

# Object-Oriented Software Engineering

Using UML, Patterns, and Java™

SECOND EDITION

Bernd  
BRUEGGE

Allen H.  
DUTOIT



# Table of Contents

|                        |               |
|------------------------|---------------|
| <b>Foreword</b>        | <b>v</b>      |
| <b>Preface</b>         | <b>vii</b>    |
| <b>Acknowledgments</b> | <b>xxxiii</b> |

---

|                                      |          |
|--------------------------------------|----------|
| <b><i>PART I</i> Getting Started</b> | <b>1</b> |
|--------------------------------------|----------|

---

|  |          |
|--|----------|
| <b><i>Chapter 1</i> Introduction to Software Engineering</b> | <b>3</b> |
| 1.1 Introduction: Software Engineering Failures              | 4        |
| 1.2 What Is Software Engineering?                            | 5        |
| 1.2.1 Modeling   | 6        |
| 1.2.2 Problem Solving  | 8        |
| 1.2.3 Knowledge Acquisition                                  | 8        |
| 1.2.4 Rationale  | 9        |
| 1.3 Software Engineering Concepts                            | 10       |
| 1.3.1 Participants and Roles                                 | 10       |
| 1.3.2 Systems and Models                                     | 12       |
| 1.3.3 Work Products  | 13       |
| 1.3.4 Activities, Tasks, and Resources                       | 13       |
| 1.3.5 Functional and Nonfunctional Requirements              | 14       |
| 1.3.6 Notations, Methods, and Methodologies                  | 15       |
| 1.4 Software Engineering Development Activities              | 16       |
| 1.4.1 Requirements Elicitation                               | 16       |

|                  |  |           |
|------------------|--|-----------|
| 1.4.2            | Analysis                                       | 16        |
| 1.4.3            | System Design                                  | 19        |
| 1.4.4            | Object Design                                  | 19        |
| 1.4.5            | Implementation                                 | 20        |
| 1.4.6            | Testing  | 20        |
| 1.5              | Managing Software Development                  | 21        |
| 1.5.1            | Communication                                  | 21        |
| 1.5.2            | Rationale Management                           | 22        |
| 1.5.3            | Software Configuration Management              | 22        |
| 1.5.4            | Project Management                             | 23        |
| 1.5.5            | Software Life Cycle                            | 23        |
| 1.5.6            | Putting It All Together                        | 23        |
| 1.6              | ARENA Case Study                               | 23        |
| 1.7              | Further Reading                                | 25        |
| 1.8              | Exercises                                      | 25        |
| <b>Chapter 2</b> | <b>Modeling with UML</b>                       | <b>29</b> |
| 2.1              | Introduction                                   | 30        |
| 2.2              | An Overview of UML                             | 31        |
| 2.2.1            | Use Case Diagrams                              | 31        |
| 2.2.2            | Class Diagrams                                 | 32        |
| 2.2.3            | Interaction Diagrams                           | 32        |
| 2.2.4            | Statechart Diagrams                            | 33        |
| 2.2.5            | Activity Diagrams                              | 33        |
| 2.3              | Modeling Concepts                              | 35        |
| 2.3.1            | Systems, Models, and Views                     | 35        |
| 2.3.2            | Data Types, Abstract Data Types, and Instances | 37        |
| 2.3.3            | Classes, Abstract Classes, and Objects         | 38        |
| 2.3.4            | Event Classes, Events, and Messages            | 40        |
| 2.3.5            | Object-Oriented Modeling                       | 41        |
| 2.3.6            | Falsification and Prototyping                  | 43        |
| 2.4              | A Deeper View into UML                         | 43        |
| 2.4.1            | Use Case Diagrams                              | 44        |
| 2.4.2            | Class Diagrams                                 | 51        |
| 2.4.3            | Interaction Diagrams                           | 59        |
| 2.4.4            | Statechart Diagrams                            | 62        |
| 2.4.5            | Activity Diagrams                              | 65        |
| 2.4.6            | Diagram Organization                           | 67        |

|  |   |            |
|--|---|------------|
| 2.4.7  | Diagram Extensions  | 69         |
| 2.5  | Further Readings  | 71         |
| 2.6  | Exercises   | 72         |
| <b>Chapter 3</b>                             | <b>Project Organization and Communication</b>                       | <b>77</b>  |
| 3.1  | Introduction: A Rocket Example                                      | 78         |
| 3.2  | An Overview of Projects   | 79         |
| 3.3  | Project Organization Concepts                                       | 83         |
| 3.3.1  | Project Organizations   | 83         |
| 3.3.2  | Roles   | 86         |
| 3.3.3  | Tasks and Work Products   | 88         |
| 3.3.4  | Schedule  | 90         |
| 3.4  | Project Communication Concepts                                      | 92         |
| 3.4.1  | Planned Communication   | 92         |
| 3.4.2  | Unplanned Communication   | 99         |
| 3.4.3  | Communication Mechanisms  | 102        |
| 3.5  | Organizational Activities   | 109        |
| 3.5.1  | Joining a Team  | 109        |
| 3.5.2  | Joining the Communication Infrastructure                            | 109        |
| 3.5.3  | Attending Team Status Meetings                                      | 110        |
| 3.5.4  | Organizing Client and Project Reviews                               | 113        |
| 3.6  | Further Readings  | 114        |
| 3.7  | Exercises   | 115        |
| <hr/> <b>PART II Dealing with Complexity</b> |   | <b>119</b> |
| <b>Chapter 4</b>                             | <b>Requirements Elicitation</b>                                     | <b>121</b> |
| 4.1  | Introduction: Usability Examples                                    | 122        |
| 4.2  | An Overview of Requirements Elicitation                             | 123        |
| 4.3  | Requirements Elicitation Concepts                                   | 125        |
| 4.3.1  | Functional Requirements   | 125        |
| 4.3.2  | Nonfunctional Requirements  | 126        |
| 4.3.3  | Completeness, Consistency, Clarity, and Correctness                 | 128        |
| 4.3.4  | Realism, Verifiability, and Traceability                            | 129        |
| 4.3.5  | Greenfield Engineering, Reengineering,<br>and Interface Engineering | 129        |

|                  |  |            |
|------------------|--|------------|
| 4.4              | Requirements Elicitation Activities                                  | 130        |
| 4.4.1            | Identifying Actors   | 130        |
| 4.4.2            | Identifying Scenarios  | 132        |
| 4.4.3            | Identifying Use Cases  | 135        |
| 4.4.4            | Refining Use Cases   | 138        |
| 4.4.5            | Identifying Relationships among Actors<br>and Use Cases              | 140        |
| 4.4.6            | Identifying Initial Analysis Objects                                 | 143        |
| 4.4.7            | Identifying Nonfunctional Requirements                               | 146        |
| 4.5              | Managing Requirements Elicitation                                    | 148        |
| 4.5.1            | Negotiating Specifications with Clients:<br>Joint Application Design | 148        |
| 4.5.2            | Maintaining Traceability   | 150        |
| 4.5.3            | Documenting Requirements Elicitation                                 | 151        |
| 4.6              | ARENA Case Study   | 153        |
| 4.6.1            | Initial Problem Statement  | 153        |
| 4.6.2            | Identifying Actors and Scenarios                                     | 155        |
| 4.6.3            | Identifying Use Cases  | 159        |
| 4.6.4            | Refining Use Cases and Identifying Relationships                     | 161        |
| 4.6.5            | Identifying Nonfunctional Requirements                               | 166        |
| 4.6.6            | Lessons Learned  | 168        |
| 4.7              | Further Readings   | 168        |
| 4.8              | Exercises  | 169        |
| <b>Chapter 5</b> | <b>Analysis</b>  | <b>173</b> |
| 5.1              | Introduction: An Optical Illusion                                    | 174        |
| 5.2              | An Overview of Analysis  | 174        |
| 5.3              | Analysis Concepts  | 176        |
| 5.3.1            | Analysis Object Models and Dynamic Models                            | 176        |
| 5.3.2            | Entity, Boundary, and Control Objects                                | 177        |
| 5.3.3            | Generalization and Specialization                                    | 178        |
| 5.4              | Analysis Activities: From Use Cases to Objects                       | 179        |
| 5.4.1            | Identifying Entity Objects   | 180        |
| 5.4.2            | Identifying Boundary Objects   | 182        |
| 5.4.3            | Identifying Control Objects  | 183        |
| 5.4.4            | Mapping Use Cases to Objects<br>with Sequence Diagrams               | 185        |

|        |   |     |
|--------|---|-----|
| 5.4.5  | Modeling Interactions among Objects with CRC Cards      | 189 |
| 5.4.6  | Identifying Associations                                | 190 |
| 5.4.7  | Identifying Aggregates                                  | 192 |
| 5.4.8  | Identifying Attributes                                  | 193 |
| 5.4.9  | Modeling State-Dependent Behavior of Individual Objects | 194 |
| 5.4.10 | Modeling Inheritance Relationships between Objects      | 195 |
| 5.4.11 | Reviewing the Analysis Model                            | 196 |
| 5.4.12 | Analysis Summary  | 197 |
| 5.5    | Managing Analysis                                       | 199 |
| 5.5.1  | Documenting Analysis                                    | 199 |
| 5.5.2  | Assigning Responsibilities                              | 200 |
| 5.5.3  | Communicating about Analysis                            | 201 |
| 5.5.4  | Iterating over the Analysis Model                       | 203 |
| 5.5.5  | Client Sign-Off   | 204 |
| 5.6    | ARENA Case Study  | 206 |
| 5.6.1  | Identifying Entity Objects                              | 206 |
| 5.6.2  | Identifying Boundary Objects                            | 211 |
| 5.6.3  | Identifying Control Objects                             | 212 |
| 5.6.4  | Modeling Interactions Among Objects                     | 212 |
| 5.6.5  | Reviewing and Consolidating the Analysis Model          | 213 |
| 5.6.6  | Lessons Learned   | 217 |
| 5.7    | Further Readings  | 218 |
| 5.8    | Exercises   | 219 |

|                  |  |            |
|------------------|--|------------|
| <b>Chapter 6</b> | <b>System Design:<br/>Decomposing the System</b> | <b>223</b> |
| 6.1              | Introduction: A Floor Plan Example               | 224        |
| 6.2              | An Overview of System Design                     | 226        |
| 6.3              | System Design Concepts                           | 228        |
| 6.3.1            | Subsystems and Classes                           | 228        |
| 6.3.2            | Services and Subsystem Interfaces                | 230        |
| 6.3.3            | Coupling and Cohesion                            | 230        |
| 6.3.4            | Layers and Partitions                            | 234        |
| 6.3.5            | Architectural Styles                             | 237        |

|                  |   |            |
|------------------|---|------------|
| 6.4              | System Design Activities: From Objects to Subsystems          | 246        |
| 6.4.1            | Starting Point: Analysis Model for<br>a Route Planning System | 246        |
| 6.4.2            | Identifying Design Goals                                      | 248        |
| 6.4.3            | Identifying Subsystems  | 252        |
| 6.5              | Further Readings  | 254        |
| 6.6              | Exercises   | 254        |
| <b>Chapter 7</b> | <b>System Design:<br/>Addressing Design Goals</b>             | <b>259</b> |
| 7.1              | Introduction: A Redundancy Example                            | 260        |
| 7.2              | An Overview of System Design Activities                       | 261        |
| 7.3              | Concepts: UML Deployment Diagrams                             | 262        |
| 7.4              | System Design Activities: Addressing Design Goals             | 264        |
| 7.4.1            | Mapping Subsystems to Processors<br>and Components            | 264        |
| 7.4.2            | Identifying and Storing Persistent Data                       | 265        |
| 7.4.3            | Providing Access Control                                      | 269        |
| 7.4.4            | Designing the Global Control Flow                             | 275        |
| 7.4.5            | Identifying Boundary Conditions                               | 277        |
| 7.4.6            | Reviewing System Design                                       | 279        |
| 7.5              | Managing System Design  | 282        |
| 7.5.1            | Documenting System Design                                     | 282        |
| 7.5.2            | Assigning Responsibilities                                    | 284        |
| 7.5.3            | Communicating about System Design                             | 285        |
| 7.5.4            | Iterating over the System Design                              | 287        |
| 7.6              | ARENA Case Study  | 288        |
| 7.6.1            | Identifying Design Goals                                      | 288        |
| 7.6.2            | Identifying Subsystems  | 289        |
| 7.6.3            | Mapping Subsystems to Processors<br>and Components            | 291        |
| 7.6.4            | Identifying and Storing Persistent Data                       | 292        |
| 7.6.5            | Providing Access Control                                      | 293        |
| 7.6.6            | Designing the Global Control Flow                             | 294        |
| 7.6.7            | Identifying Boundary Conditions                               | 296        |
| 7.6.8            | Lessons Learned   | 298        |
| 7.7              | Further Readings  | 298        |
| 7.8              | Exercises   | 299        |

|                  |   |            |
|------------------|---|------------|
| <b>Chapter 8</b> | <b>Object Design:<br/>Reusing Pattern Solutions</b>                   | <b>303</b> |
| 8.1              | Introduction: Bloopers  | 304        |
| 8.2              | An Overview of Object Design  | 305        |
| 8.3              | Reuse Concepts: Solution Objects, Inheritance, and<br>Design Patterns | 309        |
| 8.3.1            | Application Objects and Solution Objects                              | 309        |
| 8.3.2            | Specification Inheritance and Implementation Inheritance              | 309        |
| 8.3.3            | Delegation  | 312        |
| 8.3.4            | The Liskov Substitution Principle                                     | 313        |
| 8.3.5            | Delegation and Inheritance in Design Patterns                         | 313        |
| 8.4              | Reuse Activities: Selecting Design Patterns and<br>Components         | 316        |
| 8.4.1            | Encapsulating Data Stores with the Bridge Pattern                     | 317        |
| 8.4.2            | Encapsulating Legacy Components with<br>the Adapter Pattern           | 319        |
| 8.4.3            | Encapsulating Context with the Strategy Pattern                       | 321        |
| 8.4.4            | Encapsulating Platforms with the Abstract<br>Factory Pattern          | 322        |
| 8.4.5            | Encapsulating Control Flow with the Command<br>Pattern                | 325        |
| 8.4.6            | Encapsulating Hierarchies with the Composite<br>Design Pattern        | 326        |
| 8.4.7            | Heuristics for Selecting Design Patterns                              | 328        |
| 8.4.8            | Identifying and Adjusting Application Frameworks                      | 328        |
| 8.5              | Managing Reuse  | 333        |
| 8.5.1            | Documenting Reuse   | 334        |
| 8.5.2            | Assigning Responsibilities  | 336        |
| 8.6              | ARENA Case Study  | 337        |
| 8.6.1            | Applying the Abstract Factory Design Pattern                          | 337        |
| 8.6.2            | Applying the Command Design Pattern                                   | 339        |
| 8.6.3            | Applying the Observer Design Pattern                                  | 339        |
| 8.6.4            | Lessons Learned   | 340        |
| 8.7              | Further Readings  | 341        |
| 8.8              | Exercises   | 341        |



|                   |  |            |
|-------------------|--|------------|
| <b>Chapter 9</b>  | <b>Object Design:</b>  |            |
|                   | <b>Specifying Interfaces</b>   | <b>347</b> |
| 9.1               | Introduction: A Railroad Example   | 348        |
| 9.2               | An Overview of Interface Specification   | 349        |
| 9.3               | Interface Specification Concepts   | 350        |
|                   | 9.3.1 Class Implementor, Class Extender, and Class User                            | 351        |
|                   | 9.3.2 Types, Signatures, and Visibility  | 352        |
|                   | 9.3.3 Contracts: Invariants, Preconditions, and Postconditions                     | 354        |
|                   | 9.3.4 Object Constraint Language   | 355        |
|                   | 9.3.5 OCL Collections: Sets, Bags, and Sequences                                   | 359        |
|                   | 9.3.6 OCL Quantifiers: forAll and exists   | 363        |
| 9.4               | Interface Specification Activities   | 363        |
|                   | 9.4.1 Identifying Missing Attributes and Operations                                | 364        |
|                   | 9.4.2 Specifying Types, Signatures, and Visibility                                 | 366        |
|                   | 9.4.3 Specifying Pre- and Postconditions   | 367        |
|                   | 9.4.4 Specifying Invariants  | 369        |
|                   | 9.4.5 Inheriting Contracts   | 372        |
| 9.5               | Managing Object Design   | 373        |
|                   | 9.5.1 Documenting Object Design  | 373        |
|                   | 9.5.2 Assigning Responsibilities   | 379        |
|                   | 9.5.3 Using Contracts During Requirements Analysis                                 | 380        |
| 9.6               | ARENA Case Study   | 380        |
|                   | 9.6.1 Identifying Missing Operations in<br><i>TournamentStyle</i> and <i>Round</i> | 381        |
|                   | 9.6.2 Specifying the <i>TournamentStyle</i> and<br><i>Round</i> Contracts          | 382        |
|                   | 9.6.3 Specifying the <i>KnockOutStyle</i> and<br><i>KnockOutRound</i> Contracts    | 384        |
|                   | 9.6.4 Lessons Learned  | 385        |
| 9.7               | Further Readings   | 386        |
| 9.8               | Exercises  | 387        |
| <b>Chapter 10</b> | <b>Mapping Models to Code</b>  | <b>391</b> |
| 10.1              | Introduction: A Book Example   | 392        |
| 10.2              | An Overview of Mapping   | 393        |

|                   |  |            |
|-------------------|--|------------|
| 10.3              | Mapping Concepts                                     | 394        |
| 10.3.1            | Model Transformation                                 | 395        |
| 10.3.2            | Refactoring  | 396        |
| 10.3.3            | Forward Engineering                                  | 398        |
| 10.3.4            | Reverse Engineering                                  | 399        |
| 10.3.5            | Transformation Principles                            | 399        |
| 10.4              | Mapping Activities                                   | 400        |
| 10.4.1            | Optimizing the Object Design Model                   | 400        |
| 10.4.2            | Mapping Associations to Collections                  | 404        |
| 10.4.3            | Mapping Contracts to Exceptions                      | 410        |
| 10.4.4            | Mapping Object Models to a Persistent Storage Schema | 414        |
| 10.5              | Managing Implementation                              | 420        |
| 10.5.1            | Documenting Transformations                          | 420        |
| 10.5.2            | Assigning Responsibilities                           | 422        |
| 10.6              | ARENA Case Study                                     | 422        |
| 10.6.1            | ARENA Statistics                                     | 422        |
| 10.6.2            | Mapping Associations to Collections                  | 424        |
| 10.6.3            | Mapping Contracts to Exceptions                      | 426        |
| 10.6.4            | Mapping the Object Model to a Database Schema        | 428        |
| 10.6.5            | Lessons Learned                                      | 429        |
| 10.7              | Further Readings                                     | 430        |
| 10.8              | Exercises  | 430        |
| <b>Chapter 11</b> | <b>Testing</b>                                       | <b>435</b> |
| 11.1              | Introduction: Testing The Space Shuttle              | 436        |
| 11.2              | An Overview of Testing                               | 438        |
| 11.3              | Testing Concepts                                     | 442        |
| 11.3.1            | Faults, Erroneous States, and Failures               | 443        |
| 11.3.2            | Test Cases   | 446        |
| 11.3.3            | Test Stubs and Drivers                               | 448        |
| 11.3.4            | Corrections  | 449        |
| 11.4              | Testing Activities                                   | 449        |
| 11.4.1            | Component Inspection                                 | 450        |
| 11.4.2            | Usability Testing                                    | 451        |
| 11.4.3            | Unit Testing   | 453        |
| 11.4.4            | Integration Testing                                  | 463        |
| 11.4.5            | System Testing                                       | 469        |

|        |                            |     |
|--------|----------------------------|-----|
| 11.5   | Managing Testing           | 473 |
| 11.5.1 | Planning Testing           | 474 |
| 11.5.2 | Documenting Testing        | 474 |
| 11.5.3 | Assigning Responsibilities | 477 |
| 11.5.4 | Regression Testing         | 478 |
| 11.5.5 | Automating Testing         | 479 |
| 11.6   | Further Readings           | 481 |
| 11.7   | Exercises                  | 482 |

---

|                 |                        |            |
|-----------------|------------------------|------------|
| <b>PART III</b> | <b>Managing Change</b> | <b>485</b> |
|-----------------|------------------------|------------|

---

|                   |   |            |
|-------------------|---|------------|
| <b>Chapter 12</b> | <b>Rationale Management</b>                           | <b>487</b> |
| 12.1              | Introduction: Slicing Ham                             | 488        |
| 12.2              | An Overview of Rationale                              | 489        |
| 12.3              | Rationale Concepts                                    | 491        |
| 12.3.1            | Centralized Traffic Control                           | 492        |
| 12.3.2            | Defining the Problem: Issues                          | 493        |
| 12.3.3            | Exploring the Solution Space: Proposals               | 494        |
| 12.3.4            | Evaluating the Solution Space: Criteria and Arguments | 496        |
| 12.3.5            | Collapsing the Solution Space: Resolutions            | 498        |
| 12.3.6            | Implementing Resolutions: Action Items                | 498        |
| 12.3.7            | Examples of Issue-Based Models and Systems            | 499        |
| 12.4              | Rationale Activities: From Issues to Decisions        | 504        |
| 12.4.1            | CTC System Design                                     | 504        |
| 12.4.2            | Capturing Rationale in Meetings                       | 505        |
| 12.4.3            | Capturing Rationale Asynchronously                    | 513        |
| 12.4.4            | Capturing Rationale when Discussing Change            | 514        |
| 12.4.5            | Reconstructing Rationale                              | 518        |
| 12.5              | Managing Rationale                                    | 520        |
| 12.5.1            | Documenting Rationale                                 | 520        |
| 12.5.2            | Assigning Responsibilities                            | 522        |
| 12.5.3            | Heuristics for Communicating about Rationale          | 523        |
| 12.5.4            | Issue Modeling and Negotiation                        | 524        |
| 12.5.5            | Conflict Resolution Strategies                        | 525        |
| 12.6              | Further Readings                                      | 527        |
| 12.7              | Exercises   | 527        |

|                   |   |            |
|-------------------|---|------------|
| <b>Chapter 13</b> | <b>Configuration Management</b>                     | <b>531</b> |
| 13.1              | Introduction: An Aircraft Example                   | 532        |
| 13.2              | An Overview of Configuration Management             | 534        |
| 13.3              | Configuration Management Concepts                   | 535        |
| 13.3.1            | Configuration Items and CM Aggregates               | 536        |
| 13.3.2            | Versions and Configurations                         | 537        |
| 13.3.3            | Change Requests                                     | 538        |
| 13.3.4            | Promotions and Releases                             | 538        |
| 13.3.5            | Repositories and Workspaces                         | 539        |
| 13.3.6            | Version Identification Schemes                      | 539        |
| 13.3.7            | Changes and Change Sets                             | 542        |
| 13.3.8            | Configuration Management Tools                      | 542        |
| 13.4              | Configuration Management Activities                 | 544        |
| 13.4.1            | Configuration Item and CM Aggregate Identification  | 546        |
| 13.4.2            | Promotion Management                                | 548        |
| 13.4.3            | Release Management                                  | 549        |
| 13.4.4            | Branch Management                                   | 551        |
| 13.4.5            | Variant Management                                  | 556        |
| 13.4.6            | Change Management                                   | 559        |
| 13.5              | Managing Configuration Management                   | 560        |
| 13.5.1            | Documenting Configuration Management                | 561        |
| 13.5.2            | Assigning Configuration Management Responsibilities | 562        |
| 13.5.3            | Planning Configuration Management Activities        | 562        |
| 13.6              | Further Readings                                    | 563        |
| 13.7              | Exercises   | 564        |
| <b>Chapter 14</b> | <b>Project Management</b>                           | <b>567</b> |
| 14.1              | Introduction: The STS-51L Launch Decision           | 568        |
| 14.2              | An Overview of Project Management                   | 569        |
| 14.3              | Project Management Concepts                         | 576        |
| 14.3.1            | Tasks and Activities                                | 576        |
| 14.3.2            | Work Products, Work Packages, and Roles             | 577        |
| 14.3.3            | Work Breakdown Structure                            | 578        |
| 14.3.4            | Task Model  | 578        |
| 14.3.5            | Skill Matrix  | 580        |

|                   |   |            |
|-------------------|---|------------|
| 14.3.6            | Organizations   | 581        |
| 14.3.7            | Visualizing Organization Structures                       | 584        |
| 14.3.8            | The Spectrum of Organization Structures                   | 585        |
| 14.3.9            | The Software Project Management Plan                      | 586        |
| 14.4              | Project Management Activities                             | 589        |
| 14.4.1            | Planning the Project                                      | 589        |
| 14.4.2            | Organizing the Project                                    | 595        |
| 14.4.3            | Controlling the Project                                   | 600        |
| 14.4.4            | Terminating the Project                                   | 606        |
| 14.5              | Further Readings  | 608        |
| 14.6              | Exercises   | 609        |
| <b>Chapter 15</b> | <b>Software Life Cycle</b>                                | <b>613</b> |
| 15.1              | Introduction: Polynesian Navigation                       | 614        |
| 15.2              | IEEE 1074: Standard for Developing Life Cycle Processes   | 618        |
| 15.2.1            | Processes and Activities                                  | 618        |
| 15.2.2            | Life Cycle Modeling                                       | 620        |
| 15.2.3            | Project Management  | 620        |
| 15.2.4            | Pre-Development   | 621        |
| 15.2.5            | Development   | 622        |
| 15.2.6            | Post-Development  | 623        |
| 15.2.7            | Integral Processes (Cross-Development)                    | 624        |
| 15.3              | Characterizing the Maturity of Software Life Cycle Models | 625        |
| 15.4              | Life Cycle Models   | 628        |
| 15.4.1            | Sequential Activity-Centered Models                       | 629        |
| 15.4.2            | Iterative Activity-Centered Models                        | 631        |
| 15.4.3            | Entity-Centered Models                                    | 636        |
| 15.5              | Further Readings  | 639        |
| 15.6              | Exercises   | 640        |
| <b>Chapter 16</b> | <b>Methodologies:<br/>Putting It All Together</b>         | <b>643</b> |
| 16.1              | Introduction: The First Ascent of K2                      | 644        |
| 16.2              | Project Environment                                       | 647        |
| 16.3              | Methodology Issues  | 649        |
| 16.3.1            | How Much Planning?  | 649        |
| 16.3.2            | How Much Reuse?   | 650        |
| 16.3.3            | How Much Modeling?  | 651        |

|        |                                  |     |
|--------|----------------------------------|-----|
| 16.3.4 | How Much Process?                | 653 |
| 16.3.5 | How Much Control and Monitoring? | 653 |
| 16.3.6 | When to Redefine Project Goals?  | 654 |
| 16.4   | A Spectrum of Methodologies      | 654 |
| 16.4.1 | Royce's Methodology              | 655 |
| 16.4.2 | Extreme Programming              | 661 |
| 16.5   | Case Studies                     | 665 |
| 16.5.1 | XP Project: ATRACT               | 667 |
| 16.5.2 | Local King Client: FRIEND        | 670 |
| 16.5.3 | Distributed Project: JAMES       | 678 |
| 16.5.4 | Case Studies Summary             | 684 |
| 16.6   | Further Readings                 | 689 |
| 16.7   | Exercises                        | 689 |

---

**PART IV Appendices 693**

---

|                   |  |            |
|-------------------|--|------------|
| <b>Appendix A</b> | <b>Design Patterns</b>                         | <b>695</b> |
| A.1               | Abstract Factory: Encapsulating Platforms      | 696        |
| A.2               | Adapter: Wrapping Around Legacy Code           | 697        |
| A.3               | Bridge: Allowing for Alternate Implementations | 698        |
| A.4               | Command: Encapsulating Control Flow            | 699        |
| A.5               | Composite: Representing Recursive Hierarchies  | 700        |
| A.6               | Facade: Encapsulating Subsystems               | 701        |
| A.7               | Observer: Decoupling Entities from Views       | 702        |
| A.8               | Proxy: Encapsulating Expensive Objects         | 703        |
| A.9               | Strategy: Encapsulating Algorithms             | 704        |
| A.10              | Heuristics for Selecting Design Patterns       | 705        |
| <b>Appendix B</b> | <b>Glossary</b>                                | <b>707</b> |
| <b>Appendix C</b> | <b>Bibliography</b>                            | <b>739</b> |
|                   | <b>Index</b>                                   | <b>751</b> |