

Contents

| | | |
|-----------|-------------------------------------|-------------|
| | Preface | xvii |
| Chapter 1 | Reengineering Patterns | 1 |
| | Why Do We Reengineer? | 1 |
| | What's Special about Objects? | 4 |
| | The Reengineering Life Cycle | 5 |
| | Reengineering Patterns | 9 |
| | The Form of a Reengineering Pattern | 11 |
| | A Map of Reengineering Patterns | 11 |

P A R T

I

Reverse Engineering **15**

| | | |
|-----------|--|-----------|
| Chapter 2 | Setting Direction | 17 |
| | Forces | 17 |
| | Overview | 18 |
| | Pattern 2.1 <i>Agree on Maxims</i> | 18 |
| | Pattern 2.2 <i>Appoint a Navigator</i> | 19 |
| | Pattern 2.3 <i>Speak to the Round Table</i> | 20 |
| | Pattern 2.4 <i>Most Valuable First</i> | 20 |
| | Pattern 2.5 <i>Fix Problems, Not Symptoms</i> | 23 |
| | Pattern 2.6 <i>If It Ain't Broke, Don't Fix It</i> | 24 |
| | Pattern 2.7 <i>Keep It Simple</i> | 24 |

| | | |
|------------------|---|-----------|
| Chapter 3 | First Contact | 27 |
| | Forces | 27 |
| | Overview | 29 |
| | What Next | 30 |
| | Pattern 3.1 <i>Chat with the Maintainers</i> | 31 |
| | Pattern 3.2 <i>Read All the Code in One Hour</i> | 38 |
| | Pattern 3.3 <i>Skim the Documentation</i> | 44 |
| | Pattern 3.4 <i>Interview during Demo</i> | 50 |
| | Pattern 3.5 <i>Do a Mock Installation</i> | 58 |
| | | |
| Chapter 4 | Initial Understanding | 65 |
| | Forces | 65 |
| | Overview | 67 |
| | What Next | 67 |
| | Pattern 4.1 <i>Analyze the Persistent Data</i> | 68 |
| | Pattern 4.2 <i>Speculate about Design</i> | 76 |
| | Pattern 4.3 <i>Study the Exceptional Entities</i> | 84 |
| | | |
| Chapter 5 | Detailed Model Capture | 95 |
| | Forces | 95 |
| | Overview | 96 |
| | What Next | 97 |
| | Pattern 5.1 <i>Tie Code and Questions</i> | 98 |
| | Pattern 5.2 <i>Refactor to Understand</i> | 103 |
| | Pattern 5.3 <i>Step through the Execution</i> | 107 |
| | Pattern 5.4 <i>Look for the Contracts</i> | 109 |
| | Pattern 5.5 <i>Learn from the Past</i> | 113 |

P A R T

II

Reengineering

119

| | | |
|------------------|--|------------|
| Chapter 6 | Tests: Your Life Insurance! | 121 |
| | Forces | 121 |
| | Overview | 122 |
| | Pattern 6.1 <i>Write Tests to Enable Evolution</i> | 123 |
| | Pattern 6.2 <i>Grow Your Test Base Incrementally</i> | 128 |

| | | |
|-------------------|--|------------|
| Pattern 6.3 | <i>Use a Testing Framework</i> | 130 |
| Pattern 6.4 | <i>Test the Interface, Not the Implementation</i> | 136 |
| Pattern 6.5 | <i>Record Business Rules as Tests</i> | 139 |
| Pattern 6.6 | <i>Write Tests to Understand</i> | 142 |
| Chapter 7 | Migration Strategies | 147 |
| | Forces | 147 |
| | Overview | 147 |
| Pattern 7.1 | <i>Involve the Users</i> | 149 |
| Pattern 7.2 | <i>Build Confidence</i> | 151 |
| Pattern 7.3 | <i>Migrate Systems Incrementally</i> | 153 |
| Pattern 7.4 | <i>Prototype the Target Solution</i> | 155 |
| Pattern 7.5 | <i>Always Have a Running Version</i> | 157 |
| Pattern 7.6 | <i>Regression Test after Every Change</i> | 159 |
| Pattern 7.7 | <i>Make a Bridge to the New Town</i> | 160 |
| Pattern 7.8 | <i>Present the Right Interface</i> | 163 |
| Pattern 7.9 | <i>Distinguish Public from Published Interface</i> | 164 |
| Pattern 7.10 | <i>Deprecate Obsolete Interfaces</i> | 166 |
| Pattern 7.11 | <i>Conserve Familiarity</i> | 168 |
| Pattern 7.12 | <i>Use Profiler before Optimizing</i> | 169 |
| Chapter 8 | Detecting Duplicated Code | 173 |
| | Forces | 173 |
| | Overview | 174 |
| Pattern 8.1 | <i>Compare Code Mechanically</i> | 175 |
| Pattern 8.2 | <i>Visualize Code as Dotplots</i> | 180 |
| Chapter 9 | Redistribute Responsibilities | 187 |
| | Forces | 187 |
| | Overview | 188 |
| Pattern 9.1 | <i>Move Behavior Close to Data</i> | 190 |
| Pattern 9.2 | <i>Eliminate Navigation Code</i> | 199 |
| Pattern 9.3 | <i>Split Up God Class</i> | 208 |
| Chapter 10 | Transform Conditionals to Polymorphism | 215 |
| | Forces | 215 |
| | Overview | 216 |
| Pattern 10.1 | <i>Transform Self Type Checks</i> | 217 |

| | |
|--|------------|
| Pattern 10.2 <i>Transform Client Type Checks</i> | 225 |
| Pattern 10.3 <i>Factor Out State</i> | 234 |
| Pattern 10.4 <i>Factor Out Strategy</i> | 237 |
| Pattern 10.5 <i>Introduce Null Object</i> | 240 |
| Pattern 10.6 <i>Transform Conditionals into Registration</i> | 243 |
| Appendix Thumbnail patterns | 253 |
| Testing Patterns | 253 |
| A.1 <i>Retest Persistent Problems</i> | 253 |
| A.2 <i>Test Fuzzy Features</i> | 254 |
| A.3 <i>Test Old Bugs</i> | 254 |
| Refactorings | 254 |
| A.4 <i>Encapsulate Field</i> | 254 |
| A.5 <i>Extract Method</i> | 255 |
| A.6 <i>Move Method</i> | 255 |
| A.7 <i>Rename Attribute</i> | 255 |
| A.8 <i>Rename Method</i> | 255 |
| A.9 <i>Replace Conditional with Polymorphism</i> | 255 |
| Design Patterns | 256 |
| A.10 <i>Abstract Factory</i> | 256 |
| A.11 <i>Adapter</i> | 256 |
| A.12 <i>Facade</i> | 256 |
| A.13 <i>Factory Method</i> | 256 |
| A.14 <i>Flyweight</i> | 256 |
| A.15 <i>Null Object</i> | 257 |
| A.16 <i>Quantity</i> | 257 |
| A.17 <i>Singleton</i> | 257 |
| A.18 <i>State</i> | 257 |
| A.19 <i>State Patterns</i> | 257 |
| A.20 <i>Strategy</i> | 258 |
| A.21 <i>Template Method</i> | 258 |
| A.22 <i>Visitor</i> | 258 |
| References | 259 |
| Index | 267 |