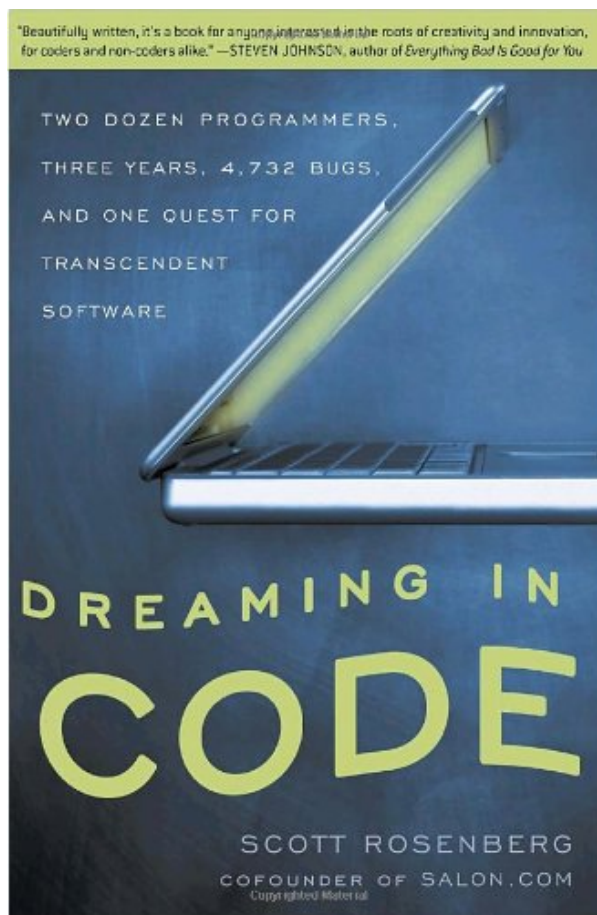


**DREAMING IN CODE: TWO DOZEN  
PROGRAMMERS, THREE YEARS, 4,732  
BUGS, AND ONE QUEST FOR  
TRANSCENDENT SOFTWARE BY SCOTT  
ROSENBERG**



**DOWNLOAD EBOOK : DREAMING IN CODE: TWO DOZEN PROGRAMMERS,  
THREE YEARS, 4,732 BUGS, AND ONE QUEST FOR TRANSCENDENT  
SOFTWARE BY SCOTT ROSENBERG PDF**



"Beautifully written, it's a book for anyone interested in the roots of creativity and innovation, for coders and non-coders alike." —STEVEN JOHNSON, author of *Everything Bad Is Good for You*

TWO DOZEN PROGRAMMERS,  
THREE YEARS, 4,732 BUGS,  
AND ONE QUEST FOR  
TRANSCENDENT  
SOFTWARE



DREAMING IN  
CODE

SCOTT ROSENBERG  
COFOUNDER OF SALON.COM  
Copyrighted Material

Click link bellow and free register to download ebook:  
**DREAMING IN CODE: TWO DOZEN PROGRAMMERS, THREE YEARS, 4,732 BUGS, AND  
ONE QUEST FOR TRANSCENDENT SOFTWARE BY SCOTT ROSENBERG**

[DOWNLOAD FROM OUR ONLINE LIBRARY](#)

# **DREAMING IN CODE: TWO DOZEN PROGRAMMERS, THREE YEARS, 4,732 BUGS, AND ONE QUEST FOR TRANSCENDENT SOFTWARE BY SCOTT ROSENBERG PDF**

Well, still puzzled of the best ways to get this book *Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software* By Scott Rosenberg right here without going outside? Just connect your computer or gadget to the internet and also begin downloading and install *Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software* By Scott Rosenberg Where? This web page will reveal you the web link page to download *Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software* By Scott Rosenberg You never fret, your favourite book will certainly be earlier yours now. It will certainly be a lot easier to enjoy reading *Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software* By Scott Rosenberg by on the internet or getting the soft data on your gizmo. It will despite that you are and just what you are. This publication *Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software* By Scott Rosenberg is created for public and also you are just one of them who can take pleasure in reading of this publication [Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software By Scott Rosenberg](#)

## Amazon.com Review

In the 80s, Tracy Kidder's *The Soul of a New Machine* attempted to define the story of the development of a minicomputer: from the new science to the business and nascent culture of electronic hardware and software that was characteristic of that time. Scott Rosenberg's *Dreaming in Code* draws on Kidder's model as it attempts to document the state of software, the Internet, and everything circa 2006 through the lens of Chandler, an as-yet-unfinished software application for the management of personal information.

The Chandler project--driven by Mitch Kapor, the founder of Lotus Development and main designer of its 1-2-3 spreadsheet, and later co-founder of the Electronic Frontier Foundation--isn't the primary point of *Dreaming in Code*, though reading about software people and their social behavior is at least as interesting as reading about that of meerkats or monkeys. Rather, Chandler is a rhetorical device with which Rosenberg takes on the big questions: How do software development teams work (or not)? Why does the reuse of software modules rarely work altogether correctly? Does open-source development by volunteers on the Internet lead to innovation or just insanely bifurcated chaos? Chandler helps his readers think more clearly about all of these issues; however, "answers" to these questions are, of course, not to be had, which is one of his points.

The problem with books about technical subjects that aspire to appeal to a general audience, particularly computers and software, is that such subjects are so far outside the realm of familiarity of most people that the prose bogs down in analogy and metaphor. Rosenberg manages to avoid too much of that and deliver a readable account of software development and culture. --David Wall

#### From Publishers Weekly

Software is easy to make, except when you want it to do something new," Rosenberg observes—but the catch is that "the only software worth making is software that does something new." This two-tiered insight comes from years of observing a team led by Mitch Kapor (the creator of the Lotus 1-2-3 spreadsheet) in its efforts to create a "personal information manager" that can handle to-do lists as easily as events scheduling and address books. Rosenberg's fly-on-the-wall reporting deftly charts the course taken by Kapor's Open Source Applications Foundation, while acknowledging that every software programmer finds his or her own unique path to a brick wall in the development process. (The software is still in development even now.) With equal enthusiasm, Rosenberg digs into the history of the computer industry's efforts to make programming a more efficient process. Though there's a lot of technical information, it's presented in very accessible terms, primarily through the context of project management. Even readers whose computer expertise ends at retrieving their e-mail will be able to enjoy digressions into arcane subjects like object-oriented programming. (Jan.)

Copyright © Reed Business Information, a division of Reed Elsevier Inc. All rights reserved.

#### From Booklist

Programmers practice a most abstract and incomprehensible art; yet, as end users, we take them for granted and demand perfection from the software they create. Rosenberg, a theater and movie critic turned technology columnist and founder of the Web magazine Salon.com, attempts to shed light on the day-to-day realities of what turns out to be a Herculean task: getting computers to do what we want them to. He spent three years following the course of development of a software program code named Chandler, a combination calendar, to-do list, e-mail manager, and personal database. This open-source project was the dream child of Mitch Kapor, the creator of Lotus 1-2-3, who envisioned a simple, elegant interface capable of easy storage and retrieval of any number of personal data. The practical matter of creation was another story, however, and we learn how the fits and starts of software engineering make even creating a "simple" program an arduous task. Although this is not edge-of-your-seat stuff, it is highly instructive for anyone planning on "managing" a software project. David Siegfried

Copyright © American Library Association. All rights reserved

# **DREAMING IN CODE: TWO DOZEN PROGRAMMERS, THREE YEARS, 4,732 BUGS, AND ONE QUEST FOR TRANSCENDENT SOFTWARE BY SCOTT ROSENBERG PDF**

[Download: DREAMING IN CODE: TWO DOZEN PROGRAMMERS, THREE YEARS, 4,732 BUGS, AND ONE QUEST FOR TRANSCENDENT SOFTWARE BY SCOTT ROSENBERG PDF](#)

**Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software By Scott Rosenberg** When writing can transform your life, when composing can enhance you by offering much cash, why don't you try it? Are you still very baffled of where understanding? Do you still have no concept with what you are visiting create? Now, you will certainly require reading *Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software By Scott Rosenberg* An excellent author is an excellent reader simultaneously. You can specify just how you write relying on exactly what publications to review. This *Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software By Scott Rosenberg* could assist you to fix the issue. It can be one of the right sources to create your composing skill.

As one of the window to open up the new world, this *Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software By Scott Rosenberg* supplies its outstanding writing from the author. Published in one of the preferred authors, this book *Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software By Scott Rosenberg* turns into one of one of the most desired books recently. In fact, the book will certainly not matter if that *Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software By Scott Rosenberg* is a best seller or otherwise. Every publication will still give finest sources to get the user all finest.

However, some people will certainly seek for the best seller publication to check out as the initial reference. This is why; this *Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software By Scott Rosenberg* is presented to satisfy your necessity. Some individuals like reading this book *Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software By Scott Rosenberg* because of this prominent book, however some love this as a result of preferred author. Or, numerous additionally like reading this publication *Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software By Scott Rosenberg* since they truly have to read this publication. It can be the one that really like reading.

# **DREAMING IN CODE: TWO DOZEN PROGRAMMERS, THREE YEARS, 4,732 BUGS, AND ONE QUEST FOR TRANSCENDENT SOFTWARE BY SCOTT ROSENBERG PDF**

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 it's so hard to bend computers to our will, Scott Rosenberg spent three years following a team of maverick software developers—led by Lotus 1-2-3 creator Mitch Kapor—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 behavior— especially their own.

- Sales Rank: #262521 in Books
- Brand: Three Rivers Press
- Model: 3626272
- Published on: 2008-02-26
- Released on: 2008-02-26
- Original language: English
- Number of items: 1
- Dimensions: 8.00" h x .90" w x 5.20" l, .66 pounds
- Binding: Paperback
- 416 pages

## Features

- Great product!

## Amazon.com Review

In the 80s, Tracy Kidder's *The Soul of a New Machine* attempted to define the story of the development of a minicomputer: from the new science to the business and nascent culture of electronic hardware and software that was characteristic of that time. Scott Rosenberg's *Dreaming in Code* draws on Kidder's model as it attempts to document the state of software, the Internet, and everything circa 2006 through the lens of Chandler, an as-yet-unfinished software application for the management of personal information.

The Chandler project--driven by Mitch Kapor, the founder of Lotus Development and main designer of its 1-2-3 spreadsheet, and later co-founder of the Electronic Frontier Foundation--isn't the primary point of *Dreaming in Code*, though reading about software people and their social behavior is at least as interesting as reading about that of meerkats or monkeys. Rather, Chandler is a rhetorical device with which Rosenberg takes on the big questions: How do software development teams work (or not)? Why does the reuse of software modules rarely work altogether correctly? Does open-source development by volunteers on the Internet lead to innovation or just insanely bifurcated chaos? Chandler helps his readers think more clearly about all of these issues; however, "answers" to these questions are, of course, not to be had, which is one of his points.

The problem with books about technical subjects that aspire to appeal to a general audience, particularly computers and software, is that such subjects are so far outside the realm of familiarity of most people that the prose bogs down in analogy and metaphor. Rosenberg manages to avoid too much of that and deliver a readable account of software development and culture. --David Wall

From Publishers Weekly

Software is easy to make, except when you want it to do something new," Rosenberg observes—but the catch is that "the only software worth making is software that does something new." This two-tiered insight comes from years of observing a team led by Mitch Kapor (the creator of the Lotus 1-2-3 spreadsheet) in its efforts to create a "personal information manager" that can handle to-do lists as easily as events scheduling and address books. Rosenberg's fly-on-the-wall reporting deftly charts the course taken by Kapor's Open Source Applications Foundation, while acknowledging that every software programmer finds his or her own unique path to a brick wall in the development process. (The software is still in development even now.) With equal enthusiasm, Rosenberg digs into the history of the computer industry's efforts to make programming a more efficient process. Though there's a lot of technical information, it's presented in very accessible terms, primarily through the context of project management. Even readers whose computer expertise ends at retrieving their e-mail will be able to enjoy digressions into arcane subjects like object-oriented programming. (Jan.)

Copyright © Reed Business Information, a division of Reed Elsevier Inc. All rights reserved.

From Booklist

Programmers practice a most abstract and incomprehensible art; yet, as end users, we take them for granted and demand perfection from the software they create. Rosenberg, a theater and movie critic turned technology columnist and founder of the Web magazine Salon.com,<sup>A</sup> attempts to shed light on the day-to-day realities of what turns out to be a Herculean task: getting computers to do what we want them to. He spent three years following the course of development of a software program code named Chandler, a combination calendar, to-do list, e-mail manager, and personal database. This open-source project was the dream child of Mitch Kapor, the creator of Lotus 1-2-3, who envisioned a simple, elegant interface capable of easy storage and retrieval of any number of personal data. The practical matter of creation was another story, however, and we learn how the fits and starts of software engineering make even creating a "simple" program an arduous task. Although this is not edge-of-your-seat stuff, it is highly instructive for anyone planning on "managing" a software project. David Siegfried

Copyright © American Library Association. All rights reserved

Most helpful customer reviews

73 of 73 people found the following review helpful.

All Too Familiar

By Andrew Otwell

If you're in software development in any capacity, you'll find almost everything in this book is depressingly familiar: schedules dragging on endlessly, vague ideas standing for clear feature descriptions, and a remarkable lack of interest in learning from the past. You meet the indecisive manager, the frustrated designer, and the cowboy coder (the one who loses interest in things as soon as they finally work, determined to gut and rewrite thousands of lines of code for the sake of incremental "elegance").

"Dreaming in Code" really isn't a description of the act of programming, it's more about the difficulty of designing software with other people. It's not at all like designing, say, a new car, where real-world constraints (like the laws of physics and the behavior of materials) come into play. In software, nearly anything is possible. Unfortunately, or a "what comes out of that isn't usually the thrill of possibility, but the paralysis of choice. And that's what Rosenberg captures so well here: the endless, endless meetings



discussing what *could* be done, how features *might* work, or what sorts of things "users" *might* want. And it is always that nebulous term "the user" here, the Chandler project never seems to bother thinking much about exactly who will use their product, under what circumstances.

It's not giving anything away to mention that the Chandler project doesn't end at the end of the book. Amazingly, the project is *still* grinding away after four years, releasing incremental versions of their calendar application (each loudly proclaiming that it's an "experimentally usable version"), even as they get passed by on a regular basis by new, fully-usable, web-based calendars like the one from Google.

It's not exactly a fun read--it is essentially a case study of a very long series of meetings--but it should be required reading for anyone involved in software development.

46 of 49 people found the following review helpful.

recommended for non programmers

By Steve Johnson

Ansel Adams wrote, "There is nothing worse than a sharp image of a fuzzy concept." And such is the case with the Chandler project. After four years of development, they have delivered only a 0.6 release with no general availability in sight.

In *Dreaming in Code*, author Scott Rosenberg follows a group of programmers tasked with creating a new product over a three-year stint. Along the way the book explores disciplines in development (and the lack of), the history of computing (particularly its truths and folklore), and explains why software engineering isn't a science but an art. A common misconception even among developers is that software is similar to construction when, as becomes clear in the book, developing software is more like cooking. Programming methodologies are as plentiful as cookbooks but both are limited by the realities of artistry. A chef can make miracles from a pantry full of ingredients; a cook cannot.

If you're involved with a development team as a marketer, there is much here that will illuminate your team's dysfunction. Rosenberg reintroduces us to concepts that have been known since *The Mythical Man Month* and *The Soul of a New Machine* but apparently not understood, remembered, or believed. Strongly recommended.

118 of 133 people found the following review helpful.

Good writing, hobbled by bad choice of project

By Yaron

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.

To pick some nits, there are errors that betray Rosenberg as an outsider. "Foo" and "bar", for instance, usually aren't stand-ins for variable names, they're stand-ins for \*values\*; variable names are decided on almost immediately once the need for one becomes known. [UPDATE: this caused some controversy in the comments. It's true that "foo" and "bar" can also represent variables, but I still contend that it's only in theoretical discussions, when nothing is known about those variables. The book (p. 196) calls them "placeholders" during real-life coding, which I don't think is often true. FURTHER UPDATE, AFTER MORE COMMENTS: Okay, okay, I guess I was wrong. You guys win!] Rosenberg also, I think, makes too big a deal of software's need for precise language: many other engineering fields, and the legal profession, require precise writing, with small errors potentially leading to catastrophe. Rosenberg also overreaches when trying his hand at software philosophy, declaring "The only software worth making is software that does something new" (tell that to the OpenOffice people).

All that aside, this is an entertaining book with some interesting insights, and it would be a great read for anyone who's thinking of going into programming - hopefully it won't scare them off.

See all 101 customer reviews...

# **DREAMING IN CODE: TWO DOZEN PROGRAMMERS, THREE YEARS, 4,732 BUGS, AND ONE QUEST FOR TRANSCENDENT SOFTWARE BY SCOTT ROSENBERG PDF**

In getting this **Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software By Scott Rosenberg**, you could not still go by strolling or using your electric motors to guide stores. Get the queuing, under the rainfall or hot light, as well as still look for the unidentified publication to be during that publication store. By visiting this page, you could just search for the Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software By Scott Rosenberg as well as you could find it. So currently, this time around is for you to go with the download link and acquisition Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software By Scott Rosenberg as your personal soft data book. You could read this publication Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software By Scott Rosenberg in soft file only and save it as your own. So, you don't need to fast place guide Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software By Scott Rosenberg right into your bag all over.

## Amazon.com Review

In the 80s, Tracy Kidder's *The Soul of a New Machine* attempted to define the story of the development of a minicomputer: from the new science to the business and nascent culture of electronic hardware and software that was characteristic of that time. Scott Rosenberg's *Dreaming in Code* draws on Kidder's model as it attempts to document the state of software, the Internet, and everything circa 2006 through the lens of Chandler, an as-yet-unfinished software application for the management of personal information.

The Chandler project--driven by Mitch Kapor, the founder of Lotus Development and main designer of its 1-2-3 spreadsheet, and later co-founder of the Electronic Frontier Foundation--isn't the primary point of *Dreaming in Code*, though reading about software people and their social behavior is at least as interesting as reading about that of meerkats or monkeys. Rather, Chandler is a rhetorical device with which Rosenberg takes on the big questions: How do software development teams work (or not)? Why does the reuse of software modules rarely work altogether correctly? Does open-source development by volunteers on the Internet lead to innovation or just insanely bifurcated chaos? Chandler helps his readers think more clearly about all of these issues; however, "answers" to these questions are, of course, not to be had, which is one of his points.

The problem with books about technical subjects that aspire to appeal to a general audience, particularly computers and software, is that such subjects are so far outside the realm of familiarity of most people that the prose bogs down in analogy and metaphor. Rosenberg manages to avoid too much of that and deliver a readable account of software development and culture. --David Wall

## From Publishers Weekly

Software is easy to make, except when you want it to do something new," Rosenberg observes—but the catch is that "the only software worth making is software that does something new." This two-tiered insight comes from years of observing a team led by Mitch Kapor (the creator of the Lotus 1-2-3 spreadsheet) in its efforts to create a "personal information manager" that can handle to-do lists as easily as events scheduling and address books. Rosenberg's fly-on-the-wall reporting deftly charts the course taken by Kapor's Open

Source Applications Foundation, while acknowledging that every software programmer finds his or her own unique path to a brick wall in the development process. (The software is still in development even now.) With equal enthusiasm, Rosenberg digs into the history of the computer industry's efforts to make programming a more efficient process. Though there's a lot of technical information, it's presented in very accessible terms, primarily through the context of project management. Even readers whose computer expertise ends at retrieving their e-mail will be able to enjoy digressions into arcane subjects like object-oriented programming. (Jan.)

Copyright © Reed Business Information, a division of Reed Elsevier Inc. All rights reserved.

From Booklist

Programmers practice a most abstract and incomprehensible art; yet, as end users, we take them for granted and demand perfection from the software they create. Rosenberg, a theater and movie critic turned technology columnist and founder of the Web magazine Salon.com, attempts to shed light on the day-to-day realities of what turns out to be a Herculean task: getting computers to do what we want them to. He spent three years following the course of development of a software program code named Chandler, a combination calendar, to-do list, e-mail manager, and personal database. This open-source project was the dream child of Mitch Kapor, the creator of Lotus 1-2-3, who envisioned a simple, elegant interface capable of easy storage and retrieval of any number of personal data. The practical matter of creation was another story, however, and we learn how the fits and starts of software engineering make even creating a "simple" program an arduous task. Although this is not edge-of-your-seat stuff, it is highly instructive for anyone planning on "managing" a software project. David Siegfried

Copyright © American Library Association. All rights reserved

Well, still puzzled of the best ways to get this book *Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software* By Scott Rosenberg right here without going outside? Just connect your computer or gadget to the internet and also begin downloading and install *Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software* By Scott Rosenberg Where? This web page will reveal you the web link page to download *Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software* By Scott Rosenberg You never fret, your favourite book will certainly be earlier yours now. It will certainly be a lot easier to enjoy reading *Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software* By Scott Rosenberg by on the internet or getting the soft data on your gizmo. It will despite that you are and just what you are. This publication *Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software* By Scott Rosenberg is created for public and also you are just one of them who can take pleasure in reading of this publication [Dreaming In Code: Two Dozen Programmers, Three Years, 4,732 Bugs, And One Quest For Transcendent Software](#) By Scott Rosenberg