The Unified Software Development Process

Ivar Jacobson
Grady Booch
James Rumbaugh

Rational Software Corporation
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
Chapter 4: An Architecture-Centric Process 59

4.1 Architecture in Brief 60

4.2 Why We Need Architecture 62
4.2.1 Understanding the System 62
4.2.2 Organizing Development 63
4.2.3 Fostering Reuse 63
4.2.4 Evolving the System 64

4.3 Use Cases and Architecture 65

4.4 The Steps to an Architecture 69
4.4.1 The Architecture Baseline Is a “Small, Skinny” System 69
4.4.2 Using Architecture Patterns 71
4.4.3 Describing Architecture 74
4.4.4 The Architect Creates the Architecture 76

4.5 Finally, an Architecture Description! 77
4.5.1 The Architectural View of the Use-Case Model 78
4.5.2 The Architectural View of the Design Model 78
4.5.3 The Architectural View of the Deployment Model 81
4.5.4 The Architectural View of the Implementation Model 83

4.6 Three Interesting Concepts 83
4.6.1 What Is Architecture? 83
4.6.2 How Is It Obtained? 83
4.6.3 How Is It Described? 83

4.7 References 84

Chapter 5: An Iterative and Incremental Process 85

5.1 Iterative and Incremental in Brief 86
5.1.1 Develop in Small Steps 87
5.1.2 What Iteration Is Not 88

5.2 Why Iterative and Incremental Development? 89
5.2.1 Mitigating Risks 89
5.2.2 Getting a Robust Architecture 91
5.2.3 Handling Changing Requirements 91
5.2.4 Allowing for Tactical Changes 92
5.2.5 Achieving Continuous Integration 92
5.2.6 Attaining Early Learning 93
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.2.5</td>
<td>139</td>
</tr>
<tr>
<td>7.2.6</td>
<td>140</td>
</tr>
<tr>
<td><strong>7.3 Workers</strong></td>
<td><strong>140</strong></td>
</tr>
<tr>
<td>7.3.1</td>
<td>140</td>
</tr>
<tr>
<td>7.3.2</td>
<td>141</td>
</tr>
<tr>
<td>7.3.3</td>
<td>142</td>
</tr>
<tr>
<td>7.3.4</td>
<td>142</td>
</tr>
<tr>
<td><strong>7.4 Workflow</strong></td>
<td><strong>143</strong></td>
</tr>
<tr>
<td>7.4.1</td>
<td>144</td>
</tr>
<tr>
<td>7.4.2</td>
<td>153</td>
</tr>
<tr>
<td>7.4.3</td>
<td>153</td>
</tr>
<tr>
<td>7.4.4</td>
<td>160</td>
</tr>
<tr>
<td>7.4.5</td>
<td>166</td>
</tr>
<tr>
<td><strong>7.5 Summary of the Requirements Workflow</strong></td>
<td><strong>171</strong></td>
</tr>
<tr>
<td><strong>7.6 References</strong></td>
<td><strong>172</strong></td>
</tr>
<tr>
<td><strong>Chapter 8: Analysis</strong></td>
<td><strong>173</strong></td>
</tr>
<tr>
<td><strong>8.1 Introduction</strong></td>
<td><strong>173</strong></td>
</tr>
<tr>
<td><strong>8.2 Analysis in Brief</strong></td>
<td><strong>176</strong></td>
</tr>
<tr>
<td>8.2.1</td>
<td>176</td>
</tr>
<tr>
<td>8.2.2</td>
<td>177</td>
</tr>
<tr>
<td>8.2.3</td>
<td>178</td>
</tr>
<tr>
<td><strong>8.3 The Role of Analysis in the Software Life Cycle</strong></td>
<td><strong>179</strong></td>
</tr>
<tr>
<td><strong>8.4 Artifacts</strong></td>
<td><strong>181</strong></td>
</tr>
<tr>
<td>8.4.1</td>
<td>181</td>
</tr>
<tr>
<td>8.4.2</td>
<td>181</td>
</tr>
<tr>
<td>8.4.3</td>
<td>186</td>
</tr>
<tr>
<td>8.4.4</td>
<td>190</td>
</tr>
<tr>
<td>8.4.5</td>
<td>193</td>
</tr>
<tr>
<td><strong>8.5 Workers</strong></td>
<td><strong>194</strong></td>
</tr>
<tr>
<td>8.5.1</td>
<td>194</td>
</tr>
<tr>
<td>8.5.2</td>
<td>194</td>
</tr>
<tr>
<td>8.5.3</td>
<td>195</td>
</tr>
<tr>
<td><strong>8.6 Workflow</strong></td>
<td><strong>196</strong></td>
</tr>
<tr>
<td>8.6.1</td>
<td>196</td>
</tr>
<tr>
<td>8.6.2</td>
<td>203</td>
</tr>
<tr>
<td>8.6.3</td>
<td>207</td>
</tr>
<tr>
<td>8.6.4</td>
<td>211</td>
</tr>
<tr>
<td><strong>8.7 Summary of Analysis</strong></td>
<td><strong>213</strong></td>
</tr>
<tr>
<td><strong>8.8 References</strong></td>
<td><strong>214</strong></td>
</tr>
</tbody>
</table>
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
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
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