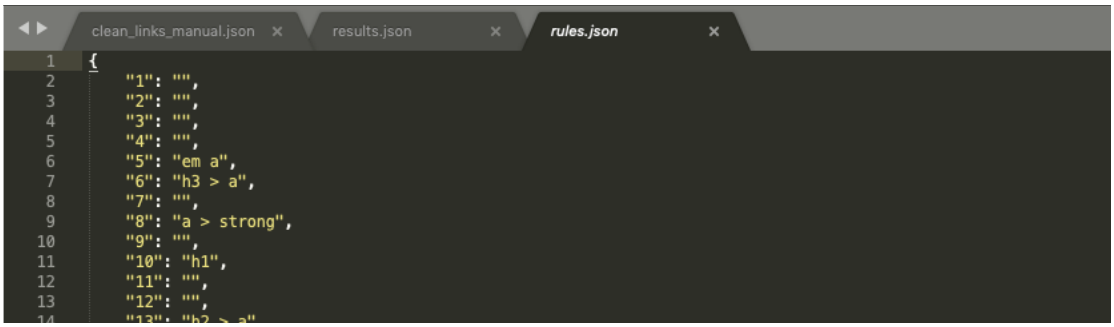
- list focused on one particular technology or platform
- list focused on one particular year
- list focused on free books
- Quora and Reddit threads

I ended up with almost 110 HTML files. I went on opening all the files on my browser, open my chrome inspector, found and wrote the CSS selector matching book titles in the article. This took me around 1hours, almost 30 seconds per page.

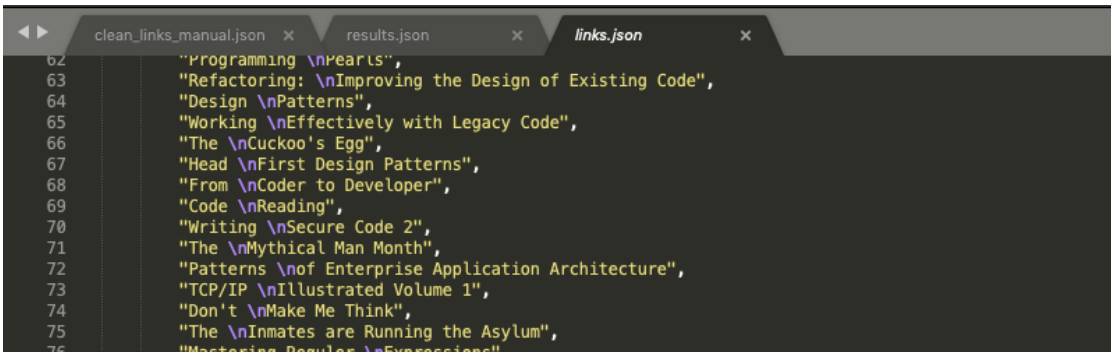
This also allowed me to discard even more non-relevant pages, and I discarded a lot. In the end I compiled around 70

lists into this one.

At this moment I had this big JSON file referencing the HTML page previously scrapped, and a CSS selector.



Using Python with Beautiful soup, I've extracted every text inside DOM elements that matched the CSS selector. I ended up with a huge list of books, not usable without some post-processing.



To find the most quoted programming books I needed to normalize my results.

I had to play with all the different variation like "{title} by {author}" or "{title} - {author}".

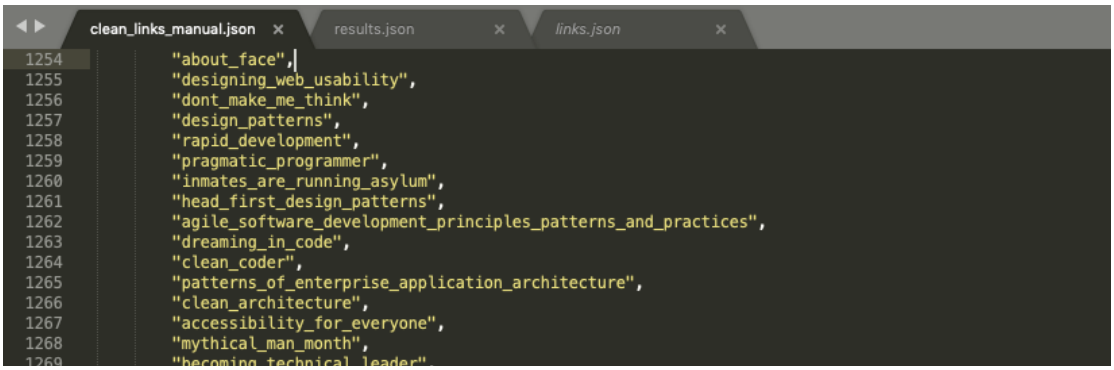
Or "{title}:{subtitle}" and "{title}", or even all the one containing edition number.

I ended up doing it using this simple custom Python function:

```
def clean_link(link):
    link = link.encode().decode('ascii',
errors='ignore')
    link = link.replace("'", '')
    link = link.lower()
    link = ' '.join([w for w in link.split('
') if w not in ['the', 'a']])
    link = link.split('by')[0]
    link = link.split(':')[0]
    link = link.split('(')[0]
    link = ' '.join(link.split())
    link = link.replace('-', '_')
    link = ''.join([c for c in link if
c.isalpha() or c == '_' or c == ' '])
    link = link.strip()
    link = link.replace(' ', '_')
    link = ''.join([c for c in link if
c.isalpha() or c == '_'])
    return link
```

and quite a bit of manual cleaning.

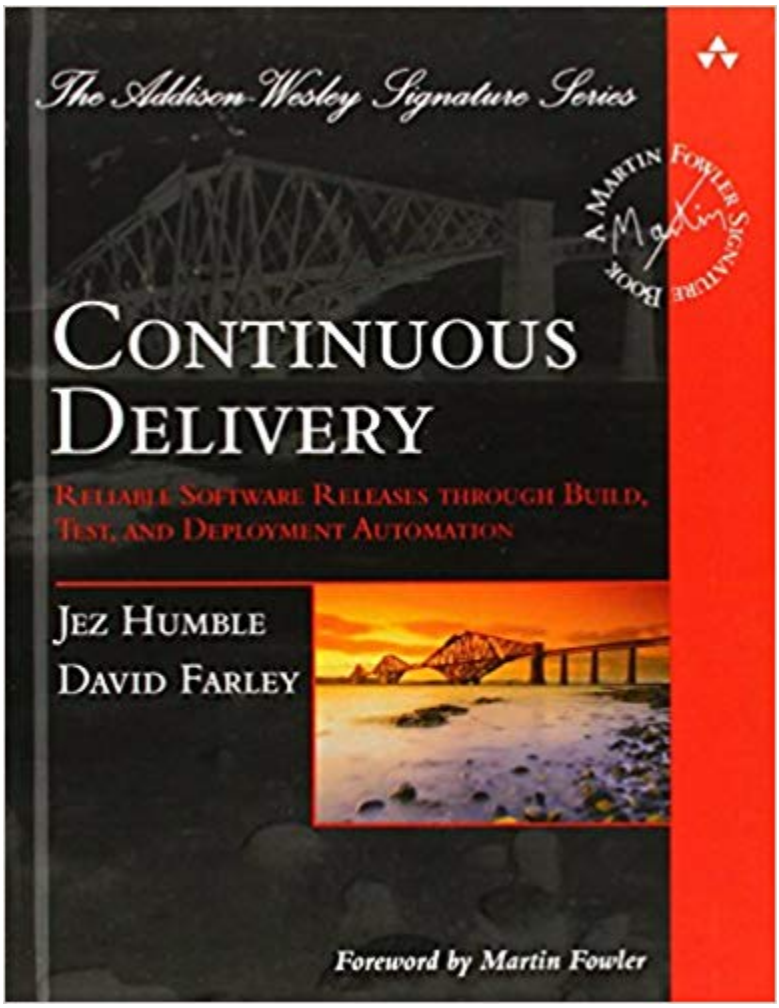
My list now looked like this:



From there it was easy to compute the most recommended books. You can find all the data used to process this list on this repo. Now let's take a look at the list:

25 most recommended programming books of all-time

25. Continuous Delivery by Jez Humble & David Farley (8.8% recommended)

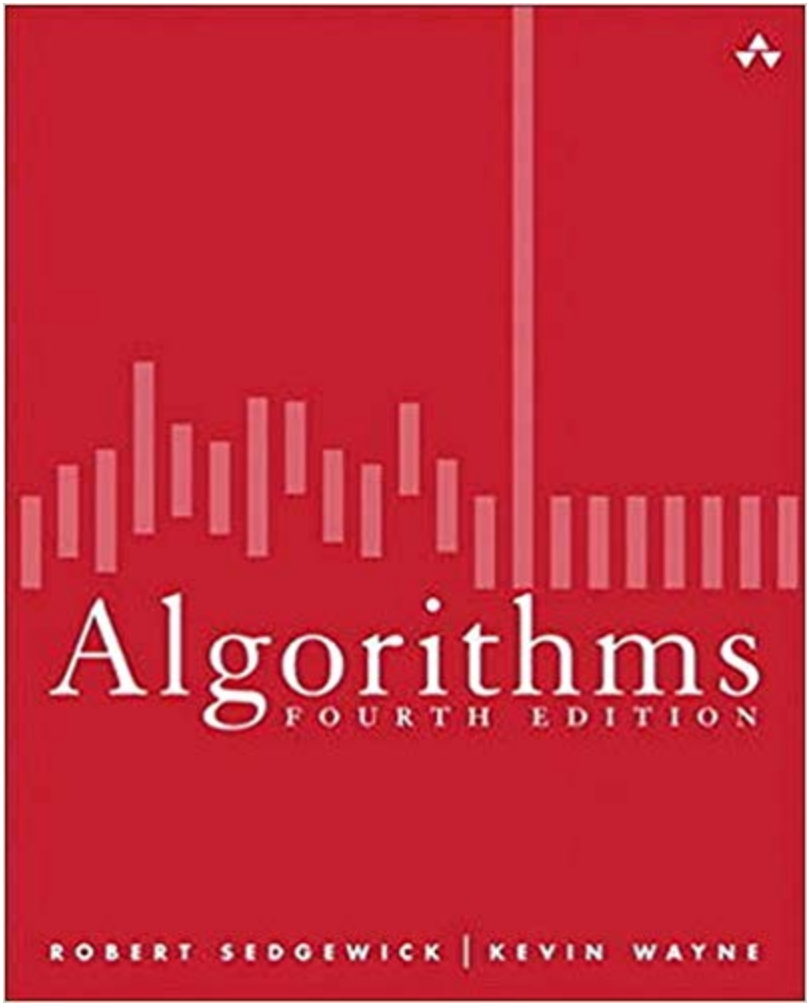


"Getting software released to users is often a painful, risky, and time-consuming process. This groundbreaking new book

sets out the principles and technical practices that enable rapid, incremental delivery of high quality, valuable new functionality to users. Through automation of the build, deployment, and testing process, and improved collaboration between developers, testers, and operations, delivery teams can get changes released in a matter of hours— sometimes even minutes–no matter what the size of a project or the complexity of its code base.

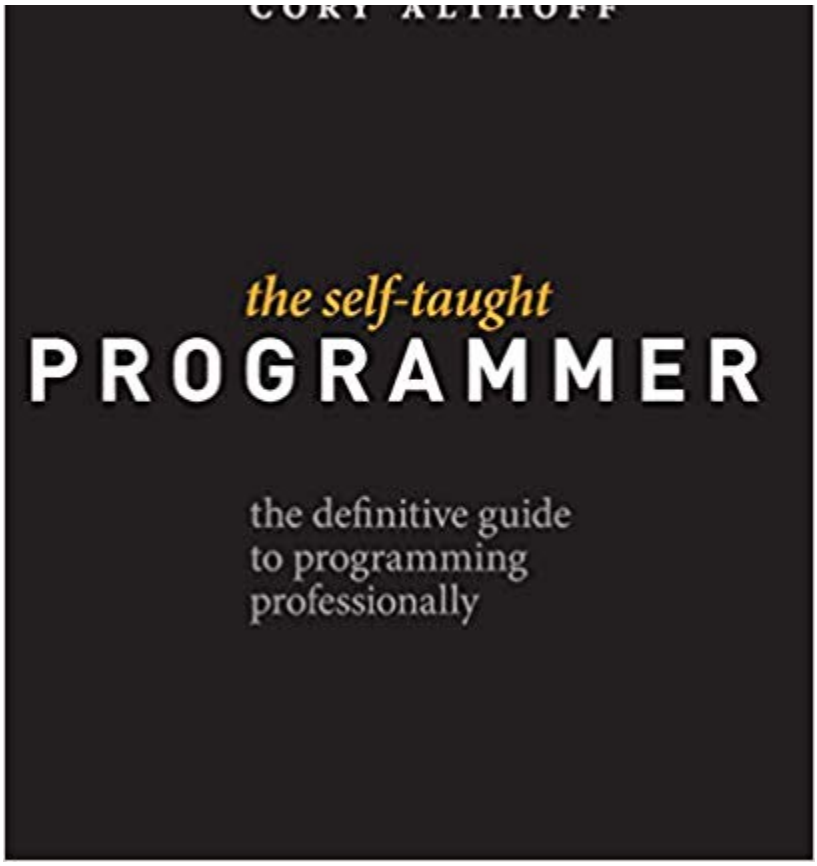
Jez Humble and David Farley begin by presenting the foundations of a rapid, reliable, low-risk delivery process. Next, they introduce the “deployment pipeline,” an automated process for managing all changes, from check-in to release. Finally, they discuss the “ecosystem” needed to support continuous delivery, from infrastructure, data and configuration management to governance.” Amazon.com

24. Algorithms **by Robert Sedgewick & Kevin Wayne (8.8% recommended)**



“The algorithms in this book represent a body of knowledge developed over the last 50 years that has become indispensable, not just for professional programmers and computer science students but for any student with interests in science, mathematics, and engineering, not to mention students who use computation in the liberal arts.”Amazon.com

23. The Self-Taught Programmer **by Cory Althoff (8.8% recommended)**



“I am a self-taught programmer. After a year of self-study, I learned to program well enough to land a job as a software engineer II at eBay.

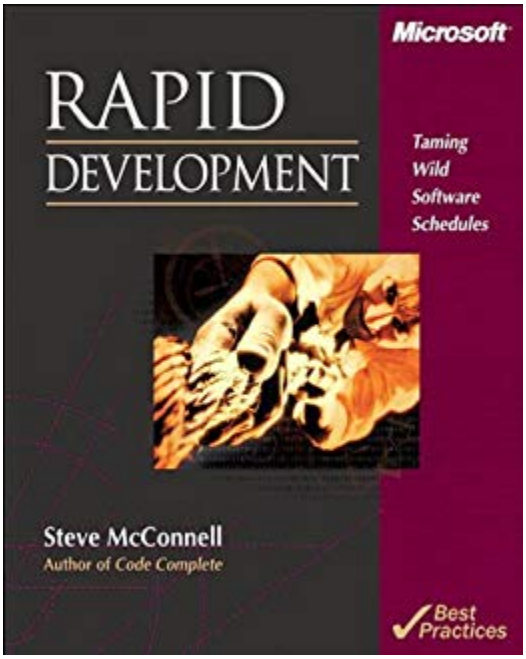
Once I got there, I realized I was severely under-prepared. I was overwhelmed by the amount of things I needed to know but hadn’t learned yet. My journey learning to program, and my experience at my first job as a software engineer were the inspiration for this book.

This book is not just about learning to program; although you will learn to code. If you want to program professionally, it is not enough to learn to code; that is why, in addition to helping you learn to program, I also cover the rest of the things you need to know to program professionally that classes and

books don't teach you.

"The Self-taught Programmer" is a roadmap, a guide to take you from writing your first Python program, to passing your first technical interview. The path is there. Will you take it?"Amazon.com

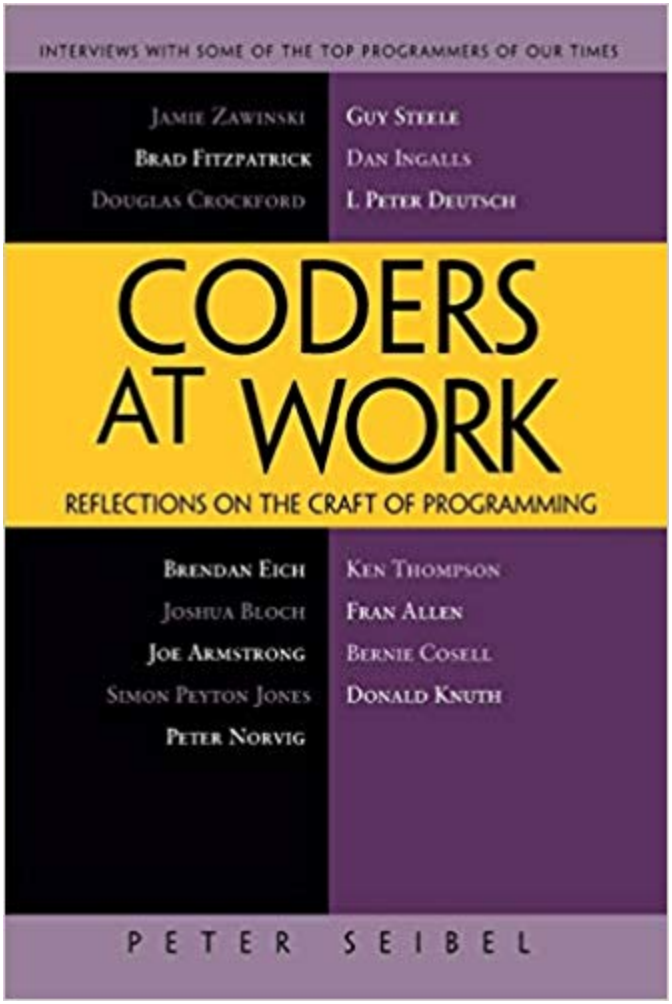
22. Rapid Development **by Steve McConnell (8.8% recommended)**



"Corporate and commercial software-development teams all want solutions for one important problem—how to get their high-pressure development schedules under control. In RAPID DEVELOPMENT, author Steve McConnell addresses that concern head-on with overall strategies, specific best practices, and valuable tips that help shrink and control development schedules and keep projects moving. Inside, you'll find: - A rapid-development strategy that can be applied

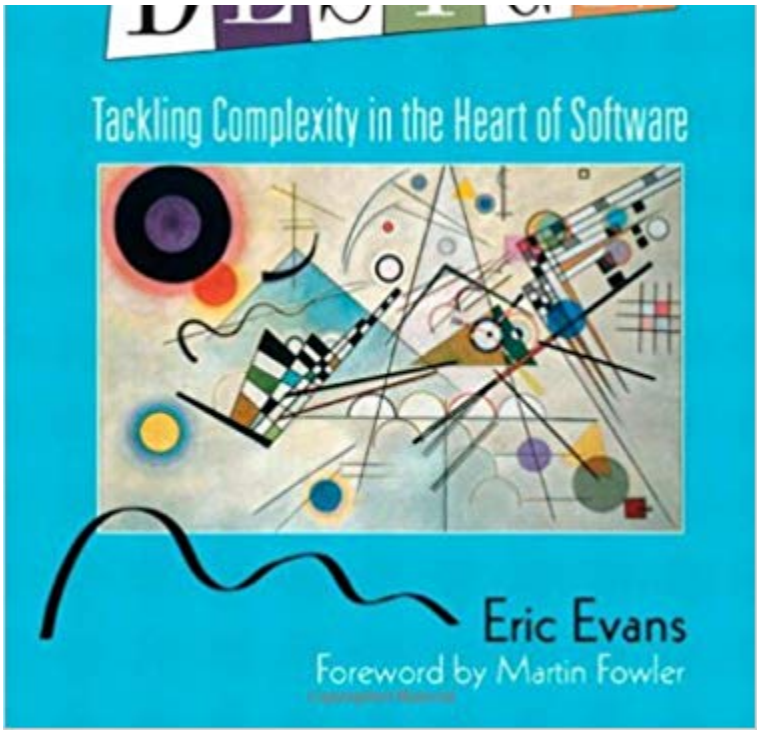
to any project and the best practices to make that strategy work - Candid discussions of great and not-so-great rapid-development practices—estimation, prototyping, forced overtime, motivation, teamwork, rapid-development languages, risk management, and many others - A list of classic mistakes to avoid for rapid-development projects, including creeping requirements, shortchanged quality, and silver-bullet syndrome - Case studies that vividly illustrate what can go wrong, what can go right, and how to tell which direction your project is going - RAPID DEVELOPMENT is the real-world guide to more efficient applications development."Amazon.com

21. Coders at Work **by Peter Seibel (10.2% recommended)**



“This is a who’s who in the programming world - a fascinating look at how some of the best in the world do their work. Patterned after the best selling Founders at Work, the book represents two years of interviews with some of the top programmers of our times.”Amazon.com

20. Domain-Driven Design **by Eric Evans (10.2% recommended)**



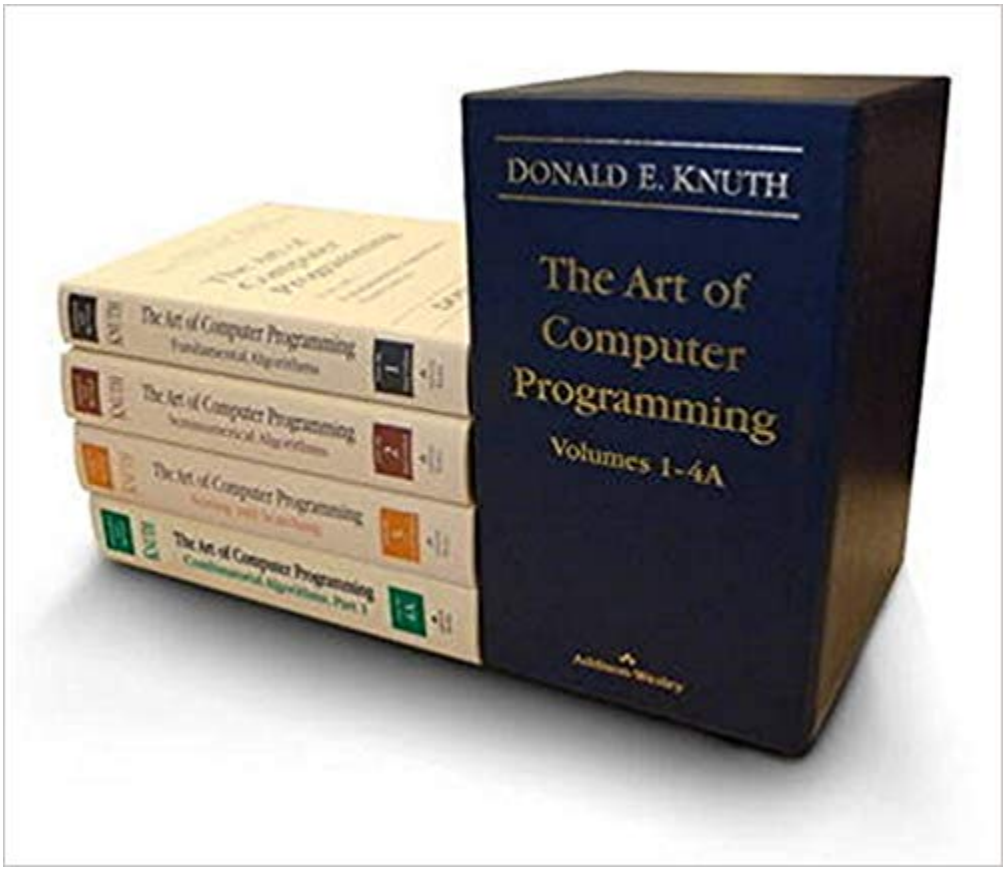
“Leading software designers have recognized domain modeling and design as critical topics for at least twenty years, yet surprisingly little has been written about what needs to be done or how to do it. Although it has never been clearly formulated, a philosophy has developed as an undercurrent in the object community, which I call “domain-driven design”.

I have spent the past decade focused on developing complex systems in several business and technical domains. I’ve tried best practices in design and development process as they have emerged from the leaders in the object-oriented development community. Some of my projects were very successful; a few failed. A feature common to the successes was a rich domain model that evolved through iterations of design and became part of the fabric of the project.

This book provides a framework for making design decisions

and a technical vocabulary for discussing domain design. It is a synthesis of widely accepted best practices along with my own insights and experiences. Projects facing complex domains can use this framework to approach domain-driven design systematically."Amazon.com

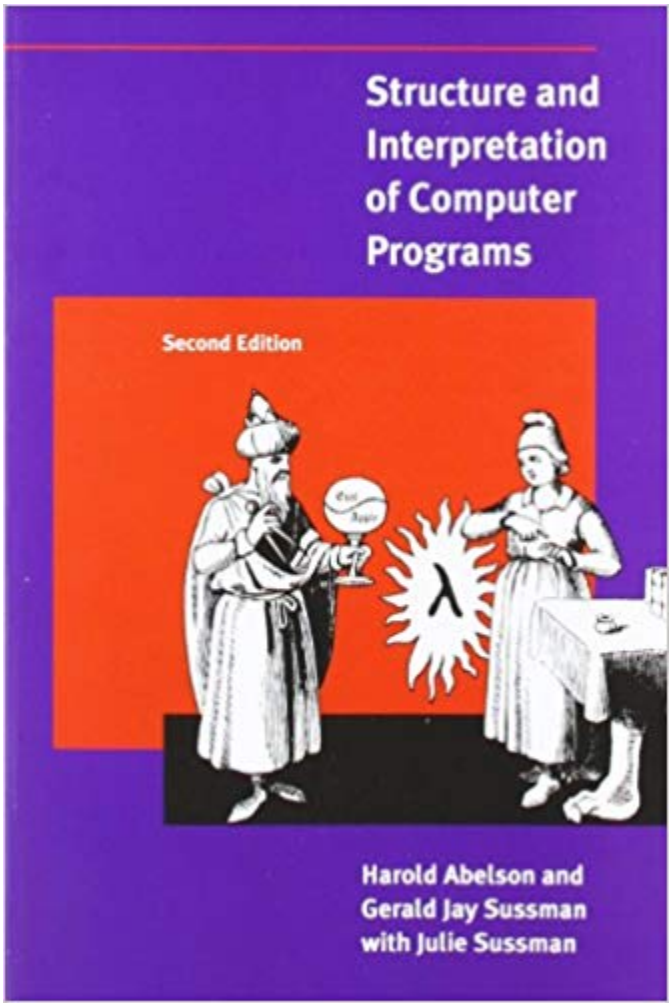
19. The Art of Computer Programming **by Donald E. Knuth(10.2% recommended)**



"Countless readers have spoken about the profound personal influence of Knuth's work. Scientists have marveled at the beauty and elegance of his analysis, while ordinary programmers have successfully applied his "cookbook" solutions to their day-to-day problems. All have admired Knuth

for the breadth, clarity, accuracy, and good humor found in his books."Amazon.com

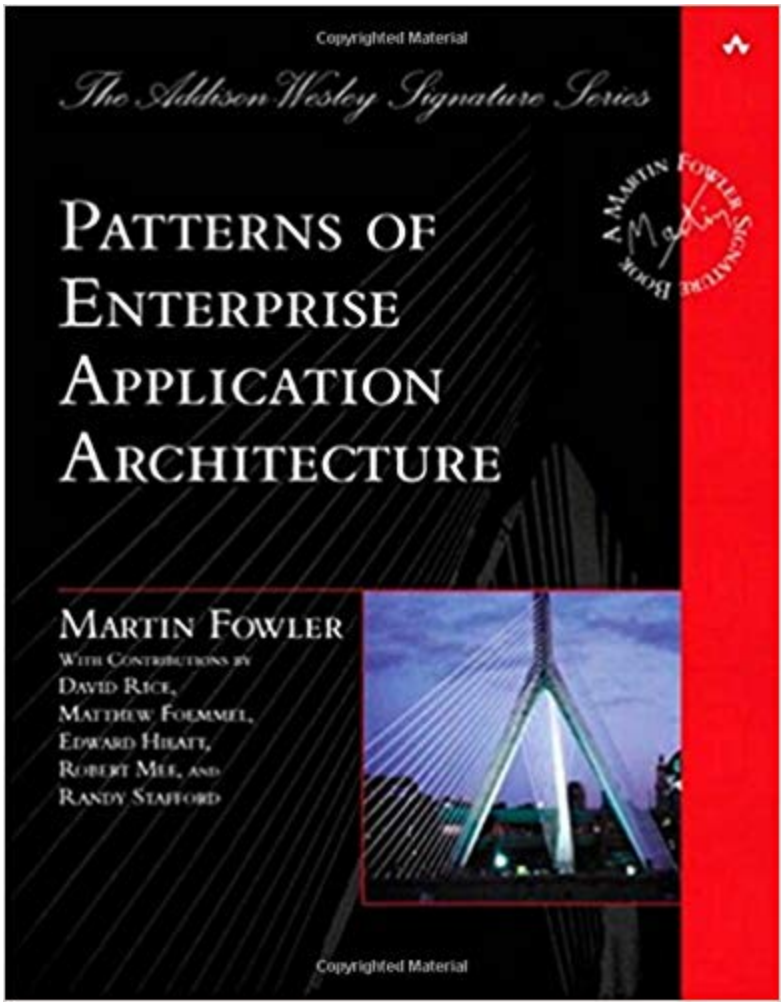
18. Structure and Interpretation of Computer Programs **by Harold Abelson / Gerald Jay Sussman / Julie Sussman (13.2% recommended)**



"Structure and Interpretation of Computer Programs has had a dramatic impact on computer science curricula over the past decade. This long-awaited revision contains changes throughout the text. There are new implementations of most of the major programming systems in the book, including the interpreters

and compilers, and the authors have incorporated many small changes that reflect their experience teaching the course at MIT since the first edition was published. A new theme has been introduced that emphasizes the central role played by different approaches to dealing with time in computational models: objects with state, concurrent programming, functional programming and lazy evaluation, and nondeterministic programming. There are new example sections on higher-order procedures in graphics and on applications of stream processing in numerical programming, and many new exercises. In addition, all the programs have been reworked to run in any Scheme implementation that adheres to the IEEE standard."Amazon.com

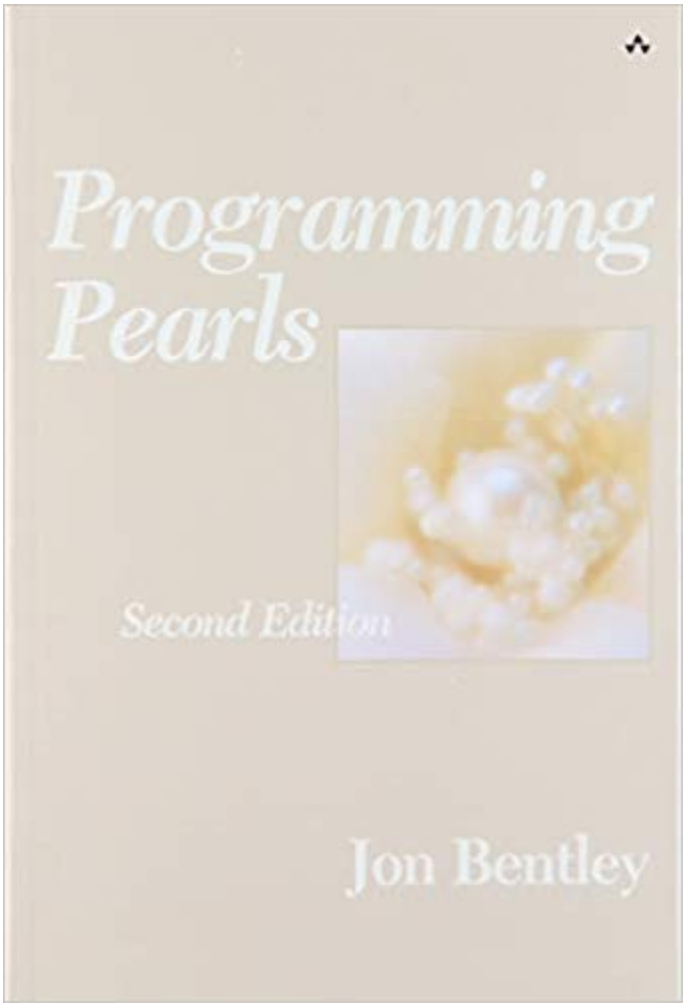
17. Patterns of Enterprise Application Architecture **by Martin Fowler (14.7% recommended)**



"The practice of enterprise application development has benefited from the emergence of many new enabling technologies. Multi-tiered object-oriented platforms, such as Java and .NET, have become commonplace. These new tools and technologies are capable of building powerful applications, but they are not easily implemented. Common failures in enterprise applications often occur because their developers do not understand the architectural lessons that experienced object developers have learned."Amazon.com

16. Programming Pearls **by Jon Bentley (16.1%**

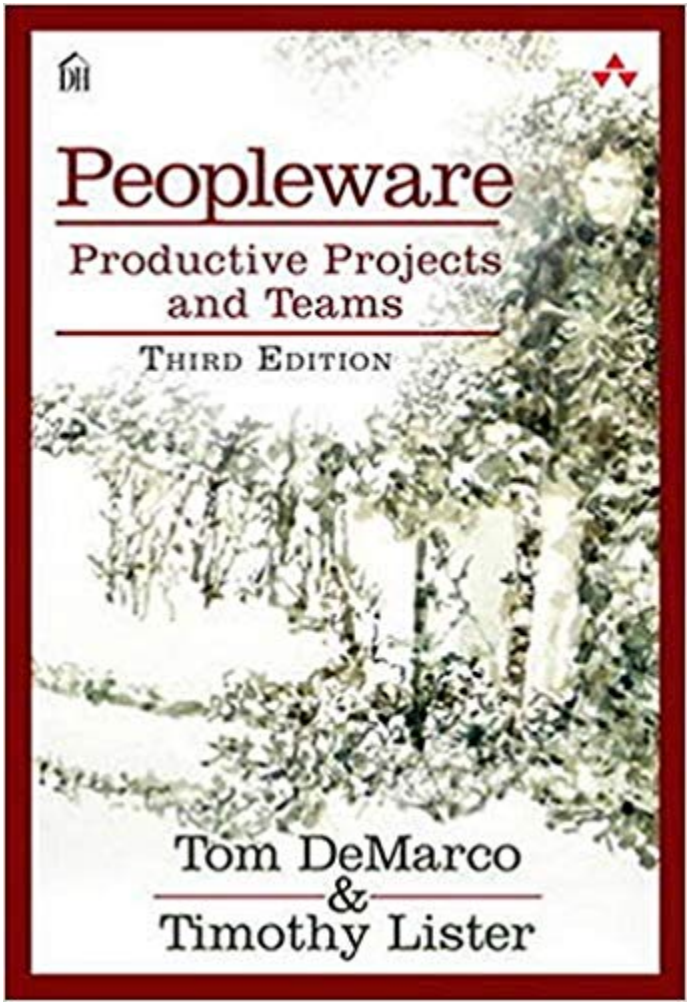
recommended)



“Computer programming has many faces. Fred Brooks paints the big picture in The Mythical Man Month; his essays underscore the crucial role of management in large software projects. At a finer grain, Steve McConnell teaches good programming style in Code Complete. The topics in those books are the key to good software and the hallmark of the professional programmer. Unfortunately, though, the workmanlike application of those sound engineering principles isn’t always thrilling – until the software is completed on time and works without surprise.

The columns in this book are about a more glamorous aspect of the profession: programming pearls whose origins lie beyond solid engineering, in the realm of insight and creativity. Just as natural pearls grow from grains of sand that have irritated oysters, these programming pearls have grown from real problems that have irritated real programmers. The programs are fun, and they teach important programming techniques and fundamental design principles."Amazon.com

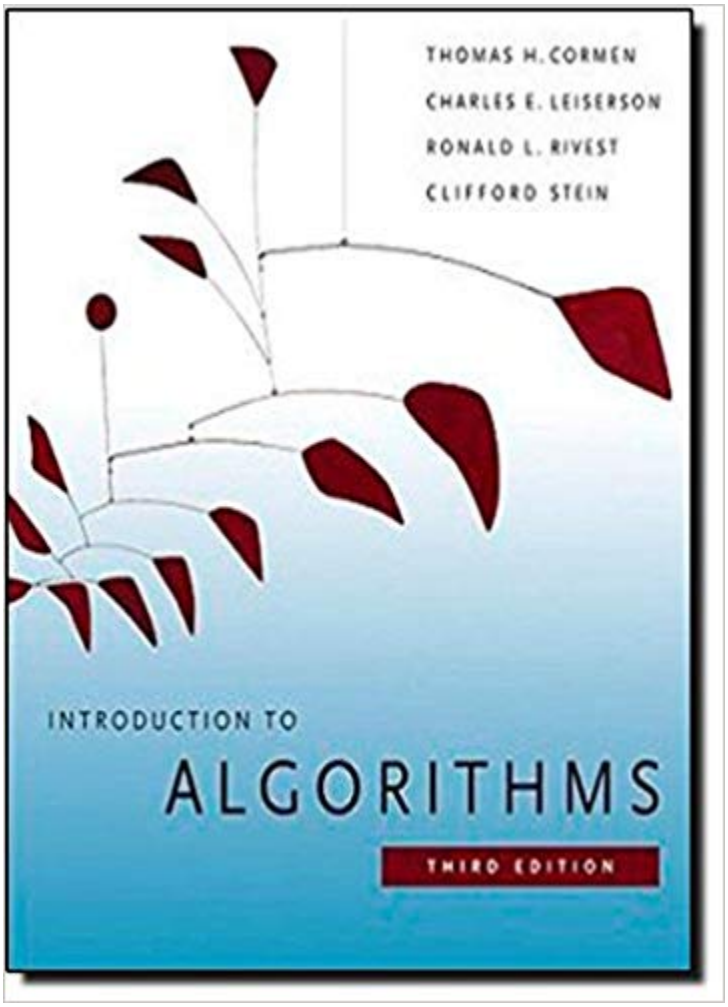
15. Peopleware **by Tom DeMarco & Tim Lister (17.6% recommended)**



"The unique insight of

this longtime bestseller is that the major issues of software development are human, not technical. They're not easy issues; but solve them, and you'll maximize your chances of success."Amazon.com

14. Introduction to Algorithms **by Thomas H. Cormen / Charles E. Leiserson / Ronald L. Rivest / Clifford Stein (17.6% recommended)**

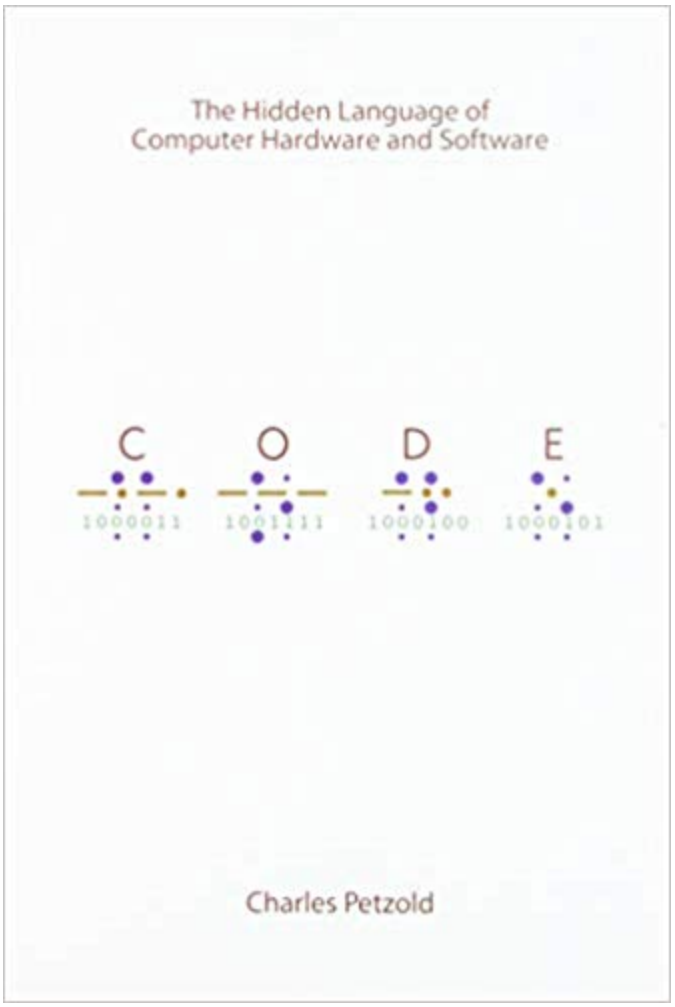


"Some books on algorithms are rigorous but incomplete; others cover masses of material but lack rigor. Introduction to Algorithms uniquely combines rigor and comprehensiveness.

The book covers a broad range of algorithms in depth, yet makes their design and analysis accessible to all levels of readers. Each chapter is relatively self-contained and can be used as a unit of study. The algorithms are described in English and in a pseudocode designed to be readable by anyone who has done a little programming. The explanations have been kept elementary without sacrificing depth of coverage or mathematical rigor.

The first edition became a widely used text in universities worldwide as well as the standard reference for professionals. The second edition featured new chapters on the role of algorithms, probabilistic analysis and randomized algorithms, and linear programming. The third edition has been revised and updated throughout. It includes two completely new chapters, on van Emde Boas trees and multithreaded algorithms, substantial additions to the chapter on recurrence (now called “Divide-and-Conquer”), and an appendix on matrices. It features improved treatment of dynamic programming and greedy algorithms and a new notion of edge-based flow in the material on flow networks. Many exercises and problems have been added for this edition. The international paperback edition is no longer available; the hardcover is available worldwide.”Amazon.com

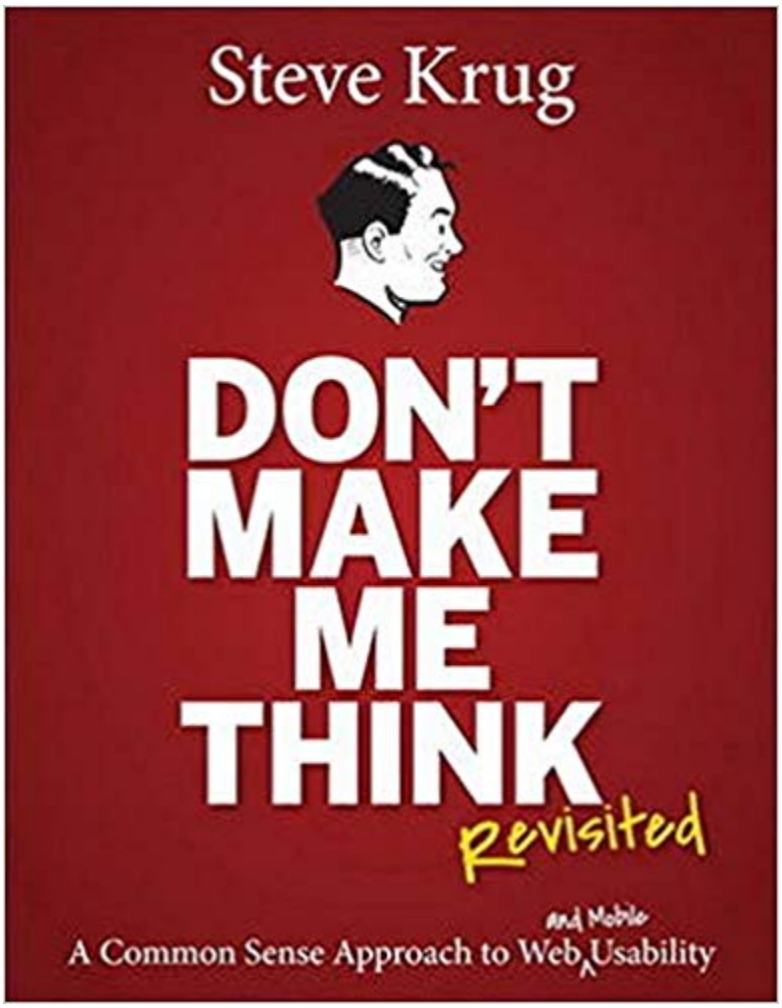
13. Code **by Charles Petzold (19.1% recommended)**



“What do flashlights, the British invasion, black cats, and seesaws have to do with computers? In CODE, they show us the ingenious ways we manipulate language and invent new means of communicating with each other. And through CODE, we see how this ingenuity and our very human compulsion to communicate have driven the technological innovations of the past two centuries. Using everyday objects and familiar language systems such as Braille and Morse code, author Charles Petzold weaves an illuminating narrative for anyone who’s ever wondered about the secret inner life of computers and other smart machines. It’s a cleverly illustrated and eminently

comprehensible story—and along the way, you'll discover you've gained a real context for understanding today's world of PCs, digital media, and the Internet. No matter what your level of technical savvy, CODE will charm you—and perhaps even awaken the technophile within."Amazon.com

12. Don't Make Me Think **by Steve Krug (19.1% recommended)**



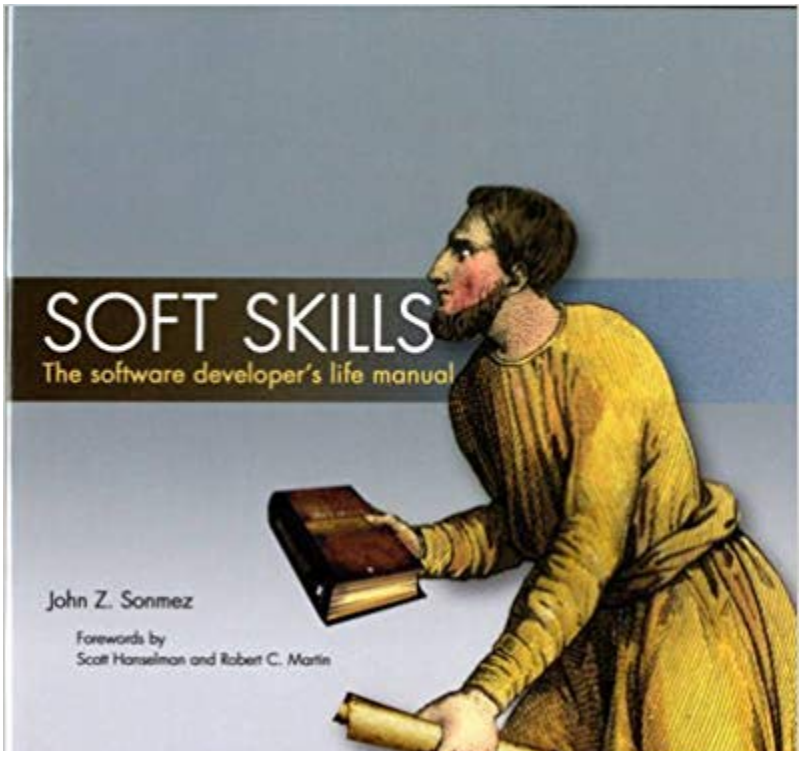
"Since Don't Make Me Think was first published in 2000, hundreds of thousands of Web designers and developers have relied on usability guru Steve Krug's guide to help them

understand the principles of intuitive navigation and information design. Witty, commonsensical, and eminently practical, it's one of the best-loved and most recommended books on the subject.

Now Steve returns with fresh perspective to reexamine the principles that made Don't Make Me Think a classic–with updated examples and a new chapter on mobile usability. And it's still short, profusely illustrated…and best of all–fun to read.

If you've read it before, you'll rediscover what made Don't Make Me Think so essential to Web designers and developers around the world. If you've never read it, you'll see why so many people have said it should be required reading for anyone working on Web sites."Amazon.com

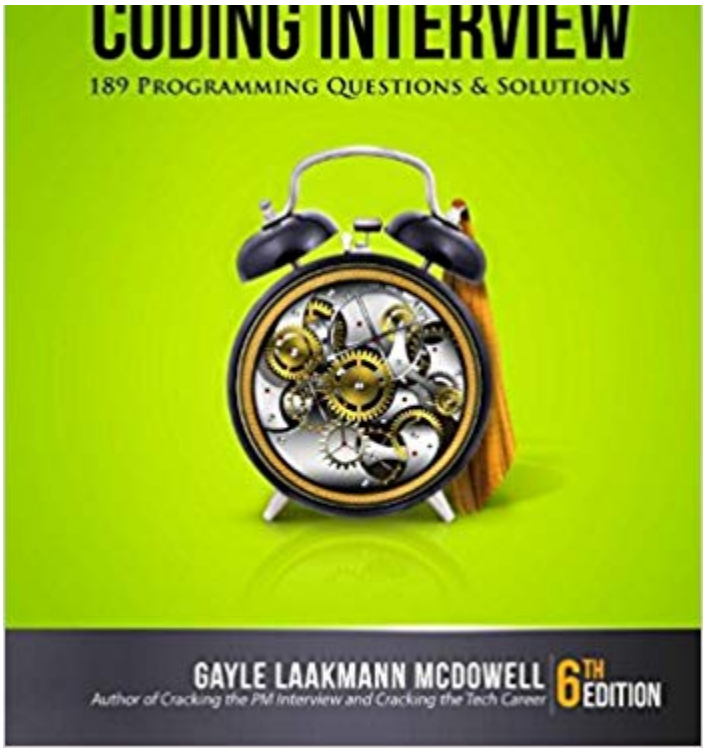
11. Soft Skills **by John Sonmez (22% recommended)**



“For most software developers, coding is the fun part. The hard bits are dealing with clients, peers, and managers, staying productive, achieving financial security, keeping yourself in shape, and finding true love. This book is here to help.

Soft Skills: The software developer’s life manual is a guide to a well-rounded, satisfying life as a technology professional. In it, developer and life coach John Sonmez offers advice to developers on important “soft” subjects like career and productivity, personal finance and investing, and even fitness and relationships. Arranged as a collection of 71 short chapters, this fun-to-read book invites you to dip in wherever you like. A Taking Action section at the end of each chapter shows you how to get quick results. Soft Skills will help make you a better programmer, a more valuable employee, and a happier, healthier person.”Amazon.com

10. Cracking the Coding Interview **by Gayle Laakmann McDowell (22% recommended)**



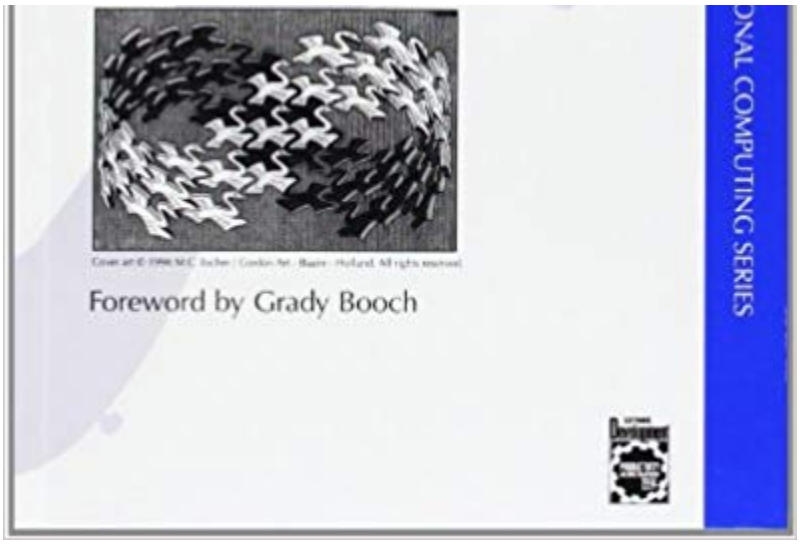
“I am not a recruiter. I am a software engineer. And as such, I know what it’s like to be asked to whip up brilliant algorithms on the spot and then write flawless code on a whiteboard. I’ve been through this as a candidate and as an interviewer.

Cracking the Coding Interview, 6th Edition is here to help you through this process, teaching you what you need to know and enabling you to perform at your very best. I’ve coached and interviewed hundreds of software engineers. The result is this book.

Learn how to uncover the hints and hidden details in a question, discover how to break down a problem into manageable chunks, develop techniques to unstick yourself when stuck, learn (or re-learn) core computer science concepts, and practice on 189 interview questions and solutions.

These interview questions are real; they are not pulled out of computer science textbooks. They reflect what's truly being asked at the top companies, so that you can be as prepared as possible. WHAT'S INSIDE? - 189 programming interview questions, ranging from the basics to the trickiest algorithm problems. - A walk-through of how to derive each solution, so that you can learn how to get there yourself. - Hints on how to solve each of the 189 questions, just like what you would get in a real interview. - Five proven strategies to tackle algorithm questions, so that you can solve questions you haven't seen. - Extensive coverage of essential topics, such as big O time, data structures, and core algorithms. - A behind the scenes look at how top companies like Google and Facebook hire developers. - Techniques to prepare for and ace the soft side of the interview: behavioral questions. - For interviewers and companies: details on what makes a good interview question and hiring process."Amazon.com

9. Design Patterns **by by Erich Gamma / Richard Helm / Ralph Johnson / John Vlissides (25% recommended)**



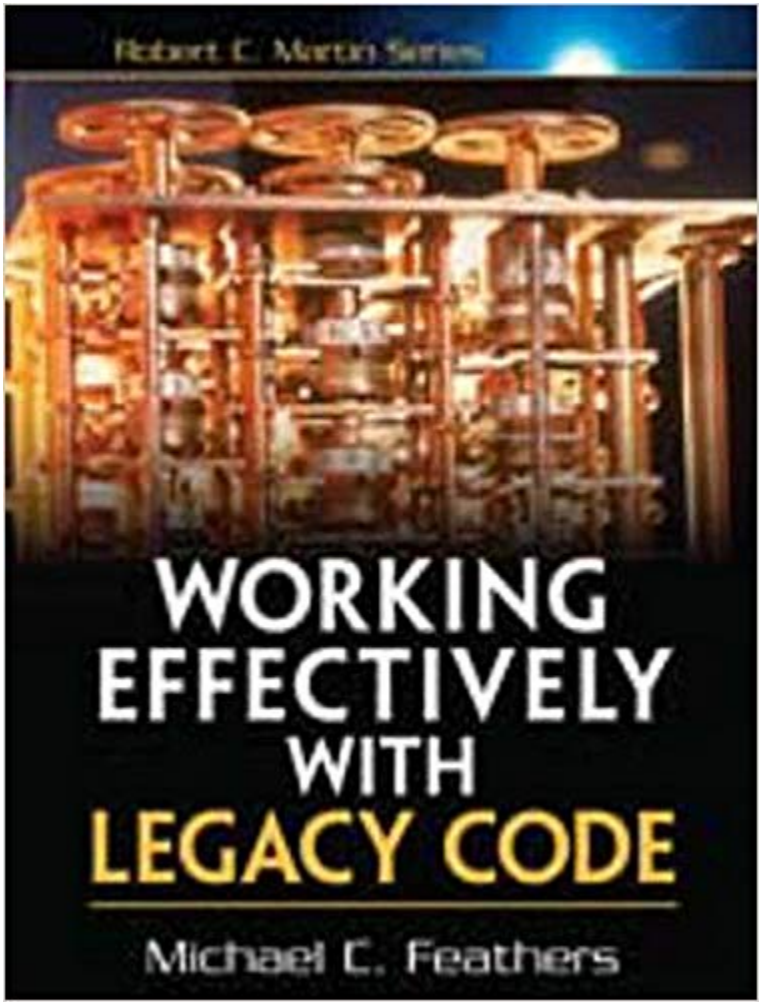
“Capturing a wealth of experience about the design of object-oriented software, four top-notch designers present a catalog of simple and succinct solutions to commonly occurring design problems. Previously undocumented, these 23 patterns allow designers to create more flexible, elegant, and ultimately reusable designs without having to rediscover the design solutions themselves.

The authors begin by describing what patterns are and how they can help you design object-oriented software. They then go on to systematically name, explain, evaluate, and catalog recurring designs in object-oriented systems. With Design Patterns as your guide, you will learn how these important patterns fit into the software development process, and how you can leverage them to solve your own design problems most efficiently.

Each pattern describes the circumstances in which it is applicable, when it can be applied in view of other design constraints, and the consequences and trade-offs of using the

pattern within a larger design. All patterns are compiled from real systems and are based on real-world examples. Each pattern also includes code that demonstrates how it may be implemented in object-oriented programming languages like C++ or Smalltalk."Amazon.com

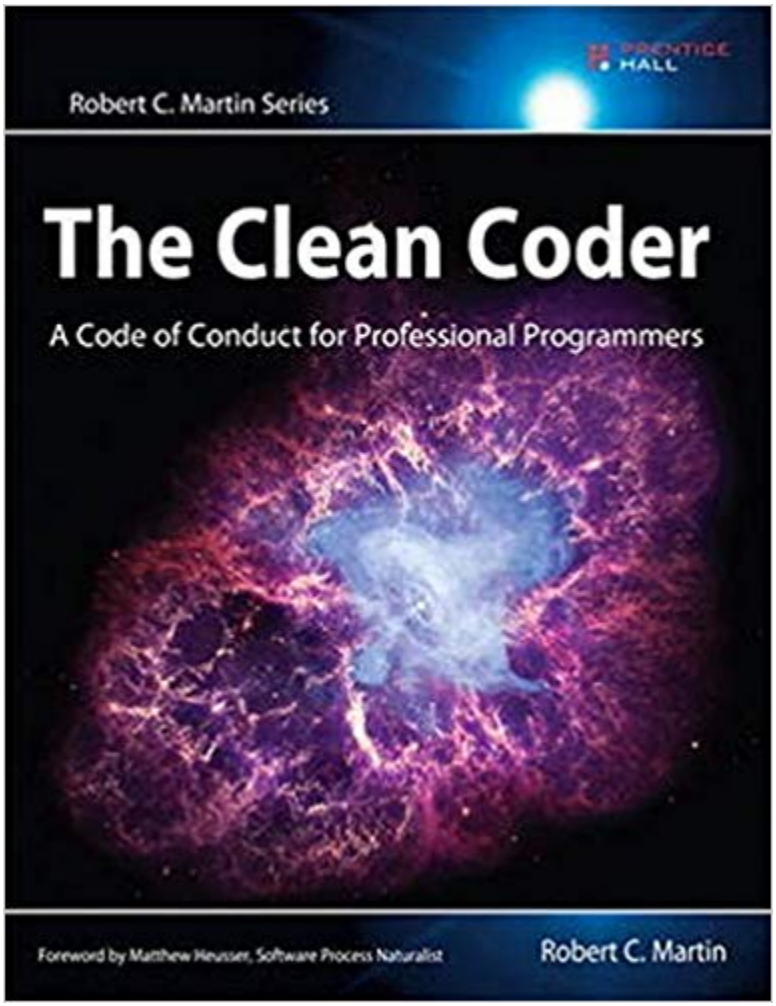
8. Working Effectively with Legacy Code **by Michael Feathers (26.4% recommended)**



"In this book, Michael Feathers offers start-to-finish strategies for working more effectively with large, untested legacy code bases. This book draws on material Michael created for his own renowned

Object Mentor seminars: techniques Michael has used in mentoring to help hundreds of developers, technical managers, and testers bring their legacy systems under control. This book also includes a catalog of twenty-four dependency-breaking techniques that help you work with program elements in isolation and make safer changes."Amazon.com

7. The Clean Coder **by Robert Martin (27.9% recommended)**



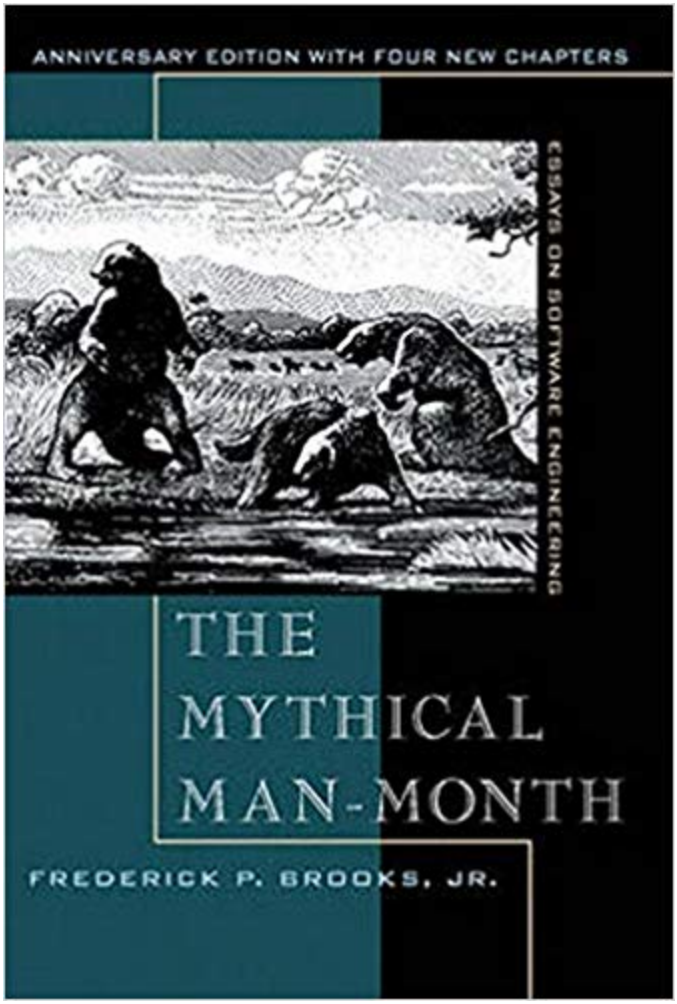
"Programmers who endure and succeed amidst swirling

uncertainty and nonstop pressure share a common attribute: They care deeply about the practice of creating software. They treat it as a craft. They are professionals.

In The Clean Coder: A Code of Conduct for Professional Programmers, legendary software expert Robert C. Martin introduces the disciplines, techniques, tools, and practices of true software craftsmanship. This book is packed with practical advice–about everything from estimating and coding to refactoring and testing. It covers much more than technique: It is about attitude. Martin shows how to approach software development with honor, self-respect, and pride; work well and work clean; communicate and estimate faithfully; face difficult decisions with clarity and honesty; and understand that deep knowledge comes with a responsibility to act.

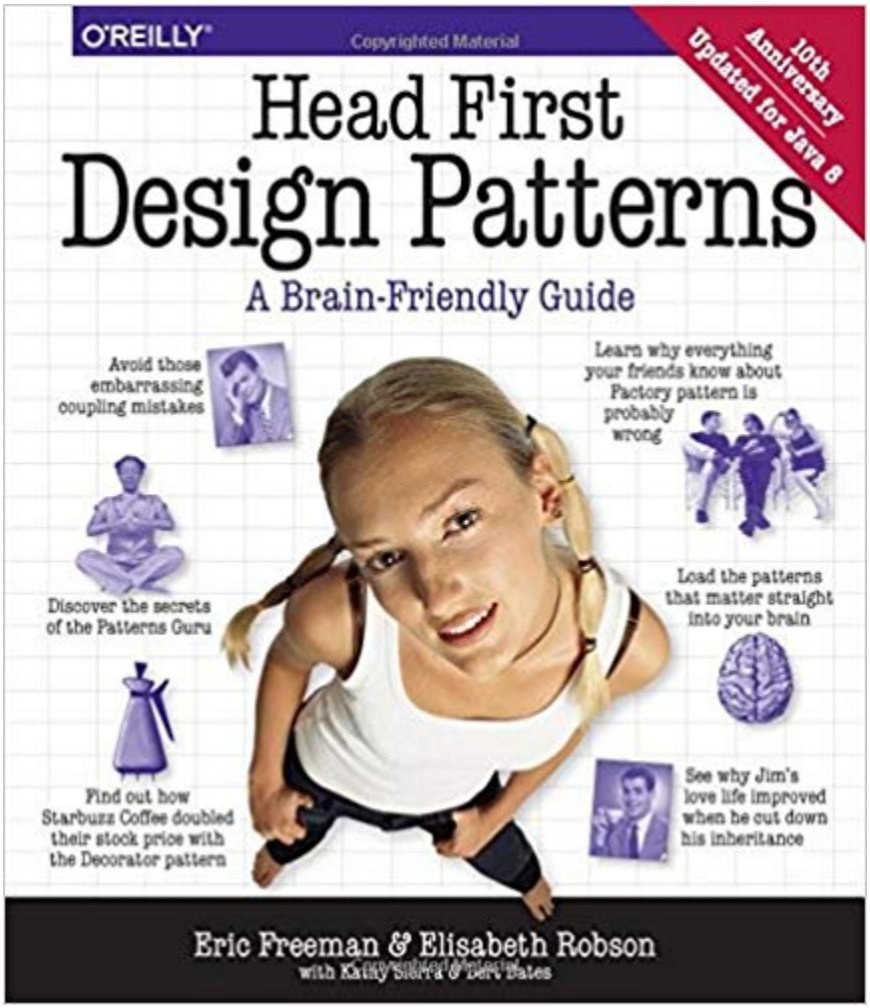
Great software is something to marvel at: powerful, elegant, functional, a pleasure to work with as both a developer and as a user. Great software isn't written by machines. It is written by professionals with an unshakable commitment to craftsmanship. The Clean Coder will help you become one of them–and earn the pride and fulfillment that they alone possess."Amazon.com

6. The Mythical Man-Month **by Frederick P. Brooks Jr (27.9% recommended)**



“Few books on software project management have been as influential and timeless as The Mythical Man-Month. With a blend of software engineering facts and thought-provoking opinions, Fred Brooks offers insight for anyone managing complex projects. These essays draw from his experience as project manager for the IBM System/360 computer family and then for OS/360, its massive software system. Now, 20 years after the initial publication of his book, Brooks has revisited his original ideas and added new thoughts and advice, both for readers already familiar with his work and for readers discovering it for the first time.”Amazon.com

5. Head First Design Patterns **by Eric Freeman / Bert Bates / Kathy Sierra / Elisabeth Robson (29.4% recommended)**



“What’s so special about design patterns?

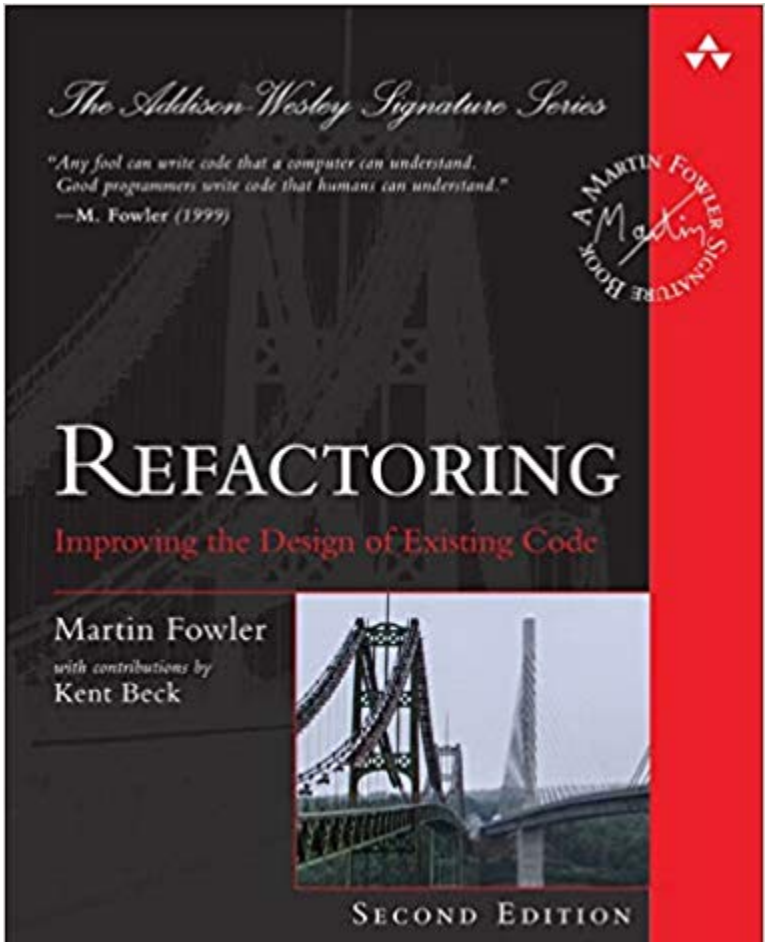
At any given moment, someone struggles with the same software design problems you have. And, chances are, someone else has already solved your problem. This edition of Head First Design Patterns—now updated for Java 8—shows you the tried-and-true, road-tested patterns used by developers to create functional, elegant, reusable, and flexible software. By the time you finish this book, you’ll be able to take

advantage of the best design practices and experiences of those who have fought the beast of software design and triumphed.

What's so special about this book?

We think your time is too valuable to spend struggling with new concepts. Using the latest research in cognitive science and learning theory to craft a multi-sensory learning experience, Head First Design Patterns uses a visually rich format designed for the way your brain works, not a text-heavy approach that puts you to sleep."Amazon.com

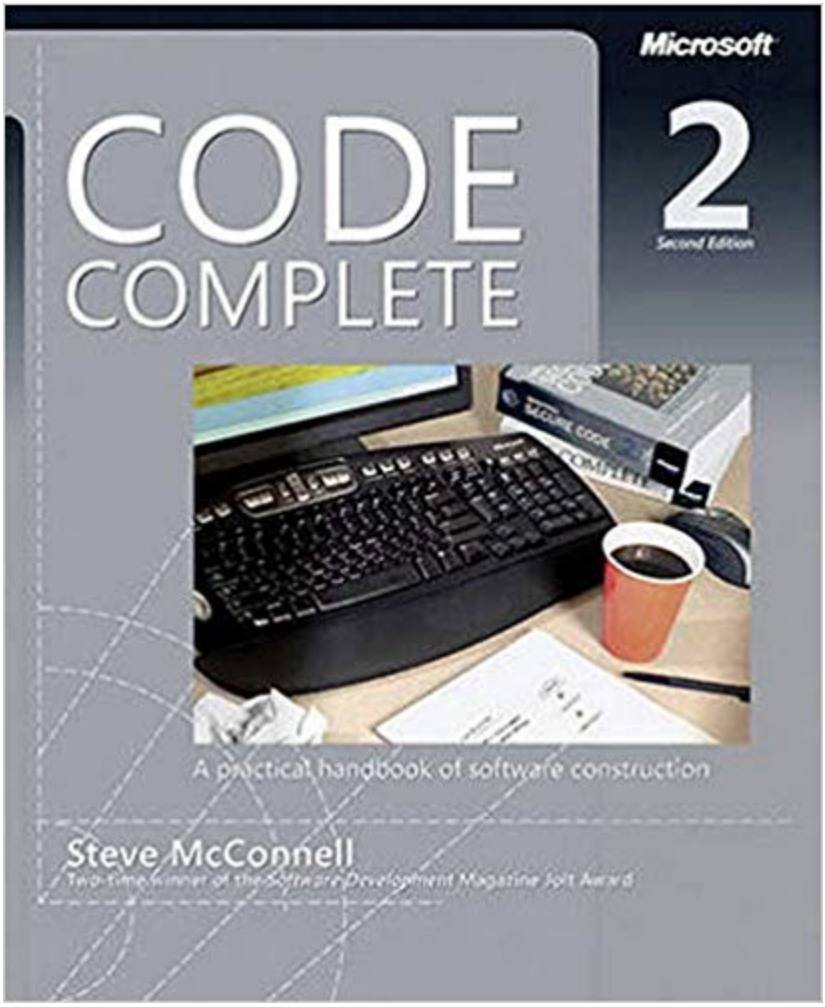
4. Refactoring **by Martin Fowler (35% recommended)**



“As the application of object technology–particularly the Java programming language–has become commonplace, a new problem has emerged to confront the software development community. Significant numbers of poorly designed programs have been created by less-experienced developers, resulting in applications that are inefficient and hard to maintain and extend. Increasingly, software system professionals are discovering just how difficult it is to work with these inherited, non-optimal applications.

For several years, expert-level object programmers have employed a growing collection of techniques to improve the structural integrity and performance of such existing software programs. Referred to as refactoring, these practices have remained in the domain of experts because no attempt has been made to transcribe the lore into a form that all developers could use…until now. In Refactoring: Improving the Design of Existing Software, renowned object technology mentor Martin Fowler breaks new ground, demystifying these master practices and demonstrating how software practitioners can realize the significant benefits of this new process. With proper training a skilled system designer” Amazon.com

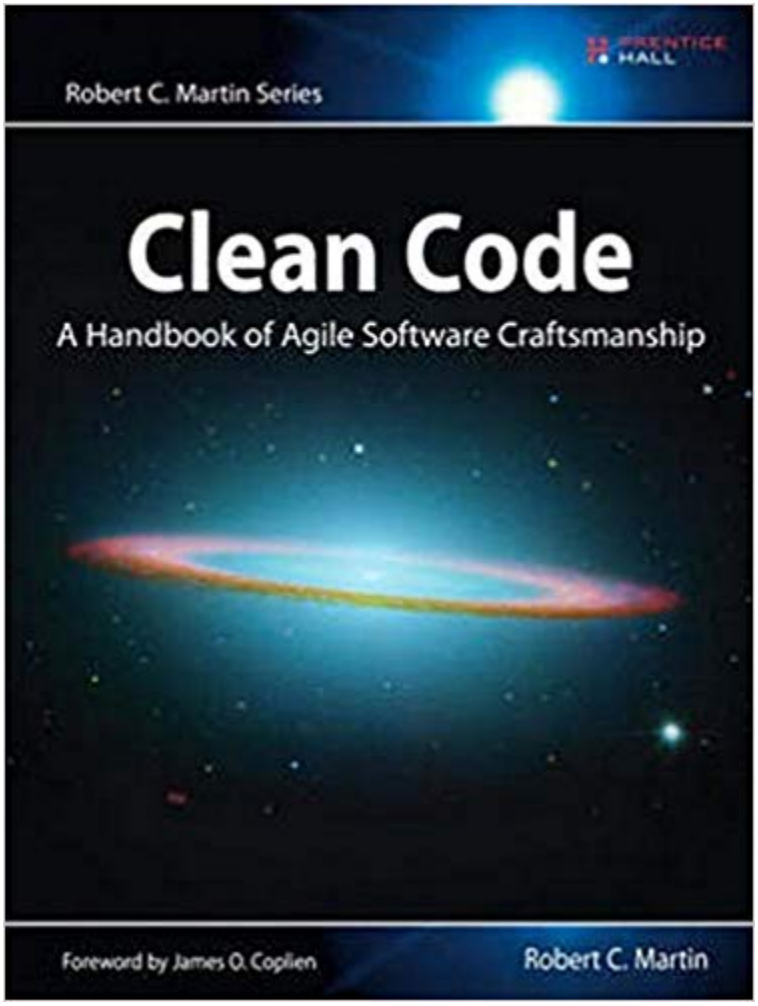
3. Code Complete **by Steve McConnell (42% recommended)**



“Widely considered one of the best practical guides to programming, Steve McConnell’s original CODE COMPLETE has been helping developers write better software for more than a decade. Now this classic book has been fully updated and revised with leading-edge practices—and hundreds of new code samples—illustrating the art and science of software construction. Capturing the body of knowledge available from research, academia, and everyday commercial practice, McConnell synthesizes the most effective techniques and must-know principles into clear, pragmatic guidance. No matter what your experience level, development environment,

or project size, this book will inform and stimulate your thinking—and help you build the highest quality code."Amazon.com

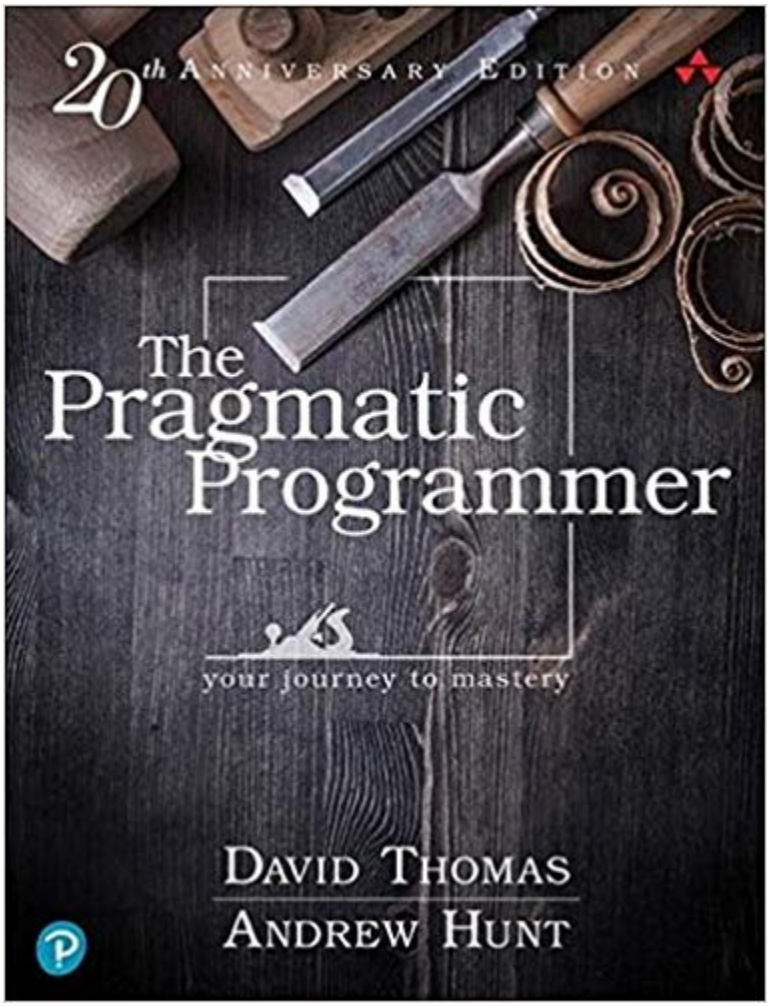
2. Clean Code **by Robert C. Martin (66% recommended)**



"Clean Code is divided into three parts. The first describes the principles, patterns, and practices of writing clean code. The second part consists of several case studies of increasing complexity. Each case study is an exercise in cleaning up code—of transforming a code base that has some problems into one that is sound and efficient. The third part is the payoff: a single

chapter containing a list of heuristics and “smells” gathered while creating the case studies. The result is a knowledge base that describes the way we think when we write, read, and clean code.”Amazon.com

1. The Pragmatic Programmer **by David Thomas & Andrew Hunt (67% recommended)**



“ The Pragmatic Programmer is one of those rare tech books you’ll read, re-read, and read again over the years. Whether you’re new to the field or an experienced practitioner, you’ll come away with fresh insights each and every time.

Dave Thomas and Andy Hunt wrote the first edition of this influential book in 1999 to help their clients create better software and rediscover the joy of coding. These lessons have helped a generation of programmers examine the very essence of software development, independent of any particular language, framework, or methodology, and the Pragmatic philosophy has spawned hundreds of books, screencasts, and audio books, as well as thousands of careers and success stories.

Now, twenty years later, this new edition re-examines what it means to be a modern programmer. Topics range from personal responsibility and career development to architectural techniques for keeping your code flexible and easy to adapt and reuse."Amazon.com

Conclusion

Although the order might suprise some, by definition, most of you must have heard of these books already.

A few additional things I learned making this list:

- Martin Fowler and Steve McConnell are the only author with several books in the list.
- Cracking to Code interview is the most recent book on the list, released in 2015.
- Python Programming, by John Zelle was the most cited book focused on one laguage. It would have #5 had I taken it into

account.

I hope you enjoyed this article.

I've made two others in the same style:

- Best Startup books
- Best Python books

I must admit, this one took a while to write. If you liked this article and feel like Twitter would like it please do not hesitate to love and retweet, it really does help :).

> Back at it again.
>
> This time I wanted to know what were the most recommended programming books ever 🏆
>
> So I've compiled more than 1200 recommendations from 68 lists and came up with this top 25 most recommended programming books of all-time📕
>
> THREAD👇
>
> — Pierre de Wulf (@PierreDeWulf) February 18, 2020

edit: Robert C. Martin also have 2 books on the list.