

# Software Engineering

---

## The Production of Quality Software Second Edition

**Shari Lawrence Pfleger**  
CONTEL TECHNOLOGY CENTER  
and SYSTEMS/SOFTWARE, INC.

**Macmillan Publishing Company**  
NEW YORK

**Collier Macmillan Canada**  
TORONTO

**Maxwell Macmillan International**  
NEW YORK OXFORD SINGAPORE SYDNEY

# C O N T E N T S

---

---

## **Chapter 1—Why Software Engineering? 1**

---

- 1.1 How Successful Have We Been? 1**
  - Our Track Record 2
- 1.2 What Is Software Engineering? 4**
  - Solving Problems 4
  - Where Does the Software Engineer Fit In? 6
  - Quality Software 7
  - Jobs and Responsibilities 10
- 1.3 A Systems Approach 11**
  - The Elements of a System 11
- 1.4 An Engineering Approach 17**
  - Building a House 17
  - Building a System 19
  - The Software Development Process 20
  - Members of the Development Team 23
- 1.5 Problems with Software Development 26**
  - Changing Constraints and Requirements 26
  - Phased Development Systems 27
  - Interactions with Other Systems 28
  - The Nature of Computer Systems Themselves 29
- 1.6 Types of Software Projects 31**
  - Degree of Difficulty of Development 31
- 1.7 Example: A Resource Tracking and Simulation System 36**
- 1.8 Chapter Summary 37**
  - Exercises 39

## **Chapter 2—Project Planning 41**

---

- 2.1 The Project Schedule 42**
  - Deliverables 43
  - Activities and Milestones 43
  - Phases of the Schedule 43
  - Work Breakdown Structure and Activity Graphs 44
  - Estimating Completion 47
- 2.2 Project Personnel 51**
  - Project Roles 51

	Personnel Characteristics	52
	Project Team Organization	58
<b>2.3</b>	<b>Cost of Building a System</b>	<b>61</b>
	Types of Costs	61
	Cost Estimation	64
	Experiential Models	64
	Static Linear Estimation Models	70
	Static Nonlinear Estimation Models	71
	Dynamic Estimation Models	83
	The State of Cost Modeling Today	89
<b>2.4</b>	<b>Tools and Techniques for Tracking Cost and Schedule</b>	<b>90</b>
<b>2.5</b>	<b>The Project Plan</b>	<b>97</b>
<b>2.6</b>	<b>Resource Tracking and Simulation Example</b>	<b>99</b>
<b>2.7</b>	<b>Chapter Summary</b>	<b>104</b>
	Exercises	110

## **Chapter 3—Requirements Analysis**

113

---

<b>3.1</b>	<b>What Is a Requirement?</b>	<b>113</b>
	Two Sets of Requirements	115
	Functional and Nonfunctional	116
<b>3.2</b>	<b>Types of Requirements</b>	<b>117</b>
<b>3.3</b>	<b>Characteristics of Requirements</b>	<b>120</b>
<b>3.4</b>	<b>How to Define Requirements</b>	<b>121</b>
	Static Descriptions of Requirements	122
	Dynamic Descriptions of Requirements	126
	Object-Oriented Specification	132
<b>3.5</b>	<b>Examples of Requirements Specification Methods</b>	<b>134</b>
	HIPO Charts	134
	Hierarchical Data Structure	135
	Data Flow Diagrams	137
	PSL/PSA	139
	Software Requirements Engineering Methodology	140
	Structured Analysis and Design Technique	144
	Structured System Analysis	146
	Gist	147
<b>3.6</b>	<b>Prototyping Requirements</b>	<b>148</b>
	Determining Customer Needs	149
<b>3.7</b>	<b>Requirements Documentation</b>	<b>150</b>
	Requirements Definition Document	150
	Requirements Specification Document	151
<b>3.8</b>	<b>Requirements Review</b>	<b>152</b>
<b>3.9</b>	<b>Choosing a Requirements Specification Method</b>	<b>153</b>

- 3.10 Resource Tracking and Simulation Example 155
- 3.11 Chapter Summary 156
- Exercises 157

## **Chapter 4—System Design**

159

- 
- 4.1 What Is System Design? 159
    - Conceptual Design 162
    - Technical Design 162
  - 4.2 Who Produces the System Design? 163
  - 4.3 Approaches to System Design 164
    - Decomposition: Function-Based Design 165
    - Composition: Object-Oriented Design 165
  - 4.4 Characteristics of Good Design 166
    - Modularity 167
    - Levels of Abstraction 167
    - Coupling 168
    - Cohesion 172
    - Control Issues 173
  - 4.5 Initial Design Strategies 176
    - Reducing Structural Complexity 176
    - Simplifying Decision Tables 177
  - 4.6 System Design Techniques and Tools 180
    - Data Flow Diagrams 180
    - Data Abstraction 184
    - Structure Charts 186
    - Prototyping Design 191
    - Application Generators 192
    - High-Level Languages and Design Tools 194
  - 4.7 Design Evaluation and Validation 195
    - Mathematical Validation 196
    - Weighted Matrices 196
    - Design Reviews 198
  - 4.8 Documentation 201
  - 4.9 Resource Tracking and Simulation Example 202
  - 4.10 Chapter Summary 206
    - Exercises 207

## **Chapter 5—Program Design**

209

- 
- 5.1 What Is Program Design? 209
  - 5.2 Design Guidelines 212
    - Top-down Approach 212
    - Bottom-up Approach 212

	Modularity and Independence	213
	Examination of Algorithms	214
	Examination of the Data Types	218
	Module Characteristics	219
	Logical vs. Physical Design	222
<b>5.3</b>	<b>Design Tools</b>	<b>222</b>
	Flow Diagrams	223
	Pseudocode	227
	Nassi-Shneiderman Charts	233
	Automated Design Tools	236
	Other Design Methods and Tools	240
<b>5.4</b>	<b>Design Quality</b>	<b>241</b>
	Structured Design	241
	Error Prevention and Correction	242
	Module Complexity	244
	Unconditional Branching	246
	Increasing the Structure	248
	Testing the Modules for Structure	250
<b>5.5</b>	<b>Design Reviews</b>	<b>253</b>
	Design Rationale	254
<b>5.6</b>	<b>Resource Tracking and Simulation Example</b>	<b>255</b>
<b>5.7</b>	<b>Chapter Summary</b>	<b>259</b>
	Exercises	259

## **Chapter 6—Program Implementation**

265

---

<b>6.1</b>	<b>Programming Standards and Procedures</b>	<b>265</b>
	Need for Standards and Procedures	265
	Matching Design with Implementation	268
<b>6.2</b>	<b>Programming Guidelines</b>	<b>269</b>
	Control Structures	269
	Algorithms	272
	Data Structures	273
	General Guidelines	276
<b>6.3</b>	<b>Documentation</b>	<b>280</b>
	Internal Documentation	280
	External Documentation	285
<b>6.4</b>	<b>Resource Tracking and Simulation Example</b>	<b>286</b>
<b>6.5</b>	<b>Chapter Summary</b>	<b>288</b>
	Exercises	289

## **Chapter 7—Program Testing**

291

---

<b>7.1</b>	<b>Defective Software</b>	<b>291</b>
	Software Errors	293
	Purpose of Testing	296

- 7.2 Unit Testing 300**
  - Program Reviews 301
  - Proving Programs Correct 302
  - Testing Programs 308
  - Generating Test Data 311
- 7.3 Integration Testing 313**
  - Bottom-up Approach 313
  - Top-down Approach 315
  - Big-bang Testing 317
  - Sandwich Testing 318
  - Comparison of Integration Strategies 319
- 7.4 Automated Testing Tools and Techniques 320**
  - Static Analysis 321
  - Dynamic Analysis 322
  - Other Automated Aids 322
- 7.5 The Test Life Cycle 323**
  - Test Plans 323
- 7.6 Estimating Software Quality 325**
  - Reliability, Availability, and Maintainability 325
  - Software Reliability 327
  - Models of Reliability 331
- 7.7 Testing the Resource Tracking and Simulation System 333**
- 7.8 Chapter Summary 336**
  - Exercises 337

## **Chapter 8—System Testing**

**341**

- 
- 8.1 Principles of System Testing 341**
    - Where System Errors Originate 343
    - Steps in System Testing 345
    - Configuration Management 349
  - 8.2 Function Testing 352**
    - Cause-and-Effect Graphs 353
  - 8.3 Performance Testing 359**
    - Test Case Coverage 359
  - 8.4 Acceptance Testing 361**
    - Types of Acceptance Tests 361
    - Results of Acceptance Tests 362
  - 8.5 Installation Testing 363**
  - 8.6 Test Tools 363**
  - 8.7 Test Team 364**
    - Professional Testers 365
    - Analysts 365
    - System Designers 366
    - Configuration Management Specialists 366
    - Users 366

- 8.8 Test Documentation 367**
  - Test Plans 367
  - Test Specification and Evaluation 370
  - Test Description 371
  - Test Analysis Report 373
- 8.9 Resource Tracking and Simulation System Example 375**
- 8.10 Chapter Summary 377**
  - Exercises 378

## **Chapter 9—System Delivery**

**381**

- 
- 9.1 Training 381**
    - Types of Training 383
    - Training Aids 385
    - Guidelines for Training 387
  - 9.2 Documentation 388**
    - Considering the Audience 388
    - User Help and Troubleshooting 395
  - 9.3 Resource Tracking and Simulation Example 399**
  - 9.4 Chapter Summary 403**
    - Exercises 404

## **Chapter 10—Maintenance**

**407**

- 
- 10.1 The Changing System 407**
    - Types of Systems 409
    - Changes During the System Life Cycle 412
    - The System Life Span 415
  - 10.2 The Nature of Maintenance 416**
    - Maintenance Activities 417
    - The Flow of Maintenance 418
  - 10.3 The Problems of Maintenance 421**
    - Factors Contributing to the Difficulty of Maintenance 421
    - Problems of Morale 424
    - Problems of Compromise 424
    - The Cost of Maintenance 425
  - 10.4 Techniques For Improving Maintenance 429**
    - Configuration Management 429
    - Maintenance Graphs and Metrics 433
    - Automated Maintenance Tools 436
  - 10.5 Resource Tracking and Simulation Example 442**
  - 10.6 Chapter Summary 443**
    - Exercises 444

---

**Chapter 11—Examining The Development Process** **447**

---

- 11.1 Software Reuse 447**
  - Types of Reuse 449
  - Reuse Technology and Component Retrieval 450
  - Experience with Reuse 451
  - Benefits of Reuse 453
- 11.2 Software Rejuvenation 453**
  - Redocumentation 455
  - Restructuring 456
  - Reverse Engineering 456
  - Reengineering 458
  - The Future of Rejuvenation 459
- 11.3 Life Cycle CASE 460**
  - Underlying Model 461
  - Requirements Analysis 461
  - Design and Implementation 464
  - Life Cycle Issues 464
- 11.4 Software Metrics and Process Maturity 465**
  - Software Metrics 465
  - Process Maturity 466
  - Steps to Take in Using Metrics 475
  - Expected Benefits of Process Maturity and Metrics 477
- 11.5 Chapter Summary 477**
  - Exercises 478

---

**Chapter 12—What Can Go Wrong** **479**

---

- 12.1 Difficulties of Estimation 481**
  - Estimation Before System Design 481
  - The Way We Estimate 482
  - Models and Metrics 483
  - Use of a Measurement Team 485
- 12.2 Dealing with Phased Development 486**
  - Configuration Management 486
  - User Comfort 486
- 12.3 Changing Schedule 487**
- 12.4 Changing Requirements 488**
  - Prototyping 488
  - Communication Problems 489
  - Technological Change 490
  - Vendor Promises 491
- 12.5 Personnel Problems 492**
  - Difficulties of Peer Review 492
  - Inflexible Work Styles 493



	Peer Preferences	493
	The Learning Curve	494
<b>12.6</b>	<b>Productivity</b>	<b>494</b>
	Use of Automated Tools	494
	Office Environment	495
<b>12.7</b>	<b>The Context of Training</b>	<b>496</b>
<b>12.8</b>	<b>Keeping Documentation Up-to-Date</b>	<b>498</b>
<b>12.9</b>	<b>Chapter Summary</b>	<b>498</b>
<b>12.10</b>	<b>Epilog</b>	<b>499</b>
	Exercises	500
	<i>References</i>	<b>503</b>
	<i>Index</i>	<b>513</b>