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

This is the digital version of the printed book (Copyright © 1996). Based on an award-winning doctoral thesis at Carnegie Mellon University, Measuring and Managing Performance in Organizations presents a captivating analysis of the perils of performance measurement systems. In the book's foreword, Peopleware authors Tom DeMarco and Timothy Lister rave, "We believe this is a book that needs to be on the desk of just about anyone who manages anything." Because people often react with unanticipated sophistication when they are being measured, measurement-based management systems can become dysfunctional, interfering with achievement of intended results. Fortunately, as the author shows, measurement dysfunction follows a pattern that can be identified and avoided. The author's findings are bolstered by interviews with eight recognized experts in the use of measurement to manage computer software development: David N. Card, of Software Productivity Solutions; Tom DeMarco, of the Atlantic Systems Guild; Capers Jones, of Software Productivity Research; John Musa, of AT&T Bell Laboratories; Daniel J. Paulish, of Siemens Corporate Research; Lawrence H. Putnam, of Quantitative Software Management; E. O. Tilford, Sr., of Fissure; plus the anonymous Expert X. A practical model for analyzing measurement projects solidifies the text–don't start without it!

This collection explores the cultural fascination with social media forms of self-portraiture, "selfies," with a specific interest in online self-imaging strategies in a Western context. This book examines the selfie as a social and technological phenomenon but also engages with digital self-portraiture as representation: as work that is committed to rigorous object-based analysis. The scholars in this volume consider the topic of online self-portraiture—both its social function as a technology-driven form of visual communication, as well as its thematic, intellectual, historical, and aesthetic intersections with the history of art and visual culture. This book will be of interest to scholars of photography, art history, and media studies.

A noted blogger and author of Joel on Software explains how companies can find and hire the best programmers available, offers practical suggestions and tips on how to identify great developers, sort resumes, interview candidates effectively, and more. (All Users)

* 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 volume brings together a number of seminal studies presented at the International Conference on Great Indian Epics held in February 2019 at the Jawaharlal Nehru University, New Delhi by scholars from various parts of the world. Each article adds a new dimension to the subject with historical scholarship and critical interpretation, reflecting comprehensiveness, unity, clarity and rightness of perception. This definitive work adds to our knowledge of the epics and their infinite influence. Please note: Taylor & Francis does not sell or distribute the Hardback in India, Pakistan, Nepal, Bhutan, Bangladesh and Sri Lanka.

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, Spolky 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.

"Raymond Chen is the original raconteur of Windows." --Scott Hanselman, ComputerZen.com "Raymond has been at Microsoft for many years and has seen many nuances of Windows that others could only ever hope to get a glimpse of. With this book, Raymond shares his knowledge, experience, and anecdotal stories, allowing all of us to get a better understanding of the operating system that affects millions of people every day. This book has something for everyone, is a casual read, and I highly recommend it!" --Jeffrey Richter, Author/Consultant, Cofounder of Wintellect "Very interesting read. Raymond tells the inside story of why Windows is the way it is." --Eric Gunnerson, Program Manager, Microsoft Corporation "Absolutely essential reading for understanding the history of Windows, its intricacies and quirks, and why they came about." --Matt Pietrek, MSDN Magazine's Under the Hood Columnist "Raymond Chen has become something of a legend in the software industry, and in this book you'll discover why. From his high-level reminiscences on the design of the Windows Start button to his low-level discussions of GlobalAlloc that only your inner-geek could love, The Old New Thing is a captivating collection of anecdotes that will help you to truly appreciate the difficulty inherent in designing and writing quality software." --Stephen Toub, Technical Editor, MSDN Magazine Why does Windows work the way it does? Why is Shut Down on the Start menu? (And why is there a Start button, anyway?) How can I tap into the dialog loop? Why does the GetWindowText function behave so strangely? Why are registry files called "hives"? Many of Windows' quirks have perfectly logical explanations, rooted in history. Understand them, and you'll be more productive and a lot less frustrated. Raymond Chen--who's spent more than a decade on Microsoft's Windows development team--reveals the "hidden Windows" you need to know. Chen's engaging style, deep insight, and thoughtful humor have made him one of the world's premier technology bloggers. Here he brings together behind-the-scenes explanations, invaluable technical advice, and illuminating anecdotes that bring Windows to life--and help you make the most of it. A few of the things you'll find inside: What vending machines can teach you about effective user interfaces A deeper understanding of window and dialog management Why performance optimization can be so counterintuitive A peek at the underbelly of COM objects and the Visual C++ compiler Key details about backwards compatibility--what Windows does and why Windows program security holes most developers don't know about How to make your program a better Windows citizen

* 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.

Shares a message of hope and inspiration for using one's faith as a cornerstone to build a happy, secure, and fulfilled life.

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.

Iris recognition is one of the highest accuracy techniques used in biometric systems. The accuracy of the iris recognition system is measured by False Reject Rate (FRR), which measures the authenticity of a user who is incorrectly rejected by the system due to changes in iris features (such as aging and health condition) and external factors that affect iris image, for instance, high noise rate. External factors such as technical fault, occlusion, and source of lighting that causes the image acquisition to produce distorted iris images create error, hence are incorrectly rejected by the biometric system. FRR can be reduced using wavelets and Gabor filters, cascaded classifiers, ordinal measures, multiple biometric modalities, and a selection of unique iris features. Nonetheless, in the long duration of the matching process, existing methods were unable to identify the authenticity of the user since the iris structure itself produces a template changed due to aging. In fact, the iris consists of unique features such as crypts, furrows, collarette, pigment blotches, freckles, and pupils that are distinguishable among humans. Earlier research was done by selecting unique iris features. However, these had low accuracy levels. A new way of identifying and matching the iris template using the nature-inspired algorithm is described in this book. It provides an overview of iris recognition that is based on nature-inspired environment technology. The book is useful for students from universities, polytechnics, community colleges; practitioners; and industry practitioners.

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

As a young man, I faced many adversities while struggling to find myself through a brutal nineteen-year war with drug and alcohol addiction, all the while suffering from anxiety, depression, and PTSD. During that war I was kidnapped, dodged a disturbing death on multiple occasions, suffered a minor stroke due to head trauma, and survived the internal warfare that almost ended with me taking my own life. Tormented by dark spirits and enlightened by the good, I was gifted to another opportunity at living a meaningful life. With God's underserving Grace and incomprehensible Mercy, I share my experience, strength, and hope with you, to prove recovery from addiction is not only 100% possible, but we can in turn, aid and assist our brothers and sisters in our communities. Joel Carroll is an advocate for men, women and children, who battle with drug addiction and suffer from mental illness. In 2013 he transformed his life, from an alcoholic and an addict, to a man who aids and assists others during their times of struggle. He also transformed from a liar and a thief, to an honest and giving man who loves his family dearly. After graduating from the Salvation Army of Tucson's six-month rehabilitation program, Joel has dedicated his life to serving God and the communities he once ravaged.

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?

The impression you make as a leader has the power to build, enhance, or sabotage your authority and ideas. Discover how to convey the essence of leadership with every interaction. Every communication leaders make—speaking, writing, posting, sharing, and even listening—has the power to either secure or sabotage their impact. But wanting to inspire and engage their team and knowing how to do it are two different things. In this book, Joel Schwartzberg suggests mindsets, tactics, tips, and examples to help readers reach that goal using the most powerful leadership tool available: a leader's voice. Whether managers are giving speeches, telling stories, sending emails, posting messages, recording videos, or running Zoom meetings, these are essential tools for establishing authority and galvanizing an audience. Readers will learn how to inspire instead of inform, communicate with purpose and power, and sell—not just share—their most important ideas.

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.

A "NOVEL" APPROACH TO ERP IMPLEMENTATION In today's business world, an effective and holistic enterprise resource planning (ERP) implementation can mean offering a wide range of services that include software implementation, multi-channel ecommerce solutions, and customized development projects. But how can you make effective decisions for your business without having deep knowledge about all these different technology solutions? In THE BIG COMMITMENT: SOLVING THE MYSTERIES OF YOUR ERP IMPLEMENTATION, Joel Patterson breaks it down by showing you how technology consultants like his company, The Vested Group, help put the system, processes, and training in place to provide business leaders with reliable and actionable information. THE BIG COMMITMENT walks you through the process of finding and working with the right partner for your ERP needs. Along the way, you'll get an authentic view of the core values, business philosophies, and vision for the future of a leader who helps businesses succeed.

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

International Community Development Practice provides readers with practice-based examples of good community development, demonstrating its value for strengthening people power and improving the effectiveness of development agencies, whether these be governmental, non-governmental or private sector. The chapters focus upon the making of the community development profession and the eight core competences required of the professional practitioner, as outlined by the International Association for Community Development (IACD), whatever their job title or host agency, in order to be able to undertake community development. These are concerned with the ability of the practitioner to: Put ethics and values into practice Engage with communities Ensure participatory planning Organize for change Support learning for change Promote diversity and inclusion Build leadership and infrastructure Develop and improve policy and practice From a policy perspective, the book will reassert the role of community development approaches as related to a wide variety of global challenges, including poverty amelioration, climate change, human rights, peace building and social, environmental, political and economic development. From a practice perspective, the book will reassert the importance of high levels of professional competence building upon decades of experience in the field around the world by development practitioners working in community work, social work, health, adult education, environmental protection, local economic development, urban design, cultural work and other disciplines concerned to support effective community development.

This book is a practical guide for individuals responsible for creating products that are safe, effective, usable, and satisfying in the hands of the intended users. The contents are intended to reduce the number of use errors involving medical devices that have led to injuries and deaths. The book presents the strong connection between user interface requirements and risk management for medical devices and instructs readers how to develop specific requirements that are sufficiently comprehensive and detailed to produce good results – a user-friendly product that is likely to be used correctly. The book's tutorial content is complemented by many real-world examples of user interface requirements, including ones pertaining to an inhaler, automated external defibrillator, medical robot, and mobile app that a patient might use to manage her diabetes. The book is intended for people representing a variety of product development disciplines who have responsibility for producing safe, effective, usable, and satisfying medical devices, including those who are studying or working in human factors engineering, psychology, mechanical engineering, biomedical engineering, systems engineering, software programming, technical writing, industrial design, graphic design, and regulatory affairs.

Writing and running software is now as much a part of science as telescopes and test tubes, but most researchers are never taught how to do either well. As a result, it takes them longer to accomplish simple tasks than it should, and it is harder for them to share their work with others than it needs to be. This book introduces the concepts, tools, and skills that researchers need to get more done in less time and with less pain. Based on the practical experiences of its authors, who collectively have spent several decades teaching software skills to scientists, it covers everything graduate-level researchers need to automate their workflows, collaborate with colleagues, ensure that their results are trustworthy, and publish what they have built so that others can build on it. The book assumes only a basic knowledge of Python as a starting point, and shows readers how it, the Unix shell, Git, Make, and related tools can give them more time to focus on the research they actually want to do. Research Software Engineering with Python can be used as the main text in a one-semester course or for self-guided study. A running example shows how to organize a small research project step by step; over a hundred exercises give readers a chance to practice these skills themselves, while a glossary defining over two hundred terms will help readers find their way through the terminology. All of the material can be re-used under a Creative Commons license, and all royalties from sales of the book will be donated to The Carpentries, an organization that teaches foundational coding and data science skills to researchers worldwide.

Heterostructured (HS) materials represent an emerging class of materials that are expected to become a major research field for the communities of materials, mechanics, and physics in the next couple of decades. One of the biggest advantages of HS materials is that they can be produced by large-scale industrial facilities and technologies and therefore can be commercialized without the scaling up and high-cost barriers that are often encountered by other advanced materials. This book collects recent papers on the progress in the field of HS materials, especially their fundamental physics. The papers are arranged in a sequence of chapters that will help new researchers entering the field to have a quick and comprehensive understanding of HS materials, including the fundamentals and recent progress in their processing, characterization, and properties.

NEW YORK TIMES BESTSELLER USA TODAY BESTSELLER NATIONAL INDIE BESTSELLER THE WASHINGTON POST BESTSELLER Recommended by Entertainment Weekly, Real Simple, NPR, Slate, and Oprah Magazine #1 Library Reads Pick—October 2020 #1 Indie Next Pick—October 2020 BOOK OF THE YEAR (2020) FINALIST—Book of The Month Club A "Best Of" Book From: Oprah Mag * CNN * Amazon * Amazon Editors * NPR * Goodreads * Bustle * PopSugar * BuzzFeed * Barnes & Noble * Kirkus Reviews * Lambda Literary * Nerdette * The Nerd Daily * Polygon * Library Reads * io9 * Smart Bitches Trashy Books * LiteraryHub * Medium * BookBub * The Mary Sue * Chicago Tribune * NY Daily News * SyFy Wire * Powells.com * Bookish * Book Riot * Library Reads Voter Favorite * In the vein of The Time Traveler's Wife and Life After Life, The Invisible Life of Addie LaRue is New York Times bestselling author V. E. Schwab's genre-defying tour de force. A Life No One Will Remember. A Story You Will Never Forget. France, 1714: in a moment of desperation, a young woman makes a Faustian bargain to live forever—and is cursed to be forgotten by everyone she meets. Thus begins the extraordinary life of Addie LaRue, and a dazzling adventure that will play out across centuries and continents, across history and art, as a young woman learns how far she will go to leave her mark on the world. But everything changes when, after nearly 300 years, Addie stumbles across a young man in a hidden bookstore and he remembers her name. At the Publisher's request, this title is being sold without Digital Rights Management Software (DRM) applied.

"I love the shit out of my daughter. I want to change the world, with words. This started as affirmations for Lilah, which turned into prayers, which turned into love letters, which turned into essays, which turned into poems, which then turned into all of the above.This has been written in the same vein as Claudia Rankine's Citizen, for context. I wanted to write something that could document my experience as a single father, a single Black father, raising a soon-to-be 2-year-old Afro-Latina in the 45th era. I wanted her to know me, and know herself, with the hope that it would help fathers learn their own daughters, and learn themselves, too." -Joel L. DanielsJoel l. Daniels is a storyteller, born and raised in the Bronx. He was the recipient of the Bronx Council of the Arts BRIO Award for poetry, and his work has been featured in the Columbia Journal, Café.com, The Boston Globe, CNN Money, The Towner, Fatherly, Thought Catalog, Philadelphia Printworks, The Smoking Section, Blavity, Huffington Post, BBC Radio, RCRD LBL, URB, BRM, AllHipHop, The Source, RESPECT, and HipHopDX.He's spoken/performed at the Apollo Theater, Joe's Pub, Rockwood Music Hall, Columbia University, Lehman College, City Tech, The National Black Theater, NYU, Webster Hall, Pianos, and Brooklyn Bowl.

Allen's study of the Books of Joel, Obadiah, Jonah, and Micah constitute a volume in The New International Commentary on the Old Testament. Like its companion series on the New Testament, this commentary devotes considerable care to achieving a balance between technical information and homiletic-devotional interpretation.

Follows the lives of Ana Alvarado and Derek Brooks as they create and relate to the artificial intelligences they helped design.

"Two thumbs up" —Gregory V. Wilson, Dr. Dobbs Journal (October 2004) No one can disparage the ability to write good code. At itshighest levels, it is an art. But no one can confuse writing good code with developing goodsoftware. The difference—in terms of challenges, skills, andcompensation—is immense. Coder to Developer helps you excel at the manynon-coding tasks entailed, from start to finish, in just about anysuccessful development project. What's more, it equips you with themindset and self-assurance required to pull it all together, sothat you see every piece of your work as part of a coherentprocess. Inside, you'll find plenty of technical guidance on suchtopics 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 sourcecode for Download Tracker, a tool for organizing your collection ofdownloaded code, that's used for examples throughout this book. Thecode is provided in various states of completion, reflecting everystage of development, so that you can dig deep into the actualprocess of building software. But you'll also develop "softer"skills, in areas such as team management, open sourcecollaboration, user and developer documentation, and intellectualproperty protection. If you want to become someone who can delivernot just good code but also a good product, this book is the placeto start. If you must build successful software projects, it'sessential reading.

*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

Joel on SoftwareAnd 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 CapacityApress

Joel's arresting imagery has shaped the church's eschatological vision of a day of wrath. Amos's ringing indictments have periodically awakened the conscience of God's people. Twenty-five-hundred years later, those prophetic words still speak powerfully. This Tyndale commentary examines the two books' literary features, historical context, theology, and ethics.

The Art of UNIX Programming poses the belief that understanding the unwritten UNIX engineering tradition and mastering its design patterns will help programmers of all stripes to become better programmers. This book attempts to capture the engineering wisdom and design philosophy of the UNIX, Linux, and Open Source software development community as it has evolved over the past three decades, and as it is applied today by the most experienced programmers. Eric Raymond offers the next generation of "hackers" the unique opportunity to learn the connection between UNIX philosophy and practice through careful case studies of the very best UNIX/Linux programs.

We are living in very decisive times where the human being is simply betting their lives in eternity in an alarming way. It is necessary that we pause to question ourselves, what is happening to us? If we go back to see our history, we see that we have only had around twenty-five to thirty years of peace. It seems to be that we have always been at war: at war between countries, at war with ethics, at war between social classes, at war with gangs, at war between families, and even at war with ourselves. There is a point where you find yourself arguing with your own self and hate is growing all around us and we ask ourselves what is going on with humanity? Who could have the answer to this question? Is there someone that will end the enmity? I believe all these questions deserve a reasonable answer so we can unmask the one who is to blame.

A noted journalist chronicles three years in the lives of a team of maverick software developers, led by Lotus 1-2-3 creator Mitch Kapor, intent on creating a revolutionary personal information manager to challenge Microsoft Outlook. Reprint. 30,000 first printing.

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.

Describes influential business philosophies and marketing ideas from the past twenty years and examines why they did not work.

Build Prometheus ecosystems with metric-centric visualization, alerting, and querying Key Features Integrate Prometheus with Alertmanager and Grafana for building a complete monitoring system Explore PromQL, Prometheus' functional query language, with easy-to-follow examples Learn how to deploy Prometheus components using Kubernetes and traditional instances Book Description Prometheus is an open source monitoring system. It provides a modern time series database, a robust query language, several metric visualization possibilities, and a reliable alerting solution for traditional and cloud-native infrastructure. This book covers the fundamental concepts of monitoring and explores Prometheus architecture, its data model, and how metric aggregation works. Multiple test environments are included to help explore different configuration scenarios, such as the use of various exporters and integrations. You’ll delve into PromQL, supported by several examples, and then apply that knowledge to alerting and recording rules, as well as

how to test them. After that, alert routing with Alertmanager and creating visualizations with Grafana is thoroughly covered. In addition, this book covers several service discovery mechanisms and even provides an example of how to create your own. Finally, you'll learn about Prometheus federation, cross-sharding aggregation, and also long-term storage with the help of Thanos. By the end of this book, you'll be able to implement and scale Prometheus as a full monitoring system on-premises, in cloud environments, in standalone instances, or using container orchestration with Kubernetes. What you will learn Grasp monitoring fundamentals and implement them using Prometheus Discover how to extract metrics from common infrastructure services Find out how to take full advantage of PromQL Design a highly available, resilient, and scalable Prometheus stack Explore the power of Kubernetes Prometheus Operator Understand concepts such as federation and cross-shard aggregation Unlock seamless global views and long-term retention in cloud-native apps with Thanos Who this book is for If you're a software developer, cloud administrator, site reliability engineer, DevOps enthusiast or system admin looking to set up a fail-safe monitoring and alerting system for sustaining infrastructure security and performance, this book is for you. Basic networking and infrastructure monitoring knowledge will help you understand the concepts covered in this book.