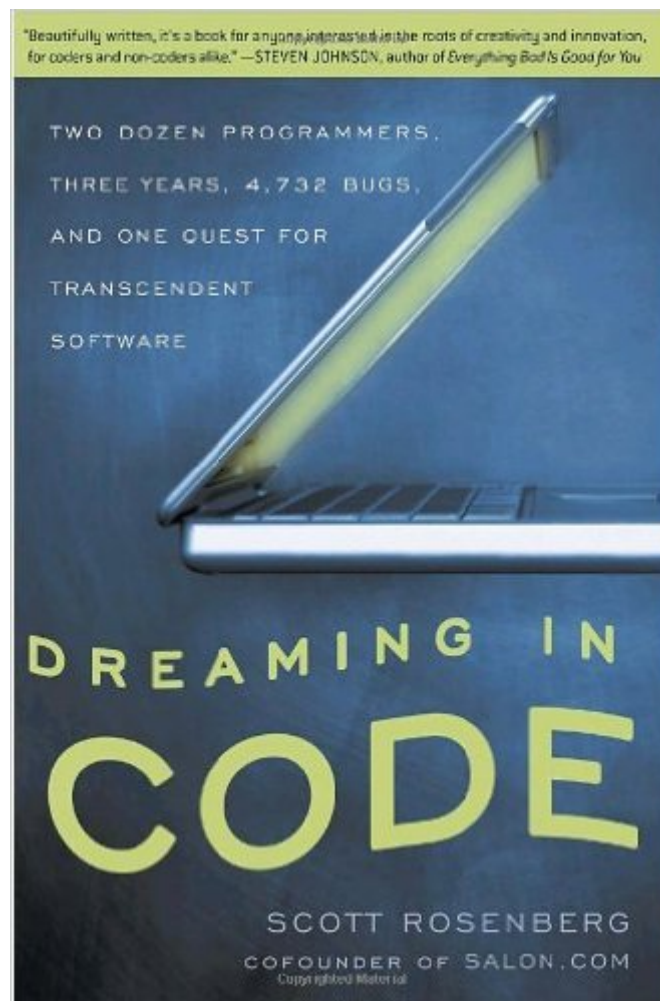# Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software

Synopsis

Our civilization runs on software. Yet the art of creating it continues to be a dark mystery, even to the experts. To find out why itâ ™s so hard to bend computers to our will, Scott Rosenberg spent three years following a team of maverick software developersâ ”led by Lotus 1-2-3 creator Mitch Kaporâ ”designing a novel personal information manager meant to challenge market leader Microsoft Outlook. Their story takes us through a maze of abrupt dead ends and exhilarating breakthroughs as they wrestle not only with the abstraction of code, but with the unpredictability of human behaviorâ ” especially their own.

Book Information

Paperback: 416 pages

Publisher: Crown Business; Reprint edition (February 26, 2008)

Language: English

ISBN-10: 1400082471

ISBN-13: 978-1400082476

Product Dimensions: 5.2 x 0.9 x 8 inches

Shipping Weight: 9.6 ounces (View shipping rates and policies)

Average Customer Review: 3.9 out of 5 starsÂ Â See all reviewsÂ (101 customer reviews)

Best Sellers Rank: #89,473 in Books (See Top 100 in Books)   #2 inÂ Books > Computers & Technology > Programming > Languages & Tools > Debugging   #101 inÂ Books > Textbooks > Computer Science > Software Design & Engineering   #103 inÂ Books > Business & Money > Industries > Computers & Technology

Customer Reviews

Santa came through this year with a slightly advance copy of "Dreaming in Code", which tries to do for software engineering what "The Soul of a New Machine" did for computer engineering, following a single project through to its attempted conclusion. Software development is a story that's very rarely told, considering how dramatically software has changed all of our lives in the last 30 years. Author Scott Rosenberg does a good job of conveying the difficulties in software engineering, and the inevitable headaches and drama that come with incomplete plans and shifting specs (and they're always incomplete and shifting).Where Rosenberg went wrong, unfortunately, is his choice of project to follow. Mitch Kapor's Chandler is quite atypical of software projects: it's driven entirely by one man's quixotic vision, and never has to encounter the usual give-and-take with VC's or upper management that help to clarify a plan. Kapor comes off as an untethered idealist (Al Gore

makes the obligatory cameo at the office), and his project is afflicted by the same we-are-the-world unseriousness as his politics. Most notably, Kapor decides there should be no central repository for data (because, hey, down with authority and all that): instead, every item will just be represented, Napster-style, across users' personal computers. It's a costly decision that I don't think would have been made if it were more than just Kapor running the show.Actually, I think the strongest part of the book is when Rosenberg abandons the project entirely in the middle section to delve into the history of the programming discipline, noting everyone from Donald Knuth to 37signals' Jason Fried. It's a useful, lucid introduction to the field that contains stories I hadn't seen before.

I have been following the Chandler project with interest from the beginning and have checked out most of the releases. Even to a distant observer,the glacial progress of the project, the lack of clarity about the objectives, and the mid-course changes that have reduced it to a pale shadow of the original vision, have been baffling.So, when I noticed Rosenberg's book, I immediately bought it and devoured it in a couple of hours. While the main focus of the book is on Chandler, the OSAF and the personalities there, it does talk quite a bit about the difficulties of software development, history of programming languages, methodologies etc quite a bit. I suspect the actual Chandler & OSAF content page wise might be only 50-60%. The book's general approach is like this - start out with something that the OSAF tried to do, then drift into a related topic from the history of software development; it alternates between discussions of the development of Chandler and various other topics. Very rarely does Scott Rosenberg actually examine why certain development efforts at OSAF failed in depth - he relies too much on failures of 'proprietary' software projects as a way of explaining the failures of the Chandler project. There is no open source context to the discussion most of the time - examining some aspect of software development of an open source project and drawing parallels from the failure of some proprietary software development effort is confusing. One of the most puzzling things about a lot of software doomsday scenario books is that, in spite of the fact that not all projects fail, they never try to figure out what made the successful ones work. Maybe nothing sells like a disaster. This book is no exception.

Dreaming in Code: Two Dozen Programmers, Three Years, 4,732 Bugs, and One Quest for Transcendent Software 2015 International Residential Code for One- and Two- Family Dwellings and Significant Changes to the 2015 International Residential Code 2012 International Residential Code for One- and Two- Family Dwellings (International Code Council Series) 2009 International Residential Code For One-and-Two Family Dwellings: Soft Cover Version (International Code

Council Series) 2009 International Residential Code For One-and-Two Family Dwellings: Looseleaf Version (International Code Council Series) How Many Bugs in a Box?: A Pop-up Counting Book (David Carter's Bugs) An Illustrated Guide for z/Architecture Assembler Programmers: A compact reference for application programmers Delphi Nuts & Bolts for Experienced Programmers: For Experienced Programmers Desert to Dream: A Dozen Years of Burning Man Photography, Revised Edition 2012 International Plumbing Code (Includes International Private Sewage Disposal Code) (International Code Council Series) The House of the Four Winds (One Dozen Daughters) Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection: Obfuscation, Watermarking, and Tamperproofing for Software Protection Software Engineering Classics: Software Project Survival Guide/ Debugging the Development Process/ Dynamics of Software Development (Programming/General) Cthulhu Lies Dreaming: Twenty-three Tales of the Weird and Cosmic The Clean Coder: A Code of Conduct for Professional Programmers (Robert C. Martin Series) One Dream Only (One Two Three volume 2) Application Debugging: An MVS Abend Handbook for Cobol, Assembly, PL/I, and Fortran Programmers (Prentice-Hall Software Series) Code Talker: The First and Only Memoir By One of the Original Navajo Code Talkers of WWII The Agile Samurai: How Agile Masters Deliver Great Software (Pragmatic Programmers) Jobs Rated Almanac: The Best and Worst Jobs - 250 in All - Ranked by More Than a Dozen Vital Factors Including Salary, Stress, Benefits, and More (Jobs Rated Almanac, 6th Ed, 2002)

[Dmca](#)