Preface v

Chapter 1 Introduction to Principles, Maxims, and Decision Frameworks 1
  Some Software Engineering History 2
  Background on Object-Oriented Technology 6
  Today’s Object-Oriented Perspective 8
  Basic Principles and Maxims 9
    The Principle of System Architecture 10
    The Principle of Decision Making 10
    Management Communication 11
  Project Management Decision Frameworks 11
    Establish the Goals for the Use of Object-Oriented Technology 11
    Use Object-Oriented Technology in Software Development 12
  Background on Case Studies 13
  Summary 15
  Additional Reading About the History of Software Engineering 16

Chapter 2 Establish Project Goals and Objectives 19
  Goals and Objectives 19
  Projects—Transitioning from Current to Desired Situation 20
  Framework for Defining Project Goals and Objectives 21
    Understand the Business Mission and Business Processes 23
    Understand Business and Team Values for Software Development 26
    State the Desired Situation in Terms of Goals and Objectives for
    Products, Processes, and Resources 27
    Assess the Current Situation 29
  Assessment of the KandA Widget Company—An Example 31
  Case Study Results 33
  Summary 36
  Additional Reading About Project Goals and Objectives 37
Chapter 3  Determining Benefits of Object-Oriented Technology  39

What Is an Object?  39
A Class Describes a Set of Objects  42
Objects Form Applications, Frameworks, and Components  43
What Does It Mean to Be Object-Oriented?  45
What About Objects Helps Build Systems?  49
Objects Separate Interface from Implementation  50
Object Models Closely Map the Real World  50
Objects Come in Different Sizes  51
Objects Live All the Way Down  51
Additional Benefits of Objects  52
Objects Are a Means of Attacking Complexity  52
Objects Build Systems Resilient to Change  53
Objects Allow Partial Systems to Work  54
Objects Represent Natural Units for Reuse  55
Framework for Determining How Projects Benefit from Using Object-Oriented Technology  56
  Determine Whether and How Object-Oriented Technology Contributes
to Obtaining What You Value in Software Development  56
  Determine Projects Needed to Achieve Valued Objectives Through the
    Use of Object-Oriented Technology  58
Case Study Results  58
Summary  60
Additional Reading About Objects  61

Chapter 4  Make an Initial Commitment to Use Object-Oriented Technology  63

Framework for Making an Initial Commitment to Objects  64
Receive an Initial Introduction to Objects  64
Obtain an Initial Management Commitment to the Use of Objects  66
  Make the Business Case  66
  Manage the Expectations of Management  68
Select an Initial Process Model and Software Development Environment  70
Select and Set Up an Initial Pilot Project  71
  Guidelines for Choosing a Pilot Project to Develop a Software System  71
  Mechanics for Setting Up the Initial Pilot Project  73
Decide Who Will Staff the First Project  74
Decide on the Training Plan  75
Chapter 5  **What Is a Process Model?**  85
   Process Model Definitions  85
   Example Process Models  87
      Waterfall Model  87
      Spiral Model  89
      Recursive/Parallel Model  90
   There Is Not One Process Model for Objects  91
   Strategies for Developing with Objects  94
      Iterative Development  94
      Incremental Development  95
      Prototyping  96
      Consuming Reusable Assets  102
      Producing Reusable Assets  104
   Case Study Results  105
      Product Process Models  105
      Reusing  106
      Prototyping  108
      Illustration 1: Time-Driven Process Model  110
      Illustration 2: Customer Information System  111
      Illustration 3: CASE Tools  112
   Summary  114
   Additional Reading About Process Models and Prototyping  115

Chapter 6  **Select a Product Process Model**  117
   Additional Strategies for Developing with Objects  117
      Partitioning  117
      Integrating and Managing Changes  121
      Documenting  122
Quality Assurance Strategies When Developing with Objects 124
How Quality Relates to Other Development Strategies 124
Uncovering Defects During Development 125
Testing Object Implementations 127
Recording Defects 132
Framework for Selecting a Product Process Model 134
Agree on Maxims 135
Verify Software Development Goals and Objectives 136
Select Project Strategies, Activities, and Their Ordering 136
Select Methods for Each Activity 137
Case Study Results 137
Product Process Models 137
Example Partitions 138
Summary 141
Additional Reading About Product Process Models That Use Object-Oriented Technology 142

Chapter 7 Plan and Control a Project 143
What Is a Project Schedule? 144
Planning Under Uncertainty 147
State Clearly What You Know and What You Do Not Know 148
State Clearly What You Will Do to Eliminate Unknowns 149
Make Sure That All Early Milestones Can Be Met 149
Plan to Replan 150
Planning Resources 150
Framework for Planning and Controlling 151
Identify Required Milestones 152
Identify System Capabilities 153
Identify Tasks 154
Estimate the Cost of Each Task 157
Account for Costs 160
Monitor and Control Project Execution 161
Summary 166
Additional Reading About Planning and Controlling Projects 168
Chapter 8  **Case Studies of Process Models**  169
  Milestone-Driven Process Model  169
      Illustration 1: Measuring Progress  170
      Illustration 2: Documenting Progress  174
  Security-Driven Process Model  180
      Illustration 3: Control Based on Need to Know  180
  Just-Do-It Process Model  182
      Illustration 4: No Formal Design  182
  Creating a Reusable Asset Process Model  185
      Illustration 5: Create a Reusable Framework  185
  Concept Development Process Model  190
      Illustration 6: Available Prototypes  190
      Illustration 7: Prototyping Partnership  196
  Summary  200
  Additional Reading About Case Study Experiences  201

Chapter 9  **What Is Reuse?**  203
  Why the Hype About Reuse?  203
  Reuse in Software Development  207
      Cloning Is Not Strategic Reuse  209
      Reuse with Objects  210
  The Value of Reuse  212
      The Reuse Producer/Consumer Equation  212
      Reuse Claims  214
  Case Study Results  216
      Illustration 1: Reusable Assets Need Support  217
      Illustration 2: Reusing a Binary Asset  218
  Summary  219
  Additional Reading About Reuse Definitions and Expectations  220

Chapter 10  **Reuse Process Models**  221
  Framework for Selecting a Reuse Process Model  221
  Define Reuse  224
Chapter 11  **Organizational Models for Reuse**  251

Organizational Model Maxims  251
- Reuse in the Virtual Hallway  252
- Reuse as a Form of Technology Transfer  252
- Reuse Is a Maintenance Responsibility  253

Reuse Organizational Models  253
- Ad Hoc Model  254
- Supply and Demand Model  255
- Expert Services Model  256
- Product Center Model  259
- COTS Model  261

Corporate Reuse Issues—Large and Small  262
- Multilevel Reuse Efforts  262
- Motivating Reuse  264
- Initiating a Corporate Reuse Program  266
- Reuse Maturity  268
- A Small-Company Question  268

Illustrations of Reuse  269
- The Reuse Process of a Software Vendor  269
- Reuse in Japan  271

Case Study Results  274
Summary  275

Additional Reading About Reuse Organizational Models  275
Chapter 12  Select a Team Structure  277
What Is a Team?  277
A Team Has Structure  280
Kinds of Teams  282
  Application Productization Team  282
  Application Prototyping Team  285
  Framework Team  285
  Cross-Project Team  287
  Reuse Team  289
  Maintenance Team  291
Retaining Team Members  292
Framework for Selecting a Team Structure  292
  Decide Which Teams Are Needed  293
  Identify the Roles Needed for Each Team  294
  Decide on the Style for Managing Each Team  295
  Determine How Communication Will Take Place  296
  Find Team Members  296
Summary  298
Additional Reading About Team Structure  299

Chapter 13  Case Studies of Teams  301
Illustration 1: Subteams Based on Enterprise-wide Analysis  302
Illustration 2: Geographically Distributed Teams  305
Illustration 3: Framework Team Support Obligations  308
Illustration 4: Enterprise-wide Framework Team  309
Illustration 5: Two Pollinator Team Examples  310
Illustration 6: The Star Trek Team Model  311
Summary  314

Chapter 14. Expectations for a Software Development Environment  315
Users as Developers  316
  There Are Different Kinds of Developers  316
  Development Is Done at Different Levels  317
Coordination of People, Processes, and Resources  319
Portability and Interoperability  321
  Cross-Platform Portability  321
  Distributed Open Systems  322
Client/Server Architectures  324
Contents

Retaining an Investment in Prior Technology 327
  Keep As-Is and Coexist 327
  Keep Temporarily Until Complete Transition 328
Illustration: Reengineering a Mainframe World to a Client/Server Open Systems Architecture 330
Summary 332
Additional Reading About Expectations for a Software Development Environment 332

Chapter 15  Analysis and Design Methods and Tools 335
Process Models, Methods, and Notations 335
What Is Analysis? 337
What Is Design? 339
Analysis and Design Techniques in Support of Object-Oriented Concepts 340
Choosing Analysis and Design Methods 344
Summary 350
Additional Reading About Analysis and Design Methods 350

Chapter 16  Languages, Libraries, Tools, and Databases 353
Choosing a Programming Language 353
  Multiple Levels of Encapsulation 356
  Bounded and Unbounded Polymorphism 356
  Single and Multiple Inheritance 357
  Static and Dynamic Type Checking and Method Binding 357
  Runtime Support for Object Lifetimes 358
  Hybrid versus Pure Languages 358
Libraries 359
Programming Tools 359
  Implementation Strategy 360
  Interaction Strategy 364
Object-Oriented Databases 368
  How to Choose a Database 370
Summary 372
Additional Reading About Software Development Environment Choices 372
Chapter 17  **Select a Software Development Environment**  375

Framework for Selecting a Software Development Environment  375
  Assess the Current Resource Situation  376
  Understand the Software to Be Developed  379
  Determine the Characteristics of the User Environment  379
  Establish a Program for Evaluating Options  380

Case Study Results  382
  Software Development Environment Camps  382
  Choice of Programming Languages  386
  Choice of Databases  387
  Illustration 1: Languages Face-off Using Multiple Comparisons  388
  Illustration 2: Languages Face-Off Using a Single Large Comparison  389
  Illustration 3: Methods Face-off Using a Survey Approach  390
  Illustration 4: Champions Face-off Face to Face  392

Summary  393

Additional Reading About Evaluating Software Development Environments  394

Chapter 18  **What Is in a Training Plan?**  395

Subject Areas  396
  Concepts About Object-Oriented Technology  396
  Object-Oriented Analysis  397
  Object-Oriented Design  398
  Framework Design  399
  Implementation Environment  400
  Project Management  401
  Other Subject Areas  401

Subject Area Proficiency Levels  405
  Proficiency Levels  405
  Time to Attain a Proficiency Level  406

Training Formats  408
  Prepared, Face-to-Face Presentations  409
  Mentoring  411
  Self-Study  412

Training Plan  413
Case Study Results 416
Summary 418
Additional Reading About Training 419

Chapter 19  Set Up a Training Plan 421
Framework for Setting Up a Training Plan 421
Determine Skills of Team Members 422
Define Desired Skills of Team Members 423
Define Training Resources and Activities 423
Case Study Results 424
Illustration 1: The Reading Club 424
Illustration 2: Continuing Education Programs 426
Illustration 3: Mentoring Success Story 428
Illustration 4: A Popular Idea—Quick Start Training Projects 430
Illustration 5: A Bad Situation That Got Better 432
Summary 435

Chapter 20  What Is Measurement? 437
What Is a Measure? 437
The Goal-Question-Metric Approach 440
Measuring Size, Productivity, and Effort 443
Length Measure: Lines of Code 445
Length Measure: Number of Object Abstractions 446
Functionality Measure: Function Points 447
Measuring Quality 452
Reporting Quality in Terms of Defects 454
Measuring Complexity 455
Inherent and Added Complexities 456
Object-Oriented Design and Code Complexity Measures 456
Measuring Reuse 458
Measuring Potential Reuse 459
Influence of Reuse on Size Measures 459
Influence of Reuse on Effort Measures 460
Framework for Setting Up a Software Measurement Program 462
  Define Project Goals and Objectives 463
  Determine Measures Based on Goals and Objectives 464
  Use Measurement Data to Assess, Predict, and Control 464
  Update the Measurement Plan 465
Summary 466
Additional Reading About Measurement 467

Chapter 21 **Failing with Objects** 469
Good Project Management Leads to Success 470
Poor Project Management Leads to Failure 470
The Last Word 474

Appendices
  Appendix A Primary Case Study Data 475
  Appendix B Suggested Survey Questions for Project Managers 481
  Appendix C Team Member Job Descriptions 493

Glossary of Terms 503
References 519
Index 529