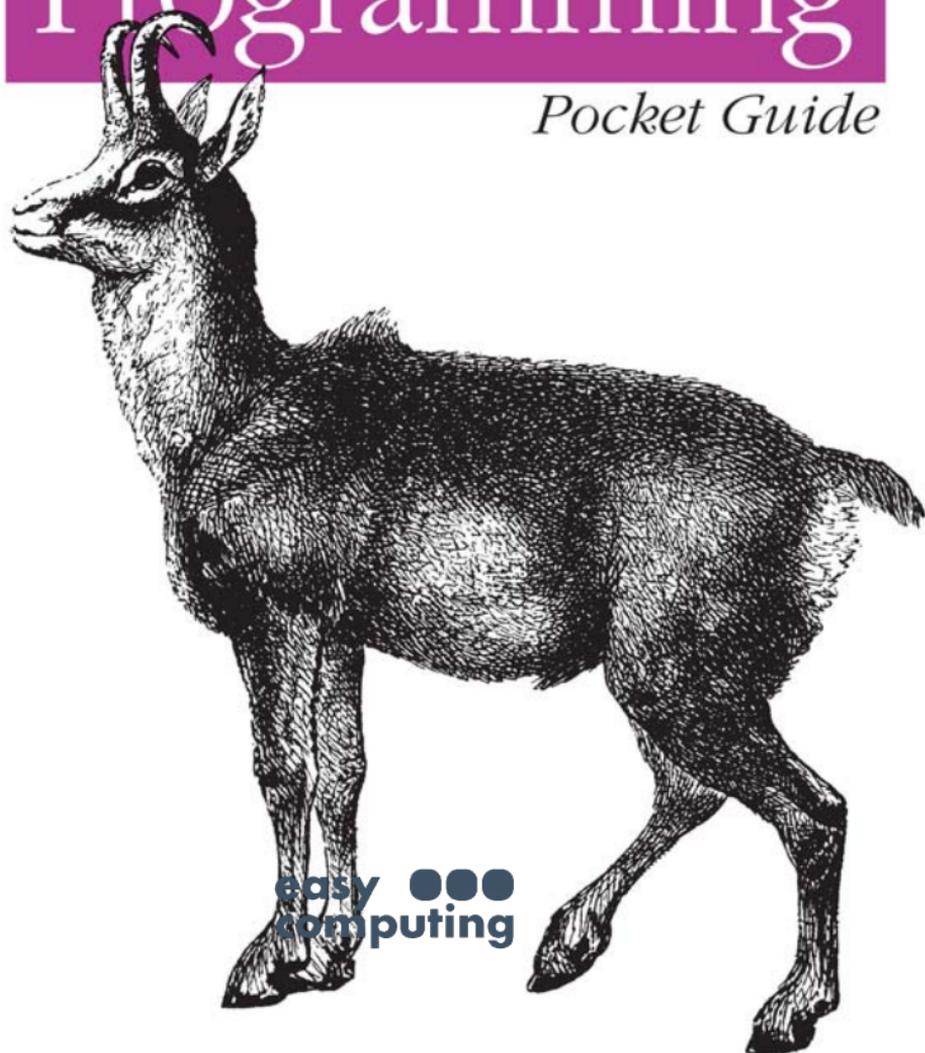


Team-Based Software Development

Extreme Programming

Pocket Guide



easy ●●●
computing

O'REILLY®

chromatic
Foreword by Ward Cunningham

Extreme Programming Pocket Guide



Is your software project coming in late, over budget, under featured, with dubious quality? Are you spending late hours debugging the same mistakes over and over again? Do your goals change week after week?

Maybe not. Maybe your project is perfect. The rest of us need something more.

Extreme Programming, or XP, is a new approach to software development that blends widely accepted best practices—constant testing, code reviews, customer involvement—with a focus on simplicity and craftsmanship. While these methods resonate with developers, actually practicing them can bring challenges and constant demands on patience.

Extreme Programming Pocket Guide explains XP's core practices as they fit into a coherent whole. Whether you're a developer, a customer, or a manager, or whether you're starting a project from scratch or trying to improve an existing one, XP has something to teach you. It's time to enjoy writing software again.

"This book is a worthy follower of our manifesto, 'Extreme Programming: Embrace Change.' It paints the same picture for a larger audience. It is straightforward, factual, concise, and confident, and it exploits our most recent experience. This is what we expect of any refactoring."

—Ward Cunningham, co-inventor of XP



Visit O'Reilly on
the Web at
www.oreilly.com



Extreme Programming

Pocket Guide

chromatic

easy ●●●
computing

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo

Extreme Programming Pocket Guide

by chromatic

Copyright © 2003 O'Reilly Media, Inc. All rights reserved.
Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North,
Sebastopol, CA 95472.

O'Reilly Media, Inc. books may be purchased for educational,
business, or sales promotional use. Online editions are also available
for most titles (*safari.oreilly.com*). For more information, contact our
corporate/institutional sales department: (800) 998-9938 or
corporate@oreilly.com.

Editor: Tatiana Apandi Diaz
Production Editor: Genevieve d'Entremont
Cover Designer: Emma Colby
Interior Designer: David Futato

Printing History:

July 2003: First Edition.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly Media, Inc. The *Pocket Guide* series designations, *Extreme Programming Pocket Guide*, the image of a chamois, and related trade dress are trademarks of O'Reilly Media, Inc. Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc. was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

easy 
computing

Contents

Foreword	ix
Preface	xi

Part I. Why XP?

Who Cares About Process, Anyway?	2
The XP Equation	3
XP Values	7
Communication	7
Feedback	8
Simplicity	10
Courage	10
Assuming Sufficiency	10
Sufficient Time	11
Sufficient Resources	11
Constant Cost of Change	12
Developer Effectiveness	13
Freedom to Experiment	14

Part II. Extreme Programming Practices

Coding Practices	16
Coding Practice 1: Code and Design Simply	16
Coding Practice 2: Refactor Mercilessly	18

Coding Practice 3: Develop Coding Standards	21
Coding Practice 4: Develop a Common Vocabulary	23
Developer Practices	25
Developer Practice 1: Adopt Test-Driven Development	25
Developer Practice 2: Practice Pair Programming	29
Developer Practice 3: Adopt Collective Code Ownership	32
Developer Practice 4: Integrate Continually	34
Business Practices	36
Business Practice 1: Add a Customer to the Team	36
Business Practice 2: Play the Planning Game	38
Business Practice 3: Release Regularly	41
Business Practice 4: Work at a Sustainable Pace	43

Part III. XP Events

Iteration Planning	45
Stories and Tasks	45
Estimates and Schedules	46
The First Iteration	49
The Iteration	50
Releasing	51

Part IV. Extreme Programming Artifacts

Story Cards	53
Task Cards	55
The Bullpen	56

Part V. Roles in Extreme Programming

The Customer	59
--------------	----

Customer Rights	60
Customer Responsibilities	61
The Developer	62
Developer Rights	62
Developer Responsibilities	63
Supplementary Roles	63
The Tracker	63
The Coach	64

Part VI. Coding, XP Style

Do the Simplest Thing That Could Possibly Work	65
You Aren't Gonna Need It	67
Once and Only Once	69

Part VII. Adopting XP

Before You Start	72
Eliminating Fear and Working Together	72
Starting Feedback	73
Including Managers and Customers	76
Now That You're Extreme	78

Part VIII. Further Resources

XP Resources	79
Index	83



easy ●●●
computing

Foreword

This is the most important Extreme Programming book written to date. It is the most complete and concise overview of the words all Extreme Programmers use to describe what they do. Until now, those wishing to understand Extreme Programming had to piece together the lore of the movement from innovative but now overly voluminous sources. The community has grown to where it needs this book.

Extreme Programming's roots started with a few people on a few projects taking a fresh look at what they do. It could have been any number of such small groups—the ideas of Extreme Programming have been invented over and over. But one group took advantage of the unique properties of the World Wide Web to articulate, refine, and expand their ideas in an ever enlarging community.

My own web site, the Portland Pattern Repository's WikiWikiWeb, gave voice to this community at its founding. Wiki lets readers become writers and trusts them to organize their writing to meet their own needs. The result is a collection of terms that have very precise and universally accepted meanings, even though they may not appear so at first glance.

That is why we need this book now. Our movement is blessed with a wonderful series of books that explains the counterintuitive logic of the methods from every perspective. These books served the second generation of Extreme Programmers who were willing to buck the prevailing wisdom in

order to get to a better place. But now Extreme Programming is becoming the prevailing wisdom, and not every convert wants to (or has time to) relive its founding. We're going to need stacks of this book.

This is the guide for you, the third generation of Extreme Programmers. You probably chose Extreme Programming based on its reputation. You don't need to be convinced, but you do need to understand. Be you sponsor, manager, customer, tester, user, or developer, if you are involved in an Extreme Programming project, then you are an Extreme Programmer. Welcome to the team. Here is your handbook. Now let's get to work.

—Ward Cunningham
Cunningham & Cunningham, Inc.

easy ●●●
computing

Preface

There's a word for software that cannot be changed after delivery. That word is hardware.

In the early days of computer programming, processor time was expensive. If you had an error that kept your program from running, it could be days or weeks before you had the chance to try again. Any change could ripple through the rest of your program. To save time and money, you'd have to be completely sure your program would work before it reached the computer. You could spend hours poring over your code.

The obvious lesson was *change is painful and expensive*.

A few decades later, *Extreme Programming* (hereafter called *XP*) claims otherwise. It's possible to develop high-quality software despite—or even because of—change. XP's great assumption is that a little bit of planning, a little bit of coding, and a little bit of testing let you decide if you're right or wrong while it's still cheap to change your mind. You still need some idea where you're going, but you don't have to commit to an exact itinerary. You can change your mind along the way without spending a fortune.

XP is a software development method that emphasizes simplicity, feedback, courage, and communication. It's partly a reaction to the pervasive belief that change is bad and avoidable. Kent Beck  *Explained* (Addison-Wesley) as a collection of 12 fundamental XP practices. Few of these practices are new—they've

teamwork
 bullpens and, 56–58
 required for coding standards, 23
 supported by pair programming, 31
test suites, required by continual integration, 35
test-driven development, 25–29, 74
 achieving simplicity of implementation, 66
 feedback process and, 9
 required by
 collective code ownership, 33
 continual integration, 35
 refactoring, 21
 regular releases, 42
 supported by
 customers on team, 38
 pair programming, 31
 planning game, 41
 simple code and design, 18
Thomas, Dave, 69
time
 actual vs. estimated, 47
 estimating, for stories/tasks, 46–49
 managing projects in terms of, 3–6
 sufficiency of, 11
tracking schedules, 51, 63
 electronically vs. on paper, 53
 task cards and, 56

U

unit tests, 26–28
 adding, 74
Univ. of Michigan School of Information war rooms study, 81

University of Utah pair programming study, 80
unrequested features, avoid adding, 17

V

vacation time and productivity, 44
values of XP, 7–10
velocity of developers, 47–50
 schedule trackers and, 63
vocabulary, common (see common vocabulary)

W

war rooms
 effective teamwork and, 56–58
 study of, done at Univ. of Michigan, 81
Wells, Don, 79
whole team approach, 36–38
Wiki web site (Ward Cunningham's), 79
Williams, Laurie, 80
working overtime vs. reducing scope, 43

X

XP

adopting, 71–78
artifacts used with, 53–58
business practices, 36–44
coding practices, 16–24
coding style for, 65–70
communication within projects, 7
constant cost of change, ensuring, 12
developer effectiveness, aiding, 13

easy 
computing

XP (*continued*)

- developer practices, 25–36, 74
- feedback between customers/
developers, 8
- freedom to experiment,
encouraging, 14
- goals, 6
- mailing lists for, 80
- managing software
projects, 3–6
- reasons for using, 1–14
- resources, 79–81
- roles in projects, 59–64

- sufficient development
time, 11
- sufficient resources, 11
- values, xi, 7–10
- variables, 3

XP Magazine (online), 79

Y

- “Yesterday’s Weather”
principle, 47
- “You Aren’t Gonna Need It”
rule, 17, 67

easy ●●●
computing