

specification by example agile

specification by example agile is a collaborative approach to defining requirements and ensuring software quality through concrete examples rather than abstract descriptions. This methodology aligns closely with agile principles, emphasizing shared understanding, continuous feedback, and iterative development. By using real-world scenarios and examples, specification by example agile helps teams clarify requirements, reduce misunderstandings, and create executable specifications that guide development and testing efforts. This article explores the concept in depth, covering its origins, core practices, benefits, and practical implementation within agile projects. Additionally, it discusses common challenges and best practices for maximizing the effectiveness of specification by example agile in complex software environments. The following sections offer a comprehensive overview designed for professionals seeking to enhance their agile development processes through this proven approach.

- Understanding Specification by Example Agile
- Core Practices of Specification by Example
- Benefits of Specification by Example in Agile
- Implementing Specification by Example Agile
- Common Challenges and Solutions

Understanding Specification by Example Agile

Specification by example agile is a method of capturing and defining software requirements through concrete, illustrative examples that demonstrate how a system should behave under various conditions. Unlike traditional requirements documentation that often relies on abstract statements, this approach uses real-life scenarios to bridge the communication gap between stakeholders, developers, and testers. The essence of specification by example agile lies in creating a shared understanding that drives development and testing efforts in tandem.

Origins and Evolution

The concept originated as a response to the shortcomings of conventional requirements gathering practices, especially within agile environments where adaptability and collaboration are paramount. It draws inspiration from behavior-driven development (BDD) and acceptance test-driven development

(ATDD), focusing on executable specifications that serve as living documentation throughout the software lifecycle. Specification by example agile has evolved into a core practice for agile teams seeking to improve clarity, reduce rework, and deliver software that truly meets user needs.

Key Terminology

Understanding the terminology is essential to grasping specification by example agile. Some key terms include:

- **Example:** A concrete scenario illustrating expected system behavior.
- **Specification:** A detailed description of functionality, often expressed through examples.
- **Executable Specification:** A specification that can be run as a test to validate system behavior.
- **Collaboration:** The joint effort of cross-functional team members to define and refine examples.

Core Practices of Specification by Example

Specification by example agile relies on a set of core practices that guide teams in effectively capturing, refining, and using examples as specifications. These practices foster collaboration, clarity, and continuous validation of requirements.

Collaborative Specification Workshops

One of the foundational practices is holding workshops that bring together business stakeholders, developers, testers, and other relevant parties. During these sessions, participants discuss requirements and collaboratively define concrete examples that illustrate expected behavior. This practice ensures shared understanding and aligns priorities early in the development process.

Living Documentation

Examples developed during specification workshops are maintained as living documentation that evolves alongside the software. This documentation is not static; it reflects the current state of requirements and serves as a reference for development, testing, and future enhancements. Living documentation supports transparency and traceability within agile teams.

Automated Acceptance Testing

Examples created through specification by example agile are often automated as acceptance tests. These automated tests verify that the software behaves as expected, providing immediate feedback to the team. Automation integrates seamlessly with continuous integration pipelines, enabling frequent validation and reducing the risk of defects.

Frequent Refinement and Feedback

Regularly revisiting and refining examples based on feedback from development and testing activities help keep specifications relevant and accurate. This continuous cycle of feedback and refinement is a hallmark of agile methodologies and is critical to the success of specification by example agile.

Benefits of Specification by Example in Agile

Adopting specification by example agile offers numerous advantages that contribute to higher quality software and more efficient development processes. These benefits address common pain points associated with traditional requirements management.

Improved Communication and Collaboration

By focusing on concrete examples, specification by example agile facilitates clear communication among stakeholders with diverse backgrounds. This shared language reduces misunderstandings and aligns expectations, which is essential for agile teams operating in fast-paced environments.

Early and Continuous Validation

Executable specifications enable teams to validate requirements early and continuously throughout the development cycle. This early detection of issues minimizes costly rework and helps ensure that the delivered software meets business needs.

Reduced Ambiguity and Rework

Using specific examples to define requirements eliminates vague or ambiguous statements that can lead to errors. Clear, testable examples serve as unambiguous acceptance criteria, reducing the likelihood of rework caused by misinterpretation.

Enhanced Quality and Confidence

Automated acceptance tests derived from specification by example agile act as a safety net that guards against regressions and defects. This contributes to higher software quality and gives teams greater confidence in their releases.

Documentation That Evolves With the Product

Unlike traditional documentation that quickly becomes outdated, living documentation maintained through specification by example agile remains relevant and useful throughout the product lifecycle, supporting ongoing development and maintenance.

Implementing Specification by Example Agile

Successful implementation of specification by example agile requires careful planning, cultural alignment, and the adoption of appropriate tools and techniques. This section outlines practical steps to integrate this methodology into agile workflows.

Establishing Cross-Functional Teams

Forming teams that include business analysts, developers, testers, and product owners ensures diverse perspectives are considered when creating examples. This cross-functional collaboration is critical to capturing comprehensive and accurate specifications.

Facilitating Specification Workshops

Regularly scheduled workshops provide a structured environment for discussing requirements and defining examples. Facilitators play a key role in guiding discussions, resolving conflicts, and ensuring productive outcomes.

Choosing the Right Tools

Tools that support behavior-driven development frameworks, such as Cucumber or SpecFlow, enable the creation and automation of executable specifications. Selecting tools integrated with the team's existing development and testing infrastructure is essential for smooth adoption.

Integrating With Agile Practices

Specification by example agile should be embedded within existing agile

ceremonies and workflows, such as sprint planning, backlog refinement, and continuous integration. This integration ensures that examples drive development and testing activities consistently.

Training and Coaching

Providing training and coaching helps team members understand the principles and practices of specification by example agile. Continuous learning supports the development of skills necessary to write effective examples and collaborate efficiently.

Common Challenges and Solutions

While specification by example agile offers substantial benefits, teams may encounter challenges during adoption and execution. Identifying these obstacles and applying appropriate solutions can enhance effectiveness.

Challenge: Resistance to Change

Teams accustomed to traditional requirements documentation may resist adopting specification by example agile. Overcoming this requires clear communication of benefits, pilot projects to demonstrate value, and leadership support.

Challenge: Insufficient Collaboration

Lack of engagement from key stakeholders can undermine the collaborative nature of specification by example agile. Encouraging participation through facilitated workshops and demonstrating the impact of collaboration can mitigate this issue.

Challenge: Maintaining Living Documentation

Failing to keep examples up to date can lead to obsolete specifications and reduced trust. Establishing ownership, integrating updates into agile workflows, and automating synchronization help maintain living documentation.

Challenge: Tooling and Automation Complexity

Choosing and configuring tools for executable specifications can be complex and time-consuming. Investing in training, selecting appropriate frameworks, and gradually expanding automation efforts ease this transition.

Challenge: Writing Effective Examples

Poorly written examples may be ambiguous or incomplete. Providing guidelines, templates, and peer reviews ensures that examples are clear, relevant, and testable.

Best Practices to Overcome Challenges

- Engage leadership to champion specification by example agile adoption.
- Foster a collaborative culture through regular communication and team-building activities.
- Invest in continuous training and knowledge sharing.
- Start small with pilot projects to demonstrate quick wins.
- Automate acceptance tests incrementally to build confidence.

Frequently Asked Questions

What is Specification by Example in Agile?

Specification by Example is a collaborative approach to defining requirements and business-oriented functional tests for software products, where examples are used to illustrate and validate expected behavior, ensuring clear communication among stakeholders.

How does Specification by Example improve Agile development?

Specification by Example improves Agile development by fostering shared understanding through concrete examples, reducing ambiguity, enabling early validation of requirements, and facilitating automated testing, which leads to higher quality and faster feedback cycles.

What are the key benefits of Specification by Example?

Key benefits include enhanced collaboration between business and technical teams, improved requirement clarity, reduced defects, better alignment of development with business needs, and support for automated acceptance testing.

How do you create effective examples in Specification by Example?

Effective examples should be realistic, focused on business value, easy to understand by all stakeholders, cover various scenarios including edge cases, and be precise enough to support automated testing.

What tools support Specification by Example in Agile projects?

Popular tools include Cucumber, SpecFlow, JBehave, Concordion, and FitNesse, which enable writing executable specifications using examples and integrating them with automated testing frameworks.

How does Specification by Example relate to Behavior-Driven Development (BDD)?

Specification by Example is a foundational practice of BDD; both emphasize using concrete examples to define behavior, improve communication, and drive automated acceptance testing in Agile environments.

Who should be involved in the Specification by Example process?

Cross-functional teams including product owners, business analysts, developers, testers, and other stakeholders should collaborate to define, review, and automate specifications by example.

What challenges might teams face when adopting Specification by Example?

Challenges include initial resistance to change, difficulty in writing good examples, maintaining up-to-date specifications, integrating tools with existing workflows, and ensuring ongoing collaboration among diverse team members.

How can Specification by Example help in reducing defects in Agile projects?

By defining clear, testable examples upfront that reflect business requirements, Specification by Example helps catch misunderstandings early, guides development, supports automated acceptance testing, and thus reduces defects and rework.

Additional Resources

1. *Specification by Example: How Successful Teams Deliver the Right Software*

This foundational book by Gojko Adzic introduces the concept of Specification by Example (SBE) and explains how teams can use realistic examples to clarify requirements and guide development. It covers practical techniques for collaborative specification, automated testing, and living documentation. The book is essential for agile teams looking to improve communication and deliver software that meets business needs.

2. *Bridging the Communication Gap: Specification by Example and Agile Acceptance Testing*

Author Gojko Adzic focuses on closing the gap between business and technical teams through Specification by Example. The book emphasizes the importance of collaboration in defining acceptance criteria and demonstrates how automated acceptance tests can serve as living documentation. It provides actionable advice for improving requirements clarity and reducing misunderstandings in agile projects.

3. *Agile Testing: A Practical Guide for Testers and Agile Teams*

By Lisa Crispin and Janet Gregory, this book discusses various agile testing practices, including Specification by Example, to ensure high-quality software delivery. It highlights the role of testers within agile teams and how they can collaborate on defining and automating acceptance tests. The book is a comprehensive resource for embedding testing activities seamlessly into agile workflows.

4. *Discover to Deliver: Agile Product Planning and Analysis*

This book by Ellen Gottesdiener and Mary Gorman covers collaborative techniques for agile product discovery and defining requirements. It includes detailed discussions on Specification by Example as a way to capture and validate requirements with concrete examples. The authors provide practical guidance for continuous stakeholder engagement and iterative delivery.

5. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*

Dean Leffingwell explores agile requirements practices, including the use of examples to specify acceptance criteria and guide development. The book integrates lean thinking with agile requirements management and discusses how Specification by Example fits into larger-scale agile frameworks. It is particularly useful for teams scaling agile practices across an organization.

6. *BDD in Action: Behavior-Driven Development for Agile Teams*

John Ferguson Smart's book delves into Behavior-Driven Development (BDD), a practice closely related to Specification by Example. It explains how to write executable specifications using examples that describe system behavior from the user's perspective. The book offers practical advice for implementing BDD within agile teams to improve collaboration and automate acceptance testing.

7. *Specification by Example Patterns*

This concise book by Gojko Adzic presents a collection of patterns and best practices for applying Specification by Example effectively. It covers topics such as example mapping, living documentation, and dealing with ambiguous requirements. The patterns help teams adopt SBE techniques to improve communication, reduce defects, and accelerate feedback cycles.

8. Lean-Agile Acceptance Test-Driven Development: Better Software through Collaboration

Ken Pugh focuses on Acceptance Test-Driven Development (ATDD), a methodology that complements Specification by Example by using acceptance tests as the basis for development. The book emphasizes collaboration between customers, developers, and testers to define clear requirements and validate functionality early. It provides step-by-step guidance to implement ATDD in agile projects.

9. From User Stories to Acceptance Tests: A Practical Guide for Agile Teams

This practical guide explains how to transform user stories into detailed acceptance tests using Specification by Example principles. It helps agile teams create clear, testable requirements through collaboration and example-driven discussions. The book includes real-world examples and techniques to improve requirements quality and support continuous delivery.

Specification By Example Agile

Find other PDF articles:

<https://explore.gcts.edu/textbooks-suggest-004/Book?dataid=JSe65-1104&title=teaching-textbooks-math-grade-9.pdf>

specification by example agile: 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

specification by example agile: The Agile Testing Collection Janet Gregory, Lisa Crispin, 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 agile: 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.

specification by example agile: System Engineering Analysis, Design, and Development Charles S. Wasson, 2015-11-16 Praise for the first edition: This excellent text will be useful to every system engineer (SE) regardless of the domain. It covers ALL relevant SE material and does so in a very clear, methodical fashion. The breadth and depth of the author's presentation of SE principles and practices is outstanding. —Philip Allen This textbook presents a comprehensive, step-by-step guide to System Engineering analysis, design, and development via an integrated set of concepts, principles, practices, and methodologies. The methods presented in this text apply to any type of human system -- small, medium, and large organizational systems and system development projects delivering engineered systems or services across multiple business sectors such as medical,

transportation, financial, educational, governmental, aerospace and defense, utilities, political, and charity, among others. Provides a common focal point for “bridging the gap” between and unifying System Users, System Acquirers, multi-discipline System Engineering, and Project, Functional, and Executive Management education, knowledge, and decision-making for developing systems, products, or services Each chapter provides definitions of key terms, guiding principles, examples, author’s notes, real-world examples, and exercises, which highlight and reinforce key SE&D concepts and practices Addresses concepts employed in Model-Based Systems Engineering (MBSE), Model-Driven Design (MDD), Unified Modeling Language (UMLTM) / Systems Modeling Language (SysMLTM), and Agile/Spiral/V-Model Development such as user needs, stories, and use cases analysis; specification development; system architecture development; User-Centric System Design (UCSD); interface definition & control; system integration & test; and Verification & Validation (V&V) Highlights/introduces a new 21st Century Systems Engineering & Development (SE&D) paradigm that is easy to understand and implement. Provides practices that are critical staging points for technical decision making such as Technical Strategy Development; Life Cycle requirements; Phases, Modes, & States; SE Process; Requirements Derivation; System Architecture Development, User-Centric System Design (UCSD); Engineering Standards, Coordinate Systems, and Conventions; et al. Thoroughly illustrated, with end-of-chapter exercises and numerous case studies and examples, Systems Engineering Analysis, Design, and Development, Second Edition is a primary textbook for multi-discipline, engineering, system analysis, and project management undergraduate/graduate level students and a valuable reference for professionals.

specification by example agile: More Agile Testing Janet Gregory, Lisa Crispin, 2015 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).

specification by example agile: The Art of Agile Development James Shore, Shane Warden, 2021-10-12 Most companies developing software employ something they call Agile. But there's widespread misunderstanding of what Agile is and how to use it. If you want to improve your software development team's agility, this comprehensive guidebook's clear, concrete, and detailed guidance explains what to do and why, and when to make trade-offs. In this thorough update of the classic Agile how-to guide, James Shore provides no-nonsense advice on Agile adoption, planning, development, delivery, and management taken from over two decades of Agile experience. He brings the latest ideas from Extreme Programming, Scrum, Lean, DevOps, and more into a cohesive whole. Learn how to successfully bring Agile development to your team and organization--or discover why Agile might not be for you. This book explains how to: Improve agility: create the conditions necessary for Agile to succeed and scale in your organization Focus on value: work as a team,

understand priorities, provide visibility, and improve continuously Deliver software reliably: share ownership, decrease development costs, evolve designs, and deploy continuously Optimize value: take ownership of product plans, budgets, and experiments--and produce market-leading software

specification by example agile: 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

specification by example agile: Specs2 Essentials: Practical Testing and Specification for Scala William E Clark, 2025-08-24 Specs2 Essentials: Practical Testing and Specification for Scala is a concise, practical guide to specification-driven testing in the Scala ecosystem. It opens with core concepts and the theoretical foundations of executable specifications, illuminates how Scala's expressive type system and Specs2's modular design enable clear, maintainable tests, and situates Specs2 among other popular testing libraries through real-world adoption examples and continuous-integration strategies relevant to both greenfield and established projects. The book delivers hands-on guidance for everyday use and advanced scenarios alike: seamless build and CI integration, dynamic configuration, and patterns for writing elegant, reusable specifications. Coverage includes strategies for scaling suites (parallel execution and test partitioning), cross-platform targets (Scala.js and Scala Native), multiple specification styles, property-based and data-driven testing, expressive matchers, and robust handling of concurrency and asynchronous behavior—plus techniques to produce living documentation that communicates with technical and nontechnical stakeholders. Beyond practical recipes, Specs2 Essentials explores isolation and mocking patterns (native and third-party), resource and integration-test management, and extensibility through custom plugins, test runners, and metadata-driven execution. It concludes with instrumentation for traceability, maintainability practices to keep test suites healthy, and governance approaches for compliance—equipping engineers, architects, and quality leads to build reliable Scala systems using modern, executable specification techniques.

specification by example agile: A Modern Enterprise Architecture Approach Dr Mehmet Yildiz, 2019-10-07 The revised version of this book to provide essential guidance, compelling ideas, and unique ways to Enterprise Architects so that they can successfully perform complex enterprise modernisation initiatives transforming from chaos to coherence. This is not an ordinary theory book describing Enterprise Architecture in detail. There are myriad of books on the market and in libraries discussing details of enterprise architecture. My aim here is to highlight success factors and reflect lessons learnt from the field within enterprise modernisation and transformation context. As a practising Senior Enterprise Architect, myself, I read hundreds of those books and articles to learn different views. They have been valuable to me to establish my foundations in the earlier phase of my profession. However, what is missing now is a concise guidance book showing Enterprise Architects the novel approaches, insights from the real-life experience and experimentations, and pointing out the differentiating technologies for enterprise modernisation. If only there were such a guide when I started engaging in modernisation and transformation programs. The biggest lesson learned is the business outcome of the enterprise modernisation. What genuinely matters for

business is the return on investment of the enterprise architecture and its monetising capabilities. The rest is the theory because nowadays sponsoring executives, due to economic climate, have no interest, attention, or tolerance for non-profitable ventures. I am sorry for disappointing some idealistic Enterprise Architects, but with due respect, it is the reality, and we cannot change it. This book deals with reality rather than theoretical perfection. Anyone against this view on this climate must be coming from another planet. In this concise, uncluttered and easy-to-read book, I attempt to show the significant pain points and valuable considerations for enterprise modernisation using a structured approach and a simple narration especially considering my audience from non-English speaking backgrounds. The architectural rigour is still essential. We cannot compromise the rigour aiming to the quality of products and services as a target outcome. However, there must be a delicate balance among architectural rigour, business value, and speed to the market. I applied this pragmatic approach to multiple substantial transformation initiatives and complex modernisations programs. The key point is using an incrementally progressing iterative approach to every aspect of modernisation initiatives, including people, processes, tools, and technologies as a whole. Starting with a high-level view of enterprise architecture to set the context, I provided a dozen of distinct chapters to point out and elaborate on the factors which can make a real difference in dealing with complexity and producing excellent modernisation initiatives. As eminent leaders, Enterprise Architects are the critical talents who can undertake this massive mission using their people and technology skills, in addition to many critical attributes such as calm and composed approach. Let's keep in mind that as Enterprise Architects, we are architects, not firefighters! I have full confidence that this book can provide valuable insights and some 'aha' moments for talented architects like yourself to tackle this enormous mission of turning chaos to coherence.

specification by example agile: *Writing Great Specifications* Kamil Nicieja, 2017-10-25

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 Nicieja 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

specification by example agile: Kanban in Action Joakim Sunden, Marcus Hammarberg, 2014-02-18 Summary Kanban in Action is a down-to-earth, no-frills, get-to-know-the-ropes introduction to kanban. It's based on the real-world experience and observations from two kanban coaches who have introduced this process to dozens of teams. You'll learn the principles of why

kanban works, as well as nitty-gritty details like how to use different color stickies on a kanban board to help you organize and track your work items. About the Book Too much work and too little time? If this is daily life for your team, you need kanban, a lean knowledge-management method designed to involve all team members in continuous improvement of your process. Kanban in Action is a practical introduction to kanban. Written by two kanban coaches who have taught the method to dozens of teams, the book covers techniques for planning and forecasting, establishing meaningful metrics, visualizing queues and bottlenecks, and constructing and using a kanban board. Written for all members of the development team, including leaders, coders, and business stakeholders. No experience with kanban is required. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. What's Inside How to focus on work in process and finish faster Examples of successful implementations How team members can make informed decisions About the Authors Marcus Hammarberg is a kanban coach and software developer with experience in BDD, TDD, Specification by Example, Scrum, and XP. Joakim Sundén is an agile coach at Spotify who cofounded the first kanban user groups in Europe. Table of Contents PART 1 LEARNING KANBAN Team Kanbaneros gets started PART 2 UNDERSTANDING KANBAN Kanban principles Visualizing your work Work items Work in process Limiting work in process Managing flow PART 3 ADVANCED KANBAN Classes of service Planning and estimating Process improvement Using metrics to guide improvements Kanban pitfalls Teaching kanban through games

specification by example agile: 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.

specification by example agile: Software Methodologies Capers Jones, 2017-07-06 This comprehensive reference uses a formal and standard evaluation technique to show the strengths and weakness of more than 60 software development methodologies such as agile, DevOps, RUP, Waterfall, TSP, XP and many more. Each methodology is applied to an application of 1000 function points using the Java language. Each methodology produces a characteristic set of results for development schedules, productivity, costs, and quality. The intent of the book is to show readers the optimum kinds of methodologies for the projects they are concerned with and to warn them about counter indications and possible harm from unsuitable methodologies.

specification by example agile: Testable JavaScript Mark Trostler, 2013-02-15 That's right: JavaScript code is testable, despite the lack of unit test standards for this language. With this book, you'll discover how to devise excellent tests and real-world testing plans for your existing JavaScript code. Developers starting out on a new project will greatly benefit from these guidelines. Testing a language that runs within many different implementations and environments is inherently difficult—and since most JavaScript environments are UI-based, automated testing becomes even more difficult. This guide shows developers and their managers how to overcome these problems to test JavaScript code efficiently. Topics include: Which types of Javascript testing are most effective When and why should you run tests on Javascript code Code patterns and anti-patterns that promote and inhibit Javascript testing Integrating unit tests with your development and build environments (automated test running and reporting) Code coverage: how to get it, interpret it, and automatically generate it

specification by example agile: Human-Computer Interaction. Theory, Design, Development and Practice Masaaki Kurosu, 2016-07-04 The 3-volume set LNCS 9731, 9732, and 9733 constitutes the refereed proceedings of the 18th International Conference on Human-Computer Interaction, HCI 2016, held in Toronto, ON, Canada, in July 2016. The total of 1287 papers and 186 posters

presented at the HCII 2016 conferences and were carefully reviewed and selected from 4354 submissions. The papers thoroughly cover the entire field of Human-Computer Interaction, addressing major advances in knowledge and effective use of computers in a variety of application areas. The volumes constituting the full 27-volume set of the conference proceedings.

specification by example agile: Software Reliability Techniques for Real-World

Applications Roger K. Youree, 2023-01-04 Authoritative resource providing step-by-step guidance for producing reliable software to be tailored for specific projects Software Reliability Techniques for Real-World Applications is a practical, up to date, go-to source that can be referenced repeatedly to efficiently prevent software defects, find and correct defects if they occur, and create a higher level of confidence in software products. From content development to software support and maintenance, the author creates a depiction of each phase in a project such as design and coding, operation and maintenance, management, product production, and concept development and describes the activities and products needed for each. Software Reliability Techniques for Real-World Applications introduces clear ways to understand each process of software reliability and explains how it can be managed effectively and reliably. The book is supported by a plethora of detailed examples and systematic approaches, covering analogies between hardware and software reliability to ensure a clear understanding. Overall, this book helps readers create a higher level of confidence in software products. In Software Reliability Techniques for Real-World Applications, readers will find specific information on: Defects, including where defects enter the project system, effects, detection, and causes of defects, and how to handle defects Project phases, including concept development and planning, requirements and interfaces, design and coding, and integration, verification, and validation Roadmap and practical guidelines, including at the start of a project, as a member of an organization, and how to handle troubled projects Techniques, including an introduction to techniques in general, plus techniques by organization (systems engineering, software, and reliability engineering) Software Reliability Techniques for Real-World Applications is a practical text on software reliability, providing over sixty-five different techniques and step-by-step guidance for producing reliable software. It is an essential and complete resource on the subject for software developers, software maintainers, and producers of software.

specification by example agile: Software Requirements & Specifications Mr. Rohit

Manglik, 2023-05-23 Focuses on requirement engineering processes, use case modeling, and creating specifications that guide software design and validation.

specification by example agile: End-User Computing: Concepts, Methodologies, Tools, and Applications Clarke, Steve, 2008-02-28 Covers the important concepts, methodologies, technologies, applications, social issues, and emerging trends in this field. Provides researchers, managers, and other professionals with the knowledge and tools they need to properly understand the role of end-user computing in the modern organization.

specification by example agile: Perspectives on the Future of Software Engineering

Jürgen Münch, Klaus Schmid, 2013-06-13 The dependence on quality software in all areas of life is what makes software engineering a key discipline for today's society. Thus, over the last few decades it has been increasingly recognized that it is particularly important to demonstrate the value of software engineering methods in real-world environments, a task which is the focus of empirical software engineering. One of the leading protagonists of this discipline worldwide is Prof. Dr. Dr. h.c. Dieter Rombach, who dedicated his entire career to empirical software engineering. For his many important contributions to the field he has received numerous awards and recognitions, including the U.S. National Science Foundation's Presidential Young Investigator Award and the Cross of the Order of Merit of the Federal Republic of Germany. He is a Fellow of both the ACM and the IEEE Computer Society. This book, published in honor of his 60th birthday, is dedicated to Dieter Rombach and his contributions to software engineering in general, as well as to empirical software engineering in particular. This book presents invited contributions from a number of the most internationally renowned software engineering researchers like Victor Basili, Barry Boehm, Manfred Broy, Carlo Ghezzi, Michael Jackson, Leon Osterweil, and, of course, by Dieter Rombach

himself. Several key experts from the Fraunhofer IESE, the institute founded and led by Dieter Rombach, also contributed to the book. The contributions summarize some of the most important trends in software engineering today and outline a vision for the future of the field. The book is structured into three main parts. The first part focuses on the classical foundations of software engineering, such as notations, architecture, and processes, while the second addresses empirical software engineering in particular as the core field of Dieter Rombach's contributions. Finally, the third part discusses a broad vision for the future of software engineering.

specification by example agile: Software Engineering for Modern Web Applications: Methodologies and Technologies Brandon, Daniel M., 2008-06-30 This book presents current, effective software engineering methods for the design and development of modern Web-based applications--Provided by publisher.

Related to specification by example agile

Interview and Book Review: Specification by Example (InfoQ13y) Unlock the full InfoQ experience by logging in! Stay updated with your favorite authors and topics, engage with content, and download exclusive resources. Senyo Simpson discusses how Rust's core

Interview and Book Review: Specification by Example (InfoQ13y) Unlock the full InfoQ experience by logging in! Stay updated with your favorite authors and topics, engage with content, and download exclusive resources. Senyo Simpson discusses how Rust's core

How CIOs in the UAE lead by example in agile transformation, with the skillset to prove it (CIO3y) With support from the top of government, and by tapping into a deep reservoir of talent to swiftly implement innovations, CIOs in the UAE serve as a global model of success as digital transformation

How CIOs in the UAE lead by example in agile transformation, with the skillset to prove it (CIO3y) With support from the top of government, and by tapping into a deep reservoir of talent to swiftly implement innovations, CIOs in the UAE serve as a global model of success as digital transformation

Back to Home: <https://explore.gcts.edu>