

More Joel On Software Further Thoughts On Diverse And Occasionally Related Matters That Will Prove Of Interest To Software Developers Designers And Or Ill Luck Work With Them In Some Capacity

In a world that divides us, technology creates connection. Cell phones, e-mail, digital cameras, personal Web sites—they all join us, however tenuously, to what we value. Is connectivity what we're willing to pay for? Should technology be our servant or a tool that helps us do other things? What can we really learn from Napster? What would intelligent standards for touch-screen user interface look like? How does technology evolve, and what drives that evolution? For Dan Bricklin, technology cannot exist independently of the lives and needs of those who use it. For more than a decade he has shared his thoughts on this essential interdependence in blogs, podcasts, and essays. This volume compiles those observations, putting together case histories and new reflections for a fascinating study of how people and technology affect one another. Whether you're a software developer or a student of human nature, you'll find yourself drawn into this most intriguing discourse—because you are its subject.

"Two thumbs up" —Gregory V. Wilson, Dr. Dobbs Journal (October 2004) No one can disparage the ability to write good code. At its highest levels, it is an art. But no one can confuse writing good code with developing good software. The difference—in terms of challenges, skills, and compensation—is immense. Coder to Developer helps you excel at the many non-coding tasks entailed, from start to finish, in just about any successful development project. What's more, it equips you with the mindset and self-assurance required to pull it all together, so that you see every piece of your work as part of a coherent process. Inside, you'll find plenty of technical guidance on such topics as: Choosing and using a source code control system Code generation tools—when and why Preventing bugs with unit testing Tracking, fixing, and learning from bugs Application activity logging Streamlining and systematizing the build process Traditional installations and alternative approaches To pull all of this together, the author has provided the source code for Download Tracker, a tool for organizing your collection of downloaded code, that's used for examples throughout this book. The code is provided in various states of completion, reflecting every stage of development, so that you can dig deep into the actual process of building software. But you'll also develop "softer" skills, in areas such as team management, open source collaboration, user and developer documentation, and intellectual property protection. If you want to become someone who can deliver not just good code but also a good product, this book is the place to start. If you must build successful software projects, it's essential reading.

A "good" programmer can outproduce five, ten, and sometimes more run-of-the-mill programmers. The secret to success for any software company then is to hire the good programmers. But how to do that? In Joel on Hiring, Joel Spolsky draws from his experience both at Microsoft and running his own successful software company based in New York City. He writes humorously, but seriously about his methods for sorting resumes, for finding great candidates, and for interviewing, in person and by phone. Joel's methods are not complex, but they do get to the heart of the matter: how to recognize a great developer when you see one. How would like to spot future trends before the competition? We all know the rules for success in our business or professions, yet we also know that these rules—paradigms—can change at any time. What Joel Barker does in Paradigms: The Business of Discovering the Future is explain how to spot paradigm shifts, how they unfold, and how to profit from them. Through the power of this method—paradigm spotting—you can: find the people in your organization most likely to spot a new trend help your key people adept when a massive change is occurring

learn to effectively grapple with your "intractable problems" and improve your results incalculably. In addition, Paradigms is full of concrete examples of paradigm shifts and predictions for the future, and contains a new introduction detailing recent developments and pointing out areas to watch for paradigm shifts.

Programming from the Ground Up uses Linux assembly language to teach new programmers the most important concepts in programming. It takes you a step at a time through these concepts: * How the processor views memory * How the processor operates * How programs interact with the operating system * How computers represent data internally * How to do low-level and high-level optimization Most beginning-level programming books attempt to shield the reader from how their computer really works. Programming from the Ground Up starts by teaching how the computer works under the hood, so that the programmer will have a sufficient background to be successful in all areas of programming. This book is being used by Princeton University in their COS 217 "Introduction to Programming Systems" course.

*Uniquely and squarely focuses on the needs of a startup ISV *Several leading companies in their market segment are actually micro-ISVs, including Fog Creek (FogBugz) and Sun Belt Software (Counter-Spy). It's possible to be small AND successful, and this book is perfect for those who wish to try *Volume of Micro ISVs is increasing, signifying a deep, broad audience for this book

Champion Your Best Ideas! Every time you communicate, you're trying to do something, change something, or move someone to action. You're trying to make a point. But the only way to make a point is to have a point. And the surprising truth is, very few communicators know their points or even understand what a point is, rendering them pointless. Communications expert Joel Schwartzberg says a point is not just a topic, an idea, or a theme. A real point is a proposition of value. It's a contention you can propose, argue, illustrate, and prove. In this concise and practical book, you'll learn to identify your point, strengthen it, stick to it, and sell it. Whether you want to improve your impact in speeches, staff meetings, pitches, emails, PowerPoint presentations, or any other communication setting, Schwartzberg's novel approach teaches you how to go from simply sharing a thought to making a difference. Which would you rather do?

* Covers three years of the best essays. * Essays range from technical to humorous, but are always tangible. * Beautifully written and extremely timely. * Google lists 183,000 links for "Joel on Software". * Spolsky is one of the most popular programmers around today, with legions of followers.

For most software developers, coding is the fun part. The hard bits are dealing with clients, peers, and managers and 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 subjects like career and productivity, personal finance and investing, and even fitness and relationships. Arranged as a collection of 71 short chapters, this fun listen invites you to dip in wherever you like. A "Taking Action" section at the end of each chapter tells you how to get quick results. Soft Skills will help make you a better programmer, a more valuable employee, and a happier, healthier person. This is the eagerly-anticipated revision to one of the seminal books in the field of software architecture which clearly defines and explains the topic.

This extraordinary book explains the engine that has catapulted the Internet from backwater to ubiquity—and reveals that it is sputtering precisely because of its runaway success. With the unwitting help of its users, the generative Internet is on a path to a lockdown, ending its cycle of innovation—and facilitating unsettling new kinds of control.

iPods, iPhones, Xboxes, and TiVos represent the first wave of Internet-centered products that can't be easily modified by anyone except their vendors or selected partners. These "tethered appliances" have already been used in remarkable but little-known ways: car GPS systems have been reconfigured at the demand of law enforcement to eavesdrop on the occupants at all times, and digital video recorders have been ordered to self-destruct thanks to a lawsuit against the manufacturer thousands of miles away. New Web 2.0 platforms like Google mash-ups and Facebook are rightly touted—but their applications can be similarly monitored and eliminated from a central source. As tethered appliances and applications eclipse the PC, the very nature of the Internet—its "generativity," or innovative character—is at risk. The Internet's current trajectory is one of lost opportunity. Its salvation, Zittrain argues, lies in the hands of its millions of users. Drawing on generative technologies like Wikipedia that have so far survived their own successes, this book shows how to develop new technologies and social structures that allow users to work creatively and collaboratively, participate in solutions, and become true "netizens."

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 Use construction practices that are right-weight for your project Debug problems quickly and effectively Resolve critical construction issues early and correctly Build quality into the beginning, middle, and end of your project

Draws on exclusive interviews to profile the acclaimed music artist's life and career, discussing such topics as his Long Island suburb upbringing, entry into the 1970s music scene and relationships with his closest associates. 100,000 first printing.

In the 1970s, author Joel Paris was one of the first doctors in his hospital to prescribe lithium to a psychiatric patient. In the wake of the drug's success, both in that case and countless others, why this book? As Dr. Paris' historical examination of bipolar diagnosis and critique of the spectrum demonstrates, medicine has often been prone to fads that are assumed correct until proven wrong by science. This book opens discussion about the overdiagnosis of bipolar disorder and the negative impact of this development on clinical care. Dr. Paris explores why patients are being classified as bipolar on dubious grounds and are being prescribed drugs they do not need. He explains the differences between bipolar disorder and depression without mania, personality disorders characterized by unstable mood, and impulsive disorders. A separate chapter discusses the unique issues present in the field of child psychiatry. Fads remain popular as long as they answer elusive and complex questions.

Unfortunately, the bipolar spectrum being used to explain a wide variety of psychopathological phenomena has caused classic bipolar disorder to become almost lost in the shuffle. Combining research findings and personal experiences, Dr. Paris documents the damage of overdiagnosis and explores alternative treatments patients could benefit from.

Most programmers' fear of user interface (UI) programming comes from their fear of doing UI design. They think that UI design is like graphic design—the mysterious process by which creative, latte-drinking, all-black-wearing people produce cool-looking, artistic pieces. Most programmers see themselves as analytic, logical thinkers instead—strong at reasoning, weak on artistic judgment, and incapable of doing UI design. In this brilliantly readable book, author Joel Spolsky proposes simple, logical rules that can be applied without any artistic talent to improve any user interface, from traditional GUI applications to websites to consumer electronics. Spolsky's primary axiom, the importance of bringing the program model in line with the user model, is both rational and simple. In a fun and entertaining way, Spolsky makes user interface design easy for programmers to grasp. After reading *User Interface Design for Programmers*, you'll know how to design interfaces with the user in mind. You'll learn the important principles that underlie all good UI design, and you'll learn how to perform usability testing that works.

Learn how declaring God's love will bless you with favor and fulfillment in this uplifting book--perfect for anyone who is determined to find success and spiritual inspiration. God helps you accomplish what you couldn't manage on your own. With His blessings, you stand out in the crowd and get breaks that you didn't deserve. The psalmist said, "God's favor surrounds us like a shield." That means that everywhere you go, you have an advantage, a divine empowerment, and a key to open up the right doors. With Joel's encouragement, you'll see how God's goodness uplifts you every day. He wants you to reach new levels of fulfillment, new levels of increase, new levels of promotion, new levels of victory. You have been called out, set apart, and chosen to live a distinctively favored life. When you realize you have been marked for blessings, you will feel the force of His favor and overcome challenges that you can't face on your own.

Managing Humans is a selection of the best essays from Michael Lopp's popular website *Rands in Repose*(www.randsinrepose.com). Lopp is one of the most sought-after IT managers in Silicon Valley, and draws on his experiences at Apple, Netscape, Symantec, and Borland. This book reveals a variety of different approaches for creating innovative, happy development teams. It covers handling conflict, managing wildly differing personality types, infusing innovation into insane product schedules, and figuring out how to build lasting and useful engineering culture. The essays are biting, hilarious, and always informative.

Learn Bosque Programming helps you to get acquainted with the regularized programming paradigm and how it is applied in Bosque. You'll learn how to write programs and apply the key and unique features that make Bosque an attractive alternative for developers who seek improved productivity without having to worry about side-effects.

* Will appeal to the same (large) audience as *Joel on Software* * Contains exclusive commentary by Joel * Lots of free publicity both because of Joel's influence in the community and the influence of the contributors

This unique text provides a holistic systems approach to project portfolio management which includes people, processes, tools, and techniques that work synergistically to produce portfolio decisions with the best chance of success. Accompanied by decision support software and

advanced decision making techniques, it guides readers step-by-step through the entire project portfolio management process. This professional guide is also ideal for executive continuing education programs, and as a primary text for graduate level academic courses.

This book addresses the rising productivity gap between the global frontier and other firms, and identifies a number of structural impediments constraining business start-ups, knowledge diffusion and resource allocation (such as barriers to up-scaling and relatively high rates of skill mismatch).

Joel, Apress, Blogs, and Blooks ...I was learning the hard way about how to be a publisher and probably spending way too much time looking at web sites and programming than I should have in response to that. Anyway, one day I came across this web site called , which was run by a guy with strong opinions and an unusual, clever writing style, along with a willingness to take on the conventional wisdom. In particular, he was writing this ongoing series about how bad most user interfaces were—mostly because programmers by and large knew, as Joel and I would say, using the same Yiddish-derived NYC vernacular that we both share, “bupkis” about what users really want. And I, like many, was hooked both by the series and the occasional random essay that Joel wrote. And then I had this epiphany: I'm a publisher, I like reading his stuff, why not turn it into a book?... Read the complete Foreword — Gary Cornell, Cofounder, Apress Since the release of the bestselling title Joel on Software in 2004, requests for a sequel have been relentless. So, we went back to the famed JoelonSoftware.com archives and pulled out a new batch of favorites, many of which have been downloaded over one million times. With Joel's newest book, More Joel on Software, you'll get an even better (not to mention updated) feast of Joel's opinions and impressions on software development, software design, running a software business, and so much more. This is a new selection of essays from the author's web site, <http://www.joelonsoftware.com>. Joel Spolsky started his weblog in March 2000 in order to offer his insights, based on years of experience, on how to improve the world of programming. This weblog has become infamous among the programming world, and is linked to more than 600 other web sites and translated into 30+ languages! Spolsky's extraordinary writing skills, technical knowledge, and caustic wit have made him a programming guru. With the success of Joel on Software, there has been a strong demand for additional gems and advice, and this book is the answer to those requests. Containing a collection of all-new articles from the original, More Joel on Software has even more of an edge than the original, and the tips for running a business or managing people have far broader application than the software industry. We feel it is safe to say that this is the most useful book you will buy this year.

Ajith, the CEO of India's most popular coffee chain, Good Morning Inc., is outrageously shocked to hear the pitch from a tech startup: fire all managers and replace them with AI. Though he wants to dismiss the idea, the pitch is ferociously compelling. Before Ajith could take a firm decision, an unexpected tragedy occurs at one of the coffee stores. Ajith loses control over his company overnight. He stages a secretive reverse coup to regain his company. At the same time, the office of the WHO issues a warning about a global respiratory pandemic to the Indian government, but the prime minister's office ignores it. Instead, it spends its focus on the developments of Good Morning Inc. Why would the PMO's office be interested in the incidents of a corporate entity? Rahul is excited to meet his blind date at one of the coffee shops of Good Morning Inc. What he doesn't know is that he is just a pawn whose actions are about to trigger an avalanche of consequences in the lives of many people, related and unrelated to Good Morning Inc. Will Ajith ever regain his company? Will he become the first CEO in the world to successfully use artificial intelligence instead of human leaders to manage its workforce?

The United States is supposed to offer economic opportunity to everyone. It shouldn't take a worldwide pandemic and nationwide protests to bring economic and racial inequality to the

forefront of problems we desperately need to solve. But now that the opportunity is here, what should we do? How can we create more equality, opportunity, and growth for everyone? Not someday, but what can government and the private sector do right now to disrupt a status quo that almost everyone wants to change? In *Common Sense*, the New York Times best-selling author Joel Greenblatt offers an investor's perspective on building an economy that truly works for everyone. With dry wit and engaging storytelling, he makes a lively and provocative case for disruptive new approaches—some drawn from personal experience, some from the outside looking in. How can leading corporations immediately disrupt our education establishment while creating high-paying job opportunities for those currently left behind? If we want a living wage for everyone, how can we afford it while using an existing program to get it done now? If we subsidize banks, what simple changes can we make to the way we capitalize and regulate them to help grow the economy, increase access, and create more jobs (while keeping the risks and benefits where they belong)? Greenblatt also explains how dramatically increasing immigration would be like giving every American a giant bonus and the reason Australia might be the best place to learn about saving for retirement. Not everyone will agree with what Greenblatt has to say—but all of us can benefit from the conversations he aims to start.

More Joel on Software Further Thoughts on Diverse and Occasionally Related Matters That Will Prove of Interest to Software Developers, Designers, and Managers, and to Those Who, Whether by Good Fortune or Ill Luck, Work with Them in Some Capacity
Apress

In 2005, Joel Greenblatt published a book that is already considered one of the classics of finance literature. In *The Little Book that Beats the Market*—a New York Times bestseller with 300,000 copies in print—Greenblatt explained how investors can outperform the popular market averages by simply and systematically applying a formula that seeks out good businesses when they are available at bargain prices. Now, with a new Introduction and Afterword for 2010, *The Little Book that Still Beats the Market* updates and expands upon the research findings from the original book. Included are data and analysis covering the recent financial crisis and model performance through the end of 2009. In a straightforward and accessible style, the book explores the basic principles of successful stock market investing and then reveals the author's time-tested formula that makes buying above average companies at below average prices automatic. Though the formula has been extensively tested and is a breakthrough in the academic and professional world, Greenblatt explains it using 6th grade math, plain language and humor. He shows how to use his method to beat both the market and professional managers by a wide margin. You'll also learn why success eludes almost all individual and professional investors, and why the formula will continue to work even after everyone "knows" it. While the formula may be simple, understanding why the formula works is the true key to success for investors. The book will take readers on a step-by-step journey so that they can learn the principles of value investing in a way that will provide them with a long term strategy that they can understand and stick with through both good and bad periods for the stock market. As the *Wall Street Journal* stated about the original edition, "Mr. Greenblatt...says his goal was to provide advice that, while sophisticated, could be understood and followed by his five children, ages 6 to 15. They are in luck. His 'Little Book' is one of the best, clearest guides to value investing out there."

Data science libraries, frameworks, modules, and toolkits are great for doing data

science, but they're also a good way to dive into the discipline without actually understanding data science. In this book, you'll learn how many of the most fundamental data science tools and algorithms work by implementing them from scratch. If you have an aptitude for mathematics and some programming skills, author Joel Grus will help you get comfortable with the math and statistics at the core of data science, and with hacking skills you need to get started as a data scientist. Today's messy glut of data holds answers to questions no one's even thought to ask. This book provides you with the know-how to dig those answers out. Get a crash course in Python Learn the basics of linear algebra, statistics, and probability—and understand how and when they're used in data science Collect, explore, clean, munge, and manipulate data Dive into the fundamentals of machine learning Implement models such as k-nearest Neighbors, Naive Bayes, linear and logistic regression, decision trees, neural networks, and clustering Explore recommender systems, natural language processing, network analysis, MapReduce, and databases

Unleash the untapped power inside you! You were designed with the ability and inner strength to stretch to the next level, to stand strong and overcome the opposition, to learn new skills, to believe bigger, dream bigger, and to take on new challenges. In *You Are Stronger than You Think*, #1 New York Times bestselling author Joel Osteen encourages you to stand firm when the difficulty is not turning around. When the pressure mounts, when you feel overwhelmed, when the struggle looks too big, or the goal is out of reach, you have to remember that is not your destiny. You are more powerful than you think. You need to see yourself the right way and realize what's already inside you. You were made to outlast the competition, to go further than you thought you could, and live a life beyond what limits you. As you read *You Are Stronger than You Think*, you will find the power to break the cycle of defeat, the courage to set a new standard for your family, the hope to conquer your fear, and the confidence to accomplish every dream. You will discover that you have more strength, more talent, more determination, and more endurance than you ever imagined. Nothing can hold you back...because *You Are Stronger than You Think!*

Life scientists today urgently need training in bioinformatics skills. Too many bioinformatics programs are poorly written and barely maintained--usually by students and researchers who've never learned basic programming skills. This practical guide shows postdoc bioinformatics professionals and students how to exploit the best parts of Python to solve problems in biology while creating documented, tested, reproducible software. Ken Youens-Clark, author of *Tiny Python Projects* (Manning), demonstrates not only how to write effective Python code but also how to use tests to write and refactor scientific programs. You'll learn the latest Python features and tools—including linters, formatters, type checkers, and tests—to create documented and tested programs. You'll also tackle 14 challenges in Rosalind, a problem-solving platform for learning bioinformatics and programming. Create command-line Python programs to document and validate parameters Write tests to verify refactor programs and confirm they're correct Address bioinformatics ideas using Python data structures and modules such as Biopython Create reproducible shortcuts and workflows using makefiles Parse essential bioinformatics file formats such as FASTA and FASTQ Find patterns of text using regular expressions Use higher-order functions in Python like `filter()`, `map()`, and `reduce()`

Most software project problems are sociological, not technological. Peopleware is a book on managing software projects.

Author of the bestselling text Supply Chain Management, John T. Mentzer's companion book Fundamentals of Supply Chain Management: Twelve Drivers of Competitive Advantage has been developed as a supplemental text for any course dealing with strategy and supply chains. Written in an entertaining, accessible style, Mentzer identifies twelve drivers of competitive advantage as clear strategic points managers can use in their companies. Research from more than 400 books, articles, and papers, as well as interviews with over fifty executives in major global companies, inform these twelve drivers. The roles of all of the traditional business functions—marketing, sales, logistics, information systems, finance, customer services, and management—in supply chain management are also addressed.

Praise for the Second Edition: "The authors present an intuitive and easy-to-read book. ... accompanied by many examples, proposed exercises, good references, and comprehensive appendices that initiate the reader unfamiliar with MATLAB." —Adolfo Alvarez Pinto, International Statistical Review "Practitioners of EDA who use MATLAB will want a copy of this book. ... The authors have done a great service by bringing together so many EDA routines, but their main accomplishment in this dynamic text is providing the understanding and tools to do EDA. —David A Huckaby, MAA Reviews Exploratory Data Analysis (EDA) is an important part of the data analysis process. The methods presented in this text are ones that should be in the toolkit of every data scientist. As computational sophistication has increased and data sets have grown in size and complexity, EDA has become an even more important process for visualizing and summarizing data before making assumptions to generate hypotheses and models. Exploratory Data Analysis with MATLAB, Third Edition presents EDA methods from a computational perspective and uses numerous examples and applications to show how the methods are used in practice. The authors use MATLAB code, pseudo-code, and algorithm descriptions to illustrate the concepts. The MATLAB code for examples, data sets, and the EDA Toolbox are available for download on the book's website. New to the Third Edition Random projections and estimating local intrinsic dimensionality Deep learning autoencoders and stochastic neighbor embedding Minimum spanning tree and additional cluster validity indices Kernel density estimation Plots for visualizing data distributions, such as beanplots and violin plots A chapter on visualizing categorical data

DRUGS ACROSS THE SPECTRUM encourages you to examine the motivation for drug use, social implications of drug use, legal ramifications, and factors affecting how drugs interact with the human body. It provides a look at the history and culture surrounding drug use and abuse, key information regarding specific types of drugs, and a review of drug treatment, education, and prevention approaches and programs. Utilizing active learning features like Fact or Fiction and Thinking Critically questions, Goldberg offers an engaging book that helps readers personally understand the issues of drugs in society. Available with InfoTrac Student Collections

<http://gocengage.com/infotrac>. Important Notice: Media content referenced within the product description or the product text may not be available in the ebook version.

Peter Seibel interviews 15 of the most interesting computer programmers alive today in Coders at Work, offering a companion volume to Apress's highly acclaimed best-seller

Founders at Work by Jessica Livingston. As the words “at work” suggest, Peter Seibel focuses on how his interviewees tackle the day-to-day work of programming, while revealing much more, like how they became great programmers, how they recognize programming talent in others, and what kinds of problems they find most interesting. Hundreds of people have suggested names of programmers to interview on the Coders at Work web site: www.codersatwork.com. The complete list was 284 names. Having digested everyone’s feedback, we selected 15 folks who’ve been kind enough to agree to be interviewed: Frances Allen: Pioneer in optimizing compilers, first woman to win the Turing Award (2006) and first female IBM fellow Joe Armstrong: Inventor of Erlang Joshua Bloch: Author of the Java collections framework, now at Google Bernie Cosell: One of the main software guys behind the original ARPANET IMPs and a master debugger Douglas Crockford: JSON founder, JavaScript architect at Yahoo! L. Peter Deutsch: Author of Ghostscript, implementer of Smalltalk-80 at Xerox PARC and Lisp 1.5 on PDP-1 Brendan Eich: Inventor of JavaScript, CTO of the Mozilla Corporation Brad Fitzpatrick: Writer of LiveJournal, OpenID, memcached, and Perlbal Dan Ingalls: Smalltalk implementor and designer Simon Peyton Jones: Coinventor of Haskell and lead designer of Glasgow Haskell Compiler Donald Knuth: Author of The Art of Computer Programming and creator of TeX Peter Norvig: Director of Research at Google and author of the standard text on AI Guy Steele: Coinventor of Scheme and part of the Common Lisp Gang of Five, currently working on Fortress Ken Thompson: Inventor of UNIX Jamie Zawinski: Author of XEmacs and early Netscape/Mozilla hacker Education Networks is a critical analysis of the emerging intersection among the global power elite, information and communication technology, and schools. Joel Spring documents and examines the economic and political interests and forces—including elite networks, the for-profit education industry, data managers, and professional educators—that are pushing the use of ICT for online instruction, test preparation and tutoring, data management, instructional software packages, and more, and looks closely at the impact this is having on schools, students, and learning. Making a distinction between “mind” (as socially constructed) and “brain” (as a physiological entity), Spring draws on recent findings from comparative psychology on the possible effects of ICT on the social construction of the minds of students and school managers, and from neuroscience regarding its effect on students’ brains. Throughout, the influence of elite networks and powerful interest groups is linked to what is happening to children in classrooms. In conclusion Spring offers bold suggestions to change the course of the looming technological triumph of ICT in the “brave new world” of schooling.

The third edition of this essential text has been updated and expanded with new material that reflects the most recent developments in the field, and explores our current understanding of a broad range of topics related to aging and older adulthood. Fresh edition includes updated content such as revised case histories and reworked material on key concepts and research applications Retains the winning format of the second edition, with chapter contents framed by individual histories Dual models add cohesiveness to the presentation of theory Thematic structure facilitates reader comprehension Instructor resources provided online upon publication at <http://www.wiley.com/go/erber> www.wiley.com/go/erber/a A New York Times bestseller! “Lively and absorbing. . .” — The New York Times Book

Review "Engrossing." —Wall Street Journal "Entertaining and well-researched . . ."

—Houston Chronicle Three noted Texan writers combine forces to tell the real story of the Alamo, dispelling the myths, exploring why they had their day for so long, and explaining why the ugly fight about its meaning is now coming to a head. Every nation needs its creation myth, and since Texas was a nation before it was a state, it's no surprise that its myths bite deep. There's no piece of history more important to Texans than the Battle of the Alamo, when Davy Crockett and a band of rebels went down in a blaze of glory fighting for independence from Mexico, losing the battle but setting Texas up to win the war. However, that version of events, as *Forget the Alamo* definitively shows, owes more to fantasy than reality. Just as the site of the Alamo was left in ruins for decades, its story was forgotten and twisted over time, with the contributions of Tejanos--Texans of Mexican origin, who fought alongside the Anglo rebels--scrubbed from the record, and the origin of the conflict over Mexico's push to abolish slavery papered over. *Forget the Alamo* provocatively explains the true story of the battle against the backdrop of Texas's struggle for independence, then shows how the sausage of myth got made in the Jim Crow South of the late nineteenth and early twentieth century. As uncomfortable as it may be to hear for some, celebrating the Alamo has long had an echo of celebrating whiteness. In the past forty-some years, waves of revisionists have come at this topic, and at times have made real progress toward a more nuanced and inclusive story that doesn't alienate anyone. But we are not living in one of those times; the fight over the Alamo's meaning has become more pitched than ever in the past few years, even violent, as Texas's future begins to look more and more different from its past. It's the perfect time for a wise and generous-spirited book that shines the bright light of the truth into a place that's gotten awfully dark.

[Copyright: 7ae194f66fbe7177f5b97d55b219043e](https://www.joelonsoftware.com/2013/08/07/forget-the-alamo/)