

Rapid Development Taming Wild Software Schedules

Yeah, reviewing a books **Rapid Development Taming Wild Software Schedules** could increase your near links listings. This is just one of the solutions for you to be successful. As understood, achievement does not recommend that you have wonderful points.

Comprehending as well as concurrence even more than extra will pay for each success. neighboring to, the broadcast as well as acuteness of this Rapid Development Taming Wild Software Schedules can be taken as capably as picked to act.

Leading a Software Development Team - Richard Whitehead 2001

Practical advice on leading a software development team, aimed at software engineers who have become project leaders.

[Design It!](#) - Michael Keeling 2017-10-18

Don't engineer by coincidence- design it like you mean it!

Filled with practical techniques, Design It! is the perfect introduction to software architecture for

programmers who are ready to grow their design skills. Lead your team as a software architect, ask the right stakeholders the right questions, explore design options, and help your team implement a system that promotes the right -ilities. Share your design decisions, facilitate collaborative design workshops that are fast, effective, and fun-and develop more awesome software! With dozens of design methods, examples, and practical know-

how, Design It! shows you how to become a software architect. Walk through the core concepts every architect must know, discover how to apply them, and learn a variety of skills that will make you a better programmer, leader, and designer. Uncover the big ideas behind software architecture and gain confidence working on projects big and small. Plan, design, implement, and evaluate software architectures and collaborate with your team, stakeholders, and other architects. Identify the right stakeholders and understand their needs, dig for architecturally significant requirements, write amazing quality attribute scenarios, and make confident decisions. Choose technologies based on their architectural impact, facilitate architecture-centric design workshops, and evaluate architectures using lightweight, effective methods. Write lean architecture descriptions people love to read. Run an architecture design studio, implement the

architecture you've designed, and grow your team's architectural knowledge. Good design requires good communication. Talk about your software architecture with stakeholders using whiteboards, documents, and code, and apply architecture-focused design methods in your day-to-day practice. Hands-on exercises, real-world scenarios, and practical team-based decision-making tools will get everyone on board and give you the experience you need to become a confident software architect.

Utopian Entrepreneur - Brenda Laurel 2001

A guide to doing socially positive work in the context of business.

More Effective Agile - Steve McConnell 2019-08-24

In this comprehensive yet accessible overview for software leaders, the author presents an impactful, action-oriented prescription-covering the practical considerations needed to ensure you reap the full benefits of effective Agile

Product Design and

Downloaded from
test.unicaribe.edu.do on
by guest

Development - Karl T. Ulrich
2003

Treating such contemporary design and development issues as identifying customer needs, design for manufacturing, prototyping, and industrial design, *Product Design and Development, 3/e*, by Ulrich and Eppinger presents in a clear and detailed way a set of product development techniques aimed at bringing together the marketing, design, and manufacturing functions of the enterprise. The integrative methods in the book facilitate problem solving and decision making among people with different disciplinary perspectives, reflecting the current industry trend to perform product design and development in cross-functional teams.

How to Be a Programmer -

Robert L. Read 2009-01-13

This book summarizes so many things we need to know as a programmer, from a programmer's perspective. Starting from the basic technical skills one must acquire, to managerial skills to

manage a team of programmers. Emphases are put on the ethics of working as a programmer and as a member of the team. Inside this book you'll find tips on how to learn communication language among your peers, how to talk to non-engineers, and how to deal with difficult people. This book also shows us how to take a break when needed, and how to recognize when to go home, and how to communicate and negotiate with your boss, so that you won't end up working for 50 to 60 hours a week. This is a very good book, one that should be a mandatory for wannabe and professional programmers. If you happened to be a manager who supervises a hive of programmers, this book should provide you with useful insights into their minds and habits.

Value-Based Software

Engineering - Stefan Biffel

2006-02-23

The IT community has always struggled with questions concerning the value of an organization's investment in

software and hardware. It is the goal of value-based software engineering (VBSE) to develop models and measures of value which are of use for managers, developers and users as they make tradeoff decisions between, for example, quality and cost or functionality and schedule – such decisions must be economically feasible and comprehensible to the stakeholders with differing value perspectives. VBSE has its roots in work on software engineering economics, pioneered by Barry Boehm in the early 1980s. However, the emergence of a wider scope that defines VBSE is more recent. VBSE extends the merely technical ISO software engineering definition with elements not only from economics, but also from cognitive science, finance, management science, behavioral sciences, and decision sciences, giving rise to a truly multi-disciplinary framework. Biffel and his co-editors invited leading researchers and structured

their contributions into three parts, following an introduction into the area by Boehm himself. They first detail the foundations of VBSE, followed by a presentation of state-of-the-art methods and techniques. The third part demonstrates the benefits of VBSE through concrete examples and case studies. This book deviates from the more anecdotal style of many management-oriented software engineering books and so appeals particularly to all readers who are interested in solid foundations for high-level aspects of software engineering decision making, i.e., to product or project managers driven by economics and to software engineering researchers and students. Requirements by Collaboration - Ellen Gottesdiener 2002 "I spend much time helping organizations capture requirements and even more time helping them recover from not capturing requirements. Many of them have gone through some motions regarding

requirements as if they were sleepwalking. It's time to wake up and do it right-and this book is going to be their alarm clock." - Jerry Weinberg, author of numerous books on productivity enhancement "In today's complex, fast-paced software development environment, collaboration-the intense peer-to-peer conversations that result in products, decisions, and knowledge sharing-is absolutely essential to success. But all too often, attempts to collaborate degenerate into agonizing meetings or ineffectual bull sessions. Ellen's wonderful book will help you bridge the gap-turning the agony of meetings into the ecstasy of effective collaboration." - Jim Highsmith, a pioneer in adaptive software development methods "Requirements by Collaboration presents a wealth of practical tools and techniques for facilitating requirements development workshops. It is suitable-no, essential reading-for requirements workshop

facilitators. It will help both technical people and customer representatives participate in these critical contributions to software success." - Karl Wieggers, Principal Consultant, Process Impact, author of Software Requirements "The need for this particular book, at this particular time, is crystal clear. We have entered a new age where software development must be viewed as a form of business problem solving. That means direct user participation in developing 'requirements, ' or more accurately, in jointly working the business problem. That, in turn, means facilitated sessions. In this book, Ellen Gottesdiener provides a wealth of practical ideas for ensuring that you have exactly the right stuff for this all-important area of professional art." - Ronald G. Ross, Principal, Business Rule Solutions, LLC, Executive Editor, www.BRCommunity.com "Gottesdiener's years of software development experience coupled with her straight-forward writing style

make her book a perfect choice for either a senior developer or a midlevel project manager. In addition to her technical experience, her knowledge of group dynamics balance the book by educating the reader on how to manage conflict and personality differences within a requirements team-something that is missing from most requirements textbooks...It is a required 'handbook' that will be referred to again and again." - Kay Christian, ebusiness Consultant, Conifer, Colorado "Requirements by Collaboration is a 'must read' for any system stakeholder. End users and system analysts will learn the significant value they can add to the systems development process. Management will learn the tremendous return they may receive from making a modest time/people investment in facilitated sessions. Facilitators will discover ways to glean an amazing amount of high-quality information in a relatively brief time." - Russ Schwartz, Computer System Quality Consultant, Global

Biotechnology Firm "In addition to showing how requirements are identified, evaluated, and confirmed, Ellen provides important guidance based on her own real-world experience for creating and managing the workshop environment in which requirements are generated. This book is an engaging and invaluable resource for project teams and sponsors, both business and IT, who are committed to achieving results in the most productive manner possible." - Hal Thilmony, Senior Manager, Business Process Improvement (Finance), CiscoSystems, Inc. "Project managers should read this book for assistance with planning the requirements process. Experienced facilitators will enrich their knowledge. New facilitators can use this book to get them up to speed and become more effective in less time." - Rob Stroober, Competence Development Manager and Project Manager, Deloitte & Touche Consultdata, The Netherlands "While many

books discuss the details of software requirement artifacts (for example, use cases), Ellen's new book zeros in on effective workshop techniques and tools used to gather the content of these artifacts. As a pioneer in requirements workshops, she shares her real-life experiences in a comprehensive and easy-to-read book with many helpful examples and diagrams." - Bill Bird, Aera Energy LLC

"Requirements by Collaboration is absolutely full of guidance on the most effective ways to use workshops in requirements capture. This book will help workshop owners and facilitators to determine and gain agreement on a sound set of requirements, which will form a solid foundation for the development work that is to follow." - Jennifer Stapleton, Software Process Consultant and author of DSDM: The Method in Practice "This book provides an array of techniques within a clear, structured process, along with excellent examples of how and when to

use them. It's an excellent, practical, and really useful handbook written by a very experienced author!" - Jean-Anne Kirk, Director DSDM Consortium and IAF Professional Development

"Ellen has written a detailed, comprehensive, and practical handbook for facilitating groups in gathering requirements. The processes she outlines give the facilitator tools to bring together very different perspectives from stakeholders elegantly and with practical, useable results." - Jo Nelson, Principal, ICA Associates, Inc., Chair, IAF (2001-2002) Requirements by Collaboration: Workshops for Defining Needs focuses on the human side of software development--how well we work with our customers and teammates. Experience shows that the quality and degree of participation, communication, respect, and trust among all the stakeholders in a project can strongly influence its success or failure. Ellen Gottesdiener points out that such qualities are especially

important when defining user requirements and she shows in this book exactly what to do about that fact. Gottesdiener shows specifically how to plan and conduct requirements workshops. These carefully organized and facilitated meetings bring business managers, technical staff, customers, and users into a setting where, together, they can discover, evolve, validate, verify, and agree upon their product needs. Not only are their requirements more effectively defined through this collaboration, but the foundation is laid for good teamwork throughout the entire project. Other books focus on how to build the product right. Requirements by Collaboration focuses instead on what must come first--the right product to build.

Rapid Application

Development - James Martin
1991

Agile Project Management with Scrum - Ken Schwaber

2004-02-11

The rules and practices for

Scrum—a simple process for managing complex projects—are few, straightforward, and easy to learn. But Scrum’s simplicity itself—its lack of prescription—can be disarming, and new practitioners often find themselves reverting to old project management habits and tools and yielding lesser results. In this illuminating series of case studies, Scrum co-creator and evangelist Ken Schwaber identifies the real-world lessons—the successes and failures—culled from his years of experience coaching companies in agile project management. Through them, you’ll understand how to use Scrum to solve complex problems and drive better results—delivering more valuable software faster. Gain the foundation in Scrum theory—and practice—you need to: Rein in even the most complex, unwieldy projects Effectively manage unknown or changing product requirements Simplify the chain of command with self-managing

development teams Receive clearer specifications—and feedback—from customers Greatly reduce project planning time and required tools Build—and release—products in 30-day cycles so clients get deliverables earlier Avoid missteps by regularly inspecting, reporting on, and fine-tuning projects Support multiple teams working on a large-scale project from many geographic locations Maximize return on investment!

201 Principles of Software Development - Alan Mark Davis 1995

Software -- Software Engineering.

100 Principles of Game Design - DESPAIN 2012-12-18

Game designers spend their lives solving extraordinary problems and facing mind-bending paradoxes. It's their job to make a meticulous plan for "spontaneous fun" players will want to experience over and over again. Pressure is heaped on with demands for innovation and blockbuster status. So designers find

themselves facing an abyss of problems, pressure, and possibilities, armed only with their brains and an assortment of design principles they picked up over years of experience. For the first time, 100 Principles of Game Design gathers some of the best of these big ideas into one toolkit. Seasoned designers will be glad they don't have to hold it all in their heads anymore, and beginning design students can use the book to learn the tools of the trade. When the going gets tough, everyone can turn to this book for guidance, inspiration, or just to remind them of what works. Collected from every popular school of thought in game design, these core principles are organized by theme: innovation, creation, balancing, and troubleshooting.

- Includes advances from the world's leading authorities on game design, some explained by the creators themselves
- A reference book of finite, individual principles for easy access, providing a jumping off point for further research
- Principles originating in fields

as diverse as architecture, psychiatry, and economics, but shown here as they apply to game design • Richly designed with illustrations and photos, making each principle easy to understand and memorable • Timeless approach includes feedback loops, game mechanics, prototyping, economies of scale, user-centered design, and much more Professional designers and instructors at one of the world's leading game design institutions lay out the building blocks of diverse knowledge required to design even the simplest of games.

Practices of an Agile

Developer - Venkat

Subramaniam 2006-04-04

These are the proven, effective agile practices that will make you a better developer. You'll learn pragmatic ways of approaching the development process and your personal coding techniques. You'll learn about your own attitudes, issues with working on a team, and how to best manage your learning, all in an iterative, incremental, agile style. You'll

see how to apply each practice, and what benefits you can expect. Bottom line: This book will make you a better developer.

Requirements Engineering: Foundation for Software Quality - Roel Wieringa
2010-06-17

This volume constitutes the refereed proceedings of the International Working Conference REFSQ 2010, held in Essen, Germany, in June/July 2010.

Code Complete - Steve

McConnell 2004-06-09

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

Digital Adaptation - Paul Boag 2014

This book is a practical resource on how to help senior

management understand the Web and adapt the business, culture, teams and workflows accordingly. No fluff, no theory—just techniques and strategies that worked in practice, and showed results. --

Adaptive Code - Gary McLean Hall 2017-04-18

Write code that can adapt to changes. By applying this book's principles, you can create code that accommodates new requirements and unforeseen scenarios without significant rewrites. Gary McLean Hall describes Agile best practices, principles, and patterns for designing and writing code that can evolve more quickly and easily, with fewer errors, because it doesn't impede change. Now revised, updated, and expanded, *Adaptive Code, Second Edition* adds indispensable practical insights on Kanban, dependency inversion, and creating reusable abstractions. Drawing on over a decade of Agile consulting and development experience, McLean Hall has updated his best-seller with

deeper coverage of unit testing, refactoring, pure dependency injection, and more. Master powerful new ways to:

- Write code that enables and complements Scrum, Kanban, or any other Agile framework
- Develop code that can survive major changes in requirements
- Plan for adaptability by using dependencies, layering, interfaces, and design patterns
- Perform unit testing and refactoring in tandem, gaining more value from both
- Use the “golden master” technique to make legacy code adaptive
- Build SOLID code with single-responsibility, open/closed, and Liskov substitution principles
- Create smaller interfaces to support more-diverse client and architectural needs
- Leverage dependency injection best practices to improve code adaptability
- Apply dependency inversion with the Stairway pattern, and avoid related anti-patterns

About You
This book is for programmers of all skill levels seeking more-practical insight into design patterns, SOLID principles,

unit testing, refactoring, and related topics. Most readers will have programmed in C#, Java, C++, or similar object-oriented languages, and will be familiar with core procedural programming techniques.

Design - Build - Run - Dave Ingram 2009-02-23

This unique and critical book shares no-fail secrets for building software and offers tried-and-true practices and principles for software design, development, and testing for mission-critical systems that must not fail. A veteran software architect walks you through the lifecycle of a project as well as each area of production readiness—functionality, availability, performance and scalability, operability, maintainability, and extensibility, and highlights their key concepts.

Peopleware - Tom DeMarco 2013

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

Rapid Development - Steve McConnell 1996

Project managers, technical leads, and Windows programmers throughout the industry share an important concern--how to get their development schedules under control. *Rapid Development* addresses that concern head-on with philosophy, techniques, and tools that help shrink and control development schedules and keep projects moving. The style is friendly and conversational--and the content is impressive.

The Best Software Writing I

- Avram Joel Spolsky

2006-11-30

* 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

Software Estimation - Steve

McConnell 2006-02-22

Often referred to as the "black art" because of its complexity and uncertainty, software estimation is not as difficult or

puzzling as people think. In fact, generating accurate estimates is straightforward—once you understand the art of creating them. In his highly anticipated book, acclaimed author Steve McConnell unravels the mystery to successful software estimation—distilling academic information and real-world experience into a practical guide for working software professionals. Instead of arcane treatises and rigid modeling techniques, this guide highlights a proven set of procedures, understandable formulas, and heuristics that individuals and development teams can apply to their projects to help achieve estimation proficiency. Discover how to: Estimate schedule and cost—or estimate the functionality that can be delivered within a given time frame Avoid common software estimation mistakes Learn estimation techniques for you, your team, and your organization * Estimate specific project activities—including

development, management, and defect correction Apply estimation approaches to any type of project—small or large, agile or traditional Navigate the shark-infested political waters that surround project estimates When many corporate software projects are failing, McConnell shows you what works for successful software estimation.

Rapid Development - Steve McConnell 1996-07-02

Corporate and commercial software-development teams all want solutions for one important problem—how to get their high-pressure development schedules under control. In **RAPID DEVELOPMENT**, author Steve McConnell addresses that concern head-on with overall strategies, specific best practices, and valuable tips that help shrink and control development schedules and keep projects moving. Inside, you'll find: A rapid-development strategy that can be applied to any project and the best practices to make that strategy work Candid

discussions of great and not-so-great rapid-development practices—estimation, prototyping, forced overtime, motivation, teamwork, rapid-development languages, risk management, and many others A list of classic mistakes to avoid for rapid-development projects, including creeping requirements, shortchanged quality, and silver-bullet syndrome Case studies that vividly illustrate what can go wrong, what can go right, and how to tell which direction your project is going **RAPID DEVELOPMENT** is the real-world guide to more efficient applications development.

Software Business - João M. Fernandes 2015

This book contains the refereed proceedings of the 6th International Conference on Software Business, ICSOB 2015, held in Braga, Portugal, in June 2015. The theme of the event was "Enterprising Cities" focusing on a noticeable spillover of software within other industries enabling new business models: Companies bundle their physical products

and software services into solutions and start to sell independent software products in addition to physical products. The 16 full, five short, and three doctoral symposium papers accepted for ICSOB were selected from 42 submissions. The papers span a wide range of issues related to contemporary software business--from strategic aspects that include external reuse, ecosystem participation, and acquisitions to operational challenges associated with running software business.

Professional Software Development - Steve McConnell 2004

bull; Renowned software expert Steve McConnell presents his latest thoughts on the condition of the software engineering profession bull; Helps software developers regain the sight of the big-picture reasons why their jobs matter bull; A thinking man's guide to the current state of software

Failsafe IS Project Delivery - Andrew Holmes 2018-02-05

This title was first published in

2001. Based on research and practical experience, this text highlights the contributory factors leading to project failure. The buisness-IT culture gap, information politics, escalating commitment, the problem of high investment and low return, and accountability for failure are all covered.

Designing Software Architectures - Humberto Cervantes 2016-04-29

Designing Software Architectures will teach you how to design any software architecture in a systematic, predictable, repeatable, and cost-effective way. This book introduces a practical methodology for architecture design that any professional software engineer can use, provides structured methods supported by reusable chunks of design knowledge, and includes rich case studies that demonstrate how to use the methods. Using realistic examples, you'll master the powerful new version of the proven Attribute-Driven Design (ADD) 3.0 method and will

learn how to use it to address key drivers, including quality attributes, such as modifiability, usability, and availability, along with functional requirements and architectural concerns. Drawing on their extensive experience, Humberto Cervantes and Rick Kazman guide you through crafting practical designs that support the full software life cycle, from requirements to maintenance and evolution. You'll learn how to successfully integrate design in your organizational context, and how to design systems that will be built with agile methods. Comprehensive coverage includes Understanding what architecture design involves, and where it fits in the full software development life cycle Mastering core design concepts, principles, and processes Understanding how to perform the steps of the ADD method Scaling design and analysis up or down, including design for pre-sale processes or lightweight architecture reviews

Recognizing and optimizing critical relationships between analysis and design Utilizing proven, reusable design primitives and adapting them to specific problems and contexts Solving design problems in new domains, such as cloud, mobile, or big data **Agile Software Development Quality Assurance** - Stamelos, Ioannis G. 2007-02-28 "This book provides the research and instruction used to develop and implement software quickly, in small iteration cycles, and in close cooperation with the customer in an adaptive way, making it possible to react to changes set by the constant changing business environment. It presents four values explaining extreme programming (XP), the most widely adopted agile methodology"--Provided by publisher. *After the Gold Rush* - Steve McConnell 1999 A new addition to the Best Practices series presents an incisive reflection on and look at the future of software engineering, offering a

collection of original essays on critical trends that will shape the industry. Original. (Beginner).

Applied Software Project Management - Andrew Stellman 2005-11-18

"If you're looking for solid, easy-to-follow advice on estimation, requirements gathering, managing change, and more, you can stop now: this is the book for you."--Scott Berkun, Author of The Art of Project Management What makes software projects succeed? It takes more than a good idea and a team of talented programmers. A project manager needs to know how to guide the team through the entire software project. There are common pitfalls that plague all software projects and rookie mistakes that are made repeatedly--sometimes by the same people! Avoiding these pitfalls is not hard, but it is not necessarily intuitive. Luckily, there are tried and true techniques that can help any project manager. In Applied Software Project Management, Andrew Stellman

and Jennifer Greene provide you with tools, techniques, and practices that you can use on your own projects right away. This book supplies you with the information you need to diagnose your team's situation and presents practical advice to help you achieve your goal of building better software. Topics include: Planning a software project Helping a team estimate its workload Building a schedule Gathering software requirements and creating use cases Improving programming with refactoring, unit testing, and version control Managing an outsourced project Testing software Jennifer Greene and Andrew Stellman have been building software together since 1998. Andrew comes from a programming background and has managed teams of requirements analysts, designers, and developers. Jennifer has a testing background and has managed teams of architects, developers, and testers. She has led multiple large-scale outsourced projects. Between

the two of them, they have managed every aspect of software development. They have worked in a wide range of industries, including finance, telecommunications, media, nonprofit, entertainment, natural-language processing, science, and academia. For more information about them and this book, visit stellman-greene.com

Software Project Survival Guide - Steve McConnell 1998
Looks at a successful software project and provides details for software development for clients using object-oriented design and programming.

The Economics of Software Quality - Capers Jones 2012
Poor quality continues to bedevil large-scale development projects, but few software leaders and practitioners know how to measure quality, select quality best practices, or cost-justify their usage. In *The Economics of Software Quality*, leading software quality experts Capers Jones and Jitendra Subramanyam show how to systematically measure the

economic impact of quality and how to use this information to deliver far more business value. Using empirical data from hundreds of software organizations, Jones and Subramanyam show how integrated inspection, static analysis, and testing can achieve defect removal rates exceeding 95 percent. They offer innovative guidance for predicting and measuring defects and quality; choosing defect prevention, pre-test defect removal, and testing methods; and optimizing post-release defect reporting and repair. This book will help you Prove that improved software quality translates into strongly positive ROI and greatly reduced TCO Drive better results from current investments in debugging and prevention Use quality techniques to stay on schedule and on budget Avoid "hazardous" metrics that lead to poor decisions Important note: The audio and video content included with this enhanced eBook can be viewed only using iBooks on an iPad,

iPhone, or iPod touch.

The Manager's Path - Camille

Fournier 2017-03-13

Managing people is difficult wherever you work. But in the tech industry, where management is also a technical discipline, the learning curve can be brutal—especially when there are few tools, texts, and frameworks to help you. In this practical guide, author Camille Fournier (tech lead turned CTO) takes you through each stage in the journey from engineer to technical manager. From mentoring interns to working with senior staff, you'll get actionable advice for approaching various obstacles in your path. This book is ideal whether you're a new manager, a mentor, or a more experienced leader looking for fresh advice. Pick up this book and learn how to become a better manager and leader in your organization. Begin by exploring what you expect from a manager Understand what it takes to be a good mentor, and a good tech lead Learn how to manage individual members while remaining focused on the

entire team Understand how to manage yourself and avoid common pitfalls that challenge many leaders Manage multiple teams and learn how to manage managers Learn how to build and bootstrap a unifying culture in teams

Micro-ISV - Robert Walsh
2006-11-21

*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

Solid Code - Donis Marshall
2009-02-18

Get best-in-class engineering practices to help you write more-robust, bug-free code.

Two Microsoft .NET development experts share real-world examples and proven methods for optimizing the software development life cycle—from avoiding costly

programming pitfalls to making your development team more efficient. Managed code developers at all levels will find design, prototyping, implementation, debugging, and testing tips to boost the quality of their code—today. Optimize each stage of the development process—from design to testing—and produce higher-quality applications. Use metaprogramming to reduce code complexity, while increasing flexibility and maintainability Treat performance as a feature—and manage it throughout the development life cycle Apply best practices for application scalability Employ preventative security measures to ward off malicious attacks Practice defensive programming to catch bugs before run time Incorporate automated builds, code analysis, and testing into the daily engineering process Implement better source-control management and check-in procedures Establish a quality-driven, milestone-based project rhythm—and improve your results!

Practical Project Initiation -
Karl Wiegiers 2007-08-08
Zero in on key project-initiation tasks—and build a solid foundation for successful software development. In this concise guide, critically-acclaimed author Karl E. Wiegiers fills a void in project management literature by focusing on the activities that are essential—but often overlooked—for launching any project. Drawing on his extensive experience, Karl shares lessons learned, proven practices, and tools for getting your project off to the right start—and steering it to ultimate success. Lay a foundation for project success—discover how to:
Effectively charter a project
Define meaningful criteria for project success and product releases
Negotiate achievable commitments for project teams and stakeholders
Identify and document potential barriers to success—and manage project risks
Apply the Wideband Delphi method for more accurate estimation
Measure project performance and avoid

common metrics traps
Systematically apply lessons
learned to future projects
Companion Web site includes:
Worksheets from inside the
book Project document
templates Resources for
project initiation and process
improvement

MOS Study Guide for Microsoft Access Expert

Exam MO-500 - Paul
McFedries 2020-04-08
Advance your everyday
proficiency with Access 2019.
And earn the credential that
proves it! Demonstrate your
expertise with Microsoft
Access! Designed to help you
practice and prepare for
Microsoft Office Specialist
(MOS): Access 2019
certification, this official Study
Guide delivers: In-depth
preparation for each MOS
objective Detailed procedures
to help build the skills
measured by the exam Hands-
on tasks to practice what
you've learned Practice files
and sample solutions Sharpen
the skills measured by these
objectives: Create and manage
databases Build tables Create

queries Create forms Create
reports About MOS A Microsoft
Office Specialist (MOS)
certification validates your
proficiency with Microsoft
Office programs,
demonstrating that you can
meet globally recognized
performance standards. Hands-
on experience with the
technology is required to
successfully pass Microsoft
Certification exams.

Debugging the Development Process - Steve Maguire 1994-01

The author explains how he
organized and supervised
effective software development
teams at the Microsoft
company to come up with
timely and high-quality
commercial applications,
offering a candid look at the
group dynamics of software
development. Original.
(Advanced).

Project Management Handbook - Jürg Kuster 2015-06-08

This practical handbook offers
a comprehensive guide to
efficient project management.
It pursues a broad, well-

structured approach, suitable for most projects, and allows newcomers, experienced project managers and decision-makers to find valuable input that matches their specific needs. The Project Management Compass guides readers through various sections of the book; templates and checklists offer additional support. The handbook's innovative structure combines concepts from systems engineering, management psychology, and process dynamics. This international edition will allow to share the authors' experience gained in many years of project work and over 2,000 project management and leadership seminars conducted for BWI Management Education in Zurich, Switzerland. This is an excellent handbook for practical project management in today's world. Prof. Dr. Heinz Schelle, Honorary Chairman of the GPM (German Project Management Association) The authors' many years in practical experience in setting up, implementing and

managing projects shines through in this book. The book also reflects the current trend towards increased social competence. I am therefore pleased to recommend this book as a basis for certification in project management. Dr. Hans Knöpfel, Honorary President of the SPM (Swiss Project Management Association)

Crystal Clear - Alistair Cockburn 2004-10-19

Carefully researched over ten years and eagerly anticipated by the agile community, *Crystal Clear: A Human-Powered Methodology for Small Teams* is a lucid and practical introduction to running a successful agile project in your organization. Each chapter illuminates a different important aspect of orchestrating agile projects. Highlights include Attention to the essential human and communication aspects of successful projects Case studies, examples, principles, strategies, techniques, and guiding properties Samples of work products from real-world

projects instead of blank templates and toy problems
Top strategies used by software teams that excel in delivering quality code in a timely fashion
Detailed introduction to emerging best-practice techniques, such as Blitz Planning, Project 360°, and the essential Reflection Workshop
Question-and-answer with the author about how he arrived at these recommendations, including where they fit with CMMI, ISO,

RUP, XP, and other methodologies
A detailed case study, including an ISO auditor's analysis of the project
Perhaps the most important contribution this book offers is the Seven Properties of Successful Projects. The author has studied successful agile projects and identified common traits they share. These properties lead your project to success; conversely, their absence endangers your project.