Contents

Preface xvii

Part I: The Unified Software Development Process 1

Chapter 1: The Unified Process: Use-Case Driven, Architecture-Centric, Iterative, and Incremental 3

1.1 The Unified Process in a Nutshell 4
1.2 The Unified Process Is Use-Case Driven 5
1.3 The Unified Process Is Architecture-Centric 6
1.4 The Unified Process Is Iterative and Incremental 7
1.5 The Life of the Unified Process 8
   1.5.1 The Product 9
   1.5.2 Phases within a Cycle 11
1.6 An Integrated Process 13
Chapter 2: The Four Ps: People, Project, Product, and Process in Software Development 15

2.1 People Are Crucial 16
   2.1.1 Development Processes Affect People 16
   2.1.2 Roles Will Change 17
   2.1.3 Turning "Resources" into "Workers" 18

2.2 Projects Make the Product 19

2.3 Product Is More Than Code 20
   2.3.1 What Is a Software System? 20
   2.3.2 Artifacts 21
   2.3.3 A System Has a Collection of Models 21
   2.3.4 What Is a Model? 22
   2.3.5 Each Model Is a Self-Contained View of the System 22
   2.3.6 Inside a Model 23
   2.3.7 Relationships between Models 23

2.4 Process Directs Projects 24
   2.4.1 Process: A Template 24
   2.4.2 Related Activities Make Up Workflows 25
   2.4.3 Specializing Process 26
   2.4.4 Merits of Process 27

2.5 Tools Are Integral to Process 28
   2.5.1 Tools Impact Process 28
   2.5.2 Process Drives Tools 28
   2.5.3 Balance Process and Tools 29
   2.5.4 Visual Modeling Supports UML 29
   2.5.5 Tools Support the Whole Life Cycle 30

2.6 References 31

Chapter 3: A Use-Case–Driven Process 33

3.1 Use-Case–Driven Development in Brief 35

3.2 Why Use Cases? 37
   3.2.1 To Capture the Value Adding Requirements 37
   3.2.2 To Drive the Process 38
   3.2.3 To Devise the Architecture and More... 39

3.3 Capturing the Use Cases 40
   3.3.1 The Use-Case Model Represents the Functional Requirements 40
   3.3.2 Actors Are the Environment of the System 41
   3.3.3 Use Cases Specify the System 41

3.4 Analysis, Design, and Implementation to Realize the Use Cases 42
   3.4.1 Creating the Analysis Model from the Use Cases 43
   3.4.2 Each Class Must Fulfill All Its Collaboration Roles 48
5.3 The Iterative Approach is Risk-Driven 94
5.3.1 Iterations Alleviate Technical Risks 95
5.3.2 Management Is Responsible for Nontechnical Risks 97
5.3.3 Dealing with Risks 97
5.4 The Generic Iteration 98
5.4.1 What an Iteration Is 98
5.4.2 Planning the Iterations 100
5.4.3 Sequencing the Iterations 100
5.5 The Result of an Iteration Is an Increment 101
5.6 Iterations over the Life Cycle 102
5.7 Models Evolve from Iterations 105
5.8 Iterations Challenge the Organization 106
5.9 References 106

Part II: The Core Workflows 109

Chapter 6: Requirements Capture: From Vision to Requirements 111
6.1 Why Requirements Capture Is Difficult 112
6.2 The Purpose of the Requirements Workflow 113
6.3 Overview of Requirements Capture 113
6.4 The Role of Requirements in the Software Life Cycle 118
6.5 Understanding the System Context Using a Domain Model 119
6.5.1 What Is a Domain Model? 119
6.5.2 Developing a Domain Model 121
6.5.3 Use of the Domain Model 121
6.6 Understanding the System Context Using a Business Model 122
6.6.1 What Is a Business Model? 122
6.6.2 How to Develop a Business Model 124
6.6.3 Find Use Cases from a Business Model 126
6.7 Supplementary Requirements 128
6.8 Summary 130
6.9 References 130

Chapter 7: Capturing the Requirements as Use Cases 131
7.1 Introduction 131
7.2 Artifacts 133
7.2.1 Artifact: Use-Case Model 133
7.2.2 Artifact: Actor 134
7.2.3 Use Case 135
7.2.4 Artifact: Architecture Description (View of the Use-Case Model) 139
Chapter 8: Analysis 173

8.1 Introduction 173

8.2 Analysis in Brief 176
8.2.1 Why Analysis Is not Design or Implementation 176
8.2.2 The Purpose of Analysis: Summary 177
8.2.3 Concrete Examples of When to Employ Analysis 178

8.3 The Role of Analysis in the Software Life Cycle 179

8.4 Artifacts 181
8.4.1 Artifact: Analysis Model 181
8.4.2 Artifact: Analysis Class 181
8.4.3 Artifact: Use-Case Realization—Analysis 186
8.4.4 Artifact: Analysis Package 190
8.4.5 Artifact: Architecture Description (View of the Analysis Model) 193

8.5 Workers 194
8.5.1 Worker: Architect 194
8.5.2 Worker: Use-Case Engineer 194
8.5.3 Worker: Component Engineer 195

8.6 Workflow 196
8.6.1 Activity: Architectural Analysis 196
8.6.2 Activity: Analyze a Use Case 203
8.6.3 Activity: Analyze a Class 207
8.6.4 Activity: Analyze a Package 211

8.7 Summary of Analysis 213

8.8 References 214
Chapter 9: Design  215

9.1 Introduction  215
9.2 The Role of Design in the Software Life Cycle  216
9.3 Artifacts  217
  9.3.1 Artifact: Design Model  217
  9.3.2 Artifact: Design Class  218
  9.3.3 Artifact: Use-Case Realization—Design  221
  9.3.4 Artifact: Design Subsystem  224
  9.3.5 Artifact: Interface  226
  9.3.6 Artifact: Architecture Description (View of the Design Model)  226
  9.3.7 Artifact: Deployment Model  227
  9.3.8 Artifact: Architecture Description (View of the Deployment Model)  228
9.4 Workers  229
  9.4.1 Worker: Architect  229
  9.4.2 Worker: Use-Case Engineer  230
  9.4.3 Worker: Component Engineer  230
9.5 Workflow  231
  9.5.1 Activity: Architectural Design  232
  9.5.2 Activity: Design a Use Case  249
  9.5.3 Activity: Design a Class  255
  9.5.4 Activity: Design a Subsystem  263
9.6 Summary of Design  265
9.7 References  266

Chapter 10: Implementation  267

10.1 Introduction  267
10.2 The Role of Implementation in the Software Life Cycle  268
10.3 Artifacts  269
  10.3.1 Artifact: Implementation Model  269
  10.3.2 Artifact: Component  269
  10.3.3 Artifact: Implementation Subsystem  272
  10.3.4 Artifact: Interface  274
  10.3.5 Artifact: Architecture Description (View of the Implementation Model)  275
  10.3.6 Artifact: Integration Build Plan  276
10.4 Workers  277
  10.4.1 Worker: Architect  277
  10.4.2 Worker: Component Engineer  277
  10.4.3 Worker: System Integrator  279
10.5 Workflow  279
  10.5.1 Activity: Architectural Implementation  280
  10.5.2 Activity: Integrate System  283
Chapter 11: Test 295

11.1 Introduction 295
11.2 The Role of Testing in the Software Life Cycle 296
11.3 Artifacts 297
   11.3.1 Artifact: Test Model 297
   11.3.2 Artifact: Test Case 297
   11.3.3 Artifact: Test Procedure 300
   11.3.4 Artifact: Test Component 302
   11.3.5 Artifact: Plan Test 302
   11.3.6 Artifact: Defect 302
   11.3.7 Artifact: Evaluate Test 302
11.4 Workers 303
   11.4.1 Worker: Test Designer 303
   11.4.2 Worker: Component Engineer 303
   11.4.3 Worker: Integration Tester 303
   11.4.4 Worker: System Tester 304
11.5 Workflow 304
   11.5.1 Activity: Plan Test 305
   11.5.2 Activity: Design Test 306
   11.5.3 Activity: Implement Test 309
   11.5.4 Activity: Perform Integration Test 310
   11.5.5 Activity: Perform System Test 311
   11.5.6 Activity: Evaluate Test 311
11.6 Summary of Testing 313
11.7 References 313

Part III: Iterative and Incremental Development 315

Chapter 12: The Generic Iteration Workflow 317

12.1 The Need for Balance 318
12.2 The Phases Are the First Division of Work 319
   12.2.1 Inception Phase Establishes Feasibility 319
   12.2.2 Elaboration Phase Focuses on “Do-Ability” 320
   12.2.3 Construction Phase Builds the System 321
   12.2.4 Transition Phase Moves into the User Environment 322
12.3 The Generic Iteration Revisited 322
   12.3.1 Core Workflows Repeat in Each Iteration 322
   12.3.2 Workers Participate in the Workflows 323

12.4 Planning Precedes Doing 324
   12.4.1 Plan the Four Phases 325
   12.4.2 Plan the Iterations 326
   12.4.3 Think Long Term 327
   12.4.4 Plan the Evaluation Criteria 327

12.5 Risks Affect Project Planning 328
   12.5.1 Manage a Risk List 328
   12.5.2 Risks Affect the Iteration Plan 329
   12.5.3 Schedule Risk Action 329

12.6 Use-Case Prioritization 330
   12.6.1 Risks Specific to a Particular Product 331
   12.6.2 Risk of Not Getting the Architecture Right 331
   12.6.3 Risk of Not Getting Requirements Right 332

12.7 Resources Needed 333
   12.7.1 Projects Differ Widely 334
   12.7.2 A Typical Project Looks Like This 335
   12.7.3 Complex Projects Have Greater Needs 335
   12.7.4 New Product Line Calls for Experience 336
   12.7.5 Paying the Cost of the Resources Used 337

12.8 Assess the Iterations and Phases 338
   12.8.1 Criteria Not Achieved 338
   12.8.2 The Criteria Themselves 339
   12.8.3 The Next Iteration 339
   12.8.4 Evolution of the Model Set 340

Chapter 13: Inception Launches the Project 341

13.1 The Inception Phase in Brief 341

13.2 Early in the Inception Phase 342
   13.2.1 Before the Inception Phase Begins 342
   13.2.2 Planning the Inception Phase 343
   13.2.3 Expanding the System Vision 344
   13.2.4 Setting the Evaluation Criteria 344

13.3 The Archetypal Inception Iteration Workflow 346
   13.3.1 Introduction to the Five Core Workflows 346
   13.3.2 Fitting the Project into the Development Environment 348
   13.3.3 Finding Critical Risks 348

13.4 Execute the Core Workflows, Requirements to Test 348
   13.4.1 Capture the Requirements 350
   13.4.2 Analysis 352
13.4.3 Design 353
13.4.5 Test 354

13.5 Make the Initial Business Case 354
13.5.1 Outline Business Bid 354
13.5.2 Estimate Return on Investment 356

13.6 Assess the Iteration(s) in the Inception Phase 356

13.7 Planning the Elaboration Phase 357

13.8 The Deliverables for the Inception Phase 358

Chapter 14: The Elaboration Phase Makes the Architectural Baseline 359

14.1 The Elaboration Phase in Brief 359

14.2 Early in the Elaboration Phase 360
14.2.1 Planning the Elaboration Phase 361
14.2.2 Building the Team 361
14.2.3 Modifying the Development Environment 361
14.2.4 Setting Evaluation Criteria 361

14.3 The Archetypal Elaboration Iteration Workflow 362
14.3.1 Capture and Refine Most of the Requirements 363
14.3.2 Develop the Architectural Baseline 364
14.3.3 Iterate While the Team Is Small 364

14.4 Execute the Core Workflows—Requirements to Test 364
14.4.1 Capture the Requirements 365
14.4.2 Analysis 367
14.4.3 Design 372
14.4.4 Implementation 374
14.4.5 Test 376

14.5 Make the Business Case 377
14.5.1 Prepare the Business Bid 378
14.5.2 Update Return on Investment 378

14.6 Assess the Iterations in the Elaboration Phase 378

14.7 Planning the Construction Phase 379

14.8 The Key Deliverables 380

Chapter 15: Construction Leads to Initial Operational Capability 381

15.1 The Construction Phase in Brief 382

15.2 Early in the Construction Phase 382
15.2.1 Staffing the Phase 383
15.2.2 Setting the Evaluation Criteria 383

15.3 The Archetypal Construction Iteration Workflow 384
15.4 Execute the Core Workflows—Requirements to Testing  385
  15.4.1 Requirements  387
  15.4.2 Analysis  388
  15.4.3 Design  389
  15.4.4 Implementation  390
  15.4.5 Test  391
15.5 Controlling the Business Case  393
15.6 Assess the Iterations and the Construction Phase  393
15.7 Planning the Transition Phase  393
15.8 The Key Deliverables  394

Chapter 16: Transition Completes Product Release  395
  16.1 The Transition Phase in Brief  396
  16.2 Early in the Transition Phase  397
    16.2.1 Planning the Transition Phase  397
    16.2.2 Staffing the Transition Phase  399
    16.2.3 Setting the Evaluation Criteria  399
  16.3 The Core Workflows Play a Small Role in this Phase  400
  16.4 What We Do in the Transition Phase  401
    16.4.1 Getting the Beta Release Out  401
    16.4.2 Installing the Beta Release  402
    16.4.3 Responding to the Test Results  402
    16.4.4 Adapting the Product to Varied User Environments  403
    16.4.5 Completing the Artifacts  404
    16.4.6 When Does the Project End?  404
  16.5 Completing the Business Case  405
    16.5.1 Controlling Progress  405
    16.5.2 Review of the Business Plan  405
  16.6 Assess the Transition Phase  406
    16.6.1 Assess the Iterations and the Phase  406
    16.6.2 Postmortem of the Project  407
  16.7 Planning the Next Release or Generation  407
  16.8 The Key Deliverables  407

Chapter 17: Making the Unified Process Work  409
  17.1 The Unified Process Helps You Deal with Complexity  409
    17.1.1 The Life Cycle Objectives  410
    17.1.2 The Life Cycle Architecture  410
    17.1.3 Initial Operational Capability  411
    17.1.4 Product Release  411
17.2 The Major Themes 411
17.3 Management Leads Conversion to Unified Process 412
  17.3.1 The Case for Action 413
  17.3.2 The Reengineering Directive Persuades 413
  17.3.3 Implementing the Transition 414
17.4 Specializing the Unified Process 416
  17.4.1 Tailoring the Process 416
  17.4.2 Filling in the Process Framework 417
17.5 Relate to the Broader Community 418
17.6 Get the Benefits of the Unified Process 418
17.7 References 419

Appendix A: Overview of the UML 421
A.1 Introduction 421
  A.1.1 Vocabulary 422
  A.1.2 Extensibility Mechanisms 422
A.2 Graphical Notation 423
  A.2.1 Structural Things 423
  A.2.2 Behavioral Things 424
  A.2.3 Grouping Things 425
  A.2.4 Annotational Things 425
  A.2.5 Dependency Relationships 425
  A.2.6 Association Relationships 425
  A.2.7 Generalization Relationships 426
  A.2.8 Extensibility Mechanisms 426
A.3 Glossary of Terms 426
A.4 References 433

Appendix B: The Unified Process-Specific Extensions of the UML 435
B.1 Introduction 435
B.2 Stereotypes 435
B.3 Tagged Values 438
B.4 Graphical Notation 439
B.5 References 439

Appendix C: General Glossary 441
C.1 Introduction 441
C.2 Terms 441

Index 451