

## Extreme Programming Explained 1999

The Agile Samurai  
Extreme Programming Explored  
Smalltalk, Objects, and Design  
Extreme Programming and Agile Methods - XP/  
Agile Universe 2004  
Agile Software Development  
Implementation Patterns  
Smalltalk Best Practice Patterns  
Extreme Programming Applied  
Working Effectively with Legacy Code  
The Way of the Web Tester  
Extreme Programming Explained  
Extreme Programming and Agile Processes in Software Engineering  
Extreme Programming EXplained  
Managing Software Requirements  
Agile Processes in Software Engineering and Extreme Programming  
Extreme Programming and Agile Processes in Software Engineering  
Extreme Programming and Agile Processes in Software Engineering  
Extreme Programming Refactored  
The Art of Agile Development  
Agile Processes in Software Engineering and Extreme Programming  
Agile Project Management For Dummies  
Test-driven Development  
Extreme Programming Perspectives  
Agile Processes in Software Engineering and Extreme Programming  
Extreme Programming Pocket Guide  
Refactoring to Patterns  
The Business of Software  
Lean Software Development  
Refactoring  
Extreme .NET  
Kent Beck's Guide to Better  
Smalltalk  
Planning Extreme Programming  
Extreme Programming and Agile Processes in Software Engineering  
Extreme Programming Installed  
Domain-Driven Design Distilled  
EuroSPI 2004  
Extreme Programming and Agile Processes in Software Engineering  
xUnit Test Patterns  
Clean Agile  
User Stories Applied

### The Agile Samurai

Provides information on eXtreme programming, or XP, a software development methodology.

### Extreme Programming Explored

“Agile Software Development is a highly stimulating and rich book. The author has a deep background and gives us a tour de force of the emerging agile methods.”  
—Tom Gilb  
The agile model of software development has taken the world by storm. Now, in Agile Software Development, Second Edition, one of agile’s leading pioneers updates his Jolt Productivity award-winning book to reflect all that’s been learned about agile development since its original introduction. Alistair Cockburn begins by updating his powerful model of software development as a “cooperative game of invention and communication.” Among the new ideas he introduces: harnessing competition without damaging collaboration; learning lessons from lean manufacturing; and balancing strategies for communication. Cockburn also explains how the cooperative game is played in business and on engineering projects, not just software development. Next, he systematically illuminates the agile model, shows how it has evolved, and answers the questions developers and project managers ask most often, including · Where does agile development fit in our organization? · How do we blend agile ideas with other ideas? · How do we extend agile ideas more broadly? Cockburn takes on crucial misconceptions that cause agile projects to fail. For example, you’ll learn why encoding project management strategies into fixed processes can lead to ineffective strategy decisions and costly mistakes. You’ll also find a thoughtful discussion of the

controversial relationship between agile methods and user experience design. Cockburn turns to the practical challenges of constructing agile methodologies for your own teams. You'll learn how to tune and continuously reinvent your methodologies, and how to manage incomplete communication. This edition contains important new contributions on these and other topics: · Agile and CMMI · Introducing agile from the top down · Revisiting "custom contracts" · Creating change with "stickers" In addition, Cockburn updates his discussion of the Crystal methodologies, which utilize his "cooperative game" as their central metaphor. If you're new to agile development, this book will help you succeed the first time out. If you've used agile methods before, Cockburn's techniques will make you even more effective.

### **Smalltalk, Objects, and Design**

Get more out of your legacy systems: more performance, functionality, reliability, and manageability Is your code easy to change? Can you get nearly instantaneous feedback when you do change it? Do you understand it? If the answer to any of these questions is no, you have legacy code, and it is draining time and money away from your development efforts. In this book, Michael Feathers offers start-to-finish strategies for working more effectively with large, untested legacy code bases. This book draws on material Michael created for his renowned Object Mentor seminars: techniques Michael has used in mentoring to help hundreds of developers, technical managers, and testers bring their legacy systems under control. The topics covered include Understanding the mechanics of software change: adding features, fixing bugs, improving design, optimizing performance Getting legacy code into a test harness Writing tests that protect you against introducing new problems Techniques that can be used with any language or platform—with examples in Java, C++, C, and C# Accurately identifying where code changes need to be made Coping with legacy systems that aren't object-oriented Handling applications that don't seem to have any structure This book also includes a catalog of twenty-four dependency-breaking techniques that help you work with program elements in isolation and make safer changes.

### **Extreme Programming and Agile Methods - XP/Agile Universe 2004**

The field of software engineering is characterized by speed and turbulence in many regards. While new ideas are proposed almost on a yearly basis, very few of them live for a decade or a longer. Lightweight software development methods were a new idea in the latter part of the 1990s. Now, ten years later, they are better known as agile software development methods, and an active community driven by practitioners has formed around the new way of thinking. Agile software development is currently being embraced by the research community as well. As a sign of increased research activity, most research-oriented conferences have an agile software development track included in the conference program. The XP conference series established in 2000 was the first conference dedicated to agile processes in software engineering. The idea of the conference is to offer a unique setting for advancing the state of the art in research and practice of agile processes. This year's conference was the tenth consecutive edition of this

international event. Due to the diverse nature of different activities during the conference, XP is claimed to be more of an experience rather than a regular conference. It offers several different ways to interact and strives to create a truly collaborative environment where new ideas and exciting findings can be presented and shared. This is clearly visible from this year's program as well.

### **Agile Software Development**

This book contains most of the papers presented at the 4th International Conference on Extreme Programming and Agile Processes in Software Engineering (XP 2003), held in Genoa, Italy, May 2003. The XP 2000 series of conferences were started in 2000 to promote the change of new ideas, research and applications in the emerging field of agile methodologies for software development. Over the years, the conference has become the main world forum for all major advances in this important field. Also this year the contributions to Agile Methodologies and Extreme Programming were substantial. They demonstrate that the topic is continuing to gain more and more momentum. In spite of some criticism of agile methodologies, everyone agrees that they address some unresolved needs of software practitioners. People still do not know how to develop software on time, with the desired features, and within the given budget! This volume is divided into several thematic sections, easing reader's navigation through the content. Full papers are presented first, followed by research reports, papers from the Educational Symposium, and papers from the Ph.D. Symposium. The presentations given during three panel sessions held at the conference conclude the book. The section on Managing Agile Processes includes contributions highlighting the sometimes difficult relationship between agile methodologies and management, and includes approaches and suggestions that should facilitate the acceptance of agile methodologies at the different levels of management.

### **Implementation Patterns**

Automated testing is a cornerstone of agile development. An effective testing strategy will deliver new functionality more aggressively, accelerate user feedback, and improve quality. However, for many developers, creating effective automated tests is a unique and unfamiliar challenge. xUnit Test Patterns is the definitive guide to writing automated tests using xUnit, the most popular unit testing framework in use today. Agile coach and test automation expert Gerard Meszaros describes 68 proven patterns for making tests easier to write, understand, and maintain. He then shows you how to make them more robust and repeatable--and far more cost-effective. Loaded with information, this book feels like three books in one. The first part is a detailed tutorial on test automation that covers everything from test strategy to in-depth test coding. The second part, a catalog of 18 frequently encountered "test smells," provides trouble-shooting guidelines to help you determine the root cause of problems and the most applicable patterns. The third part contains detailed descriptions of each pattern, including refactoring instructions illustrated by extensive code samples in multiple programming languages.

### **Smalltalk Best Practice Patterns**

This book constitutes the refereed proceedings of the 7th International Conference on Extreme Programming and Agile Processes in Software Engineering, XP 2006, held in Oulu, Finland, June 2006. The book presents 16 revised full papers together with 6 experience papers, 12 poster papers and panel summaries, organized in topical sections on foundation and rationale for agile methods, effects of pair programming, quality in agile software development, and more.

### **Extreme Programming Applied**

Lean Software Development: An Agile Toolkit Adapting agile practices to your development organization Uncovering and eradicating waste throughout the software development lifecycle Practical techniques for every development manager, project manager, and technical leader Lean software development: applying agile principles to your organization In Lean Software Development, Mary and Tom Poppendieck identify seven fundamental "lean" principles, adapt them for the world of software development, and show how they can serve as the foundation for agile development approaches that work. Along the way, they introduce 22 "thinking tools" that can help you customize the right agile practices for any environment. Better, cheaper, faster software development. You can have all three—if you adopt the same lean principles that have already revolutionized manufacturing, logistics and product development. Iterating towards excellence: software development as an exercise in discovery Managing uncertainty: "decide as late as possible" by building change into the system. Compressing the value stream: rapid development, feedback, and improvement Empowering teams and individuals without compromising coordination Software with integrity: promoting coherence, usability, fitness, maintainability, and adaptability How to "see the whole"—even when your developers are scattered across multiple locations and contractors Simply put, Lean Software Development helps you refocus development on value, flow, and people—so you can achieve breakthrough quality, savings, speed, and business alignment.

### **Working Effectively with Legacy Code**

Written for Smalltalk programmers, this book is designed to help readers become more effective Smalltalk developers and object technology users.

### **The Way of the Web Tester**

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

### **Extreme Programming Explained**

The co-author of Microsoft Secrets links issues related to strategy and organization to those of managing technology, arguing that companies must chose a business model that will capitalize on good times and survive more difficult periods, and presenting the success stories of such companies as IBM, Toshiba, and Motorola. 25,000 first printing.

## **Extreme Programming and Agile Processes in Software Engineering**

This book constitutes the refereed proceedings of the 4th Conference on Extreme Programming and Agile Methods, XP/Agile Universe 2004, held in Calgary, Canada in August 2004. The 18 revised full papers presented together with summaries of workshops, panels, and tutorials were carefully reviewed and selected from 45 submissions. The papers are organized in topical sections on testing and integration, managing requirements and usability, pair programming, foundations of agility, process adaptation, and educational issues.

## **Extreme Programming EXplained**

A guide to XP leads the developer, project manager, and team leader through the software development planning process, offering real world examples and tips for reacting to changing environments quickly and efficiently.

## **Managing Software Requirements**

This book constitutes the refereed proceedings of the 8th International Conference on Agile Processes in Software Engineering and eXtreme Programming, XP 2007, held in Como, Italy in June 2007. It covers managing agile processes, extending agile methodologies, teaching and introducing agile methodologies, methods and tools, empirical studies, and methodology issue.

## **Agile Processes in Software Engineering and Extreme Programming**

The first edition of "Extreme Programming Explained" is a classic. It won awards for its then-radical ideas for improving small-team development, such as having developers write automated tests for their own code and having the whole team plan weekly. Much has changed in five years. This completely rewritten second edition expands the scope of XP to teams of any size by suggesting a program of continuous improvement based on: five core values consistent with excellence in software development; eleven principles for putting those values into action; and, thirteen primary and eleven corollary practices to help you push development past its current business and technical limitations. Whether you have a small team that is already closely aligned with your customers or a large team in a gigantic or multinational organization, you will find in these pages a wealth of ideas to challenge, inspire, and encourage you and your team members to substantially improve your software development.

## **Extreme Programming and Agile Processes in Software**

## **Engineering**

This book constitutes the refereed proceedings of the 7th International Conference on Extreme Programming and Agile Processes in Software Engineering, XP 2006, held in Oulu, Finland, June 2006. The book presents 16 revised full papers together with 6 experience papers, 12 poster papers and panel summaries, organized in topical sections on foundation and rationale for agile methods, effects of pair programming, quality in agile software development, and more.

## **Extreme Programming and Agile Processes in Software Engineering**

The problem. Risk: The basic problem. A development episode. Economics of software development. Four variables. Cost of change. Learning to drive. Four values. Basic principles. Back to basics. The solution. Quick overview. How could this work? Management strategy. Facilities strategy. Splitting business and technical responsibility. Planning strategy. Development strategy. Design strategy. Testing strategy. Implementing XP. Adopting XP. Retrofitting XP. Lifecycle of an ideal XP project. Roles for people. 20-80 rule. What makes XP hard. When you shouldn't try XP. XP at work. Conclusion.

## **Extreme Programming Refactored**

This book constitutes the refereed proceedings of the 4th International Conference on Extreme Programming and Agile Processes in Software Engineering, XP 2003, held in Genova, Italy in May 2003. The 33 revised full papers presented together with experience reports, research reports, abstracts from an educational symposium and a PhD symposium, as well as panel statements were carefully reviewed and selected for inclusion in the book. The papers are organized in topical sections on managing agile processes, methodological issues, extending agile methodologies and XP, methods and tools, teaching and education aspects, testing, pair programming, and other XP practices.

## **The Art of Agile Development**

This book is for everyone who needs to test the web. As a tester, you'll automate your tests. As a developer, you'll build more robust solutions. And as a team, you'll gain a vocabulary and a means to coordinate how to write and organize automated tests for the web. Follow the testing pyramid and level up your skills in user interface testing, integration testing, and unit testing. Your new skills will free you up to do other, more important things while letting the computer do the one thing it's really good at: quickly running thousands of repetitive tasks. This book shows you how to do three things: How to write really good automated tests for the web. How to pick and choose the right ones. \* How to explain, coordinate, and share your efforts with others. If you're a traditional software tester who has never written an automated test before, this is the perfect book for getting started. Together, we'll go through everything you'll need to start writing your own tests. If you're a developer, but haven't thought much about testing, this book will show you how to move fast without breaking stuff. You'll test RESTful web services and

legacy systems, and see how to organize your tests. And if you're a team lead, this is the Rosetta Stone you've been looking for. This book will help you bridge that testing gap between your developers and your testers by giving your team a model to discuss automated testing, and most importantly, to coordinate their efforts. The Way of the Web Tester is packed with cartoons, graphics, best practices, war stories, plenty of humor, and hands-on tutorial exercises that will get you doing the right things, the right way.

## **Agile Processes in Software Engineering and Extreme Programming**

Domain-Driven Design (DDD) software modeling delivers powerful results in practice, not just in theory, which is why developers worldwide are rapidly moving to adopt it. Now, for the first time, there's an accessible guide to the basics of DDD: What it is, what problems it solves, how it works, and how to quickly gain value from it. Concise, readable, and actionable, Domain-Driven Design Distilled never buries you in detail—it focuses on what you need to know to get results. Vaughn Vernon, author of the best-selling Implementing Domain-Driven Design, draws on his twenty years of experience applying DDD principles to real-world situations. He is uniquely well-qualified to demystify its complexities, illuminate its subtleties, and help you solve the problems you might encounter. Vernon guides you through each core DDD technique for building better software. You'll learn how to segregate domain models using the powerful Bounded Contexts pattern, to develop a Ubiquitous Language within an explicitly bounded context, and to help domain experts and developers work together to create that language. Vernon shows how to use Subdomains to handle legacy systems and to integrate multiple Bounded Contexts to define both team relationships and technical mechanisms. Domain-Driven Design Distilled brings DDD to life. Whether you're a developer, architect, analyst, consultant, or customer, Vernon helps you truly understand it so you can benefit from its remarkable power. Coverage includes What DDD can do for you and your organization—and why it's so important The cornerstones of strategic design with DDD: Bounded Contexts and Ubiquitous Language Strategic design with Subdomains Context Mapping: helping teams work together and integrate software more strategically Tactical design with Aggregates and Domain Events Using project acceleration and management tools to establish and maintain team cadence

## **Agile Project Management For Dummies**

Agile Values and Principles for a New Generation “In the journey to all things Agile, Uncle Bob has been there, done that, and has the both the t-shirt and the scars to show for it. This delightful book is part history, part personal stories, and all wisdom. If you want to understand what Agile is and how it came to be, this is the book for you.” –Grady Booch “Bob’s frustration colors every sentence of Clean Agile, but it’s a justified frustration. What is in the world of Agile development is nothing compared to what could be. This book is Bob’s perspective on what to focus on to get to that ‘what could be.’ And he’s been there, so it’s worth listening.” –Kent Beck “It’s good to read Uncle Bob’s take on Agile. Whether just beginning, or a seasoned Agilista, you would do well to read this book. I agree with

almost all of it. It's just some of the parts make me realize my own shortcomings, dammit. It made me double-check our code coverage (85.09%).” –Jon Kern Nearly twenty years after the Agile Manifesto was first presented, the legendary Robert C. Martin (“Uncle Bob”) reintroduces Agile values and principles for a new generation—programmers and nonprogrammers alike. Martin, author of Clean Code and other highly influential software development guides, was there at Agile’s founding. Now, in Clean Agile: Back to Basics, he strips away misunderstandings and distractions that over the years have made it harder to use Agile than was originally intended. Martin describes what Agile is in no uncertain terms: a small discipline that helps small teams manage small projects . . . with huge implications because every big project is comprised of many small projects. Drawing on his fifty years’ experience with projects of every conceivable type, he shows how Agile can help you bring true professionalism to software development. Get back to the basics—what Agile is, was, and should always be Understand the origins, and proper practice, of SCRUM Master essential business-facing Agile practices, from small releases and acceptance tests to whole-team communication Explore Agile team members’ relationships with each other, and with their product Rediscover indispensable Agile technical practices: TDD, refactoring, simple design, and pair programming Understand the central roles values and craftsmanship play in your Agile team’s success If you want Agile’s true benefits, there are no shortcuts: You need to do Agile right. Clean Agile: Back to Basics will show you how, whether you’re a developer, tester, manager, project manager, or customer. Register your book for convenient access to downloads, updates, and/or corrections as they become available. See inside book for details.

## **Test-driven Development**

You know what XP is, how to get it up and running, and how to plan projects using it. Now it's time to expand your use of Extreme Programming and learn the best practices of this popular discipline. In "Extreme Programming Explored, " you can read about best practices as learned from the concrete experience of successful XP developers. Author and programmer Bill Wake provides answers to practical questions about XP implementation. Using hands-on examples—including code samples written in the Java programming language--this book demonstrates the day-to-day mechanics of working on an XP team and shows well-defined methods for carrying out a successful XP project. The book is divided into three parts: Part 1, Programming--programming incrementally, test-first, and refactoring. Part 2, Team Practices--code ownership, integration, overtime, and pair programming; how XP approaches system architecture; and how a system metaphor shapes a common vision, a shared vocabulary, and the architecture. Part 3, Processes--how to write stories to plan a release; how to plan iterations; and the activities in a typical day for the customer, the programmer, and the manager of an XP project. To demonstrate how an XP team uses frequent testing, you'll learn how to develop the core of a library search system by unit testing in small increments. To show how to make code ready for major design changes, the author teaches you how to refactor a Java program that generates a Web page. To see how a system metaphor influences the shape of a system, you'll learn about the effects of different metaphors on customer service and word processing applications. To show how customers and programmers participate in release planning, the book demonstrates writing and estimating stories, and shows how the customer plans a

release. 0201733978B07052001

## **Extreme Programming Perspectives**

In 1994, Design Patterns changed the landscape of object-oriented development by introducing classic solutions to recurring design problems. In 1999, Refactoring revolutionized design by introducing an effective process for improving code. With the highly anticipated Refactoring to Patterns, Joshua Kerievsky has changed our approach to design by forever uniting patterns with the evolutionary process of refactoring. This book introduces the theory and practice of pattern-directed refactorings: sequences of low-level refactorings that allow designers to safely move designs to, towards, or away from pattern implementations. Using code from real-world projects, Kerievsky documents the thinking and steps underlying over two dozen pattern-based design transformations. Along the way he offers insights into pattern differences and how to implement patterns in the simplest possible ways. Coverage includes: A catalog of twenty-seven pattern-directed refactorings, featuring real-world code examples Descriptions of twelve design smells that indicate the need for this book's refactorings General information and new insights about patterns and refactoring Detailed implementation mechanics: how low-level refactorings are combined to implement high-level patterns Multiple ways to implement the same pattern—and when to use each Practical ways to get started even if you have little experience with patterns or refactoring Refactoring to Patterns reflects three years of refinement and the insights of more than sixty software engineering thought leaders in the global patterns, refactoring, and agile development communities. Whether you're focused on legacy or "greenfield" development, this book will make you a better software designer by helping you learn how to make important design changes safely and effectively.

## **Agile Processes in Software Engineering and Extreme Programming**

Extreme Programming (XP) is a significant departure from traditional software development methods, one that is ushering in a change for both developers and business people. It is an agile methodology, which enables highly productive teams to produce quality software from rapidly changing or unclear requirements. XP is disciplined software craftsmanship, elevating best practices in software analysis, design, testing, implementation, and project management to a new level. "Extreme Programming Applied" helps you begin using the principles behind this revolutionary concept. Even as the popularity of XP grows, many programmers and developers are still seeking practical advice on getting started. They find themselves in search of an XP roadmap, one that points to paths around the obstacles. "Extreme Programming Applied" is just that roadmap, a pragmatic guide to getting started with Extreme Programming. It helps programmers and project managers take their first steps toward applying the XP discipline. This book is not a tutorial, however. It uses real-world experience to educate readers about how to apply XP in their organizations. The authors offer guidelines for implementing XP, illustrating key points with valuable stories from successful XP pioneers.

0201616408B09172001

## Extreme Programming Pocket Guide

Extreme Programming has come a long way since its first use in the C3 project almost 10 years ago. Agile methods have found their way into the mainstream, and at the end of last year we saw the second edition of Kent Beck's book on Extreme Programming, containing a major refactoring of XP. This year, the 6th International Conference on Extreme Programming and Agile Processes in Software Engineering took place June 18-23 in Sheffield. As in the years before, XP 2005 provided a unique forum for industry and academic professionals to discuss their needs and ideas on Extreme Programming and agile methodologies. These proceedings reflect the activities during the conference which ranged from presentation of research papers, invited talks, posters and demonstrations, panels and activity sessions, to tutorials and workshops. Included are also papers from the Ph.D. and Master's Symposium which provided a forum for young researchers to present their results and to get feedback. As varied as the activities were the topics of the conference which covered the presentation of new and improved practices, empirical studies, experience reports and case studies, and last but not least the social aspects of agile methods. The papers and the activities went through a rigorous reviewing process. Each paper was reviewed by at least three Program Committee members and was discussed carefully among the Program Committee. Of 62 papers submitted, only 22 were accepted as full papers.

## Refactoring to Patterns

Stephens and Rosenberg examine XP in the context of existing methodologies and processes such as RUP, ICONIX, Spiral, RAD, DSDM, etc - and show how XP goals can be achieved using these existing processes.

## The Business of Software

Software Expert Kent Beck Presents a Catalog of Patterns Infinitely Useful for Everyday Programming Great code doesn't just function: it clearly and consistently communicates your intentions, allowing other programmers to understand your code, rely on it, and modify it with confidence. But great code doesn't just happen. It is the outcome of hundreds of small but critical decisions programmers make every single day. Now, legendary software innovator Kent Beck—known worldwide for creating Extreme Programming and pioneering software patterns and test-driven development—focuses on these critical decisions, unearthing powerful “implementation patterns” for writing programs that are simpler, clearer, better organized, and more cost effective. Beck collects 77 patterns for handling everyday programming tasks and writing more readable code. This new collection of patterns addresses many aspects of development, including class, state, behavior, method, collections, frameworks, and more. He uses diagrams, stories, examples, and essays to engage the reader as he illuminates the patterns. You'll find proven solutions for handling everything from naming variables to checking exceptions.

## Lean Software Development

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.

### **Refactoring**

Users can dramatically improve the design, performance, and manageability of object-oriented code without altering its interfaces or behavior. "Refactoring" shows users exactly how to spot the best opportunities for refactoring and exactly how to do it, step by step.

### **Extreme .NET**

More than a guide to the Smalltalk language.

### **Kent Beck's Guide to Better Smalltalk**

Filled with practical, hands-on examples, this will be the first book Microsoft developers go to when learning Agile development techniques.

### **Planning Extreme Programming**

Printed in full color. Faced with a software project of epic proportions? Tired of over-committing and under-delivering? Enter the dojo of the agile samurai, where agile expert Jonathan Rasmusson shows you how to kick-start, execute, and deliver your agile projects. Combining cutting-edge tools with classic agile practices, *The Agile Samurai* gives you everything you need to deliver something of value every week and make rolling your software into production a non-event. Get ready to kick some software project butt. By learning the ways of the agile samurai you will discover: how to create plans and schedules your customer and your team can believe in what characteristics make a good agile team and how to form your own how to gather requirements in a fraction of the time using agile user stories what to do when you discover your schedule is wrong, and how to look like a pro correcting it how to execute fiercely by leveraging the power of agile software

engineering practices By the end of this book you will know everything you need to set up, execute, and successfully deliver agile projects, and have fun along the way. If you're a project lead, this book gives you the tools to set up and lead your agile project from start to finish. If you are an analyst, programmer, tester, usability designer, or project manager, this book gives you the insight and foundation necessary to become a valuable agile team member. The Agile Samurai slices away the fluff and theory that make other books less-than-agile. It's packed with best practices, war stories, plenty of humor and hands-on tutorial exercises that will get you doing the right things, the right way. This book will make a difference.

### **Extreme Programming and Agile Processes in Software Engineering**

This book constitutes the refereed proceeding of the 11th European Software Process Improvement Conference, EuroSPI 2004, held in Trondheim, Norway in November 2004. The 18 revised full papers presented were carefully reviewed and selected from 31 submissions. The papers are organized in topical sections on agile methods and global software development, software process improvement techniques, studies of software process improvement, knowledge management, and effort estimation.

### **Extreme Programming Installed**

Write clean code that works with the help of this groundbreaking software method. Example-driven teaching is the basis of Beck's step-by-step instruction that will have readers using TDD to further their projects.

### **Domain-Driven Design Distilled**

Any way readers slice it, Extreme Programming is thought-provoking and controversial. "Extreme Programming Perspectives" collects the best papers from the world's two leading XP conferences, and presents this information in a handy, easily-digestible format that allows reader to determine whether XP is a prudent undertaking for their organization, and how their organization can optimize their XP initiatives.

### **EuroSPI 2004**

This updated edition shows you how to use the agile project management framework for success! Learn how to apply agile concepts to your projects. This fully updated book covers changes to agile approaches and new information related to the methods of managing an agile project. Agile Project Management For Dummies, 3rd Edition gives product developers and other project leaders the tools they need for a successful project. This book's principles and techniques will guide you in creating a product roadmap, self-correcting iterations of deployable products, and preparing for a product launch. Agile approaches are critical for achieving fast and flexible product development. It's also a useful tool for managing a range of business projects. Written by one of the original agile

technique thought-leaders, this book guides you and your teams in discovering why agile techniques work and how to create an effective agile environment. Users will gain the knowledge to improve various areas of project management. Define your product's vision and features Learn the steps for putting agile techniques into action Manage the project's scope and procurement Plan your team's sprints and releases Simplify reporting related to the project Agile Project Management For Dummies can help you to better manage the scope of your project as well as its time demands and costs. You'll also be prepared to skillfully handle team dynamics, quality challenges, and risks.

### **Extreme Programming and Agile Processes in Software Engineering**

For those considering Extreme Programming, this book provides no-nonsense advice on agile planning, development, delivery, and management taken from the authors' many years of experience. While plenty of books address the what and why of agile development, very few offer the information users can apply directly.

### **xUnit Test Patterns**

A classic treatise that defined the field of applied demand analysis, *Consumer Demand in the United States: Prices, Income, and Consumption Behavior* is now fully updated and expanded for a new generation. Consumption expenditures by households in the United States account for about 70% of America's GDP. The primary focus in this book is on how households adjust these expenditures in response to changes in price and income. Econometric estimates of price and income elasticities are obtained for an exhaustive array of goods and services using data from surveys conducted by the Bureau of Labor Statistics, providing a better understanding of consumer demand. Practical models for forecasting future price and income elasticities are also demonstrated. Fully revised with over a dozen new chapters and appendices, the book revisits the original Taylor-Houthakker models while examining new material as well, such as the use of quantile regression and the stationarity of consumer preference. It also explores the emerging connection between neuroscience and consumer behavior, integrating the economic literature on demand theory with psychology literature. The most comprehensive treatment of the topic to date, this volume will be an essential resource for any researcher, student or professional economist working on consumer behavior or demand theory, as well as investors and policymakers concerned with the impact of economic fluctuations.

### **Clean Agile**

Extreme programming. The circle of life. On-site customer. User stories. Acceptance tests. Sidebar: Acceptance test samples. Story estimation. Interlude: Sense of completion. Small releases. Customer defines release. Iteration planning. Quick design session. Programming. Sidebar: Code Quality. Pair programming. Unit tests. Sidebar: xUnit. Test first, by intention. Releasing changes. Do or do not. Experience improves estimates. Resources, scope, quality, time. Steering. Steering the iteration. steering the release. Handling defects. Sidebar:

Advanced Issue: Defect databases; Advanced practice: Tests as database. Conclusion. Bonus tracks: We'll try. How to estimate anything. Infrastructure. It's chet's fault. Balancing hopes and fears. Testing improves code. XPer tries Java. A Java perspective. A true story. Estimates and promises. Everything that could possibly break.

## **User Stories Applied**

This classic book is the definitive real-world style guide for better Smalltalk programming. This author presents a set of patterns that organize all the informal experience successful Smalltalk programmers have learned the hard way. When programmers understand these patterns, they can write much more effective code. The concept of Smalltalk patterns is introduced, and the book explains why they work. Next, the book introduces proven patterns for working with methods, messages, state, collections, classes and formatting. Finally, the book walks through a development example utilizing patterns. For programmers, project managers, teachers and students -- both new and experienced. This book presents a set of patterns that organize all the informal experience of successful Smalltalk programmers. This book will help you understand these patterns, and empower you to write more effective code.

[ROMANCE](#) [ACTION & ADVENTURE](#) [MYSTERY & THRILLER](#) [BIOGRAPHIES & HISTORY](#) [CHILDREN'S](#) [YOUNG ADULT](#) [FANTASY](#) [HISTORICAL FICTION](#) [HORROR](#) [LITERARY FICTION](#) [NON-FICTION](#) [SCIENCE FICTION](#)