Simple Programmer

Making the complex simple

# 10 ESSENTIAL BOOKS EVERY DEVELOPER SHOULD READ

## AUTHOR

Hi, I'm John Sonmez, the Founder of Simple Programmer and author of "Soft Skills: The Software Developer's Life Manual".

JOHN SONMEZ

I created Simple Programmer with the goal of making the complex simple.

My goal for Simple Programmer is not just to help you get a better job and make more money—although I certainly want to do that—but to help you improve in all areas of your life.

To help you build self-confidence, to be more productive, to find and reach your goals, and even to explore the wonderful world of entrepreneurship.

In short, I want to help you succeed and become a better version of yourself each and every day.

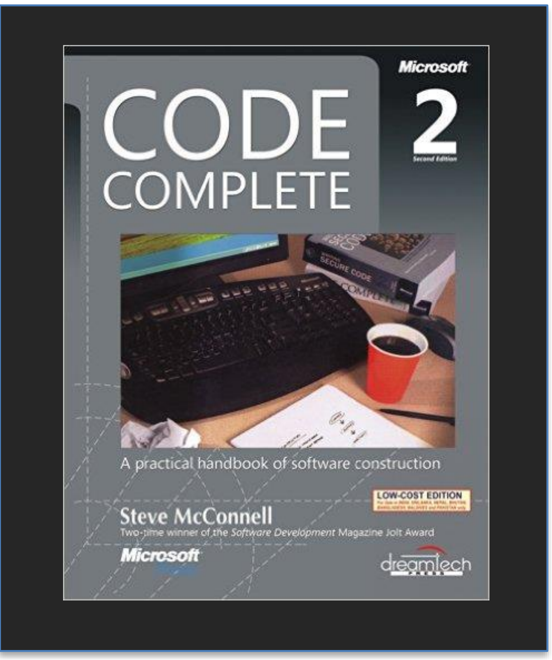
# INTRODUCTION

A lot of viewers asked me to create a **top 10 list of my favorite programming books**. After taking some time to think, I decided to create this list with my top 10 books that every software developer should read.

These are, for me, the best programming books I've read in my entire life. I also think these are mandatory reading for anyone that might be interested in improving their lives, career and software development skills.

All books were thought and listed here with the purpose of improving diferente áreas of your software developer career. Reading those books will improve your overall performance, which will definitely make a huge impact on your career and other areas as well.

## 1) Code Complete


Microsoft
CODE COMPLETE
2
Second Edition
A practical handbook of software construction
LOW-COST EDITION
Steve McConnell
Two-time winner of the Software Development Magazine Jolt Award
Microsoft
dreamtech

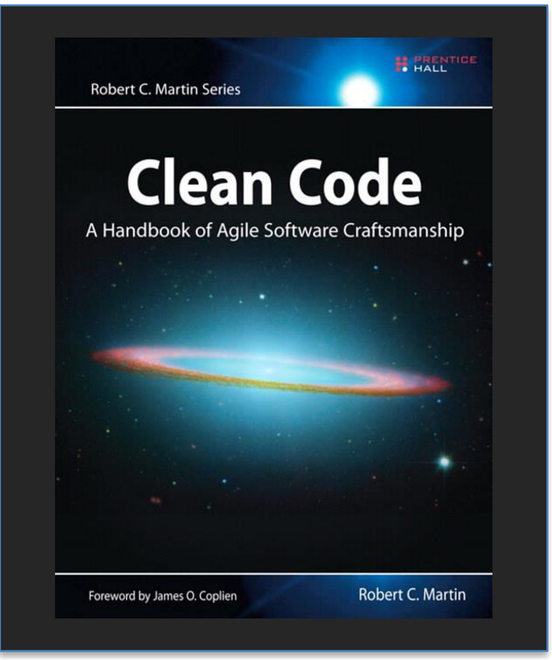
Widely considered one of the best practical guides to programming, Steve McConnell's original CODE COMPLETE has been helping developers write better software for more than a decade. Now this classic book has been fully updated and revised with leading-edge practices—and hundreds of new code samples—illustrating 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.

Discover the timeless techniques and strategies that help you: Design for minimum complexity and maximum creativity, reap the benefits of collaborative development, apply defensive programming techniques to reduce and flush out errors, exploit opportunities to refactor—or evolve—code, and do it safely.

CLICK HERE TO BUY

## 2) Clean Code


Robert C. Martin Series
PRENTICE HALL
Clean Code
A Handbook of Agile Software Craftsmanship
Foreword by James O. Coplien
Robert C. Martin

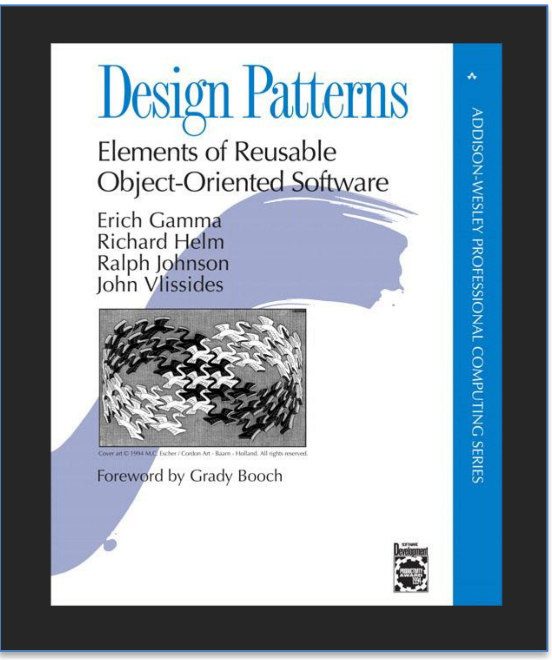
Even bad code can function. But if code isn't clean, it can bring a development organization to its knees. Every year, countless hours and significant resources are lost because of poorly written code. But it doesn't have to be that way.

Noted software expert Robert C. Martin presents a revolutionary paradigm with Clean Code: A Handbook of Agile Software Craftsmanship . Martin has teamed up with his colleagues from Object Mentor to distill their best agile practice of cleaning code “on the fly” into a book that will instill within you the values of a software craftsman and make you a better programmer—but only if you work at it.

What kind of work will you be doing? You'll be reading code—lots of code. And you will be challenged to think about what's right about that code, and what's wrong with it. More importantly, you will be challenged to reassess your professional values and your commitment to your craft.

CLICK HERE TO BUY

## 3) Design Patterns

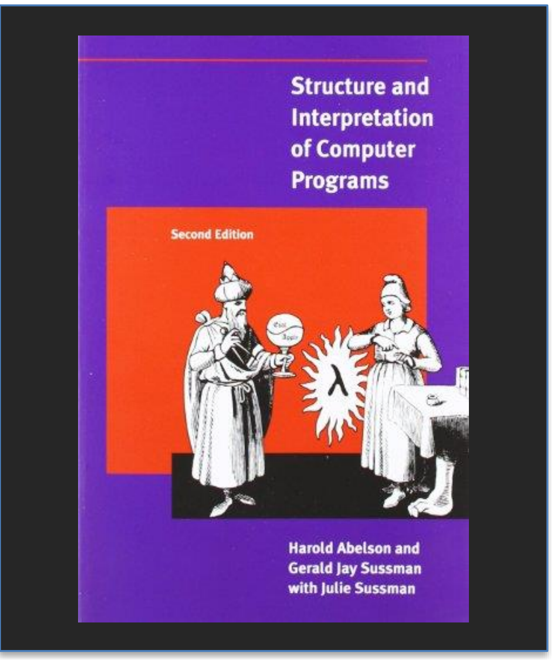

Design Patterns
Elements of Reusable Object-Oriented Software
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides
Foreword by Grady Booch
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

Design Patterns is a modern classic in the literature of object-oriented development, offering timeless and elegant solutions to common problems in software design. It describes patterns for managing object creation, composing objects into larger structures, and coordinating control flow between objects. The book provides numerous examples where using composition rather than inheritance can improve the reusability and flexibility of code. Note, though, that it's not a tutorial but a catalog that you can use to find an object-oriented design pattern that's appropriate for the needs of your particular application--a selection for virtuoso programmers who appreciate (or require) consistent, well-engineered object-oriented designs.

This is one of the best written and wonderfully insightful books that I have read in a great long while...this book establishes the legitimacy of patterns in the best way: not by argument, but by example.

CLICK HERE TO BUY

## 4) Structure and Interpretation of Computer Programs

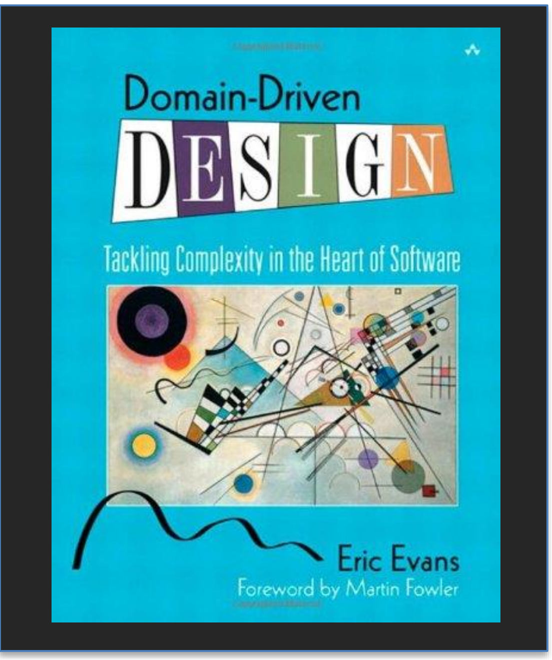

Structure and Interpretation of Computer Programs
Second Edition
Harold Abelson and Gerald Jay Sussman with Julie Sussman

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 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.

CLICK HERE TO BUY

## 5) Domain-Driven Design

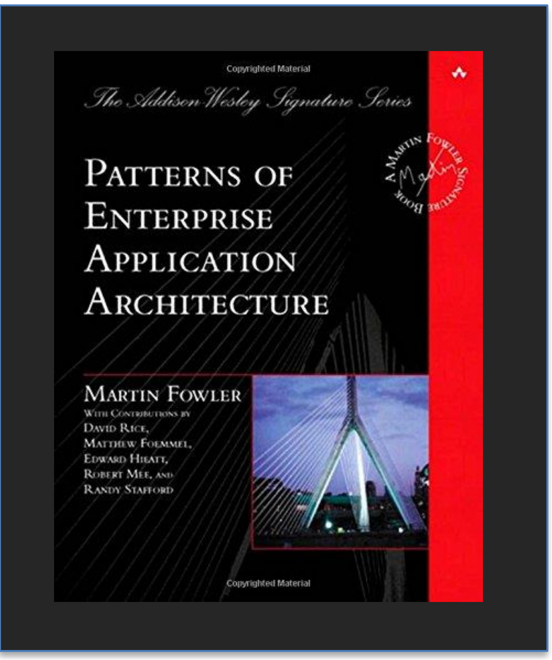

Domain-Driven
DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler

The software development community widely acknowledges that domain modeling is central to software design. Through domain models, software developers are able to express rich functionality and translate it into a software implementation that truly serves the needs of its users. But despite its obvious importance, there are few practical resources that explain how to incorporate effective domain modeling into the software development process.

Domain-Driven Design fills that need. This is not a book about specific technologies. It offers readers a systematic approach to domain-driven design, presenting an extensive set of design best practices, experience-based techniques, and fundamental principles that facilitate the development of software projects facing complex domains. Intertwining design and development practice, this book incorporates numerous examples based on actual projects to illustrate the application of domain-driven design to real-world software development.

CLICK HERE TO BUY

## 6) Patterns of Enterprise Application Architecture

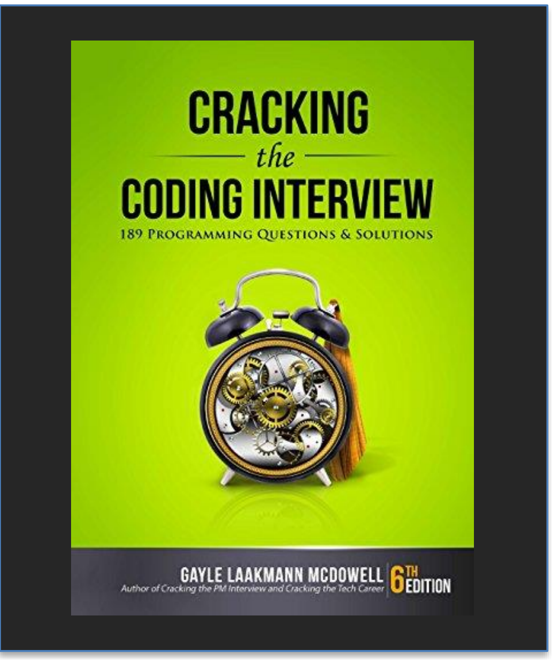


The practice of enterprise application development has benefited from the emergence of many new enabling technologies. Multi-tiered object-oriented platforms, such as Java and .NET, have become commonplace. These new tools and technologies are capable of building powerful applications, but they are not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

Patterns of Enterprise Application Architecture is written in direct response to the stiff challenges that face enterprise application developers. The author, noted object-oriented designer Martin Fowler, noticed that despite changes in technology--from Smalltalk to CORBA to Java to .NET--the same basic design ideas can be adapted and applied to solve common problems. Martin distills over forty recurring solutions into patterns. The result is an indispensable handbook of solutions that are applicable to any enterprise application platform.

CLICK HERE TO BUY

## 7) Cracking The Coding Interview


CRACKING
the
CODING INTERVIEW
189 PROGRAMMING QUESTIONS & SOLUTIONS
GAYLE LAAKMANN MCDOWELL
Author of Cracking the PM Interview and Cracking the Tech Career
6TH EDITION

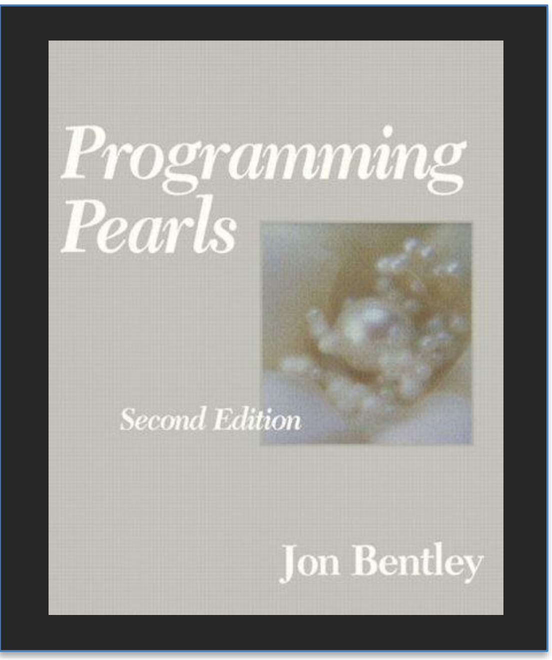
I am not a recruiter. I am a software engineer. And as such, I know what it's like to be asked to whip up brilliant algorithms on the spot and then write flawless code on a whiteboard. I've been through this as a candidate and as an interviewer.

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.

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.

CLICK HERE TO BUY

## 8) Programming Pearls



Fourteen years after it was first issued, C++ expert Jon Bentley reinvents a true classic with the second edition of his Programming Pearls.

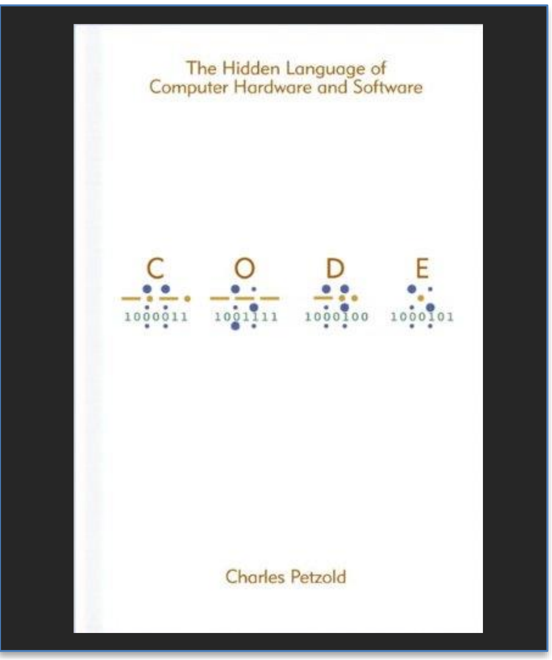
Completely revised and brought up to date with all new code examples in C and C++, this book remains an exceptional tutorial for learning to think like a programmer.

The "pearls" in question center not only on choosing the right algorithms (like binary searches, sorting techniques, or sparse arrays) but also on showing how to solve problems effectively.

Each chapter frames a particular programming task--such as sorting numbers, creating anagrams, or counting the words in a block of text--many drawn from Bentley's experiences in his long career as a developer. The book traces the process of arriving at a fast, efficient, and accurate solution, along with code profiling to discover what works best. After refining the correct answer, each chapter enumerates programming principles that you can use on your own.

CLICK HERE TO BUY

## 9) Code: The Hidden Language of Computer



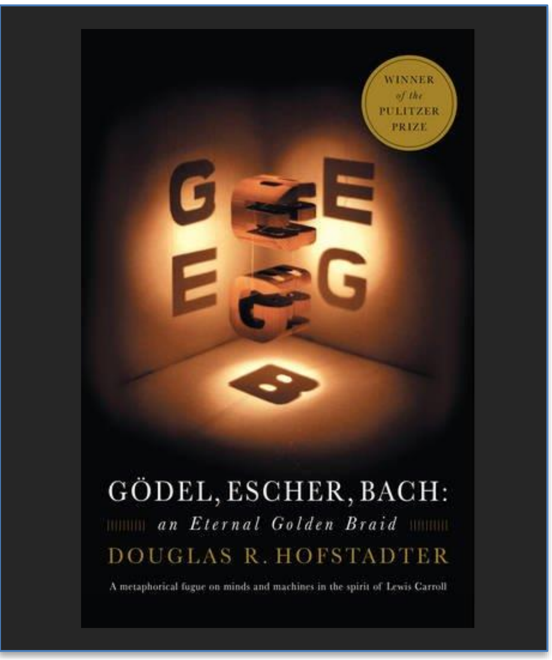
Charles Petzold's latest book, Code: The Hidden Language of Computer Hardware and Software, crosses over into general-interest nonfiction from his usual programming genre. It's a carefully written, carefully researched gem that will appeal to anyone who wants to understand computer technology at its essence.

Readers learn about number systems (decimal, octal, binary, and all that) through Petzold's patient (and frequently entertaining) prose and then discover the logical systems that are used to process them.

There's loads of historical information too. From Louis Braille's development of his eponymous raised-dot code to Intel Corporation's release of its early microprocessors, Petzold presents stories of people trying to communicate with (and by means of) mechanical and electrical devices. It's a fascinating progression of technologies, and Petzold presents a clear statement of how they fit together.

CLICK HERE TO BUY

## 10) Gödel, Escher, Bach


WINNER of the PULITZER PRIZE
GÖDEL, ESCHER, BACH:
an Eternal Golden Braid
DOUGLAS R. HOFSTADTER
A metaphorical fugue on minds and machines in the spirit of Lewis Carroll

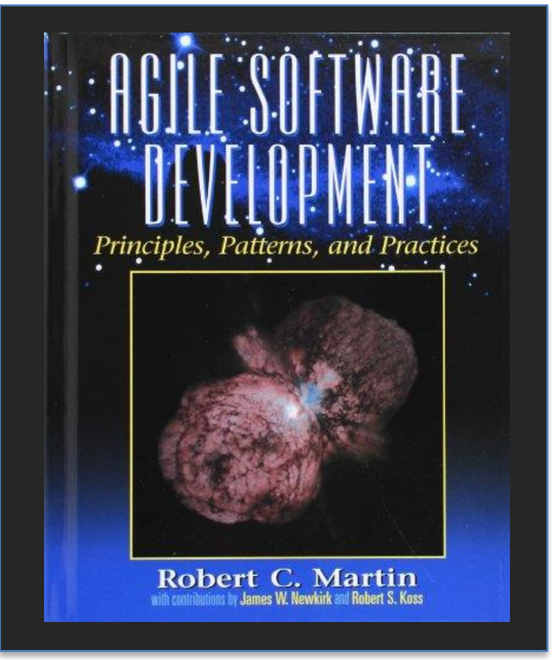
Besides being a profound and entertaining meditation on human thought and creativity, this book looks at the surprising points of contact between the music of Bach, the artwork of Escher, and the mathematics of Gödel. It also looks at the prospects for computers and artificial intelligence (AI) for mimicking human thought. For the general reader and the computer techie alike, this book still sets a standard for thinking about the future of computers and their relation to the way we think.

Douglas Hofstadter's book is concerned directly with the nature of "maps" or links between formal systems. However, according to Hofstadter, the formal system that underlies all mental activity transcends the system that supports it. If life can grow out of the formal chemical substrate of the cell, if consciousness can emerge out of a formal system of firing neurons, then so too will computers attain human intelligence. Gödel, Escher, Bach is a wonderful exploration of fascinating ideas at the heart of cognitive science: meaning, reduction, recursion, and much more.

CLICK HERE TO BUY

# BONUS BOOKS

## 11) Agile Software Development Principles


AGILE SOFTWARE DEVELOPMENT
Principles, Patterns, and Practices
Robert C. Martin
with contributions by James W. Newkirk and Robert S. Koss

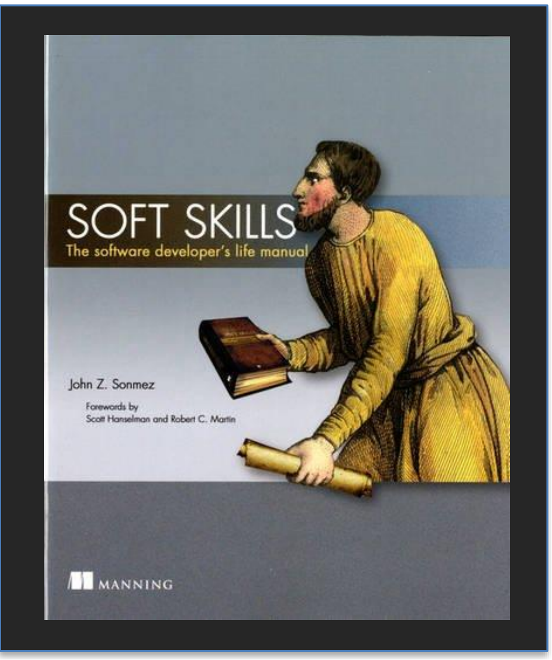
Written by a software developer for software developers, this book is a unique collection of the latest software development methods. The author includes OOD, UML, Design Patterns, Agile and XP methods with a detailed description of a complete software design for reusable programs in C++ and Java.

Using a practical, problem-solving approach, it shows how to develop an object-oriented application—from the early stages of analysis, through the low-level design and into the implementation. Walks readers through the designer's thoughts — showing the errors, blind alleys, and creative insights that occur throughout the software design process.

The book covers: Statics and Dynamics; Principles of Class Design; Complexity Management; Package Design; Analysis and Design; Patterns and Paradigm Crossings. Explains the principles of OOD, one by one, and then demonstrates them with numerous examples, completely worked-through designs, and case studies.

CLICK HERE TO BUY

## 12) Soft Skills


SOFT SKILLS
The software developer's life manual
John Z. Sonmez
Forewords by
Scott Hanselman and Robert C. Martin
MANNING

Most software development books are about...software development— this one isn't. There are plenty of books out there about writing good code and using various technologies, but I've been hard-pressed to find a book that told me how to be a good software developer.

This book isn't about what you can do. This book is about...you. That's right. It's about your career, your life, your body, your mind, and—if you believe there is such a thing—your soul. Now, I don't want you to think I'm some kind of lunatic. I'm not a transcendentalist monk sitting on the floor meditating while smoking Peyote leaves, trying to help you ascend to a higher state of consciousness. On the contrary, I think you'll find I'm a pretty down-to-earth kind of guy who just happens to think that being a software developer is about a whole lot more than writing code.

This book is divided into seven sections, each focusing on a different aspect of your life as a software developer.

CLICK HERE TO BUY