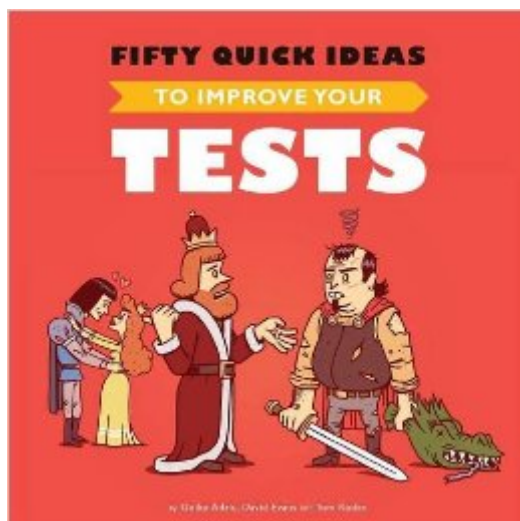


The book was found

# Fifty Quick Ideas To Improve Your Tests



## Synopsis

This book is for cross-functional teams working in an iterative delivery environment, planning with user stories and testing frequently changing software under tough time pressure. This book will help you test your software better, easier and faster. Many of these ideas also help teams engage their business stakeholders better in defining key expectations and improve the quality of their software products.

## Book Information

Paperback: 124 pages

Publisher: Neuri Consulting LLP (May 15, 2015)

Language: English

ISBN-10: 0993088112

ISBN-13: 978-0993088117

Product Dimensions: 8.5 x 0.3 x 8.5 inches

Shipping Weight: 12 ounces (View shipping rates and policies)

Average Customer Review: 4.7 out of 5 stars Â Â See all reviews Â (27 customer reviews)

Best Sellers Rank: #170,024 in Books (See Top 100 in Books) #56 in Â Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Testing #80 in Â Books > Computers & Technology > Computer Science > Systems Analysis & Design #415 in Â Books > Computers & Technology > Programming > Web Programming

## Customer Reviews

Theyâ™ve done it again. Gojko Adzic, David Evans and, in this book, Tom Roden has written another 50 quick ideas book. And this one is equally good as the previous book on user stories. If not even better. This is my review after reading the book in the worst possible manner. Iâ™ll tell you why. But even doing so I got so much out of this book and my tool belt expanded significantly. I really like the approach of these short, focus, one-topic books, starting with Gojoks book on impact mapping. They donâ™t promise to be deep dives and total coverage but rather to give you ideas (wellâ | thatâ™s in the title even), be challenged and investigate further. In this book, on testing, they have divided the ideas into 4 groups, brushing on different aspects of testing:- Generating test ideas- Designing good checks- Improving testability- Managing large test suites One of the things that struck me is how far (agile) testing have progress during my relative short period interested in the field. This is a very sober and concrete look at the new breed of testers that want to be part in design, that takes failed tests as an opportunity to learn. We have sections on measuring test half

times (how often do test change) in order to focus our testing efforts, thereâ€™s suggestions for how to involve and inform business users directly in creation of key examples etc. This is not your fathers testing and I like it! I have a confession to make: Iâ€™m not really into testing. Iâ€™m a developer and very fascinated by agile testing but the early parts of this book touch more on organizations of test efforts and exploratory testing planning etc. Thatâ€™s not my thing really. I read those parts faster.

The first thing I noticed when I opened the book to the Table of Contents is the great organization starting with generating testing ideas, which is different than the second section on designing good checks. Because these are separate sections, it delineates the difference between the two concepts. Generating testing ideas is about exploring what you need or want to get a shared understanding by the whole team. I particularly like the "Emotional Heuristics" â€“ I find that business people can relate to these ideas very well. Another idea that appealed to me was "Documenting Trust Boundaries" for dependencies. So many thoughts I have used in the past have been given a name which will help me to describe to others â€“ for example: "Donâ€™t Let the Pen be the Bottleneck" during a collaborative workshop is something Iâ€™ve been saying in my training for years, just not as succinctly. The designing good checks section is really about extracting great examples from the business users, and there are many useful tips to help. A couple of my favourites are using the "Given, When, Then" format correctly and "Donâ€™t Automate Manual Tests". The third section â€“ Improving Testability â€“ is what you really want to get your programmers reading. There are some basic things that can help get your application testable â€“ not only for automation, but also for exploratory testing. The last section is on Managing Large Test Suites, and this section might be the most important of all for me. I get asked on a regular basis about this subject, so now I have some great ideas to share and a place to refer teams to for more thoughts. There are too many good concepts for me to call out one or two.

[Download to continue reading...](#)

Managing Your Family's High-Tech Habits: (From Video-Games to the Dark Side of the Web) Keep Your Love On: Connection Communication And Boundaries The Smarter Screen: Surprising Ways to Influence and Improve Online Behavior A Lifelong Love: How to Have Lasting Intimacy, Friendship, and Purpose in Your Marriage Fear and Faith: Finding the Peace Your Heart Craves Grieving the Child I Never Knew: A Devotional for Comfort in the Loss of Your Unborn or Newly Born Child Do Your Bit to Be Physically Fit! (Healthy Habits for a Lifetime) Husband After God: Drawing Closer To God And Your Wife Quietening Your Heart: 30-Day Prayer Journal - Love Edition

Bonding with Your Child through Boundaries Be Safe on Your Bike Quieting Your Heart for the Holidays: 30-Day Prayer Journal MySQL Explained: Your Step-by-Step Guide Tinkletown: Your Favorite Place to 'Go' Why Kids Make You Fat: â and How to Get Your Body Back Kingdom Woman: Embracing Your Purpose, Power, and Possibilities Whatever the Cost: Facing Your Fears, Dying to Your Dreams, and Living Powerfully PHP: MYSQL 100 Tests, Answers & Explanations, Pass Final Exam, Job Interview Exam, Engineer Certification Exam, Examination, PHP programming, PHP in easy steps: A Beginner's Guide If You Change Your Words It Will Transform Your Life Let God Fight Your Battles: Being Peaceful in the Storm

[Dmca](#)