

How To Write A User Guide For Software

Eventually, you will unquestionably discover a extra experience and exploit by spending more cash. nevertheless when? do you believe that you require to acquire those every needs past having significantly cash? Why don't you attempt to get something basic in the beginning? That's something that will lead you to comprehend even more around the globe, experience, some places, when history, amusement, and a lot more!

It is your no question own become old to perform reviewing habit. in the middle of guides you could enjoy now is how to write a user guide for software below.

How to Write a Book: 13 Steps From a Bestselling Author **How To Write User Stories** Agile User Stories **How To Write A Book In A Weekend: Serve Humanity By Writing A Book** | Chandler Bolt | TEDxYoungstown **How to Write Good User Stories Using [3 Key] Components My Secret Book Writing Formula [Free Template]** | Brian Tracy **How To Write A Book In Google Docs** How To Write User Stories, Epics, \u0026 Personas - Dev Life **7 SECRETS TO WRITING A MUST-READ SELF-HELP BOOK** **How To Write A Book For Beginners** How To Outline A Book: Step-by-Step Book Outlining Instructions to Write a Better Book **Faster Creative Writing advice and tips from Stephen King** **How To Sell Publish a Book Publishing My 1st Children's Book (Ingramspark/Creatspace/Preoreto)** **How Much Money Does My SELF-PUBLISHED Book Earn?**

I wrote a book when I was 13. It sucked.

User Stories vs Use Cases**How to write Good User Stories (Part 2): INVEST** **How to do User Story Mapping** How to Self-Publish Your First Book: Step-by-step tutorial for beginners How to Write User Stories How To Write \u0026 Publish A Book Pt. 1 | Dr. Myles Munroe **HARSH WRITING ADVICE!** (mostly for newer writers) How to Write a Book Introduction: A Formula for More Sales How to Write a Children's Book: 8 EASY STEPS! Book Writing 101! How to Write A Book Introduction on how to write User Stories How to Write a Book: 10 Simple Steps to Self Publishing How I Write Books (A published author's imperfect process) How To Write A User Step 2: Create your user document Avoid jargon and highly technical terms except when absolutely necessary (and then be sure to define them). Avoid acronyms unless you explain what they mean. Use a commonly recognized writing style such as AP or Chicago whenever possible. If your organization has a ...

How to Build the Best User Documentation (New Guide ...

Creating the Username 1. Know that your username represents you. Your username is going to be the first thing people see when they interact... 2. Create different usernames for different services. Different places online may call for different styles of usernames. 3. Stay anonymous. Avoid using any ...

How to Create a Username: 10 Steps (with Pictures) - wikiHow

All user stories are unique and they should be complemented by story maps, diagrams, storyboards, and mockups, but below are a few best practices that can help you write an effective user story: Know Your User: Define and understand your user persona (s). Include All Stakeholders: Be sure to include ...

How to Write a Good User Story | SmartSheet

What are the steps to write great Agile User Stories? Step 1: Think of the " Who " This is the first and, maybe, the most fundamental step. Before writing a User Story you... Step 2: Think of the " What " Now we have a few groups of end users. The next step we do is define what functionality each....

How to Write a Good User Story: with Examples & Templates

How to Write a User Manual (That ' s Easy to Follow) Step 1: Identify the Audience. Understanding who your audience is can be half the battle when writing a user manual. Identifying your audience helps... Step 2: Define the Purpose of the Manual. Step 3: Identify the How-To Steps. Step 4: Formalize ...

How to Write a User Manual (That ' s Easy to Follow)

Including Essential Information 1. Organize the manual logically. The user manual should proceed in a way that the user will find most beneficial. 2. Include necessary warnings. The general warnings or cautionary information should provide information about potential... 3. Describe the device. Your ...

How to Create a User Manual (with Pictures) - wikiHow

Write directly to the user Start each step with a verb where possible. This helps make your instructions clear and concise, and ensures you use an active voice, for example: Log in to Agresso using your username and password.

Creating a 'How to' Guide - University of Bath

Then, we had an option to re-write the user story in to two User Stories - as an " Android Mobile App user " and " iOS Mobile App user ". The Product Owner prioritized the " iOS Mobile App user " over the " Android Mobile App user " since that was a User Segment with even more business value.

How to Write Good User Stories? User Story Examples ...

When writing a manual, you need to put on a " user ' s hat " and think like a user. You should have a good understanding of your users so you can understand the information they need to know, their background, and their knowledge of the product. Once you think like a user, you can write content that the users need to know.

Five Tips for Writing a User Manual - Online-learning.com

Writing a good epic and user story is the most basic and the most important task at hand when you enter the role of Product Management. Hence I am going to get right to it and give you some real tips and examples of how to write epics and user stories — best case scenarios.

How to write Epics and User stories — Best practice. | by ...

So, write User Stories on your Product Backlog — the tasks can wait till you subscribe a story to a Sprint. #6 Bump up your story mapping skills and focus on the MVP You have your user stories all listed in front of you, often as post-its scattered across a white board.

User Stories, 7 Tips to Write User Stories with Example ...

The first step in writing a good user manual is to get the actual writing process as far away from the engineers as possible. The software developer knows more than anybody what makes the software work, but that doesn't mean the developer should write the guide. On the contrary, it is a distinct disadvantage.

How to Write a User Manual for Software | Blfdraft

User stories typically follow a simple template that captures the user, and the goal that the user has, in a simple and non-technical format. As a < type of user/role >, I want < some goal > so that < some reason/benefit >. We also use a handy acronym, INVEST, to remember the best-practices of writing good user stories. A good user story should be:

The Business Analyst ' s Guide to Writing User Stories

A user story helps agile software development teams capture simplified, high-level descriptions of a user ' s requirements written from that end user ' s perspective. A user story is not a contextless feature, written in " dev " speak.

How to Write Good User Stories in Agile Software ...

Another version of writing a user story. As a [particular user], I want to [be able to perform/do something] so that [I get some form of value or benefit]. There are several templates floating on the internet to help you write a user story. But at its core, if you want to write good user stories, you need to define 3 things:

How To Write Good User Stories? With Examples & Templates

How to write user stories. Now that we ' ve listed some reasons why you should write user stories, here ' s how to actually write them. I.N.V.E.S.T. The I.N.V.E.S.T. guideline to writing user stories is almost universally accepted as the standard to work by. The acronym was made popular by Bill Wake ' s original article from 2003. Our ...

How to Write Great Agile User Stories - Sprinly

The details piece of an Agile user story spells out how particular functionality will work. Using the example of platform for location doctors, let ' s take this user story: As a user, I can create an account. Photo credit: Health Grades. We can then write out the following details: i. User clicks on account creation option. ii.

How to Write A Painless User Story - UXPin

How to write a user story What to include. Your user stories should include enough information for your product manager to decide how important the story is. They should always include:

WHAT IS THIS BOOK ABOUT? This Book Is About the " Card " (User Story: Card, Criteria, Conversation) User Stories are a great method for expressing stakeholder requirements, whether your projects follow an Agile, Iterative, or a Waterfall methodology. They are the basis for developers to deliver a suitable information technology (IT) app or application. Well-structured user stories express a single action to achieve a specific goal from the perspective of a single role. When writing user stories, stakeholders knowledgeable about the role should focus on the business result that the IT solution will enable while leaving technology decisions up to the developers. Good user stories are relevant to the project, unambiguous, and understandable to knowledge peers. The best user stories also contain crucial non-functional (quality) requirements, which are the best weapon in the war against unsatisfactory performance in IT solutions. This book presents two common user story structures to help you ensure that your user stories have all the required components and that they express the true business need as succinctly as possible. It offers five simple rules to ensure that your user stories are the best that they can be. That, in turn, will reduce the amount of time needed in user story elaboration and discussion with the development team. This book targets business professionals who are involved with an IT project. Product Owners in charge of managing a backlog, or Business Analysts working with an Agile team. Author ' s Note The term " User Story " is a relative new addition to our language and its definition is evolving. In today ' s parlance, a complete User Story has three primary components, namely the " Card ", the " Conversation ", and the " Criteria ". Different roles are responsible for creating each component. The " Card " expresses a business need. A representative of the business community is responsible for expressing the business need. Historically (and for practical reasons) the " Card " is the User Story from the perspective of the business community. Since we wrote this book specifically to address that audience, we use the term " User Story " in that context throughout. The " Conversation " is an ongoing discussion between a developer responsible for creating software that meets the business need and the domain expert(s) who defined it (e.g. the original author of the " Card "). The developer initiates the " Conversation " with the domain expert(s) to define the " Criteria " and any additional information the developer needs to create the application. There is much to be written about both the " Conversation " and the " Criteria ", but neither component is dealt with in any detail in this publication. A well-written User Story (" Card ") can drastically reduce the time needed for the " Conversation ". It reduces misinterpretations, misunderstandings, and false starts, thereby paving the way for faster delivery of working software. We chose to limit the content of this publication to the " User Story " as understood by the business community to keep the book focused and address the widest possible audience. WHO WILL BENEFIT FROM READING THIS BOOK? How organizations develop and deliver working software has changed significantly in recent years. Because the change was greatest in the developer community, many books and courses justifiably target that group. There is, however, an overlooked group of people essential to the development of software-as-an-asset that have been neglected. Many distinct roles or job titles in the business community perform business needs analysis for digital solutions. They include: - Product Owners - Business Analysts - Requirements Engineers - Test Developers - Business- and Customer-side Team Members - Agile Team Members - Subject Matter Experts (SME) - Project Leaders and Managers - Systems Analysts and Designers - AND " anyone wearing the business analysis hat ", meaning anyone responsible for defining a future IT solution TOM AND ANGELA ' S (the authors) STORY Like all good IT stories, theirs started on a project many years ago. Tom was the super techie, Angela the super SME. They fought their way through the 3-year development of a new policy maintenance system for an insurance company. They vehemently disagreed on many aspects, but in the process discovered a fundamental truth about IT projects. The business community (Angela) should decide on the business needs while the technical team ' s (Tom) ' s job was to make the technology deliver what the business needed. Talk about a revolutionary ideal! All that was left was learning how to communicate with each other without bloodshed to make the project a resounding success. Mission accomplished. They decided this epiphany was so important that the world needed to know about it. As a result, they made it their mission (and their passion) to share this ground-breaking concept with the rest of the world. To achieve that lofty goal, they married and began the mission that still defines their life. After over 30 years of living and working together 24x7x365, they are still wildly enthusiastic about helping the victims of technology learn how to ask for and get the digital (IT) solutions they need to do their jobs better. More importantly, they are more enthusiastically in love with each other than ever before!

Thoroughly reviewed and eagerly anticipated by the agile community, User Stories Applied offers a requirements process that saves time, eliminates rework, and leads directly to better software. The best way to build software that meets users' needs is to begin with "user stories": simple, clear, brief descriptions of functionality that will be valuable to real users. In User Stories Applied, Mike Cohn provides you with a front-to-back blueprint for writing these user stories and weaving them into your development lifecycle. You'll learn what makes a great user story, and what makes a bad one. You'll discover practical ways to gather user stories, even when you can't speak with your users. Then, once you've compiled your user stories, Cohn shows how to organize them, prioritize them, and use them for planning, management, and testing. User role modeling: understanding what users have in common, and where they differ Gathering stories: user interviewing, questionnaires, observation, and workshops Working with managers, trainers, salespeople and other "proxies" Writing user stories for acceptance testing Using stories to prioritize, set schedules, and estimate release costs Includes end-of-chapter practice questions and exercises User Stories Applied will be invaluable to every software developer, tester, analyst, and manager working with any agile method: XP, Scrum... or even your own home-grown approach.

This popular handbook presents a step-by-step method for clearly explaining a product, system, or procedure. The easy-to-follow text—packed with examples and illustrations—explains the unique demands of this form of writing and shows how to set up the best user model. The book covers developing a modular outline and storyboard, generating the draft, revising, developing a formal usability test, and supporting and updating user documentation. Also included are a glossary of terms, a listing of books and periodicals for additional information, and an index.

User story mapping is a valuable tool for software development, once you understand why and how to use it. This insightful book examines how this often misunderstood technique can help your team stay focused on users and their needs without getting lost in the enthusiasm for individual product features. Author Jeff Patton shows you how changeable story maps enable your team to hold better conversations about the project throughout the development process. Your team will learn to come away with a shared understanding of what you ' re attempting to build and why. Get a high-level view of story mapping, with an exercise to learn key concepts quickly Understand how stories really work, and how they come to life in Agile and Lean projects Dive into a story ' s lifecycle, starting with opportunities and moving deeper into discovery Prepare your stories, pay attention while they ' re built, and learn from those you convert to working software

This book presents unique insights and advice on defining and managing the innovation transformation journey. Using novel ideas, examples and best practices, it empowers management executives at all levels to drive cultural, technological and organizational changes toward innovation. Covering modern innovation techniques, tools, programs and strategies, it focuses on the role of the latest technologies (e.g., artificial intelligence to discover, handle and manage ideas), methodologies (including Agile Engineering and Rapid Prototyping) and combinations of these (like hackathons or gamification). At the same time, it highlights the importance of culture and provides suggestions on how to build it. In the era of AI and the unprecedented pace of technology evolution, companies need to become truly innovative in order to survive. The transformation toward an innovation-led company is difficult — it requires a strong leadership and culture, advanced technologies and well-designed programs. The book is based on the author ' s long-term experience and novel ideas, and reflects two decades of startup, consulting and corporate leadership experience. It is intended for business, technology, and innovation leaders.

WHAT IS THIS BOOK ABOUT? Communicate Business Needs in an Agile (e.g. Scrum) or Lean (e.g. Kanban) Environment Problem solvers are in demand in every organization, large and small, from a Mom and Pop shop to the federal government. Increase your confidence and your value to organizations by improving your ability to analyze, extract, express, and discuss business needs in formats supported by Agile, Lean, and DevOps. The single largest challenge facing organizations around the world is how to leverage their Information Technology to gain competitive advantage. This is not about how to program the device; it is figuring out what the device should do. The skills needed to identify and define the best IT solutions are invaluable for every role in the organization. These skills can propel you from the mail room to the boardroom by making your organization more effective and more profitable. Whether you - are tasked with defining business needs for a product or existing software, - need to prove that a digital solution works, - want to expand your User Story and requirements discovery toolkit, or - are interested in becoming a Business Analyst, this book presents invaluable ideas that you can steal. The future looks bright for those who embrace Lean concepts and are prepared to engage with the business community to ensure the success of Agile initiatives. WHAT YOU WILL LEARN Learn Step by Step When and How to Define Lean / Agile Requirements Agile, Lean, DevOps, and Continuous Delivery do not change the need for good business analysis. In this book, you will learn how the new software development philosophies influence the discovery, expression, and analysis of business needs. We will cover User Stories, Features, and Quality Requirements (a.k.a. Non-functional Requirements — NFR). User Story Splitting and Feature Drill-down transform business needs into technology solutions. Acceptance Tests (Scenarios, Scenario Outlines, and Examples) have become a critical part of many Lean development approaches. To support this new testing paradigm, you will also learn how to identify and optimize Scenarios, Scenario Outlines, and Examples in GIVEN-WHEN-THEN format (Gherkin) that are the bases for Acceptance Test Driven Development (ATDD) and Behavior Driven Development (BDD). This book presents concrete approaches that take you from day one of a change initiative to the ongoing acceptance testing in a continuous delivery environment. The authors introduce novel and innovative ideas that augment tried-and-true techniques for - discovering and capturing what your stakeholders need, - writing and refining the needs as the work progresses, and - developing scenarios to verify that the software does what it should. Approaches that proved their value in conventional settings have been redefined to ferret out and eliminate waste (a pillar of the Lean philosophy). Those approaches are fine-tuned and perfected to support the Lean and Agile movement that defines current software development. In addition, the book is chock-full of examples and exercises that allow you to confirm your understanding of the presented ideas. WHO WILL BENEFIT FROM READING THIS BOOK? How organizations develop and deliver working software has changed significantly in recent years. Because the change was greatest in the developer community, many books and courses justifiably target that group. There is, however, an overlooked group of people essential to the development of software-as-an-asset that have been neglected. Many distinct roles or job titles in the business community perform business needs analysis for digital solutions. They include: - Product Owners - Business Analysts - Requirements Engineers - Test Developers - Business- and Customer-side Team Members - Agile Team Members - Subject Matter Experts (SME) - Project Leaders and Managers - Systems Analysts and Designers - AND " anyone wearing the business analysis hat ", meaning anyone responsible for defining a future IT solution TOM AND ANGELA ' S (the authors) STORY Like all good IT stories, theirs started on a project many years ago. Tom was the super techie, Angela the super SME. They fought their way through the 3-year development of a new policy maintenance system for an insurance company. They vehemently disagreed on many aspects, but in the process discovered a fundamental truth about IT projects. The business community (Angela) should decide on the business needs while the technical team ' s (Tom) ' s job was to make the technology deliver what the business needed. Talk about a revolutionary ideal! All that was left was learning how to communicate with each other without bloodshed to make the project a resounding success. Mission accomplished. They decided this epiphany was so important that the world needed to know about it. As a result, they made it their mission (and their passion) to share this ground-breaking concept with the rest of the world. To achieve that lofty goal, they married and began the mission that still defines their life. After over 30 years of living and working together 24x7x365, they are still wildly enthusiastic about helping the victims of technology learn how to ask for and get the IT solutions they need to do their jobs better. More importantly, they are more enthusiastically in love with each other than ever before!

Without words, apps would be an unusable jumble of shapes and icons, while voice interfaces and chatbots wouldn't even exist. Words make software human—centered, and require just as much thought as the branding and code. This book will show you how to give your users clarity, test your words, and collaborate with your team. You'll see that writing is designing.

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.

This book presents unique insights and advice on defining and managing the innovation transformation journey. Using novel ideas, examples and best practices, it empowers management executives at all levels to drive cultural, technological and organizational changes toward innovation. Covering modern innovation techniques, tools, programs and strategies, it focuses on the role of the latest technologies (e.g., artificial intelligence to discover, handle and manage ideas), methodologies (including Agile Engineering and Rapid Prototyping) and combinations of these (like hackathons or gamification). At the same time, it highlights the importance of culture and provides suggestions on how to build it. In the era of AI and the unprecedented pace of technology evolution, companies need to become truly innovative in order to survive. The transformation toward an innovation-led company is difficult — it requires a strong leadership and culture, advanced technologies and well-designed programs. The book is based on the author ' s long-term experience and novel ideas, and reflects two decades of startup, consulting and corporate leadership experience. It is intended for business, technology, and innovation leaders.

WHAT IS THIS BOOK ABOUT? Communicate Business Needs in an Agile (e.g. Scrum) or Lean (e.g. Kanban) Environment Problem solvers are in demand in every organization, large and small, from a Mom and Pop shop to the federal government. Increase your confidence and your value to organizations by improving your ability to analyze, extract, express, and discuss business needs in formats supported by Agile, Lean, and DevOps. The single largest challenge facing organizations around the world is how to leverage their Information Technology to gain competitive advantage. This is not about how to program the device; it is figuring out what the device should do. The skills needed to identify and define the best IT solutions are invaluable for every role in the organization. These skills can propel you from the mail room to the boardroom by making your organization more effective and more profitable. Whether you - are tasked with defining business needs for a product or existing software, - need to prove that a digital solution works, - want to expand your User Story and requirements discovery toolkit, or - are interested in becoming a Business Analyst, this book presents invaluable ideas that you can steal. The future looks bright for those who embrace Lean concepts and are prepared to engage with the business community to ensure the success of Agile initiatives. WHAT YOU WILL LEARN Learn Step by Step When and How to Define Lean / Agile Requirements Agile, Lean, DevOps, and Continuous Delivery do not change the need for good business analysis. In this book, you will learn how the new software development philosophies influence the discovery, expression, and analysis of business needs. We will cover User Stories, Features, and Quality Requirements (a.k.a. Non-functional Requirements — NFR). User Story Splitting and Feature Drill-down transform business needs into technology solutions. Acceptance Tests (Scenarios, Scenario Outlines, and Examples) have become a critical part of many Lean development approaches. To support this new testing paradigm, you will also learn how to identify and optimize Scenarios, Scenario Outlines, and Examples in GIVEN-WHEN-THEN format (Gherkin) that are the bases for Acceptance Test Driven Development (ATDD) and Behavior Driven Development (BDD). This book presents concrete approaches that take you from day one of a change initiative to the ongoing acceptance testing in a continuous delivery environment. The authors introduce novel and innovative ideas that augment tried-and-true techniques for - discovering and capturing what your stakeholders need, - writing and refining the needs as the work progresses, and - developing scenarios to verify that the software does what it should. Approaches that proved their value in conventional settings have been redefined to ferret out and eliminate waste (a pillar of the Lean philosophy). Those approaches are fine-tuned and perfected to support the Lean and Agile movement that defines current software development. In addition, the book is chock-full of examples and exercises that allow you to confirm your understanding of the presented ideas. WHO WILL BENEFIT FROM READING THIS BOOK? How organizations develop and deliver working software has changed significantly in recent years. Because the change was greatest in the developer community, many books and courses justifiably target that group. There is, however, an overlooked group of people essential to the development of software-as-an-asset that have been neglected. Many distinct roles or job titles in the business community perform business needs analysis for digital solutions. They include: - Product Owners - Business Analysts - Requirements Engineers - Test Developers - Business- and Customer-side Team Members - Agile Team Members - Subject Matter Experts (SME) - Project Leaders and Managers - Systems Analysts and Designers - AND " anyone wearing the business analysis hat ", meaning anyone responsible for defining a future IT solution TOM AND ANGELA ' S (the authors) STORY Like all good IT stories, theirs started on a project many years ago. Tom was the super techie, Angela the super SME. They fought their way through the 3-year development of a new policy maintenance system for an insurance company. They vehemently disagreed on many aspects, but in the process discovered a fundamental truth about IT projects. The business community (Angela) should decide on the business needs while the technical team ' s (Tom) ' s job was to make the technology deliver what the business needed. Talk about a revolutionary ideal! All that was left was learning how to communicate with each other without bloodshed to make the project a resounding success. Mission accomplished. They decided this epiphany was so important that the world needed to know about it. As a result, they made it their mission (and their passion) to share this ground-breaking concept with the rest of the world. To achieve that lofty goal, they married and began the mission that still defines their life. After over 30 years of living and working together 24x7x365, they are still wildly enthusiastic about helping the victims of technology learn how to ask for and get the IT solutions they need to do their jobs better. More importantly, they are more enthusiastically in love with each other than ever before!

Without words, apps would be an unusable jumble of shapes and icons, while voice interfaces and chatbots wouldn't even exist. Words make software human—centered, and require just as much thought as the branding and code. This book will show you how to give your users clarity, test your words, and collaborate with your team. You'll see that writing is designing.

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.

Copyright code : 075834f8687cdf5564287c5b4e79fb2f1