Bridging The Communication Gap Specification By Example And Agile Acceptance Testing Gojko Adzic

Getting the books **Bridging The Communication Gap Specification By Example And Agile Acceptance Testing Gojko Adzic** now is not type of inspiring means. You could not unaided going gone book stock or library or borrowing from your friends to entre them. This is an unquestionably easy means to specifically get guide by on-line. This online pronouncement Bridging The Communication Gap Specification By Example And Agile Acceptance Testing Gojko Adzic can be one of the options to accompany you following having supplementary time.

It will not waste your time. tolerate me, the e-book will entirely freshen you extra event to read. Just invest little become old to way in this on-line publication **Bridging The Communication Gap Specification By Example And Agile Acceptance Testing Gojko Adzic** as well as evaluation them wherever you are now.

Advanced UFT 12 for Test Engineers Cookbook - Meir Bar-Tal 2014-11-28
This advanced cookbook is designed for software testers and engineers with previous automation experience and teaches UFT (QTP) developers advanced programming approaches. Knowledge of software testing and basic coding (with VBScript in particular) and familiarity with programming concepts are prerequisites.
Object Design Style Guide - Matthias Noback 2019-12-23
”Demystifies object-oriented programming, and lays out how to use it to design truly secure and performant applications.” —Charles Soetan, Plum.io Key Features Dozens of techniques for writing object-oriented code that’s easy to read, reuse, and maintain Write code that other programmers will instantly understand Design rules for constructing objects, changing and exposing state, and more Examples written in an instantly familiar pseudocode that’s easy to apply to Java, Python, C#, and any object-oriented language Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About The Book Well-written object-oriented code is easy to read, modify, and debug. Elevate your coding style by mastering the universal best practices for object design presented in this book. These clearly presented rules, which apply to any OO language, maximize the clarity and durability of your codebase and increase productivity for you and your team. In Object Design Style Guide, veteran developer Matthias Noback lays out design rules for constructing objects, defining methods, and much more. All examples use instantly familiar pseudocode, so you can follow along in the language you prefer. You’ll go case by case through important scenarios and challenges for object design and then walk through a simple web application that demonstrates how different types of objects can work together effectively. What You Will Learn Universal design rules for a wide range of objects Best practices for testing objects A catalog of common object types Changing and exposing state Test your object design skills with exercises This Book Is Written For For readers familiar with an object-oriented language and basic application architecture. About the Author Matthias Noback is a professional web developer with nearly two decades of experience. He runs his own web development, training, and consultancy company called “Noback’s Office.” Table of Contents: 1 ¦ Programming with objects: A primer 2 ¦ Creating services 3 ¦ Creating other objects 4 ¦ Manipulating objects 5 ¦ Using objects 6 ¦ Retrieving information 7 ¦ Performing tasks 8 ¦ Dividing responsibilities 9 ¦ Changing the behavior of services 10 ¦ A field guide to

objects 11 ¦ Epilogue

ATDD by Example - Markus Gärtner 2012

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.

The Book of R - Tilman M. Davies 2016-07-16

The Book of R is a comprehensive, beginner-friendly guide to R, the world’s most popular programming language for statistical analysis. Even if you have no programming experience and little more than a grounding in the basics of mathematics, you’ll find everything you need to begin using R effectively for statistical analysis. You’ll start with the basics, like how to handle data and write simple programs, before moving on to more advanced topics, like producing statistical summaries of your data and performing statistical tests and modeling. You’ll even learn how to create impressive data visualizations with R’s basic graphics tools and contributed packages, like ggplot2 and ggvis, as well as interactive 3D visualizations using the rgl package. Dozens of hands-on exercises (with downloadable solutions) take you from theory to practice, as you learn: –The fundamentals of programming in R, including how to write data frames, create functions, and use variables, statements, and loops –Statistical concepts like exploratory data analysis, probabilities, hypothesis tests, and regression modeling, and how to execute them in R –How to access R’s thousands of functions, libraries, and data sets –How to draw valid and useful conclusions from your data –How to create publication-quality graphics of your results Combining detailed explanations with real-world examples and exercises, this book will provide you with a solid understanding of both statistics and the depth of R’s functionality. Make The Book of R your doorway into the growing world of data analysis.

Fifty Quick Ideas to Improve Your User Stories - Gojko Adzic 2014-10-15

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.

Practices for Scaling Lean & Agile Development - Craig Larman 2010-01-26

Lean and Agile Development for Large-Scale Products: Key Practices for Sustainable Competitive Success Increasingly, large product-development organizations are turning to lean thinking, agile principles and practices, and large-scale Scrum to sustainably and quickly deliver value and innovation. Drawing on their long experience leading and guiding lean and agile adoptions for large, multisite, and offshore product development, internationally recognized consultant and best-selling author Craig Larman and former leader of the agile transformation at Nokia Networks Bas Vodde share the key action tools needed for success. Coverage includes Frameworks for large-scale Scrum for multihundred-person product groups Testing and building quality in Product management and the end of the “contract game” between business and R&D Envisioning a large release, and planning for multiteam development Low-quality legacy code: why it’s created, and how to stop it Continuous integration in a large multisite context Agile architecting Multisite or offshore development Contracts and outsourced development In a competitive environment that demands ever-faster cycle times and greater innovation, the practices inspired by lean thinking and agile principles are ever-more relevant. Practices for Scaling Lean & Agile Development will help people realize a lean enterprise—and deliver on the significant benefits of agility. In addition to the action tools in this text, see the companion book Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrumfor complementary foundation tools.

Rule-Governed Behavior - Steven C. Hayes 2012-12-06

Animal learning and human learning traditions have been distinguishable within psychology since the start of the discipline and are to this day. The human learning wing was interested in the development of psychological functions in human organisms and proceeded directly to their examination. The animal learning wing was not distinguished by a corresponding interest in animal behavior per se. Rather, the animal learners studied animal behavior in order to identify principles of behavior of relevance to humans as well as other organisms. The two traditions, in other words, did not differ so much on goals as on strategies. It is not by accident

that so many techniques of modem applied psychol ogy have emerged from the animal laboratory. That was one of the ultimate purposes of this work from the very beginning. The envisioned extension to humans was not just technological, however. Many animal researchers, B. F. Skinner most prominently among them, recognized that direct basic research with humans might ultimately be needed in certain areas but that it was wise first to build a strong foundation in the controlled environment of the animal laboratory. In a sense, animal learning was always in part a human research program in development.

Virtual Teams That Work - Cristina B. Gibson 2003-03-21

Virtual Teams That Work offers a much-needed, comprehensive guidebook for business leaders and managers who want to create the organizational conditions that will help virtual teams thrive. Each chapter in this important book focuses on best practices and includes case studies and illustrative examples from a wide variety of companies, including British Petroleum, Lucent Technologies, Ramtech, SoftCo, and Whirlpool Corporation. These real-life examples demonstrate how the principles identified in the book play out within virtual teams. Virtual Teams That Work shows how organizations can put in place the structure to help team members who speak different languages and have different cultural values develop effective ways of communicating when there is little opportunity for the members to meet face-to-face. The authors also reveal how organizations can implement performance management and reward systems that will motivate team members to cooperate across multiple boundaries. And they offer the information to determine which technologies best fit a variety of virtual-team tasks and the level of information technology support needed.

How to Reduce the Cost of Software Testing - Matthew Heusser 2018-09-03

Plenty of software testing books tell you how to test well; this one tells you how to do it while decreasing your testing budget. A series of essays written by some of the leading minds in software testing, How to Reduce the Cost of Software Testing provides tips, tactics, and techniques to help readers accelerate the testing process, improve the performance of the test teams, and lower costs. The distinguished team of contributors—that includes corporate test leaders, best paper authors, and keynote speakers from leading software testing conferences—supply concrete suggestions on how to find cost savings without sacrificing outcome. Detailing strategies that testers can immediately put to use to reduce costs, the book explains how to make testing nimble, how to remove bottlenecks in the testing process, and how to locate and track defects efficiently and effectively. Written in language accessible to non-technical executives, as well as those doing the testing, the book considers the latest advances in test automation, ideology, and technology. Rather than present the perspective of one or two experts in software testing, it supplies the wide-ranging perspectives of a team of experts to help ensure your team can deliver a completed test cycle in less time, with more confidence, and reduced costs.

The Cucumber Book - Matt Wynne 2017-02-17

Your customers want rock-solid, bug-free software that does exactly what they expect it to do. Yet they can't always articulate their ideas clearly enough for you to turn them into code. You need Cucumber: a testing, communication, and requirements tool-all rolled into one. All the code in this book is updated for Cucumber 2.4, Rails 5, and RSpec 3.5. Express your customers' wild ideas as a set of clear, executable specifications that everyone on the team can read. Feed those examples into Cucumber and let it guide your development. Build just the right code to keep your customers happy. You can use Cucumber to test almost any system or any platform. Get started by using the core features of Cucumber and working with Cucumber's Gherkin DSL to describe-in plain language-the behavior your customers want from the system. Then write Ruby code that interprets those plain-language specifications and checks them against your application. Next, consolidate the knowledge you've gained with a worked example, where you'll learn more advanced Cucumber techniques, test asynchronous systems, and test systems that use a database. Recipes highlight some of the most difficult and commonly seen situations the authors have helped teams solve. With these

patterns and techniques, test Ajax-heavy web applications with Capybara and Selenium, REST web services, Ruby on Rails applications, command-line applications, legacy applications, and more. Written by the creator of Cucumber and the co-founders of Cucumber Ltd., this authoritative guide will give you and your team all the knowledge you need to start using Cucumber with confidence. What You Need: Windows, Mac OS X (with XCode) or Linux, Ruby 1.9.2 and upwards, Cucumber 2.4, Rails 5, and RSpec 3.5

Emerging Innovations in Agile Software Development - Ghani, Imran 2016-01-26

Agile is a relatively recent methodology used in the development process of a project. Therefore, it is important to share new emerging knowledge with researchers and professionals interested in adopting an agile mindset. Emerging Innovations in Agile Software Development focuses on the use of agile methodologies to manage, design, develop, test and maintain software projects. Emphasizing research-based solutions for contemporary software development, this publication is designed for use by software developers, researchers, and graduate-level students in software engineering and project management programs.

Test-Driven Infrastructure with Chef - Stephen Nelson-Smith 2013-10-11

Since Test-Driven Infrastructure with Chef first appeared in mid-2011, infrastructure testing has begun to flourish in the web ops world. In this revised and expanded edition, author Stephen Nelson-Smith brings you up to date on this rapidly evolving discipline, including the philosophy driving it and a growing array of tools. You'll get a hands-on introduction to the Chef framework, and a recommended toolchain and workflow for developing your own test-driven production infrastructure. Several exercises and examples throughout the book help you gain experience with Chef and the entire infrastructure-testing ecosystem. Learn how this test-first approach provides increased security, code quality, and peace of mind. Explore the underpinning philosophy that infrastructure can and should be treated as code Become familiar with the MASCOT approach to test-driven infrastructure Understand the basics of test-driven and behavior-driven development for managing change Dive into Chef fundamentals by building an infrastructure with real examples Discover how Chef works with tools such as Virtualbox and Vagrant Get a deeper understanding of Chef by learning Ruby language basics Learn the tools and workflow necessary to conduct unit, integration, and acceptance tests

Improving Software Testing - Tim A. Majchrzak 2012-02-03

Software is continuously increasing in complexity. Paradigmatic shifts and new development frameworks make it easier to implement software – but not to test it. Software testing remains to be a topic with many open questions with regard to both technical low-level aspects and to the organizational embedding of testing. However, a desired level of software quality cannot be achieved by either choosing a technical procedure or by optimizing testing processes. In fact, it requires a holistic approach.This Brief summarizes the current knowledge of software testing and introduces three current research approaches. The base of knowledge is presented comprehensively in scope but concise in length; thereby the volume can be used as a reference. Research is highlighted from different points of view. Firstly, progress on developing a tool for automated test case generation (TCG) based on a program's structure is introduced. Secondly, results from a project with industry partners on testing best practices are highlighted. Thirdly, embedding testing into e-assessment of programming exercises is described.

Large-Scale Scrum - Craig Larman 2016-09-23

In Large-Scale Scrum , Craig Larman and Bas Vodde offer the most direct, concise, actionable guide to reaping the full benefits of agile in distributed, global enterprises. Larman and Vodde have distilled their immense experience helping geographically distributed development organizations move to agile. Going beyond their previous books, they offer today's fastest, most focused guidance: "brass tacks" advice and field-proven best practices for achieving value fast, and achieving even more value as you move forward. Targeted to enterprise project participants and stakeholders, Large-Scale Scrum offers straight-to-the-point insights for

scaling Scrum across the entire project lifecycle, from sprint planning to retrospective. Larman and Vodde help you: Implement proven Scrum frameworks for large-scale developments Scale requirements, planning, and product management Scale design and architecture Effectively manage defects and interruptions Integrate Scrum into multisite and offshore projects Choose the right adoption strategies and organizational designs This will be the go-to resource for enterprise stakeholders at all levels: everyone who wants to maximize the value of Scrum in large, complex projects.

Lean-Agile Acceptance Test-Driven-Development - Ken Pugh 2010-12-22

Within the framework of Acceptance Test-Driven-Development (ATDD), customers, developers, and testers collaborate to create acceptance tests that thoroughly describe how software should work from the customer's viewpoint. By tightening the links between customers and agile teams, ATDD can significantly improve both software quality and developer productivity. This is the first start-to-finish, real-world guide to ATDD for every agile project participant. Leading agile consultant Ken Pugh begins with a dialogue among a customer, developer, and tester, explaining the "what, why, where, when, and how" of ATDD and illuminating the experience of participating in it. Next, Pugh presents a practical, complete reference to each facet of ATDD, from creating simple tests to evaluating their results. He concludes with five diverse case studies, each identifying a realistic set of problems and challenges with proven solutions. Coverage includes • How to develop software with fully testable requirements • How to simplify and componentize tests and use them to identify missing logic • How to test user interfaces, service implementations, and other tricky elements of a software system • How to identify requirements that are best handled outside software • How to present test results, evaluate them, and use them to assess a project's overall progress • How to build acceptance tests that are mutually beneficial for development organizations and customers • How to scale ATDD to large projects

Enterprise Integration Patterns - Gregor Hohpe 2012-03-09

Enterprise Integration Patterns provides an invaluable catalog of sixty-five patterns, with real-world solutions that demonstrate the formidable of messaging and help you to design effective messaging solutions for your enterprise. The authors also include examples covering a variety of different integration technologies, such as JMS, MSMQ, TIBCO ActiveEnterprise, Microsoft BizTalk, SOAP, and XSL. A case study describing a bond trading system illustrates the patterns in practice, and the book offers a look at emerging standards, as well as insights into what the future of enterprise integration might hold. This book provides a consistent vocabulary and visual notation framework to describe large-scale integration solutions across many technologies. It also explores in detail the advantages and limitations of asynchronous messaging architectures. The authors present practical advice on designing code that connects an application to a messaging system, and provide extensive information to help you determine when to send a message, how to route it to the proper destination, and how to monitor the health of a messaging system. If you want to know how to manage, monitor, and maintain a messaging system once it is in use, get this book.

Knowledge Discovery, Knowledge Engineering and Knowledge Management - Ana Fred 2013-12-20

This book constitutes the thoroughly refereed proceedings of the 4th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K, held in Barcelona, Spain, in October 2012. The 29 best papers were carefully reviewed and selected from 347 submissions. The papers are organized in topical sections on knowledge discovery and information retrieval; knowledge engineering and ontology development; knowledge management and information sharing.

Impact Mapping - Gojko Adzic 2012-10-01

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

Beyond Requirements - Kent J. McDonald 2015-08-29

Satisfy Stakeholders by Solving the Right

Problems, in the Right Ways In Beyond Requirements , Kent J. McDonald shows how applying analysis techniques with an agile mindset can radically transform analysis from merely “gathering and documenting requirements” to an important activity teams use to build shared understanding. First, McDonald discusses the unique agile mindset, reviews the key principles underlying it, and shows how these principles link to effective analysis. Next, he puts these principles to work in four wide-ranging and thought-provoking case studies. Finally, he drills down on a full set of techniques for effective agile analysis, using examples to show how, why, and when they work. McDonald’s strategies will teach you how to understand stakeholders’ needs, identify the best solution for satisfying those needs, and build a shared understanding of your solution that persists throughout the product lifecycle. He also demonstrates how to iterate your analysis, taking advantage of what you learn throughout development, testing, and deployment so that you can continuously adapt, refine, and improve. Whether you’re an analysis practitioner or you perform analysis tasks as a developer, manager, or tester, McDonald’s techniques will help your team consistently find and deliver better solutions. Coverage includes Core concepts for analysis: needs/ solutions, outcome/output, discovery/delivery Adapting Lean Startup ideas for IT projects: customer delivery, build–measure–learn, and metrics Structuring decisions, recognizing differences between options and commitments, and overcoming cognitive biases Focusing on value: feature injection, minimum viable products, and minimum marketable features Understanding how analysis flows alongside your project’s lifecycle Analyzing users: mapping stakeholders, gauging commitment, and creating personas Understanding context: performing strategy (enterprise) analysis Clarifying needs: applying decision filters, assessing project opportunities, stating problems Investigating solutions: impact and story mapping, collaborative modeling, and acceptance criteria definition Kent J. McDonald uncovers better ways of delivering value. His experience includes work in business analysis, strategic planning, project management, and product development in the financial services, health insurance, performance marketing, human services, nonprofit, and automotive industries. He has a BS in industrial engineering from Iowa State University and an MBA from Kent State University. He is coauthor of Stand Back and Deliver: Accelerating Business Agility (Addison-Wesley, 2009).

Lean-agile Acceptance Test-driven Development - Kenneth Pugh 2011

How to scale ATDD to large projects --

Agile & Scrum Methodologies - Ajit Singh 2019-10-05

This ebook is licensed for your personal enjoyment only. This ebook may not be re-sold or given away to other people. If you would like to share this book with another person, please purchase an additional copy for each recipient. If you’re reading this book and did not purchase it, or it was not purchased for your use only, then please return to your favorite ebook retailer and purchase your own copy. Thank you for respecting the hard work of this author.

Cranked - Steve Fenton 2014-06-15

Cranked helps teams and organisations to effectively deliver software in a changeable or uncertain environment. This book will teach you all about the values, activities and practices that you need to know to delight your customers with your software product. With the techniques in this book you can: - Improve product quality - Release faster and with less errors - Focus on value - Deliver more features - Increase motivation and job satisfaction - Make your customers and end-users happy If you are already working in an agile or lean team, Cranked could accelerate you to the next level. If you are switching to agile or lean - Cranked will help you to avoid common problems in failed agile adoptions. Cranked can be used in any size of organisation to solve complex software development problems.

Executable Specifications with Scrum - Mario Cardinal 2013-07-11

Most books about specifications still assume that requirements can be known up front and won’t change much during your project. In today’s “real world,” however, you must specify and build software in the face of high and continuing uncertainty. Scrum and other agile methods have evolved to reflect this reality. Now, there’s a complete guide to specifying software in agile

environments when prerequisites are unclear, requirements are difficult to grasp, and anything about your project could change. Long-time agile coach and enterprise architect Mario Cardinal shows how to create executable specifications and use them to test software behavior against requirements. Cardinal shows how to trawl requirements incrementally, step-by-step, using a vision-centric and emergent iterative practice that is designed for agility. Writing for analysts, architects, developers, and managers, Cardinal makes a strong case for the iterative discovery of requirements. Then, he moves from theory to practice, fully explaining the technical mechanisms and empirical techniques you need to gain full value from executable specifications. You'll learn to connect specifications with software under construction, link requirements to architecture, and automate requirements verification within the Scrum framework. Above all, Cardinal will help you solve the paramount challenge of software development: not only to solve the problem right, but also to solve the right problem. You will learn how to • Establish more effective agile roles for analysts and architects • Integrate and simplify the best techniques from FIT, ATDD, and BDD • Identify "core certainties" on which your project team should rely to ensure requirements discovery • Manage uncertainty by discovering stakeholder desires through short feedback loops • Specify as you go while writing small chunks of requirements • Use storyboarding and paper prototyping to improve conversations with stakeholders • Express stakeholder desires that are requirements with user stories • Refine your user stories, and plan more effective Scrum sprints • Confirm user stories by scripting behaviors with scenarios • Transform scenarios into automated tests that easily confirm your software's expected behavior as designs emerge and specifications evolve • Ensure higher-quality software by specifying nonfunctional requirements

Agile Processes, in Software Engineering, and Extreme Programming - Helen Sharp 2016-05-14

This book contains the refereed proceedings of the 17th International Conference on Agile Software Development, XP 2016, held in Edinburgh, UK, in May 2016. While agile development has already become mainstream in industry, this field is still constantly evolving and continues to spur an enormous interest both in industry and academia. To this end, the XP conference attracts a large number of software practitioners and researchers, providing a rare opportunity for interaction between the two communities. The 14 full papers accepted for XP 2016 were selected from 42 submissions. Additionally, 11 experience reports (from 25 submissions) 5 empirical studies (out of 12 submitted) and 5 doctoral papers (from 6 papers submitted) were selected, and in each case the authors were shepherded by an experienced researcher. Generally, all of the submitted papers went through a rigorous peer-review process.

System Engineering Analysis, Design, and Development - Charles S. Wasson 2015-11-16

Praise for the first edition: "This excellent text will be useful to everysystem engineer (SE) regardless of the domain. It covers ALLrelevant SE material and does so in a very clear, methodicalfashion. The breadth and depth of the author's presentation ofSE principles and practices is outstanding." –Philip Allen This textbook presents a comprehensive, step-by-step guide toSystem Engineering analysis, design, and development via anintegrated set of concepts, principles, practices, andmethodologies. The methods presented in this text apply to any typeof human system -- small, medium, and large organizational systemsand system development projects delivering engineered systems orservices across multiple business sectors such as medical,transportation, financial, educational, governmental, aerospace anddefense, utilities, political, and charity, among others. Provides a common focal point for "bridgingthe gap" between and unifying System Users, System Acquirers,multi-discipline System Engineering, and Project, Functional, andExecutive Management education, knowledge, and decision-making fordeveloping systems, products, or services Each chapter provides definitions of key terms,guiding principles, examples, author's notes, real-worldexamples, and exercises, which highlight and reinforce key SE&Dconcepts and practices Addresses concepts employed in Model-BasedSystems

Engineering (MBSE), Model-Driven Design (MDD), UnifiedModeling Language (UMLTM) / Systems Modeling Language(SysMLTM), and Agile/Spiral/V-Model Development such asuser needs, stories, and use cases analysis; specificationdevelopment; system architecture development; User-Centric SystemDesign (UCSD); interface definition & control; systemintegration & test; and Verification & Validation(V&V) Highlights/introduces a new 21st Century SystemsEngineering & Development (SE&D) paradigm that is easy tounderstand and implement. Provides practices that are critical stagingpoints for technical decision making such as Technical StrategyDevelopment; Life Cycle requirements; Phases, Modes, & States;SE Process; Requirements Derivation; System ArchitectureDevelopment, User-Centric System Design (UCSD); EngineeringStandards, Coordinate Systems, and Conventions; et al. Thoroughly illustrated, with end-of-chapter exercises andnumerous case studies and examples, Systems EngineeringAnalysis, Design, and Development, Second Edition is a primarytextbook for multi-discipline, engineering, system analysis, andproject management undergraduate/graduate level students and avaluable reference for professionals.

The Art of Action - Stephen Bungay 2011-02-16

What do you want me to do? This question is the enduring management issue, a perennial problem that Stephen Bungay shows has an old solution that is counter-intuitive and yet common sense. The Art of Action is a thought-provoking and fresh look at how managers can turn planning into execution, and execution into results. Drawing on his experience as a consultant, senior manager and a highly respected military historian, Stephen Bungay takes a close look at the nineteenth-century Prussian Army, which built its agility on the initiative of its highly empowered junior officers, to show business leaders how they can build more effective, productive organizations. Based on a theoretical framework which has been tested in practice over 150 years, Bungay shows how the approach known as 'mission command' has been applied in businesses as diverse as pharmaceuticals and F1 racing today. The Art of Action is scholarly but engaging, rigorous but pragmatic, and shows how common sense can sometimes be surprising.

Agile Processes in Software Engineering and Extreme Programming - Juan Garbajosa 2018-05-16

This open access book constitutes the proceedings of the 19th International Conference on Agile Software Development, XP 2018, held in Porto, Portugal, in May 2018. XP is the premier agile software development conference combining research and practice, and XP 2018 provided a playful and informal environment to learn and trigger discussions around its main theme – make, inspect, adapt. The 21 papers presented in this volume were carefully reviewed and selected from 62 submissions. They were organized in topical sections named: agile requirements; agile testing; agile transformation; scaling agile; human-centric agile; and continuous experimentation.

Discovery - Gáspár Nagy 2018-02-07

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.

Verification, Validation, and Testing of Engineered Systems - Avner Engel 2010-06-15

Systems' Verification Validation and Testing (VVT) are carried out throughout systems' lifetimes. Notably, quality-cost expended on performing VVT activities and correcting system defects consumes about half of the overall engineering cost. Verification, Validation and

Testing of Engineered Systems provides a comprehensive compendium of VVT activities and corresponding VVT methods for implementation throughout the entire lifecycle of an engineered system. In addition, the book strives to alleviate the fundamental testing conundrum, namely: What should be tested? How should one test? When should one test? And, when should one stop testing? In other words, how should one select a VVT strategy and how it be optimized? The book is organized in three parts: The first part provides introductory material about systems and VVT concepts. This part presents a comprehensive explanation of the role of VVT in the process of engineered systems (Chapter-1). The second part describes 40 systems' development VVT activities (Chapter-2) and 27 systems' post-development activities (Chapter-3). Corresponding to these activities, this part also describes 17 non-testing systems' VVT methods (Chapter-4) and 33 testing systems' methods (Chapter-5). The third part of the book describes ways to model systems' quality cost, time and risk (Chapter-6), as well as ways to acquire quality data and optimize the VVT strategy in the face of funding, time and other resource limitations as well as different business objectives (Chapter-7). Finally, this part describes the methodology used to validate the quality model along with a case study describing a system's quality improvements (Chapter-8). Fundamentally, this book is written with two categories of audience in mind. The first category is composed of VVT practitioners, including Systems, Test, Production and Maintenance engineers as well as first and second line managers. The second category is composed of students and faculties of Systems, Electrical, Aerospace, Mechanical and Industrial Engineering schools. This book may be fully covered in two to three graduate level semesters; although parts of the book may be covered in one semester. University instructors will most likely use the book to provide engineering students with knowledge about VVT, as well as to give students an introduction to formal modeling and optimization of VVT strategy.

Designing Embedded Hardware - John Catsoulis 2002

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.

Real Scrum and More - Alex Manfield 2015-08-27

Scrum and other Agile methodologies are discussed in this book. Scrum can help managing Projects with tight schedules, low tolerance to bugs and the difficulty of securing capital. Scrum and other Agile methodologies provides faster and more reliable ways to get from idea to market with the least amount of overhead. Alex works as Agile Coach for an IT group in London. He started his first project as Scrum Master in India in 2005. He started as developer and specialized into management roles. Alex is PMP and PSM, and is an Agile evangelist. This book can help the beginner to get started and the advanced professional to see more from real Projects. Several Open Source &

Commercial tools are described in this book.

More Agile Testing - Janet Gregory 2014-09-30
Janet Gregory and Lisa Crispin pioneered the agile testing discipline with their previous work, Agile Testing. Now, in More Agile Testing, they reflect on all they've learned since. They address crucial emerging issues, share evolved agile practices, and cover key issues agile testers have asked to learn more about. Packed with new examples from real teams, this insightful guide offers detailed information about adapting agile testing for your environment; learning from experience and continually improving your test processes; scaling agile testing across teams; and overcoming the pitfalls of automated testing. You'll find brand-new coverage of agile testing for the enterprise, distributed teams, mobile/embedded systems, regulated environments, data warehouse/BI systems, and DevOps practices. You'll come away understanding • How to clarify testing activities within the team • Ways to collaborate with business experts to identify valuable features and deliver the right capabilities • How to design automated tests for superior reliability and easier maintenance • How agile team members can improve and expand their testing skills • How to plan "just enough," balancing small increments with larger feature sets and the entire system • How to use testing to identify and mitigate risks associated with your current agile processes and to prevent defects • How to address challenges within your product or organizational context • How to perform exploratory testing using "personas" and "tours" • Exploratory testing approaches that engage the whole team, using test charters with session- and thread-based techniques • How to bring new agile testers up to speed quickly–without overwhelming them The eBook edition of More Agile Testing also is available as part of a two-eBook collection, The Agile Testing Collection (9780134190624).

Succeeding with Agile - Mike Cohn 2010
Provides recommendations and case studies to help with the implementation of Scrum.

Bridging the Communication Gap - Gojko Adzic 2009
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.

Test Driven .NET Development with FitNesse - Gojko Adzic 2008-02-01
Test Driven .NET Development with FitNesse takes you on a journey through the wonderful world of FitNesse, a great web-based tool for software acceptance testing. FitNesse enables software developers and business people to build a shared understanding of the domain and helps produce software that is genuinely fit for purpose.

The Darien Gap - Martin Mitchinson 2008
When Balboa marched through Dariens jungles to cross the narrow isthmus in 1513, he was the first European to sight the Pacific from its eastern shores. For the next four centuries, pirates, gold miners, rebels, and political schemers all gravitated to Darien. Scotland failed miserably in its attempt to establish a colony. An American Navy expedition wandered lost in its jungle for two months with seven men dying, and countries fought to control the regions traffic and trade. Even today, Darien is best known as a roadless backwater, home to native communities, Colombian guerrillas, and the descendants of black slaves and Spanish colonists.

Specification by Example - Gojko Adzic 2011
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.

The Leadership Gap - Lolly Daskal 2017-05-30
Do people see you as the kind of leader you want to be? Are your strongest leadership qualities getting in the way of your greatness? After decades of advising and inspiring some of the

most eminent chief executives in the world, Lolly Daskal has uncovered a startling pattern: within each leader are powerful abilities that are also hidden impediments to greatness. She's witnessed many highly driven, overachieving leaders rise to prominence fueled by well-honed skill sets, only to falter when the shadow sides of the same skills emerge. Now Daskal reveals her proven system, which leaders at any level can apply to dramatically improve their results. It begins with identifying your distinctive leadership archetype and recognizing its shadow: ■ The Rebel, driven by confidence, becomes the Imposter, plagued by self-doubt. ■ The Explorer, fueled by intuition, becomes the Exploiter, master of manipulation. ■ The Truth Teller, who embraces candor, becomes the Deceiver, who creates suspicion. ■ The Hero, embodying courage, becomes the Bystander, an outright coward. ■ The Inventor, brimming with integrity, becomes the Destroyer, who is morally corrupt. ■ The Navigator, trusts and is trusted, becomes the Fixer, endlessly arrogant. ■ The Knight, for whom loyalty is everything, becomes the Mercenary, who is perpetually self-serving. Using psychology, philosophy, and her own experience, Daskal offers a breakthrough perspective on leadership. She'll take you inside some of the most cloistered boardrooms, let you in on deeply personal conversations with industry leaders, and introduce you to luminaries who've changed the world. Her insights will help you rethink everything you know to become the leader you truly want to be.

The Agile Testing Collection - Janet Gregory 2015-06-22

A Comprehensive Collection of Agile Testing Best Practices: Two Definitive Guides from Leading Pioneers Janet Gregory and Lisa Crispin haven't just pioneered agile testing, they have also written two of the field's most valuable guidebooks. Now, you can get both guides in one indispensable eBook collection: today's must-have resource for all agile testers, teams, managers, and customers. Combining comprehensive best practices and wisdom contained in these two titles, The Agile Testing Collection will help you adapt agile testing to your environment, systematically improve your skills and processes, and strengthen engagement across your entire development team. The first title, Agile Testing: A Practical Guide for Testers and Agile Teams, defines the agile testing discipline and roles, and helps you choose, organize, and use the tools that will help you the most. Writing from the tester's viewpoint, Gregory and Crispin chronicle an entire agile software development iteration, and identify and explain seven key success factors of agile testing. The second title, More Agile Testing: Learning Journeys for the Whole Team, addresses crucial emerging issues, shares evolved practices, and covers key issues that delivery teams want to learn more about. It offers powerful new insights into continuous improvement, scaling agile testing across teams and the enterprise, overcoming pitfalls of automation, testing in regulated environments, integrating DevOps practices, and testing mobile/embedded and business intelligence systems. The Agile Testing Collection will help you do all this and much more. Customize agile testing processes to your needs, and successfully transition to them Organize agile teams, clarify roles, hire new testers, and quickly bring them up to speed Engage testers in agile development, and help agile team members improve their testing skills Use tests and collaborate with business experts to plan features and guide development Design automated tests for superior reliability and easier maintenance Plan "just enough," balancing small increments with larger feature sets and the entire system Test to identify and mitigate risks, and prevent future defects Perform exploratory testing using personas, tours, and test charters with session- and thread-based techniques Help testers, developers, and operations experts collaborate on shortening feedback cycles with continuous integration and delivery Both guides in this collection are thoroughly grounded in the authors' extensive experience, and supported by examples from actual projects. Now, with both books integrated into a single, easily searchable, and cross-linked eBook, you can learn from their experience even more easily.

Specification by Example - Gojko Adzic 2011-06-02

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