

# Contents

|                     |            |
|---------------------|------------|
| <b>Introduction</b> | <b>xxi</b> |
|---------------------|------------|

## **Part I: Introduction to Practical Enterprise Development**

|  |          |
|--|----------|
| <b>Chapter 1: What is Enterprise Design?</b> | <b>3</b> |
|--|----------|

---

|  |           |
|--|-----------|
| <b>What Is Enterprise Architecture?</b>          | <b>5</b>  |
| <b>What Is Enterprise Development?</b>           | <b>6</b>  |
| Reliability                                      | 6         |
| Flexibility                                      | 6         |
| Separation of Concerns                           | 7         |
| Reusability                                      | 7         |
| Maintainability                                  | 7         |
| <b>Where Is All of the Microsoft Enterprise?</b> | <b>9</b>  |
| The COM Factor                                   | 10        |
| The Shift to Java                                | 10        |
| The .NET Revolution                              | 11        |
| <b>Summary</b>                                   | <b>14</b> |

|                                       |           |
|---------------------------------------|-----------|
| <b>Chapter 2: The Enterprise Code</b> | <b>15</b> |
|---------------------------------------|-----------|

---

|                                  |           |
|----------------------------------|-----------|
| <b>A New Way to Look At Code</b> | <b>15</b> |
| Modularity                       | 16        |
| Loosely Coupled Classes          | 17        |
| Unit Testing                     | 23        |
| Inversion of Control Containers  | 25        |
| <b>Summary</b>                   | <b>29</b> |

## **Part II: The New Code — Changing the Way You Build**

|   |           |
|---|-----------|
| <b>Chapter 3: Emancipate Your Classes</b> | <b>33</b> |
|---|-----------|

---

|  |           |
|--|-----------|
| <b>Evaluating Your Code for Dependencies</b> | <b>33</b> |
| Rigidity                                     | 45        |
| Flexibility                                  | 45        |
| Separation of Concerns                       | 45        |

# Contents

---

|   |            |
|---|------------|
| Reusability   | 45         |
| Maintainability   | 45         |
| <b>Separation of Concerns and Identifying Modularity</b>                                | <b>46</b>  |
| <b>The Dependency Inversion Principle</b>   | <b>61</b>  |
| <b>Turning Classes Inside Out Using Dependency Injection</b>                            | <b>64</b>  |
| Rigidity  | 70         |
| Flexibility   | 70         |
| Separation of Concerns  | 70         |
| Reusability   | 70         |
| Maintainability   | 70         |
| <b>Summary</b>  | <b>72</b>  |
| <br>  |            |
| <b>Chapter 4: Test Driven Development</b>   | <b>73</b>  |
| <hr/>   |            |
| <b>Tic Tac Toe and Test Driven Development: An Example</b>                              | <b>74</b>  |
| Tic Tac Toe Requirements  | 75         |
| Testing Frameworks  | 110        |
| Identifying Testable Elements   | 110        |
| Writing Unit Tests That Work and Help   | 112        |
| <b>Refactoring</b>  | <b>116</b> |
| <b>Refactoring Tools</b>  | <b>117</b> |
| ReSharper   | 117        |
| Refactor Pro  | 117        |
| <b>Dealing with Dependencies in Test Driven Development — Mocking, Stubs, and Fakes</b> | <b>118</b> |
| <b>Mocking Frameworks</b>   | <b>127</b> |
| Rhino Mocks   | 128        |
| Moq   | 128        |
| NMock   | 128        |
| <b>Summary</b>  | <b>128</b> |
| <br>  |            |
| <b>Chapter 5: Make It Simple Again — Inversion of Control</b>                           | <b>131</b> |
| <hr/>   |            |
| <b>Creating Dependencies</b>  | <b>131</b> |
| <b>Factory Pattern</b>  | <b>138</b> |
| <b>Service Locator</b>  | <b>142</b> |
| <b>Inversion of Control and IoC Containers</b>  | <b>144</b> |
| <b>What's in a Name? Dependency Injection vs. Inversion of Control</b>                  | <b>146</b> |
| Choosing Your Container   | 146        |
| <b>StructureMap</b>   | <b>150</b> |
| Wire Up Using the Fluent Interface  | 151        |
| Wire Up Using Attributes — the Plugin Family  | 153        |
| Wire Up Using Configuration Meta Data   | 157        |

---

|   |            |
|---|------------|
| <b>To XML or Not to XML, That Is the Question</b> | <b>160</b> |
| Other IoC Frameworks                              | 160        |
| <b>Summary</b>                                    | <b>160</b> |

## **Part III: Enterprise Design Patterns**

---

### **Chapter 6: Getting to the Middle of Things** **165**

|                                      |            |
|--------------------------------------|------------|
| <b>On Middleware</b>                 | <b>165</b> |
| <b>The Wild West</b>                 | <b>166</b> |
| <b>Tiered Designs</b>                | <b>166</b> |
| <b>The Internet Age</b>              | <b>168</b> |
| <b>The Enterprise Middleware Age</b> | <b>169</b> |
| <b>A WCF Web Service</b>             | <b>172</b> |
| <b>The Messaging Model</b>           | <b>182</b> |
| <b>A Brief Note on SOA</b>           | <b>183</b> |
| <b>Summary</b>                       | <b>184</b> |

---

### **Chapter 7: Writing Your Own Middleware** **185**

|  |            |
|--|------------|
| <b>Business Logic Layer</b>                    | <b>185</b> |
| <b>Patterns for Your Business</b>              | <b>186</b> |
| Transaction Script                             | 186        |
| Active Record Pattern                          | 188        |
| Domain Model Pattern                           | 190        |
| Which Pattern to Use?                          | 194        |
| <b>Serving Your Business</b>                   | <b>195</b> |
| The Service Layer                              | 195        |
| Putting Patterns into Practice                 | 196        |
| Mortgage Loan Prequalification Application     | 197        |
| Talking the Language of the Domain             | 197        |
| <b>A Little Bit about Domain-Driven Design</b> | <b>198</b> |
| Entities                                       | 198        |
| Value Objects                                  | 199        |
| Aggregates and Aggregate Roots                 | 199        |
| Talking with the Domain Expert                 | 200        |
| Building the Domain Model                      | 202        |
| Identifying the Aggregates                     | 203        |
| Building the Application                       | 204        |
| Creating the Repositories                      | 242        |
| Creating the Domain Services                   | 244        |
| <b>Summary</b>                                 | <b>250</b> |

# Contents

---

|  |            |
|--|------------|
| <b>Chapter 8: “Mining” Your Own Business</b>         | <b>251</b> |
| <b>What Is the Data Access Layer?</b>                | <b>251</b> |
| Rolling Your Own Data Access Layer                   | 252        |
| Object Relation Mapping                              | 252        |
| The Data Context                                     | 253        |
| Entity Framework                                     | 269        |
| LinqToSQL or the Entity Framework?                   | 284        |
| Mapping the Mortgage Application with NHibernate     | 299        |
| <b>Summary</b>                                       | <b>318</b> |
| <b>Chapter 9: Organizing Your Front End</b>          | <b>321</b> |
| <b>The Neglected Front End</b>                       | <b>321</b> |
| <b>Early Front-End Patterns</b>                      | <b>322</b> |
| Java Struts  | 324        |
| ASP.NET  | 325        |
| Model-View-Presenter                                 | 328        |
| Back to MVC . . . the Rails Way                      | 332        |
| <b>Summary</b>                                       | <b>336</b> |
| <b>Chapter 10: Model-View-Presenter</b>              | <b>337</b> |
| <b>The MVP Pattern — Simplified</b>                  | <b>337</b> |
| The Model  | 338        |
| The View   | 338        |
| The Presenter  | 339        |
| The MVP Mortgage Calculator — Web Sample             | 339        |
| <b>Switching Platforms — Thick Client Sample</b>     | <b>360</b> |
| Setting Up the WPF Application                       | 361        |
| <b>Summary</b>                                       | <b>365</b> |
| <b>Chapter 11: The Model-View-Controller Pattern</b> | <b>367</b> |
| <b>Back to Basics</b>                                | <b>367</b> |
| Models   | 369        |
| Controllers  | 370        |
| Views  | 370        |
| The Mortgage Loan Application                        | 371        |
| The Model  | 374        |
| The Controller                                       | 376        |
| The View   | 378        |
| A Simple Repository                                  | 382        |

---

|  |            |
|--|------------|
| Creating and Editing                       | 386        |
| Flushing Out the Model                     | 401        |
| The Full Source Code                       | 413        |
| <b>Summary</b>                             | <b>413</b> |
| <b>Chapter 12: Putting It All Together</b> | <b>415</b> |

---

|  |            |
|--|------------|
| <b>One Step Back, Many Steps Forward</b>     | <b>415</b> |
| <b>The Concepts</b>                          | <b>416</b> |
| Reliability                                  | 416        |
| Flexibility                                  | 416        |
| Separation of Concerns                       | 416        |
| Reusability                                  | 416        |
| Maintainability                              | 416        |
| The Big Picture                              | 417        |
| <b>The Code</b>                              | <b>417</b> |
| The Big Picture                              | 418        |
| <b>The Patterns</b>                          | <b>418</b> |
| Middleware                                   | 419        |
| Persistence                                  | 420        |
| The User Interface                           | 421        |
| The Big Picture                              | 422        |
| <b>Being Holistic — The Big, BIG Picture</b> | <b>422</b> |
| The Mortgage Service                         | 423        |
| <b>Final Thoughts</b>                        | <b>423</b> |
| <b>Summary</b>                               | <b>424</b> |
| <b>Appendix A: C#.NET Primer</b>             | <b>425</b> |

---

|              |            |
|--------------|------------|
| <b>Index</b> | <b>455</b> |
|--------------|------------|

