

## Specification By Example How Successful Teams Deliver The Right Software

Although service-level objectives (SLOs) continue to grow in importance, there's a distinct lack of information about how to implement them. Practical advice that does exist usually assumes that your team already has the infrastructure, tooling, and culture in place. In this book, recognized SLO expert Alex Hidalgo explains how to build an SLO culture from the ground up. Ideal as a primer and daily reference for anyone creating both the culture and tooling necessary for SLO-based approaches to reliability, this guide provides detailed analysis of advanced SLO and service-level indicator (SLI) techniques. Armed with mathematical models and statistical knowledge to help you get the most out of an SLO-based approach, you'll learn how to build systems capable of measuring meaningful SLIs with buy-in across all departments of your organization. Define SLIs that meaningfully measure the reliability of a service from a user's perspective Choose appropriate SLO targets, including how to perform statistical and probabilistic analysis Use error budgets to help your team have better discussions and make better data-driven decisions Build supportive tooling and resources required for an SLO-based approach Use SLO data to present meaningful reports to leadership and your users

"Information Systems for Business and Beyond introduces the concept of information systems, their use in business, and the larger impact they are having on our world."--BC Campus website. Specification by Example How Successful Teams Deliver the Right Software Manning Publications With Acceptance Test-Driven Development (ATDD), business customers, testers, and developers can collaborate to produce testable requirements that help them build higher quality software more rapidly. However, ATDD is still widely misunderstood by many practitioners. ATDD by Example is the first practical, entry-level, hands-on guide to implementing and successfully applying it. ATDD pioneer Markus Gärtner walks readers step by step through deriving the right systems from business users, and then implementing fully automated, functional tests that accurately reflect business requirements, are intelligible to stakeholders, and promote more effective development. Through two end-to-end case studies, Gärtner demonstrates how ATDD can be applied using diverse frameworks and languages. Each case study is accompanied by an extensive set of artifacts, including test automation classes, step definitions, and full sample implementations. These realistic examples illuminate ATDD's fundamental principles, show how ATDD fits into the broader development process, highlight tips from Gärtner's extensive experience, and identify crucial pitfalls to avoid. Readers will learn to Master the thought processes associated with successful ATDD implementation Use ATDD with Cucumber to describe software in ways businesspeople can understand Test web pages using ATDD tools Bring ATDD to Java with the FitNesse wiki-based acceptance test framework Use examples more effectively in Behavior-Driven Development (BDD) Specify software collaboratively through innovative workshops Implement more user-friendly and collaborative test automation Test more cleanly, listen to test results, and refactor tests for greater value If you're a tester, analyst, developer, or project manager, this book offers a concrete foundation for achieving real benefits with ATDD now-and it will help you reap even more value as you gain experience.

Data is at the center of many challenges in system design today. Difficult issues need to be figured out, such as scalability, consistency, reliability, efficiency, and maintainability. In addition, we have an overwhelming variety of tools, including relational databases, NoSQL datastores, stream or batch processors, and message brokers. What are the right choices for your application? How do you make sense of all these buzzwords? In this practical and comprehensive guide, author Martin Kleppmann helps you navigate this diverse landscape by examining the pros and cons of various technologies for processing and storing data. Software keeps changing, but the fundamental principles remain the same. With this book, software engineers and architects will learn how to apply those ideas in practice, and how to make full use of data in modern applications. Peer under the hood of the systems you already use, and learn how to use and operate them more effectively Make informed decisions by identifying the strengths and weaknesses of different tools Navigate the trade-offs around consistency, scalability, fault tolerance, and complexity Understand the distributed systems research upon which modern databases are built Peek behind the scenes of major online services, and learn from their architectures

Disciplined Entrepreneurship

Dive into the Future of Infrastructure

Restoring Rigor, Motivating Students, and Saving Faculty Time

Feature Engineering for Machine Learning and Data Analytics

Effective Unit Testing

A guide for Java developers

Writing Effective Use Cases

**Written by the inventors of the technology, The Java® Virtual Machine Specification, Java SE 7 Edition, is the definitive technical reference for the Java Virtual Machine. The book provides complete, accurate, and detailed coverage of the Java Virtual Machine. It fully describes the invokedynamic instruction and method handle mechanism added in Java SE 7, and gives the formal Prolog specification of the type-**

**checking verifier introduced in Java SE 6. The book also includes the class file extensions for generics and annotations defined in Java SE 5.0, and aligns the instruction set and initialization rules with the Java Memory Model.**

**A practical guide to impact mapping, a simple yet incredibly effective method for collaborative strategic planning that helps organizations make an impact with software.**

**Feature engineering plays a vital role in big data analytics. Machine learning and data mining algorithms cannot work without data. Little can be achieved if there are few features to represent the underlying data objects, and the quality of results of those algorithms largely depends on the quality of the available features. Feature Engineering for Machine Learning and Data Analytics provides a comprehensive introduction to feature engineering, including feature generation, feature extraction, feature transformation, feature selection, and feature analysis and evaluation. The book presents key concepts, methods, examples, and applications, as well as chapters on feature engineering for major data types such as texts, images, sequences, time series, graphs, streaming data, software engineering data, Twitter data, and social media data. It also contains generic feature generation approaches, as well as methods for generating tried-and-tested, hand-crafted, domain-specific features. The first chapter defines the concepts of features and feature engineering, offers an overview of the book, and provides pointers to topics not covered in this book. The next six chapters are devoted to feature engineering, including feature generation for specific data types. The subsequent four chapters cover generic approaches for feature engineering, namely feature selection, feature transformation based feature engineering, deep learning based feature engineering, and pattern based feature generation and engineering. The last three chapters discuss feature engineering for social bot detection, software management, and Twitter-based applications respectively. This book can be used as a reference for data analysts, big data scientists, data preprocessing workers, project managers, project developers, prediction modelers, professors, researchers, graduate students, and upper level undergraduate students. It can also be used as the primary text for courses on feature engineering, or as a supplement for courses on machine learning, data mining, and big data analytics.**

**This book will help you write better stories, spot and fix common issues, split stories so that they are smaller but still valuable, and deal with difficult stuff like crosscutting concerns, long-term effects and non-functional requirements. Above all, this book will help you achieve the promise of agile and iterative delivery: to ensure that the right stuff gets delivered through productive discussions between delivery team members and business stakeholders. Who is this book for? This is a book for anyone working in an iterative delivery environment, doing planning with user stories. The ideas in this book are useful both to people relatively new to user stories and those who have been working with them for years. People who work in software delivery, regardless of their role, will find plenty of tips for engaging stakeholders better and structuring iterative plans more effectively. Business stakeholders working with software teams will discover how to provide better information to their delivery groups, how to set better priorities and how to outrun the competition by achieving more with less software. What's inside? Unsurprisingly, the book contains exactly fifty ideas. They are grouped into five major parts: - Creating stories: This part deals with capturing information about stories before they get accepted into the delivery pipeline. You'll find ideas about what kind of information to note down on story cards and how to quickly spot potential problems. - Planning with stories: This part contains ideas that will help you manage the big-picture view, set milestones and organise long-term work. - Discussing stories: User stories are all about effective conversations, and this part contains ideas to improve discussions between delivery teams and business stakeholders. You'll find out how to discover hidden assumptions and how to facilitate effective conversations to ensure shared understanding. - Splitting stories: The ideas in this part will help you deal with large and difficult stories, offering several strategies for dividing them into smaller chunks that will help you learn fast and deliver value quickly. - Managing iterative delivery: This part contains ideas that will help you work with user stories in the short and mid term, manage capacity, prioritise and reduce scope to achieve the most with the least software. About the authors: Gojko Adzic is a strategic software delivery consultant who works with ambitious teams to improve the quality of their software products and processes. Gojko's book Specification by Example was awarded the #2 spot on the top 100 agile books for 2012 and won the Jolt Award for the best book of 2012. In 2011, he was voted by peers as the most influential agile testing professional, and his blog won the UK agile award for the best online publication in 2010. David Evans is a consultant, coach and trainer specialising in the field of Agile Quality. David helps organisations with strategic process improvement and coaches teams on effective agile practice. He is regularly in demand as a conference speaker and has had several articles published in international journals.**

**24 Steps to Success! Disciplined Entrepreneurship will change the way you think about starting a company. Many believe that entrepreneurship cannot be taught, but great entrepreneurs aren't born with something special - they simply make great products. This book will show you how to create a successful startup through developing an innovative product. It breaks down the necessary processes into an integrated, comprehensive, and proven 24-step framework that any industrious person can learn and apply. You will learn: Why the "F" word - focus - is crucial to a startup's success Common obstacles that**

**entrepreneurs face - and how to overcome them How to use innovation to stand out in the crowd - it's not just about technology Whether you're a first-time or repeat entrepreneur, Disciplined Entrepreneurship gives you the tools you need to improve your odds of making a product people want. Author Bill Aulet is the managing director of the Martin Trust Center for MIT Entrepreneurship as well as a senior lecturer at the MIT Sloan School of Management. For more please visit <http://disciplinedentrepreneurship.com/>**

**Human-System Integration in the System Development Process**

**How People Learn**

**The Negotiation Book**

**Specification by Example and Agile Acceptance Testing**

**ATDD by Example**

**User Story Mapping**

**Software Requirements**

Intelligent readers who want to build their own embedded computer systems-- installed in everything from cell phones to cars to handheld organizers to refrigerators-- will find this book to be the most in-depth, practical, and up-to-date guide on the market. Designing Embedded Hardware carefully steers between the practical and philosophical aspects, so developers can both create their own devices and gadgets and customize and extend off-the-shelf systems. There are hundreds of books to choose from if you need to learn programming, but only a few are available if you want to learn to create hardware. Designing Embedded Hardware provides software and hardware engineers with no prior experience in embedded systems with the necessary conceptual and design building blocks to understand the architectures of embedded systems. Written to provide the depth of coverage and real-world examples developers need, Designing Embedded Hardware also provides a road-map to the pitfalls and traps to avoid in designing embedded systems. Designing Embedded Hardware covers such essential topics as: The principles of developing computer hardware Core hardware designs Assembly language concepts Parallel I/O Analog-digital conversion Timers (internal and external) UART Serial Peripheral Interface Inter-Integrated Circuit Bus Controller Area Network (CAN) Data Converter Interface (DCI) Low-power operation This invaluable and eminently useful book gives you the practical tools and skills to develop, build, and program your own application-specific computers.

This is the official guide and reference manual for Subversion 1.6 - the popular open source revision control technology.

PDF is becoming the standard for digital documents worldwide, but it's not easy to learn on your own. With capabilities that let you use a variety of images and text, embed audio and video, and provide links and navigation, there's a lot to explore. This practical guide helps you understand how to work with PDF to construct your own documents, troubleshoot problems, and even build your own tools. You'll also find best practices for producing, manipulating, and consuming PDF documents. In addition, this highly approachable reference will help you navigate the official (and complex) ISO documentation. Learn how to combine PDF objects into a cohesive whole Use PDF's imaging model to create vector and raster graphics Integrate text, and become familiar with fonts and glyphs Provide navigation within and between documents Use annotations to overlay or incorporate additional content Build interactive forms with the Widget annotation Embed related files such as multimedia, 3D content, and XML files Use optional content to enable non-printing graphics Tag content with HTML-like structures, including paragraphs and tables

In April 1991 BusinessWeek ran a cover story entitled, "Can't Work This #@! Thing," about the difficulties many people have with consumer products, such as cell phones and VCRs. More than 15 years later, the situation is much the same--but at a very different level of scale. The disconnect between people and technology has had society-wide consequences in the large-scale system accidents from major human error, such as those at Three Mile Island and in Chernobyl. To prevent both the individually annoying and nationally significant consequences, human capabilities and needs must be considered early and throughout system design and development. One challenge for such consideration has been providing the background and data needed for the seamless integration of humans into the design process from various perspectives: human factors engineering, manpower, personnel, training, safety and health, and, in the military, habitability and survivability. This collection of development activities has come to be called human-system integration (HSI). Human-System Integration in the System Development Process reviews in detail more than 20 categories of HSI methods to provide invaluable guidance and information for system designers and developers.

The classic, landmark work on software testing The hardware and software of computing have changed markedly in the three decades since the first edition of The Art of Software Testing, but this book's powerful underlying analysis has stood the test of time. Whereas most books on software testing target particular development techniques, languages, or testing methods, The Art of Software Testing, Third Edition provides a brief but powerful and comprehensive presentation of time-proven software testing approaches. If your software development project is mission-critical, this book is an investment that will pay for itself with the first bug you find. The new Third Edition explains how to apply the book's classic principles to today's hot topics including: Testing apps for iPhones, iPads, BlackBerrys, Androids, and other mobile devices Collaborative (user) programming and testing Testing for Internet applications, e-commerce, and agile programming environments Whether you're a student looking for a testing guide you'll use for the rest of your career, or an IT manager overseeing a software development team, The Art of Software Testing, Third Edition is an expensive book that will pay for itself many times over.

A New Look

Implementing Service Level Objectives

24 Steps to a Successful Startup

Designing Embedded Hardware

How Successful Teams Deliver the Right Software

Subversion 1.6 Official Guide

Kubernetes: Up and Running

Winner! - CMI Management Book of the Year 2017 – Practical Manager category Master the art of negotiation and gain the competitive advantage Now revised and updated, the second edition of The Negotiation Book will teach you about one of the most important skills in business. We all have to negotiate at some point; whether in the office or at home and good negotiation skills can have a profound effect on our lives – both financially and personally. No other skill will give you a better chance of optimizing your success and your organization's success. Every time you negotiate, you are looking for an increased advantage. This book delivers it, whilst ensuring the other party also comes away feeling good about the deal. Nothing will put you in a stronger position to build capacity, build negotiation strategies and facilitate negotiations through to successful conclusions. The Negotiation Book: Explains the importance of planning, dynamics and strategies Will help you understand the psychology, tactics and behaviours of negotiation Teaches you how to conduct successful win-win negotiations Gives you the competitive advantage

Linda Nilson puts forward an innovative but practical and tested approach to grading--the specifications grading paradigm--which restructures assessments to streamline the grading process and greatly reduce grading time, empower students to choose the level of attainment they want to achieve, reduce antagonism between the evaluator and the evaluated, and increase student receptivity to meaningful feedback, thus facilitating the learning process - all while upholding rigor. In addition, specs grading increases students' motivation to do well by making expectations clear, lowering their stress and giving them agency in determining their course goals. Among the unique characteristics of the schema, all of which simplify faculty decision making, are the elimination of partial credit, the reliance on a one-level grading rubric and the "bundling" of assignments and tests around learning outcomes. Successfully completing more challenging bundles (or modules) earns a student a higher course grade. Specs grading works equally well in small and large class settings and encourages "authentic assessment." Used consistently over time, it can restore credibility to grades by demonstrating and making transparent to all stakeholders the learning outcomes that students achieve. Publisher Fact Sheet A concise, hands-on approach to managing & improving the critical requirements process in software development.

Get the must-have reference on standards and best practices for the delivery of a construction project. The CSI Practice Guides are a library of comprehensive references specifically and carefully designed for the construction professional. Each book examines important concepts and best practices integral to a particular aspect of the construction process. Laying the foundation for this series, The CSI Project Delivery Practice Guide provides fundamental knowledge for the documentation, administration, and successful delivery of construction projects. It also serves as the pivotal starting point for understanding CSI's core values, as well as a useful study aid for those wishing to obtain the Construction Documents Technologist certificate. This easy-to-follow guide: Is a great introduction to the construction process for the new practitioner. Functions as a ready reference for the experienced construction professional. Packaged with the book is an access code which allows access to a password-protected web site with bonus content, including a PDF of the printed book and samples of CSI format documents, such as UniFormat and SectionFormat/PageFormat. The CSI Project Delivery Practice Guide offers general information all construction professionals need for understanding their roles in the delivery of a construction project. Key principles are presented and discussed in detail to allow the reader to take full advantage of material covered in depth by the more specialized CSI Practice Guides. If you can own only one Practice Guide, this is the one to get.

Authored by two of the leading authorities in the field, this guide offers readers the knowledge and skills needed to achieve proficiency with embedded software.

Designing Data-Intensive Applications

Information Systems for Business and Beyond

Writing Great Specifications

Principles of Applied Civil Engineering Design

Site Reliability Engineering

The Big Ideas Behind Reliable, Scalable, and Maintainable Systems

The SAGE Encyclopedia of Educational Research, Measurement, and Evaluation

Ying-Kit Choi walks engineers through standard practices, basic principles, and design philosophy needed to prepare quality design and construction documents for a successful infrastructure project.

Written by the creator of SpecFlow and the author of The Cucumber for Java Book, this book provides inside information on how to get the most out of the discovery phase of Behaviour Driven Development (BDD). This practical guide demonstrates good collaboration techniques, illustrated by concrete examples. This book is written for everyone involved in the specification and delivery of software (including product owners, business analysts, developers, and testers). The book starts by explaining the reasons BDD exists in the first place and describes techniques for getting the most out of collaboration between business and delivery team members. This is the first in the BDD Books series that will guide you through the entire development process, including specific technical practices needed to successfully drive development using collaboratively-authored specifications and living documentation.

Effective awk Programming, 3rd Edition, focuses entirely on awk, exploring it in the greatest depth of the three awk titles we carry. It's an excellent companion piece to the more broadly focused second edition. This book provides complete coverage of the gawk 3.1 language as well as the most up-to-date coverage of the POSIX standard for awk available anywhere. Author Arnold Robbins clearly distinguishes standard awk features from GNU awk (gawk)-specific features, shines light into many of the "dark corners" of the language (areas to watch out for when programming), and devotes two full chapters to example programs. A brand new chapter is devoted to TCP/IP networking with gawk. He includes a summary of how the awk language evolved. The book also covers: Internationalization of gawk Interfacing to i18n at the awk level Two-way pipes TCP/IP networking via the two-way pipe interface The new PROCINFO array, which provides information about running gawk Profiling and pretty-printing awk programs In addition to covering the awk language, this book serves as the official "User's Guide" for the GNU implementation of awk (gawk), describing in an integrated fashion the extensions available to the System V Release 4 version of awk that are also available in gawk. As the official gawk User's Guide, this book will also be available electronically, and can be freely copied and distributed under the terms of the Free Software Foundation's Free Documentation License (FDL).

A portion of the proceeds from sales of this book will go to the Free Software Foundation to support further development of free and open source software. The third edition of Effective awk Programming is a GNU Manual and is published by O'Reilly & Associates under the Free Software Foundation's Free Documentation License (FDL). A portion of the proceeds from the sale of this book is donated to the Free Software Foundation to further development of GNU software. This book is also available in electronic form; you have the freedom to modify this GNU Manual, like GNU software. Copies published by the Free Software Foundation raise funds for GNU development. Legend has it that Google deploys over two billion application containers a week. How's that possible? Google revealed the secret through a project called Kubernetes, an open source cluster orchestrator (based on its internal Borg system) that radically simplifies the task of building, deploying, and maintaining scalable distributed systems in the cloud. This practical guide shows you how Kubernetes and container technology can help you achieve new levels of velocity, agility, reliability, and efficiency. Authors Kelsey Hightower, Brendan Burns, and Joe Beda—who've worked on Kubernetes at Google and other organizations—explain how this system fits into the lifecycle of a distributed application. You will learn how to use tools and APIs to automate scalable distributed systems, whether it is for online services, machine-learning applications, or a cluster of Raspberry Pi computers. Explore the distributed system challenges that Kubernetes addresses Dive into containerized application development, using containers such as Docker Create and run containers on Kubernetes, using the docker image format and container runtime Explore specialized objects essential for running applications in production Reliably roll out new software versions without downtime or errors Get examples of how to develop and deploy real-world applications in Kubernetes

Bridging the Communication Gap is a book about improving communication between customers, business analysts, developers and testers on software projects, especially by using specification by example and agile acceptance testing. These two key emerging software development practices can significantly improve the chances of success of a software project. They ensure that all project participants speak the same language, and build a shared and consistent understanding of the domain. This leads to better specifications, flushes out incorrect assumptions and ensures that functional gaps are discovered before the development starts. With these practices in place you can build software that is genuinely fit for purpose. Using Specification by Example and Gherkin

Making a Big Impact with Software Products and Projects

A Guide to the Project Management Body of Knowledge (PMBOK® Guide) - Seventh Edition and The Standard for Project Management (RUSSIAN)

A Handbook For Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage

Maintenance and Design Manual

Explore Behaviour Using Examples

Discovery

User story mapping is a valuable tool for software development, once you understand why and how to use it. This insightful book examines how this often misunderstood technique can help your team stay focused on users and their needs without getting lost in the enthusiasm for individual product features. Author Jeff Patton shows you how changeable story maps enable your team to hold better conversations about the project throughout the development process. Your team will learn to come away with a shared understanding of what you're attempting to build and why. Get a high-level view of story mapping, with an exercise to learn key concepts quickly

Understand how stories really work, and how they come to life in Agile and Lean projects Dive into a story's lifecycle, starting with opportunities and moving deeper into discovery Prepare your stories, pay attention while they're built, and learn from those you convert to working software

Describes a method of effectively specifying, testing, and delivering software, covering such topics as documentation, process patterns, and automation, along with case studies from a variety of firms.

Summary Writing Great Specifications is an example-rich tutorial that teaches you how to write good Gherkin specification documents that take advantage of the benefits of specification by example. Foreword written by Gojko Adzic. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology The clearest way to communicate a software specification is to provide examples of how it should work. Turning these story-based descriptions into a well-organized dev plan is another matter. Gherkin is a human-friendly, jargon-free language for documenting a suite of examples as an executable specification. It fosters efficient collaboration between business and dev teams, and it's an excellent foundation for the specification by example (SBE) process. About the Book Writing Great Specifications teaches you how to capture executable software designs in Gherkin following the SBE method. Written for both developers and non-technical team members, this practical book starts with collecting individual feature stories and organizing them into a full, testable spec. You'll learn to choose the best scenarios, write them in a way that anyone can understand, and ensure they can be easily updated by anyone.management. What's Inside Reading and writing Gherkin Designing story-based test cases Team Collaboration Managing a suite of Gherkin documents About the Reader Primarily written for developers and architects, this book is accessible to any member of a software design team. About the Author Kamil Nieceja is a seasoned engineer, architect, and project manager with deep expertise in Gherkin and SBE. Table of contents Introduction to specification by example and Gherkin PART 1 - WRITING EXECUTABLE SPECIFICATIONS WITH EXAMPLES The specification layer and the automation layer Mastering the Given-When-Then template The basics of scenario outlines Choosing examples for scenario

outlines The life cycle of executable specifications Living documentation PART 2 - MANAGING SPECIFICATION SUITES Organizing scenarios into a specification suite Refactoring features into abilities and business needs Building a domain-driven specification suite Managing large projects with bounded contexts

The overwhelming majority of a software system's lifespan is spent in use, not in design or implementation. So, why does conventional wisdom insist that software engineers focus primarily on the design and development of large-scale computing systems? In this collection of essays and articles, key members of Google's Site Reliability Team explain how and why their commitment to the entire lifecycle has enabled the company to successfully build, deploy, monitor, and maintain some of the largest software systems in the world. You'll learn the principles and practices that enable Google engineers to make systems more scalable, reliable, and efficient—lessons directly applicable to your organization. This book is divided into four sections:

Introduction—Learn what site reliability engineering is and why it differs from conventional IT industry practices Principles—Examine the patterns, behaviors, and areas of concern that influence the work of a site reliability engineer (SRE) Practices—Understand the theory and practice of an SRE's day-to-day work: building and operating large distributed computing systems Management—Explore Google's best practices for training, communication, and meetings that your organization can use

Larman covers how to investigate requirements, create solutions and then translate designs into code, showing developers how to make practical use of the most significant recent developments. A summary of UML notation is included

Text Processing and Pattern Matching

Competitive Engineering

The CSI Project Delivery Practice Guide

Discover the Whole Story, Build the Right Product

Producing Drawings, Specifications, and Cost Estimates for Heavy Civil Projects

Your Definitive Guide to Successful Negotiating

Summary Effective Unit Testing is written to show how to write good tests—tests that are concise and to the point, expressive, useful, and maintainable. Inspired by Roy Osherove's bestselling *The Art of Unit Testing*, this book focuses on tools and practices specific to the Java world. It introduces you to emerging techniques like behavior-driven development and specification by example, and shows you how to add robust practices into your toolkit. About Testing Test the components before you assemble them into a full application, and you'll get better software. For Java developers, there's now a decade of experience with well-crafted tests that anticipate problems, identify known and unknown dependencies in the code, and allow you to test components both in isolation and in the context of a full application. About this Book Effective Unit Testing teaches Java developers how to write unit tests that are concise, expressive, useful, and maintainable. Offering crisp explanations and easy-to-absorb examples, it introduces emerging techniques like behavior-driven development and specification by example. Programmers who are already unit testing will learn the current state of the art. Those who are new to the game will learn practices that will serve them well for the rest of their career. Purchase of the print book comes with an offer of a free PDF, ePub, and Kindle eBook from Manning. Also available is all code from the book. About the Author Lasse Koskela is a coach, trainer, consultant, and programmer. He hacks on open source projects, helps companies improve their productivity, and speaks frequently at conferences around the world. Lasse is the author of *Test Driven*, also published by Manning. What's Inside A thorough introduction to unit testing Choosing best-of-breed tools Writing tests using dynamic languages Efficient test automation Table of Contents PART 1 FOUNDATIONS The promise of good tests In search of good Test doubles PART 2 CATALOG Readability Maintainability Trustworthiness PART 3 DIVERSIONS Testable design Writing tests in other JVM languages Speeding up test execution

This User's Guide is intended to support the design, implementation, analysis, interpretation, and quality evaluation of registries created to increase understanding of patient outcomes. For the purposes of this guide, a patient registry is an organized system that uses observational study methods to collect uniform data (clinical and other) to evaluate specified outcomes for a population defined by a particular disease, condition, or exposure, and that serves one or more predetermined scientific, clinical, or policy purposes. A registry database is a file (or files) derived from the registry. Although registries can serve many purposes, this guide focuses on registries created for one or more of the following purposes: to describe the natural history of disease, to determine clinical effectiveness or cost-effectiveness of health care products and services, to measure or monitor safety and harm, and/or to measure quality of care. Registries are classified according to how their populations are defined. For example, product registries include patients who have been exposed to biopharmaceutical products or medical devices. Health services registries consist of patients who have had a common procedure, clinical encounter, or hospitalization. Disease or condition registries are defined by patients having the same diagnosis, such as cystic fibrosis or heart failure. The User's Guide was created by researchers affiliated with AHRQ's Effective Health Care Program, particularly those who participated in AHRQ's DEcIDE (Developing Evidence to Inform Decisions About Effectiveness) program. Chapters were subject to multiple internal and external independent reviews.

This guide will help readers learn how to employ the significant power of use cases to their software development efforts. It provides a practical methodology, presenting key use case concepts.

PMBOK® Guide is the go-to resource for project management practitioners. The project management profession has significantly evolved due to emerging technology, new approaches and rapid market changes. Reflecting this evolution, The Standard for Project Management enumerates 12 principles of project management and the PMBOK® Guide – Seventh Edition is structured around eight project performance domains. This edition is designed to address practitioners' current and future needs and to help them be more proactive, innovative and nimble in enabling desired project outcomes. This edition of the PMBOK® Guide: • Reflects the full range of development approaches (predictive, adaptive, hybrid, etc.); • Provides an entire section devoted to tailoring the development approach and processes; • Includes an expanded list of models, methods, and artifacts; • Focuses on not just delivering project outputs but also enabling outcomes; and • Integrates with PMStandards+™ for information and standards application content based on project type, development approach, and industry sector. Competitive Engineering documents Tom Gilb's unique, ground-breaking approach to communicating management objectives and systems engineering requirements, clearly and unambiguously. Competitive Engineering is a revelation for anyone involved in management and risk control. Already used by thousands of project managers and systems engineers around the world, this is a handbook for initiating, controlling and delivering complex projects on time and within budget. The Competitive Engineering methodology provides a practical set of tools and techniques that enable readers to effectively design, manage and deliver results in any complex organization - in engineering, industry, systems

engineering, software, IT, the service sector and beyond. Elegant, comprehensive and accessible, the Competitive Engineering methodology provides a practical set of tools and techniques that enable readers to effectively design, manage and deliver results in any complex organization - in engineering, industry, systems engineering, software, IT, the service sector and beyond. Provides detailed, practical and innovative coverage of key subjects including requirements specification, design evaluation, specification quality control and evolutionary project management Offers a complete, proven and meaningful 'end-to-end' process for specifying, evaluating, managing and delivering high quality solutions Tom Gilb's clients include HP, Intel, CitiGroup, IBM, Nokia and the US Department of Defense

The Java Virtual Machine Specification, Java SE 7 Edition

Registries for Evaluating Patient Outcomes

Sap: How to Write a Report Functional Specification

Bridging the Communication Gap

APPLYING UML & PATTERNS 3RD EDITION

Developing with PDF

A Practical Guide to Acceptance Test-Driven Development

Summary Specification by Example is an emerging practice for creating software based on realistic examples, bridging the communication gap between business stakeholders and the dev teams building the software. In this book, author Gojko Adzic distills interviews with successful teams worldwide, sharing how they specify, develop, and deliver software, without defects, in short iterative delivery cycles. About the Technology Specification by Example is a collaborative method for specifying requirements and tests. Seven patterns, fully explored in this book, are key to making the method effective. The method has four main benefits: it produces living, reliable documentation; it defines expectations clearly and makes validation efficient; it reduces rework; and, above all, it assures delivery teams and business stakeholders that the software that's built is right for its purpose. About the Book This book distills from the experience of leading teams worldwide effective ways to specify, test, and deliver software in short, iterative delivery cycles. Case studies in this book range from small web startups to large financial institutions, working in many processes including XP, Scrum, and Kanban. This book is written for developers, testers, analysts, and business people working together to build great software. Purchase of the print book comes with an offer of a free PDF, ePub, and Kindle eBook from Manning. Also available is all code from the book.

What's Inside Common process patterns How to avoid bad practices Fitting SBE in your process 50+ case studies

=====  
Table of Contents Part 1 Getting started Part 2 Key process patterns Part 3 Case studies Key benefits Key process patterns Living documentation Initiating the changes Deriving scope from goals Specifying collaboratively Illustrating using examples Refining the specification Automating validation without changing specifications Validating frequently Evolving a documentation system uSwitch RainStor Iowa Student Loan Sabre Airline Solutions ePlan Services Songkick Concluding thoughts

Author Linda Timms goes beyond the standard consulting guide to bring you constructive reliable advice for delivering effective, complete, professional functional specs on time. Filled with plain English, real-world examples, hints and tips, SAP: How to Write a Report Functional Specification provides the secrets you need to make a daunting task achievable. Whether you are a SAP project team member seconded from the business, unsure where to start with documenting business requirements a support analyst dealing with change requests and new requirements an offshore analyst/consultant wanting to up your game, get recognition for top quality work, and stand out from the crowd anyone fresh out of a SAP academy or training course, wanting to transition smoothly into a valuable project team member a graduate with one of the big management consulting firms wanting focussed reliable advice to help build your consulting career a junior consultant wanting to make a name for yourself as a professional productive good SAP resource an experienced consultant wanting to refresh your knowledge and maybe kick some bad habits a business analyst looking to move into the SAP arena a manager or team lead tasked with reviewing and signing off functional specs a project manager wanting to bring in best practices a technical analyst wanting to understand the functional side of SAP requirements SAP: How to Write a Report Functional Specification is a comprehensive guide, including a free downloadable functional specification template that will have you producing polished, high-caliber, valuable report specifications in no time!

Uncover surprises, risks, and potentially serious bugs with exploratory testing. Rather than designing all tests in advance, explorers design and execute small, rapid experiments, using what they learned from the last little experiment to inform the next. Learn essential skills of a master explorer, including how to analyze software to discover key points of vulnerability, how to design experiments on the fly, how to hone your observation skills, and how to focus your efforts. Software is full of surprises. No matter how careful or skilled you are, when you create software it can behave differently than you intended. Exploratory testing mitigates those risks. Part 1 introduces the core, essential skills of a master explorer. You'll learn to craft charters to guide your exploration, to observe what's really happening (hint: it's harder than it sounds), to identify interesting variations, and to determine what expected behavior should be when exercising software in unexpected ways. Part 2 builds on that foundation. You'll learn how to explore by varying interactions, sequences, data, timing, and configurations. Along the way you'll see how to incorporate analysis techniques like state modeling, data modeling, and defining context diagrams into your explorer's arsenal. Part 3 brings the techniques back into the context of a software project. You'll apply the skills and techniques in a variety of contexts and integrate exploration into the development cycle from the very beginning. You can apply the techniques in this book to any kind of software. Whether you work on embedded systems, Web applications, desktop applications, APIs, or something else, you'll find this book contains a wealth of concrete and practical advice about exploring your software to discover its capabilities, limitations, and risks.

The purpose of this manual is to provide clear and helpful information for maintaining gravel roads. Very little technical help is available to small agencies that are responsible for managing these roads. Gravel road maintenance has traditionally been "more of an art than a science" and very few formal standards exist. This manual contains guidelines to help answer the questions that arise concerning gravel road maintenance such as: What is enough surface crown? What is too much? What causes corrugation? The information is as nontechnical as possible without sacrificing clear guidelines and instructions on how to do the job right.

First released in the Spring of 1999, How People Learn has been expanded to show how the theories and insights from the original book can translate into actions and practice, now making a real connection between classroom activities and learning behavior. This edition includes far-reaching suggestions for research that could increase the impact that classroom teaching has on actual learning. Like the original edition, this book offers exciting new research about the mind and the brain that provides answers to a number of compelling questions. When do infants begin to learn? How do experts learn and how is this different from non-experts? What can teachers and schools do-with curricula, classroom settings, and teaching

methods--to help children learn most effectively? New evidence from many branches of science has significantly added to our understanding of what it means to know, from the neural processes that occur during learning to the influence of culture on what people see and absorb. *How People Learn* examines these findings and their implications for what we teach, how we teach it, and how we assess what our children learn. The book uses exemplary teaching to illustrate how approaches based on what we now know result in in-depth learning. This new knowledge calls into question concepts and practices firmly entrenched in our current education system. Topics include: How learning actually changes the physical structure of the brain. How existing knowledge affects what people notice and how they learn. What the thought processes of experts tell us about how to teach. The amazing learning potential of infants. The relationship of classroom learning and everyday settings of community and workplace. Learning needs and opportunities for teachers. A realistic look at the role of technology in education.

Impact Mapping

Explore It!

A User's Guide

Dive Into the Portable Document Format

Fifty Quick Ideas to Improve Your User Stories

Effective awk Programming

A Consultant's Guide to the Secrets of Effective Functional Spec Writing Including Examples and a Downloadable Template

"This book covers the basics of traditional educational testing, measurement, and evaluation theory and methodology, as well as sociopolitical issues and trends influencing the future of that research and practice"--Publisher's description.

Specification by Example

Version Control with Subversion

The Art of Software Testing

With C and GNU Development Tools

How Google Runs Production Systems

Programming Embedded Systems

Brain, Mind, Experience, and School: Expanded Edition