

# Contents

Emperor's New Code (a Story) .....	<i>xi</i>
Preface .....	<i>xv</i>
About the Authors .....	<i>xix</i>
Acknowledgments .....	<i>xxi</i>
Introduction .....	<i>xxiii</i>

## *Part I      Another Fine Mess You've Gotten Me Into               (Laurel and Hardy Take Up Programming)....1*

### **Chapter 1    XP in a Nuthouse               (Oops, We Mean Nutshell).....3**

Extreme Programming in Theory .....	4
What Problems Are Being Targeted by XP? .....	21
Extreme Programming in Practice: The Voice of eXPerience .....	23
Knocking It Down ... and Then Rebuilding It .....	26
Summary .....	28

### **Chapter 2    Where Did XP Come From? (Chrysler Knows               It Ain't Easy ...).....31**

Overview of C3 .....	33
The XP Project Life Cycle (As Illustrated by the Activities at C3) .....	34
Problems with C3 .....	53
Summary .....	56

### **Chapter 3    The Case Against XP .....57**

A Self-Referential Safety Net (Circle of Snakes) .....	57
Tailoring a Process to Local Conditions: Why XP Stands on Its Head .....	78
Summary .....	82

<i>Part II Social Aspects of XP (Mama Don't Let Your Coders Grow Up to Be Cowboys)</i> .....	83
<b>Chapter 4 Extremo Culture</b> .....	85
“XP Is Not About Mindless Hacking!” .....	86
XP Goes Mainstream .....	88
XP and the Dot-com Boom .....	92
XP As a People Process .....	94
XP Terminology .....	101
Big Words Like Constantinople and TerminationCanBeSuccess .....	103
Shooting the Messenger .....	104
Fear .....	108
Summary.....	116
<b>Chapter 5 The On-site Customer</b> .....	117
That's the Customer's Problem .....	118
On-site Customer: The Old Testament .....	121
On-site Customer: The New Testament .....	127
Summary.....	133
<b>Chapter 6 Pair Programming (Dear Uncle Joe, My Pair Programmer Has Halitosis)</b> .....	135
Pair Programming Basics .....	137
There's a Study That Proves My Point! .....	139
Wishing for the Sound of Silence.....	143
It's a Work of Love, Enforced by Coercive Means .....	144
Productivity: <code>numProgrammers/2 == numProgrammers?</code> Right? .....	146
Pair Programming Illuminated.....	154
Summary.....	160
<b>Chapter 7 Oral Documentation (Oxymoronic, or Just Plain Moronic?)</b> .....	161
“But I Thought You Said...” .....	162
Just Plain Moronic .....	170
Summary.....	179

<i>Part III</i>	<i>We Don't Write Permanent Specs and Barely Do Any Upfront Design, So . . . . .</i>	<i>181</i>
<b>Chapter 8</b>	<b>Design After First Testing.....</b>	<b>183</b>
When All You've Got Is a Hammer.....		184
XP Design Mantra: No BDUF.....		187
Problems with Unit Testing.....		188
Programming Without a Safety Net.....		197
Summary.....		199
<b>Chapter 9</b>	<b>Constant Refactoring After Programming (If It Ain't Broke, Fix It Anyway).....</b>	<b>201</b>
Refactoring Heaven.....		203
XP Design Mantra: Refactor Mercilessly.....		206
Is Up-Front Design Sufficient to Avoid Large Refactorings Later? .....		212
Refactoring with an Installed User Base.....		218
Summary.....		225
<b>Chapter 10</b>	<b>User Stories and Acceptance Tests.....</b>	<b>227</b>
Tell Me a Story, Daddy .....		229
User Stories vs. Use Cases.....		232
User Stories vs. Requirements .....		237
"Documenting" Requirements As Acceptance Tests .....		242
Summary.....		245
<i>Part IV</i>	<i>The Perpetual Coding Machine .....</i>	<i>247</i>
<b>Chapter 11</b>	<b>Software Is Never Done (The Schedule Does Not Exist Per Se).....</b>	<b>249</b>
The Schedule Doesn't Exist Per Se.....		250
Optional-Scope Contracts .....		260
Summary.....		267
<b>Chapter 12</b>	<b>Emergent Architecture and Design.....</b>	<b>269</b>
XP Design Mantra: YAGNI .....		274
Building an Infrastructure with Emergent Design.....		277

Emergent Architecture vs. Early Prototyping .....	289
Summary .....	291
 <b>Chapter 13 Embracing Change (Embrace People, Manage Change) .....</b>	 <b>293</b>
The Cost of Change Curve (aka the Cost to Fix Defects Curve) .....	295
Release Early, Release Often .....	297
Release Planning .....	300
Iteration Planning .....	301
The Perpetual Coding Machine (Embracing Change) .....	302
What Is This Change of Which You Speak? .....	307
Use Up-Front Design to Enhance Agility .....	307
Summary .....	310
 <b><i>Part V The Big Picture</i> .....</b>	 <b>311</b>
 <b>Chapter 14 Scalability .....</b>	 <b>313</b>
Painting Over the Cracks: XP on a 50-Person Project .....	314
Architectural Scalability .....	322
When XP Starts to Fail .....	328
Summary .....	335
 <b>Chapter 15 Refactoring XP .....</b>	 <b>337</b>
How to Be Agile Without Being Fragile .....	339
Extreme Programming Defanged: Taking the “Extreme” out of XP .....	343
Case Study: The Server Tools Project (Using a Defanged, Much Less Extreme but Still Very Agile Process) .....	362
Summary .....	368
 <b>Chapter 16 Conclusion: Neutralizing the Reality Distortion Field .....</b>	 <b>371</b>
Ethereal Wizardry in Action .....	372
At the End of the Day .....	379
In Closing .....	381
 <b>Index .....</b>	 <b>383</b>