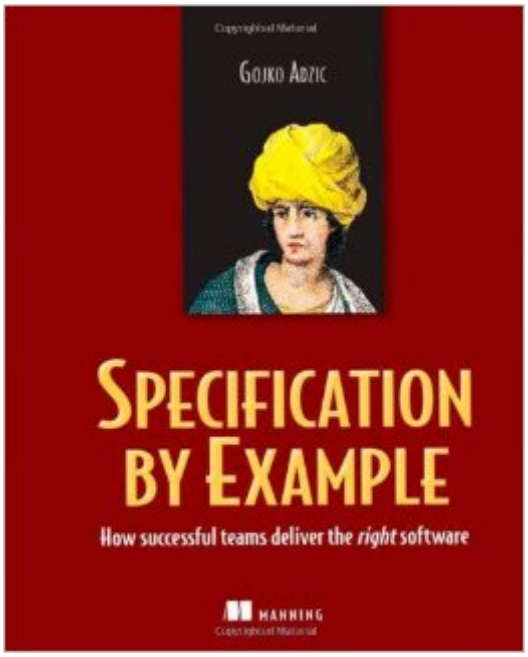
The book was found

Specification By Example: How Successful Teams Deliver The Right Software



PDF
Adobe
DOWNLOAD EBOOK

Synopsis

SummarySpecification 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 TechnologySpecification 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 BookThis 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 InsideCommon process patternsHow to avoid bad practicesFitting SBE in your process50+ case studies==========â ‹=======Table of ContentsPart 1 Getting startedKey benefitsKey process patternsLiving documentationInitiating the changesPart 2 Key process patternsDeriving scope from goalsSpecifying collaborativelyIllustrating using examplesRefining the specificationAutomating validation without changing specificationsValidating frequentlyEvolving a documentation systemPart 3 Case studiesuSwitchRainStorIowa Student LoanSabre Airline SolutionsePlan ServicesSongkickConcluding thoughts

Book Information

Paperback: 296 pages

Publisher: Manning Publications; 1 edition (June 6, 2011)

Language: English

ISBN-10: 1617290084

ISBN-13: 978-1617290084

Product Dimensions: 7.4 x 0.6 x 9.2 inches

Shipping Weight: 1.1 pounds (View shipping rates and policies)

Average Customer Review: 4.5 out of 5 starsÂ Â See all reviewsÂ (35 customer reviews)

Best Sellers Rank: #200,282 in Books (See Top 100 in Books) #52 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Tools #87 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Testing #118 inÂ Books > Computers & Technology > Computer Science > Systems Analysis & Design

Customer Reviews

One one hand, the book is not deep, reflective and well argued enough to be a timeless classic. One the other hand, it lacks concrete examples, steps and instructions to be a timely and actionable cookbook. It hangs in a midair between realms of inspirational and practical, touching on both and delivering on none.I was really looking forward to read this book after hearing an interview with the author on Devnology podcast. It pains me to admit that reading it was not a time well spent. How could the author call his approach "Specification by Example" and offer no end-to-end examples that could be studied, evaluated and replicated? Please comment with a page number(s) for such examples if you disagree, and I will be more than happy to admit my blindness.

Specification by Example is Gojko’s third book on this subject. The first book, Fitness.net, was very technical and tool oriented. The second book, Bridging the Communication Gap, was a lot more coordination oriented. Now his third book, this one, he describes practices that teams he studied have used. From that perspective, this book is the follow-up of Bridging and might go a little fast if you are totally unfamiliar with the subject.The book is divided in three parts. The first part is mainly introduction where Gojko describes the benefits and the key practices that will be described in this book. The second part is the actual description of the key practices and the third part are different case studies about different teams in different companies that have adopted specification by example.The key practices that are introduced in part one and described in part 2 are:- Deriving scope from goals- Specifying collaboratively- Illustrating using example- Refining the specification- Automating without changing the specification- Validating frequently- Evolving a documentation systemDeriving scope from goals discusses how customers main concert is not the software but solving a problem and developers shouldn’t just expect to get the requirements from the customer but work together with them to help them to solve their problem in the best way. Specifying collaboratively covers how the customer and the teams will cooperatively define the specifications that the team will be implementing later. Illustrating using examples explains how these specifications can be described best by moving from abstract requirements to concrete examples. Refining the specification then takes the essence out of the requirements and describes them in the

clearest possible way. After that, the specification can be automated without changing the specification and this chapter gives tips on how to do that. When the specifications are automated, you want to run them frequently which is described in the validate frequently chapter. Evolving a documentation system describes how the specifications become the documentation of what the system does. They stay in-sync with the system because they are continuously executed.The third part described a couple of case studies of companies that implemented specification by example. I really loved these case studies and they were written very well.I've read both of Gojko's earlier books and had high expectations for this book. I was not disappointed, it is an excellent follow-up and will be my standard book reference on Specification by Example (or A-TDD as it is also called). The book is not perfect though. As times I felt there was too much focus on documentation and too little on collaboration. Still, I'd rate this book five stars and recommend everyone in an Agile development team to read this and practice specification by example.

The book was well organized and timely. Most teams swerve to the extremes of the test automation path... this book advocates striking a pragmatic balance between spotty automated test coverage and a plethora of maintenance-magnet technical tests. SBE is shown as a treasure map.. and then each checkpoint is elaborated in a chapter. Lot of real world knowledge collated in this book...Key take-aways:* Don't let customers dictate solutions, instead challenge and extract scope/solutions via collaboration.* Don't make test automation your end-goal. Move on to living documentation (although I have no clue on how to convince teams of the benefits).* Keep specs readable by business users.* Adapt your test suites to the current reality.. Fast feedback is key even for acceptance tests.Nitpicks: Could have been a shorter book. I realized I don't like reading about case studies... maybe others would like it. I strafed over Part III. and at 50$ a pop, the price is a bit steep.

When I started to read this book my impression was rather negative - looked rather as yet another piece of marketing blah-blah-blah. But when I reached the book core (chapters 2 - 10) I changed my mind completely. Yes, it's true that the book is oversaturated with success stories to which 7 of 16 chapters are devoted - but anyway it is one of the best books on a requirement collection and maintenance I ever saw.It promotes a set of very important principles (listed in the chapter 2) that, strictly speaking, are not Specification-by-Example specific or even new - most of them are known from 70-s or early 80-s - but anyway are way too often overlooked or forgotten. To name a few of the (the most important as for me):* Deriving Scope from Goals (a specification should answer not "how?" or even "what?", but "why?" and "what for?").* Specification Refinement (specification

should contain all necessary detail but nothing more and should be expressed on an appropriate level of abstraction).* Specify Collaboratively - customers, business analyst, developers and testers should participate in the specification creation.The above mentioned principles are a must for any successful software project - a project (save the most trivial one) seriously violating them can succeed by chance only.If add to them two Specification-by-Example specific principles - Executable Specification (specification expressed as acceptance tests written not in the technical but in the business language) and Living Documentation (documentation consisting of or generated from automated acceptance tests) you may imagine which benefit Specification-by-Example may bring to your development.The book considers a usability of Specification-by-Example in different scenarios and in differently organized teams (based on the real-life experience) and provide a well-grounded advices what to do - and what refrain from - based on the conditions in which your teams operates.All this is on the bright side.On the dark side are:* Already mentioned oversaturation with success stories - yes they play as important role as background and illustrations, but may safely be shortened 2-fold at least, as for me.* The book describes advantages but is almost silent about inherent dangers and drawbacks of the proposed approach. To be honest, it avoids them not completely; there are a few words on some of them here and there - but not enough as for me. And the main dangers - fragmented, incoherent and non-uniform solution and an exponential growth of complexity in the behavioral tests - are not even mentioned explicitly (yes, there are talks about "an appropriate level of abstraction" and "key examples" - but it is not enough).* The book overestimates a "less rework" effect provided by the approach - as with any specification developed and implemented piecewise there are high chances to miss on early development stages a requirement critical for internal structure of the system inevitably causing a massive rework of the system core. Moreover one of the success stories contains the following passage "From the first day of development, the Talia team used Specification by Example and built up a living documentation system. After a year of development, they had to rewrite the core of the virtual agent engine from scratch." There is no explanation why this "rework from scratch" becomes necessary - but there are good chances that it was for the reasons explained above.Smaller things that I really like:* "Building the product right and building the right product are two very different things. We need to do both to succeed."* "Instead of a technical feature specification, we should ask for a high-level example how a feature would be useful."* "I generally don’t agree with the categorization of requirements into functional and non-functional groups, but that is probably a topic for another book." - BTW, I am likely wrong qualifying this point of view (which I myself advocate from early 90-s) as a "smaller thing".* "Scripts are not specifications" - perfectly said, scripts inevitably specify

"how" not "what for".And smaller things which I rather dislike:* Maintenance costs of the automated acceptance are underestimated - along with possible bugs in fixtures.* "It never happens" syndrome is not dealt with (I mean business customers that tend to specify a happy-path only and for any border cases cut discussion short claiming "it never happens" - needless to say that the question is not "if" but "when" it happens).* There is a statement in the Introduction/ This book has no source code and does not explain any tools section: "Once you get the communication and collaboration right, a tool might help to make it go smoother." - which is only partially true, as without an appropriate tool it is virtually impossible to obtain an executable specification and a living documentation, 2 major benefits of the Specification-by-Example approach.* Quite ambiguous/imprecise statements:o Chapter 2, Evolving a documentation system section states about a living documentation "It is as reliable as the code, but much easier to read and understand." -almost true, as this documentation is almost as reliable as code (subject to misinterpretations/bugs in fixtures, parts of/paths through of the code not covered with automated acceptance tests etc.).Despite all small (and not so small) problems mentioned above the book is really great!

Download to continue reading...

Specification by Example: How Successful Teams Deliver the Right Software Nathan Wallace's Delphi 3 Example Book (Programmer's Example Series) 42 Rules for Building a High-Velocity Inside Sales Team: Actionable Guide to Creating Inside Sales Teams that Deliver Quantum Results HBR's 10 Must Reads on Teams (with featured article “The Discipline of Teams,” by Jon R. Katzenbach and Douglas K. Smith) Consumer Reports Life Insurance Handbook: How to Buy the Right Policy from the Right Company at the Right Price Big Data in Practice: How 45 Successful Companies Used Big Data Analytics to Deliver Extraordinary Results The Agile Samurai: How Agile Masters Deliver Great Software The Agile Samurai: How Agile Masters Deliver Great Software (Pragmatic Programmers) Software Engineering Classics: Software Project Survival Guide/ Debugging the Development Process/ Dynamics of Software Development (Programming/General) Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection: Obfuscation, Watermarking, and Tamperproofing for Software Protection The Interior Design Reference & Specification Book: Everything Interior Designers Need to Know Every Day The Industrial Design Reference & Specification Book: Everything Industrial Designers Need to Know Every Day Study Guide for Fundamentals of Engineering (FE) Electrical and Computer CBT Exam: Practice over 400 solved problems based on NCEESÃ Â® FE CBT Specification Version 9.4 The Architecture Reference & Specification Book: Everything Architects Need to Know Every Day Empirical Dynamic Asset Pricing: Model Specification and Econometric Assessment Spend

Analysis and Specification Development Using Failure Interpretation Notes to a Software Team Leader: Growing Self Organizing Teams Agile Product Management: Product Owner: 27 Tips To Manage Your Product And Work With Scrum Teams (scrum, scrum master, agile development, agile software development) The College Solution: A Guide for Everyone Looking for the Right School at the Right Price (2nd Edition) Effective Data Visualization: The Right Chart for the Right Data

Dmca