Contents

Foreword xi

Introduction xiii

Acknowledgments xvii

About the Authors xix

1 **Models 1**

AntiPattern: Voyeuristic Models 2

Solution: Follow the Law of Demeter 3

Solution: Push All `find()` Calls into Finders on the Model 7

Solution: Keep Finders on Their Own Model 10

AntiPattern: Fat Models 14

Solution: Delegate Responsibility to New Classes 15

Solution: Make Use of Modules 21

Solution: Reduce the Size of Large Transaction Blocks 24

AntiPattern: Spaghetti SQL 31

Solution: Use Your Active Record Associations and Finders Effectively 32

Solution: Learn and Love the `Scope` Method 36

Solution: Use a Full-Text Search Engine 42

AntiPattern: Duplicate Code Duplication 50

Solution: Extract into Modules 50

Solution: Write a Plugin 59

Solution: Make Magic Happen with Metaprogramming 64

2 **Domain Modeling 73**

AntiPattern: Authorization Astronaut 74

Solution: Simplify with Simple Flags 76

AntiPattern: The Million-Model March 79

Solution: Denormalize into Text Fields 79

Solution: Make Use of Rails Serialization 82

3 Views 89

AntiPattern: PHPitis 91

Solution: Learn About the View Helpers That Come with Rails 92

Solution: Add Useful Accessors to Your Models 98

Solution: Extract into Custom Helpers 100

AntiPattern: Markup Mayhem 107

Solution: Make Use of the Rails Helpers 109

Solution: Use Haml 111

4 Controllers 117

AntiPattern: Homemade Keys 118

Solution: Use Clearance 119

Solution: Use Authlogic 121

AntiPattern: Fat Controller 123

Solution: Use Active Record Callbacks and Setters 123

Solution: Move to a Presenter 142

AntiPattern: Bloated Sessions 154

Solution: Store References Instead of Instances 154

AntiPattern: Monolithic Controllers 161

Solution: Embrace REST 161

AntiPattern: Controller of Many Faces 167

Solution: Refactor Non-RESTful Actions into a Separate Controller 167

AntiPattern: A Lost Child Controller 170

Solution: Make Use of Nested Resources 173

AntiPattern: Rat's Nest Resources 180

Solution: Use Separate Controllers for Each Nesting 181

AntiPattern: Evil Twin Controllers 184

Solution: Use Rails 3 Responders 186

5 Services 189

AntiPattern: Fire and Forget 190

Solution: Know What Exceptions to Look Out For 190

AntiPattern: Sluggish Services 195
Solution: Set Your Timeouts 195
Solution: Move the Task to the Background 195
AntiPattern: Pitiful Page Parsing 197
Solution: Use a Gem 198
AntiPattern: Successful Failure 201
Solution: Obey the HTTP Codes 203
AntiPattern: Kraken Code Base 207
Solution: Divide into Confederated Applications 207

6 Using Third-Party Code 211
AntiPattern: Recutting the Gem 213
Solution: Look for a Gem First 213
AntiPattern: Amateur Gemologist 214
Solution: Follow TAM 214
AntiPattern: Vendor Junk Drawer 216
Solution: Prune Irrelevant or Unused Gems 216
AntiPattern: Miscreant Modification 217
Solution: Consider Vendored Code Sacrosanct 217

7 Testing 221
AntiPattern: Fixture Blues 223
Solution: Make Use of Factories 225
Solution: Refactor into Contexts 228
AntiPattern: Lost in Isolation 236
Solution: Watch Your Integration Points 238
AntiPattern: Mock Suffocation 240
Solution: Tell, Don't Ask 241
AntiPattern: Untested Rake 246
Solution: Extract to a Class Method 248
AntiPattern: Unprotected Jewels 251
Solution: Write Normal Unit Tests Without Rails 251
Solution: Load Only the Parts of Rails You Need 254
Solution: Break Out the Atom Bomb 259

8 Scaling and Deploying 267

AntiPattern: Scaling Roadblocks 268

Solution: Build to Scale from the Start 268

AntiPattern: Disappearing Assets 271

Solution: Make Use of the System Directory 271

AntiPattern: Sluggish SQL 272

Solution: Add Indexes 272

Solution: Reassess Your Domain Model 277

AntiPattern: Painful Performance 282

Solution: Don't Do in Ruby What You Can Do in SQL 282

Solution: Move Processing into Background Jobs 286

9 Databases 291

AntiPattern: Messy Migrations 292

Solution: Never Modify the up Method on a Committed Migration 292

Solution: Never Use External Code in a Migration 293

Solution: Always Provide a down Method in Migrations 295

AntiPattern: Wet Validations 297

Solution: Eschew Constraints in the Database 298

10 Building for Failure 301

AntiPattern: Continual Catastrophe 302

Solution: Fail Fast 302

AntiPattern: Inaudible Failures 306

Solution: Never Fail Quietly 307

Index 311