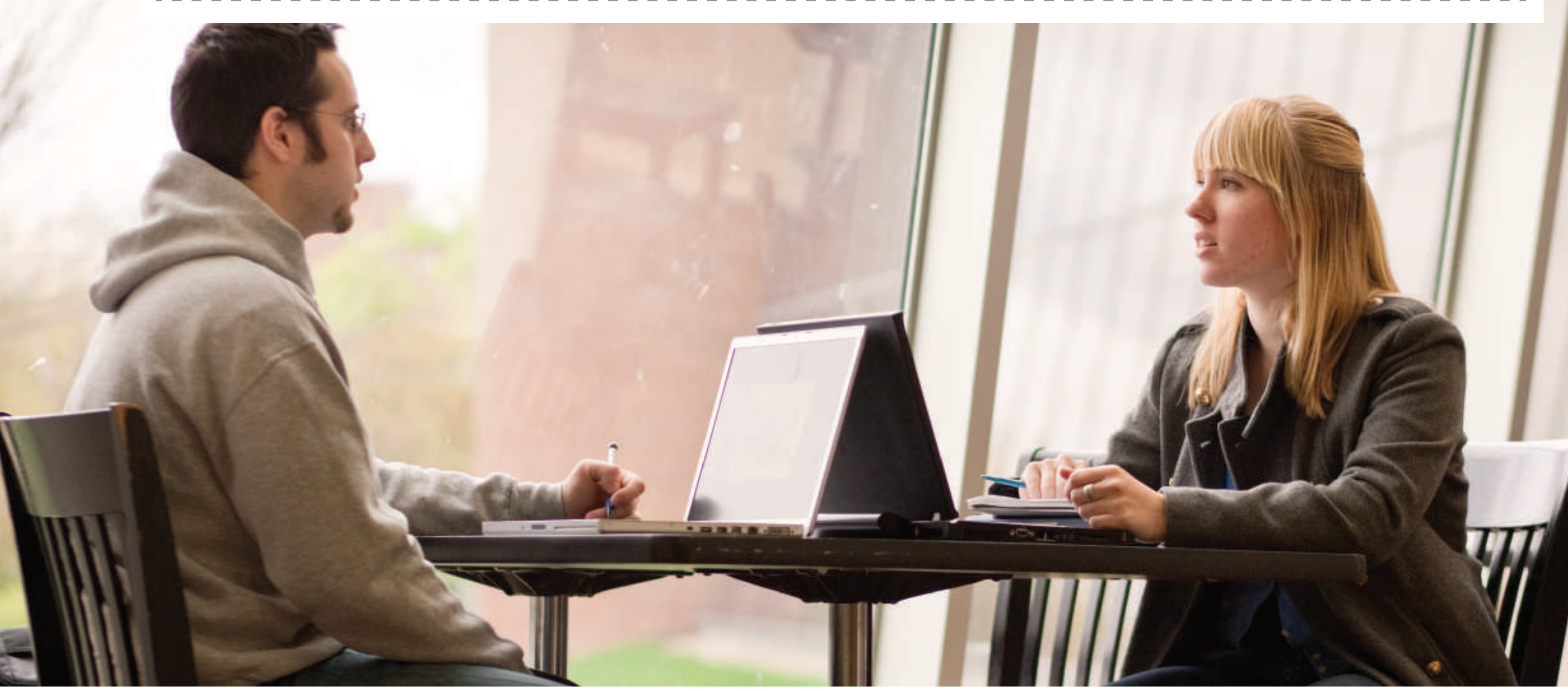
SHELLY CASHMAN SERIES®

Systems Analysis and Design

Ninth Edition

SHELLY | ROSENBLATT

Systems Analysis and Design Ninth Edition



Gary B. Shelly
Harry J. Rosenblatt

Shelly Cashman Series®
An imprint of Course Technology, Cengage Learning

COURSE TECHNOLOGY
CENGAGE Learning™

Australia • Brazil • Japan • Korea • Mexico • Singapore • Spain • United Kingdom • United States

COURSE TECHNOLOGY
CENGAGE Learning™

Systems Analysis and Design, Ninth Edition
Gary B. Shelly
Harry J. Rosenblatt

Vice President, Publisher: Nicole Pinard
Executive Editor: Kathleen McMahon
Associate Acquisitions Editor: Reed Curry
Associate Product Manager: Aimee Poirier
Editorial Assistant: Angela Giannopoulos
Director of Marketing: Elisa Roberts
Marketing Manager: Tristen Kendall
Marketing Coordinator: Adrienne Fung
Print Buyer: Julio Esperas
Director of Production: Patty Stephan
Content Project Manager: Matthew Hutchinson
Development Editor: Deb Kaufmann
Proofreader: Kim Kosmatka
Art Director: Marissa Falco
Interior and Text Design: Joel Sadagursky
Cover Design: Curio Press
Cover Photos: Tom Kates Photography
Compositor: PreMedialGlobal
Indexer: Liz Cunningham

Library of Congress Control Number: 2010943248

ISBN-13: 978-0-538-48161-8

ISBN-10: 0-538-48161-7

Course Technology
20 Channel Center Street
Boston, MA 02210
USA

Cengage Learning is a leading provider of customized learning solutions with office locations around the globe, including Singapore, the United Kingdom, Australia, Mexico, Brazil, and Japan. Locate your local office at: **international.cengage.com/region**

Cengage Learning products are represented in Canada by Nelson Education, Ltd.

Visit our Web site **www.cengage.com/ct/shellycashman** to share and gain ideas on our textbooks!

To learn more about Course Technology, visit **www.cengage.com/coursetechnology**

Purchase any of our products at your local college bookstore or at our preferred online store at **www.cengagebrain.com**

Printed in the United States of America
1 2 3 4 5 6 7 16 15 14 13 12 11

BRIEF CONTENTS

PHASE 1: SYSTEMS PLANNING 1

Chapter 1 Introduction to Systems Analysis and Design 2

Chapter 2 Analyzing the Business Case 52

Chapter 3 Managing Systems Projects 100

PHASE 2: SYSTEMS ANALYSIS 139

Chapter 4 Requirements Modeling 140

Chapter 5 Data and Process Modeling 198

Chapter 6 Object Modeling 248

Chapter 7 Development Strategies 284

PHASE 3: SYSTEMS DESIGN 333

Chapter 8 User Interface Design 334

Chapter 9 Data Design 390

Chapter 10 System Architecture 452

PHASE 4: SYSTEMS IMPLEMENTATION 505

Chapter 11 Managing Systems Implementation 506

PHASE 5: SYSTEMS SUPPORT AND SECURITY 569

Chapter 12 Managing Systems Support and Security 570

THE SYSTEMS ANALYST'S TOOLKIT 629

Toolkit Part A Communication Tools 630

Toolkit Part B CASE Tools 648

Toolkit Part C Financial Analysis Tools 668

Toolkit Part D Internet Resource Tools 688

Glossary/Index 715

Photo Credits 739

TABLE OF CONTENTS

PHASE 1: SYSTEMS PLANNING

Chapter 1

Introduction to Systems Analysis and Design

Objectives 2
Introduction 2
Chapter Introduction Case: Mountain View College Bookstore 3
The Impact of Information Technology 4
The Future 4
Systems Development 6
Systems Analysis and Design 7
Who Develops Information Systems? 7
Information System Components 7
Hardware 8
Software 8
Data 9
Processes 9
People 10
Understand the Business 10
Business Profile 10
Business Process 10
New Kinds of Companies 11
Case in Point 1.1: Cloud Nine Financial Advisors 12
Impact of the Internet 13
B2C (Business-to-Consumer) 13
B2B (Business-to-Business) 14
Business Information Systems 15
Enterprise Computing 15
Transaction Processing 15
Business Support 16
Knowledge Management 16
User Productivity 17
Information Systems Integration 17
What Information Do Users Need? 18
Top Managers 18
Middle Managers and Knowledge Workers 18
Supervisors and Team Leaders 19
Operational Employees 19
Systems Development Tools 19
Modeling 19
Prototyping 20
Computer-Aided Systems Engineering (CASE) Tools 20
Systems Development Methods 21
Structured Analysis 22
Object-Oriented Analysis 24
Agile Methods 25
Other Development Methods 26
Systems Development Guidelines 27
The Information Technology Department 28
Application Development 28
Case in Point 1.2: Global Hotels and Momma's Motels 28
Systems Support and Security 29
User Support 29
Database Administration 29
Network Administration 29
Web Support 29
Quality Assurance (QA) 29
Case in Point 1.3: What Should Lisa Do? 30
The Systems Analyst 30
Responsibilities 30
Knowledge, Skills, and Education 30
Certification 32
Career Opportunities 32
Case in Point 1.4: Just-in-Time Airfreight, Inc. 33
A Question of Ethics 34
Chapter Summary 34
Key Terms and Phrases 36
Learn It Online 37
SCR Associates Case Simulation Session 1: Introduction 38
Chapter Exercises 40
Apply Your Knowledge 41
Case Studies 43
New Century Health Clinic 43
Personal Trainer, Inc. 44
Original Kayak Adventures 45
Chapter Capstone Case: SoftWear, Limited 47

Chapter 2

Analyzing the Business Case

Objectives 52
Introduction 52
Chapter Introduction Case: Mountain View College Bookstore 53
Strategic Planning — A Framework for IT Systems Development 54
Strategic Planning Overview 54
Case in Point 2.1: Lo Carb Meals 56
A CASE Tool Example 57
The Role of the IT Department in Project Evaluation 58
The Future 58
Case in Point 2.2: Attaway Airlines, Part One 58
What Is a Business Case? 58
Information Systems Projects 59
Main Reasons for Systems Projects 59
Case in Point 2.3: Trent College 61
Factors that Affect Systems Projects 61
Internal Factors 62
External Factors 62
Project Management 64
Evaluation of Systems Requests 65
Systems Request Forms 65
Systems Review Committee 66
Overview of Feasibility 66
Operational Feasibility 67
Technical Feasibility 67
Economic Feasibility 68
Schedule Feasibility 69

Evaluating Feasibility 69
Setting Priorities 69
Factors that Affect Priority 70
Discretionary and Nondiscretionary Projects 70
Case in Point 2.4: Attaway Airlines, Part Two 71
Preliminary Investigation Overview 71
Interaction with Managers and Users 71
Planning the Preliminary Investigation 72
Step 1: Understand the Problem or Opportunity 73
Step 2: Define the Project Scope and Constraints 74
Step 3: Perform Fact-Finding 75
Step 4: Analyze Project Usability, Cost, Benefit, and Schedule Data 80
Step 5: Evaluate Feasibility 80
Step 6: Present Results and Recommendations to Management 81
A Question of Ethics 82
Chapter Summary 82
Key Terms and Phrases 83
Learn It Online 84
SCR Associates Case Simulation Session 2: Analyzing the Business Case 85
Chapter Exercises 86
Apply Your Knowledge 87
Case Studies 89
New Century Health Clinic 89
Personal Trainer, Inc. 89
Original Kayak Adventures 90
Town of Eden Bay 91
Chapter Capstone Case: SoftWear, Limited 93

Chapter 3

Managing Systems Projects

Objectives 100
Introduction 100
Chapter Introduction Case: Mountain View College Bookstore 101
Overview of Project Management 102
What Shapes a Project? 102
What Does a Project Manager Do? 103
Case in Point 3.1: Spring Forward Products 103
Project Activities and Planning Steps 103
Step 1: Create a Work Breakdown Structure 104
What Is a Gantt Chart? 104
What Is a PERT/CPM Chart? 105
Which Type of Chart Is Better? 105
Identifying Tasks in a Work Breakdown Structure 106
Case in Point 3.2: Parallel Services 107
Factors Affecting Duration 107
Case in Point 3.3: Sunrise Software 108
Displaying the Work Breakdown Structure 109
Step 2: Identify Task Patterns 110
What Are Task Patterns? 110
How Do I Use Task Boxes to Create a Model? 110
What Are the Main Types of Task Patterns? 110
How Do I Identify Task Patterns? 111
How Do I Work With Complex Task Patterns? 112
Step 3: Calculate the Critical Path 113
What Is a Critical Path? 113
How Do I Calculate the Critical Path? 113
Project Monitoring and Control 114
Monitoring and Control Techniques 114
Maintaining a Schedule 115
Reporting 115
Project Status Meetings 115
Project Status Reports 115
Project Management Examples 116
PERT/CPM Example 116
Software-Driven Example 118
Case in Point 3.4: Census 2010 121
Risk Management 121
Steps in Risk Management 121
Risk Management Software 122
Managing for Success 123
Business Issues 123
Budget Issues 123
Schedule Issues 123
The Bottom Line 124
A Question of Ethics 125
Chapter Summary 125
Key Terms and Phrases 127
Learn It Online 128
SCR Associates Case Simulation Session 3: Managing Systems Projects 129
Chapter Exercises 130
Apply Your Knowledge 131
Case Studies 133
New Century Health Clinic 133
Personal Trainer, Inc. 133
Chapter Capstone Case: SoftWear, Limited 135

PHASE 2: SYSTEMS ANALYSIS

Chapter 4

Requirements Modeling

Objectives 140
Introduction 140
Chapter Introduction Case: Mountain View College Bookstore 141
Systems Analysis Phase Overview 142
Systems Analysis Activities 142
Systems Analysis Skills 143
Team-Based Techniques: JAD, RAD, and Agile Methods 143
Joint Application Development 143
User Involvement 144
JAD Participants and Roles 144
JAD Advantages and Disadvantages 145
Rapid Application Development 145
RAD Phases and Activities 146
RAD Objectives 147
RAD Advantages and Disadvantages 147
Agile Methods 147
Agile Method Advantages and Disadvantages 149
Case in Point 4.1: North Hills College 149

Modeling Tools and Techniques 149
Functional Decomposition Diagrams 150
Business Process Modeling 150
Data Flow Diagrams 151
Unified Modeling Language 151
System Requirements Checklist 153
Output Examples 153
Input Examples 154
Process Examples 154
Performance Examples 154
Control Examples 154
Future Growth, Costs, and Benefits 155
Scalability 155
Total Cost of Ownership 155
Fact-Finding 156
Fact-Finding Overview 156
Who, What, Where, When, How, and Why? 157
The Zachman Framework 158
Interviews 159
Step 1: Determine the People to Interview 159
Step 2: Establish Objectives for the Interview 159
Step 3: Develop Interview Questions 160
Step 4: Prepare for the Interview 161
Step 5: Conduct the Interview 162
Step 6: Document the Interview 162
Step 7: Evaluate the Interview 163
Case in Point 4.2: Deep River College 163
Unsuccessful Interviews 163
Case in Point 4.3: FastPak Overnight Package System 164
Other Fact-Finding Techniques 164
Document Review 164
Observation 164
Questionnaires and Surveys 166
Sampling 167
Research 168
Interviews versus Questionnaires 169
Case in Point 4.4: CyberStuff 170
Documentation 170
The Need for Recording the Facts 170
Software Tools 170
Preview of Logical Modeling 175
A Question of Ethics 175
Chapter Summary 175
Key Terms and Phrases 177
Learn It Online 178
SCR Associates Case Simulation Session 4: Requirements Modeling 179
Chapter Exercises 180
Apply Your Knowledge 181
Case Studies 183
New Century Health Clinic 183
Personal Trainer, Inc. 184
Baxter Community College 185
Town of Eden Bay 185
Chapter Capstone Case: SoftWear, Limited 186

Chapter 5

Data and Process Modeling

Objectives 198
Introduction 198
Chapter Introduction Case: Mountain View College Bookstore 199
Overview of Data and Process Modeling Tools 200
Data Flow Diagrams 200
DFD Symbols 200
Creating a Set of DFDs 206
Guidelines for Drawing DFDs 206
Step 1: Draw a Context Diagram 208
Step 2: Draw a Diagram 0 DFD 209
Step 3: Draw the Lower-Level Diagrams 212
Case in Point 5.1: Big Ten University 217
Data Dictionary 217
Using CASE Tools for Documentation 218
Documenting the Data Elements 218
Documenting the Data Flows 220
Documenting the Data Stores 221
Documenting the Processes 222
Documenting the Entities 223
Documenting the Records 223
Data Dictionary Reports 223
Process Description Tools 224
Modular Design 224
Structured English 225
Decision Tables 226
Case in Point 5.2: Rock Solid Outfitters (Part 1) 230
Decision Trees 230
Case in Point 5.3: Rock Solid Outfitters (Part 2) 230
Logical Versus Physical Models 231
Sequence of Models 231
Four-Model Approach 231
Case in Point 5.4: Tip Top Staffing 231
A Question of Ethics 232
Chapter Summary 232
Key Terms and Phrases 233
Learn It Online 234
SCR Associates Case Simulation Session 5: Data and Process Modeling 235
Chapter Exercises 236
Apply Your Knowledge 237
Case Studies 239
New Century Health Clinic 239
Personal Trainer, Inc. 240
Chapter Capstone Case: SoftWear, Limited 241

Chapter 6

Object Modeling

Objectives 248
Introduction 248
Chapter Introduction Case: Mountain View College Bookstore 249
Overview of Object-Oriented Analysis 250
Object-Oriented Terms and Concepts 250
Objects 251
Attributes 254
Methods 254
Messages 255
Classes 256

Relationships Among Objects and Classes 258
Object Relationship Diagram 258
Object Modeling with the Unified Modeling Language 259
Use Case Modeling 259
Case in Point 6.1: Hilltop Motors 261
Use Case Diagrams 261
Class Diagrams 262
Case in Point 6.2: Train the Trainer, Inc. 264
Sequence Diagrams 264
State Transition Diagrams 265
Activity Diagrams 266
Case in Point 6.3: TravelBiz 266
CASE Tools 266
Organizing the Object Model 267
Case in Point 6.4: Cyber Associates 267
A Question of Ethics 267
Chapter Summary 268
Key Terms and Phrases 269
Learn It Online 270
SCR Associates Case Simulation Session 6: Object Modeling 271
Chapter Exercises 272
Apply Your Knowledge 273
Case Studies 275
New Century Health Clinic 275
Personal Trainer, Inc. 275
Chapter Capstone Case: SoftWear, Limited 276

Chapter 7
Development Strategies

Objectives 284
Introduction 284
Chapter Introduction Case: Mountain View College Bookstore 285
Development Strategies Overview 286
The Impact of the Internet 286
Software as a Service 286
Traditional vs. Web-Based Systems Development 287
Looking to the Future: Web 2.0 and Cloud Computing 289
Outsourcing 290
The Growth of Outsourcing 290
Outsourcing Fees 291
Outsourcing Issues and Concerns 291
Offshore Outsourcing 292
Case in Point 7.1: Turnkey Services 293
In-House Software Development Options 293
Make or Buy Decision 293
Developing Software In-House 294
Purchasing a Software Package 295
Customizing a Software Package 296
Creating User Applications 297
Role of the Systems Analyst 298
Analyzing Cost and Benefits 299
Financial Analysis Tools 299
Case in Point 7.2: Sterling Associates 299
Cost-Benefit Analysis Checklist 300
The Software Acquisition Process 301
Step 1: Evaluate the Information System Requirements 301
Step 2: Identify Potential Vendors or Outsourcing Options 304
Step 3: Evaluate the Alternatives 306
Step 4: Perform Cost-Benefit Analysis 307
Step 5: Prepare a Recommendation 307
Step 6: Implement the Solution 308
Case in Point 7.3: Doug's Sporting Goods 308
Completion of Systems Analysis Tasks 309
System Requirements Document 309
Presentation to Management 309
Transition to Systems Design 310
Preparing for Systems Design 311
Logical and Physical Design 311
Systems Design Guidelines 311
Overview 312
Case in Point 7.4: Downtown! 313
Design Trade-Offs 314
Prototyping 315
Prototyping Methods 315
Prototyping Tools 316
Limitations of Prototypes 317
Software Development Trends 317
Views from the IT Community 317
A Question of Ethics 318
Chapter Summary 319
Key Terms and Phrases 321
Learn It Online 322
SCR Associates Case Simulation Session 7: Development Strategies 323
Chapter Exercises 324
Apply Your Knowledge 325
Case Studies 327
New Century Health Clinic 327
Personal Trainer, Inc. 328
Chapter Capstone Case: SoftWear, Limited 330

PHASE 3: SYSTEMS DESIGN

Chapter 8
User Interface Design

Objectives 334
Introduction 334
Chapter Introduction Case: Mountain View College Bookstore 335
What Is a User Interface? 336
Evolution of the User Interface 336
Human-Computer Interaction 338
Case in Point 8.1: Casual Observer Software 340
Principles of User-Centered Design 341
Understand the Business 341
Maximize Graphical Effectiveness 341
Think Like a User 341

Use Models and Prototypes 342
Focus on Usability 342
Invite Feedback 342
Document Everything 342

Designing the User Interface 342
Design a Transparent Interface 343
Create an Interface that Is Easy to Learn and Use 344
Enhance User Productivity 344
Make It Easy for Users to Obtain Help or Correct Errors 345
Minimize Input Data Problems 346
Provide Feedback to Users 346
Create an Attractive Layout and Design 347
Use Familiar Terms and Images 347
Add Control Features 347

Case in Point 8.2: Boolean Toys 349

Output Design 350
Overview of Report Design 350
Types of Reports 352
User Involvement in Report Design 353
Report Design Principles 354

Case in Point 8.3: Lazy Eddie 357
Output Technology 357

Input Design 360
Source Documents and Forms 360

Case in Point 8.4: Trustworthy Insurance Company 362
Data Entry Screens 363
Input Masks 365
Validation Rules 366
Input Technology 368
Input Volume Reduction 370

Security and Control Issues 370
Output Security and Control 370
Input Security and Control 371

A Question of Ethics 372

Chapter Summary 372

Key Terms and Phrases 374

Learn It Online 375

SCR Associates Case Simulation Session 8: User Interface Design 376

Chapter Exercises 377

Apply Your Knowledge 378

Case Studies 380
New Century Health Clinic 380
Personal Trainer, Inc. 380
Video Superstore 381

Chapter Capstone Case: SoftWear, Limited 382

Chapter 9

Data Design

Objectives 390

Introduction 390

Chapter Introduction Case: Mountain View College Bookstore 391

Data Design Concepts 392
Data Structures 392
Overview of File Processing 392
The Evolution from File Systems to Database Systems 395

DBMS Components 396
Interfaces for Users, Database Administrators, and Related Systems 396
Data Manipulation Language 398
Schema 398
Physical Data Repository 398

Web-Based Database Design 398
Characteristics of Web-Based Design 398
Internet Terminology 399
Connecting a Database to the Web 400
Data Security 401

Data Design Terminology 401
Definitions 401
Key Fields 402
Referential Integrity 404

Entity-Relationship Diagrams 405
Drawing an ERD 406
Types of Relationships 406
Cardinality 408

Case in Point 9.1: TopText Publishing 410

Normalization 410
Standard Notation Format 411
Repeating Groups and Unnormalized Designs 411
First Normal Form 412
Second Normal Form 413
Third Normal Form 416
A Normalization Example 417

Case in Point 9.2: CyberToys 419

Using Codes During Data Design 422
Overview of Codes 422
Types of Codes 423
Developing a Code 424

Case in Point 9.3: DotCom Tools 425

Database Design: One Step At a Time 426

Database Models 427
A Real-World Business Example 427
Working with a Relational Database 428

Data Storage and Access 430
Strategic Tools for Data Storage and Access 430
Logical and Physical Storage 432
Data Coding and Storage 433

Data Control 435

Case in Point 9.4: SoccerMom 436

A Question of Ethics 436

Chapter Summary 437

Key Terms and Phrases 439

Learn It Online 440

SCR Associates Case Simulation Session 9: Data Design 441

Chapter Exercises 442

Apply Your Knowledge 443

Case Studies 445
New Century Health Clinic 445
Personal Trainer, Inc. 445
FastFlight Airlines 446

Chapter Capstone Case: SoftWear, Limited 447

Chapter 10

System Architecture

Objectives 452
Introduction 452
Chapter Introduction Case: Mountain View College Bookstore 453
System Architecture Checklist 454
Enterprise Resource Planning (ERP) 454
Case in Point 10.1: ABC Systems 455
Initial Cost and TCO 455
Scalability 456
Web Integration 456
Legacy System Interface Requirements 457
Processing Options 458
Security Issues 458
Planning the Architecture 458
Servers 458
Clients 459
Client/Server Architecture 461
Overview 461
Client/Server Design Styles 463
Fat and Thin Clients 464
Client/Server Tiers 464
Middleware 465
Cost-Benefit Issues 465
Client/Server Performance Issues 466
Internet-Based Architecture 467
Developing E-Commerce Solutions In-House 468
Case in Point 10.2: Small Potatoes, Inc. 469
Packaged Solutions and E-Commerce Service Providers 469
Corporate Portals 470
Cloud Computing 470
Web 2.0 472
Processing Methods 474
Online Processing 474
Batch Processing 475
Case in Point 10.3: R/Way Trucking Company 475
Combined Online and Batch Processing 475
Network Models 476
The OSI Reference Model 476
Network Protocols 477
Network Topology 477
Routers 481
Network Modeling Tools 481
Network Licensing Issues 482
Wireless Networks 482
Wireless Network Standards 482
Wireless Network Topologies 483
Wireless Trends 484
Case in Point 10.4: Spider IT Services 485
Systems Design Completion 485
System Design Specification 486
User Approval 487
Presentations 487
A Question of Ethics 488
Chapter Summary 488
Key Terms and Phrases 491
Learn It Online 492
SCR Associates Case Simulation Session 10: Systems Architecture 493
Chapter Exercises 494
Apply Your Knowledge 495
Case Studies 497
New Century Health Clinic 497
Personal Trainer, Inc. 497
Chapter Capstone Case: SoftWear, Limited 499

PHASE 4: SYSTEMS IMPLEMENTATION

Chapter 11

Managing Systems Implementation

Objectives 506
Introduction 506
Chapter Introduction Case: Mountain View College Bookstore 507
Software Quality Assurance 508
Software Engineering 508
International Organization for Standardization (ISO) 509
Overview of Application Development 511
Review the System Design 511
Application Development Tasks 511
Systems Development Tools 512
Project Management 514
Structured Application Development 514
Structure Charts 514
Cohesion and Coupling 516
Drawing a Structure Chart 517
Object-Oriented Application Development 518
Characteristics of Object-Oriented Application Development 519
Implementation of Object-Oriented Designs 520
Object-Oriented Cohesion and Coupling 520
Agile Application Development 520
An Extreme Programming (XP) Example 520
The Future of Agile Development 523
Coding 523
Programming Environments 523
Generating Code 523
Testing the System 525
Unit Testing 525
Case in Point 11.1: Your Move, Inc. 526
Integration Testing 526
System Testing 527
Case in Point 11.2: WebTest, Inc. 528
Documentation 528
Program Documentation 529
System Documentation 529
Operations Documentation 529
User Documentation 530
Management Approval 533
System Installation and Evaluation 534
Operational and Test Environments 534
Training 535
Training Plan 535
Vendor Training 536

Webinars, Podcasts, and Tutorials 537
Outside Training Resources 537
Training Tips 537
Interactive Training 539

Data Conversion 543
Data Conversion Strategies 544
Data Conversion Security and Controls 544

System Changeover 544
Direct Cutover 544
Parallel Operation 545
Pilot Operation 546
Phased Operation 546

Case in Point 11.3: Global Cooling 547

Post-Implementation Tasks 547
Post-Implementation Evaluation 547

Case in Point 11.4: Yorktown Industries 549

Final Report to Management 549

A Question of Ethics 550

Chapter Summary 550

Key Terms and Phrases 552

Learn It Online 553

SCR Associates Case Simulation Session 11: Systems Implementation 554

Chapter Exercises 555

Apply Your Knowledge 556

Case Studies 558
New Century Health Clinic 558
Personal Trainer, Inc. 558
Fanciful Crystal 559

Chapter Capstone Case: SoftWear, Limited 560

PHASE 5: SYSTEMS SUPPORT AND SECURITY

Chapter 12

Managing Systems Support and Security

Objectives 570

Introduction 570

Chapter Introduction Case: Mountain View College Bookstore 571

Overview 572

User Support 572
User Training 572
Help Desks 572
Outsourcing Issues 574

Maintenance Tasks 574
Corrective Maintenance 575
Adaptive Maintenance 576
Perfective Maintenance 577
Preventive Maintenance 577

Case in Point 12.1: Outback Outsourcing, Inc. 578

Maintenance Management 578
The Maintenance Team 578

Case in Point 12.2: Brightside Insurance, Inc. 580

Maintenance Requests 580
Establishing Priorities 582
Configuration Management 582
Maintenance Releases 583
Version Control 583
Baselines 584

System Performance Management 584
Fault Management 585
Performance and Workload Measurement 585
Capacity Planning 587
System Maintenance Tools 588

System Security Overview 589
System Security Concepts 589
Risk Management 590
Attacker Profiles and Attacks 592

Security Levels 594
Physical Security 594

Case in Point 12.3: Outer Banks County 597

Network Security 597
Application Security 601
File Security 603
User Security 603
Procedural Security 606

Case in Point 12.4: Chain Link Consulting, Inc. 606

Backup and Recovery 607
Backup Policies 607
Business Continuity Issues 608

System Obsolescence 609

Future Challenges and Opportunities 610
Predictions 610
Strategic Planning for IT Professionals 611
IT Credentials and Certification 611
Critical Thinking Skills 611

A Question of Ethics 613

Chapter Summary 613

Key Terms and Phrases 615

Learn It Online 616

SCR Associates Case Simulation Session 12: Managing Systems Support and Security 617

Chapter Exercises 618

Apply Your Knowledge 619

Case Studies 621
New Century Health Clinic 621
Personal Trainer, Inc. 621
Tarheel Industries 622
Mills Imports 623

Chapter Capstone Case: SoftWear, Limited 624

THE SYSTEMS ANALYST'S TOOLKIT

Toolkit Part A

Communication Tools

Objectives 630

Introduction 630

Toolkit Introduction Case: Mountain View College Bookstore 631

Successful Communication Strategies 632
Why, Who, What, When, and How 632
Cultural Context 632
Know Your Subject 633

Written Communications 633
- Writing Style and Readability 633
- E-Mail, Memos, and Letters 634
- Netiquette 635
- Workgroup Software 637
- Reports 637

Oral Communications 639
- Define the Audience 639
- Define the Objectives 639
- Organize the Presentation 640
- Define Any Technical Terms 640
- Prepare Presentation Aids 640
- Practice 641
- The Presentation 642
- Online Presentations 643

Managing Your Communication Skills 644

Toolkit Summary 645

Key Terms and Phrases 646

Toolkit Exercises 647

Toolkit Part B

CASE Tools

Objectives 648

Introduction 648

Toolkit Introduction Case: Mountain View College Bookstore 649

Overview of CASE Tools 650
- CASE Tools History 651
- The Marketplace for CASE Tools 651

CASE Terms and Concepts 653
- Repository 653
- Individual Tools 653

Integrated Development Environments 656
- Examples of Integrated Development Environments 656
- Pros and Cons of Integrated Development Tools 658

CASE Tool Examples 658
- Visible Analyst 658
- Rational Software 658

CASE Tool Trends 661
- New Products and Features 661
- Method-Specific CASE Tools 662

Toolkit Summary 664

Key Terms and Phrases 665

Toolkit Exercises 666

Toolkit Part C

Financial Analysis Tools

Objectives 668

Introduction 668

Toolkit Introduction Case: Mountain View College Bookstore 669

Describing Costs and Benefits 670
- Cost Classifications 672
- Managing Information Systems Costs and Charges 673
- Benefit Classifications 674

Cost-Benefit Analysis 674
- Payback Analysis 675
- Using a Spreadsheet to Compute Payback Analysis 677
- Return on Investment Analysis 678
- Using a Spreadsheet to Compute ROI 680
- Present Value Analysis 680
- Using a Spreadsheet to Calculate Present Value 683

Toolkit Summary 684

Key Terms and Phrases 686

Toolkit Exercises 687

Toolkit Part D

Internet Resource Tools

Objectives 688

Introduction 688

Toolkit Introduction Case: Mountain View College Bookstore 689

Overview 690

Planning an Internet Research Strategy 690
- Step 1. Review Your Information Requirements 690
- Step 2. Use the Proper Search Tools and Techniques 691
- Step 3. Evaluate the Results 691
- Step 4. Consider Copyright and Data Integrity Issues 691

Search Basics 692

Search Engines 693
- Search Engine Concepts 693
- Search Techniques 694
- Advanced Search Techniques 695
- Search Checklist 697

Subject Directories 698
- A Subject Directory Example 698
- Advantages and Disadvantages of Subject Directories 699

The Invisible Web 699
- Invisible Web Examples 700
- Navigation Tools for the Invisible Web 700

Internet Communication Channels 702
- Social Networking 702
- Newsgroups 702
- Newsletters, Blogs, Podcasts, and Webcasts 703
- RSS Feeds 704
- Webinars 704
- Mailing Lists 704
- Web-Based Discussion Groups 705
- Chat Rooms 705
- Instant Messaging and Text Messaging 706

Information Technology Community Resources 708
- Corporate Resources 708
- Government Resources 709
- Personal and Professional Resources 709
- Online Learning Resources 709

Toolkit Summary 712

Key Terms and Phrases 713

Toolkit Exercises 714

Glossary/Index 715

Photo Credits 739

PREFACE

The Shelly Cashman Series® offers the finest textbooks in computer education. We are proud that our previous editions of *Systems Analysis and Design* have been so well received by instructors and students. *Systems Analysis and Design, Ninth Edition* continues with the innovation, quality, and reliability you have come to expect from the Shelly Cashman Series.

Overview

Systems Analysis and Design, Ninth Edition includes exciting Video Learning Sessions, developed to maximize the learning experience. The Video Learning Sessions combined with the text offer an interactive, multimedia approach to information systems development. Many two- and four-year colleges and schools use this book in information systems, computer science, and e-commerce curriculums. The textbook emphasizes the role of the systems analyst in a dynamic, business-related environment.

Facing a challenging global marketplace, companies need strong IT resources to survive and compete effectively. Many of today's students will become the systems analysts, managers, and IT professionals of tomorrow. This textbook will help prepare them for those roles.

Using this book, students learn how to translate business requirements into information systems that support a company's short- and long-term objectives. Case studies and assignments teach analytical and problem-solving skills. Students learn about traditional structured analysis, object-oriented concepts, and agile methods. Extensive end-of-chapter exercises emphasize critical-thinking skills.

The *Ninth Edition* introduces several major new features, including four new Video Learning Sessions and a new end-of-chapter assignment called *Ready for a Challenge*, which stresses critical thinking skills. This edition also includes significant updates on topics such as agile development, IT security, and Web 2.0 trends.

Objectives of This Textbook

Systems Analysis and Design, Ninth Edition is intended for a three credit-hour introductory systems analysis and design course. This textbook is designed to:

- Enhance critical thinking skills with the new Ready for a Challenge feature at the end of each chapter. The scenario-based tasks and sample answers help students develop perception, organization, analysis, problem-solving, and decision-making skills that they can take to the workplace.
- Explain systems analysis and design using an appealing full-color format, numerous screen shots and illustrations, and an easy-to-read style that invites students to learn.
- Introduce project management concepts early in the systems development process, with a new chapter that explains project management tools and techniques.
- Challenge students with a Question of Ethics mini-case in each chapter that asks them to respond to real-life ethical issues in an IT environment.
- Provide multi-method coverage, including a comparison of structured, object-oriented, and agile systems development methods.
- Emphasize the importance of planning, implementing, and managing an effective IT security program.
- Explain how IT supports business requirements in today's intensely competitive environment, and describe major IT developments and trends.

- Describe a systems analyst's job in a typical business organization, and show students how to use various tools and techniques to improve their skills and manage their careers.
- Provide students with a comprehensive Systems Analyst's Toolkit that highlights four major cross-functional tools, including: Communications Tools, CASE Tools, Financial Analysis Tools, and Internet Resource Tools.

Video Learning Sessions

Eighteen multimedia Video Learning Sessions describe key systems analysis skills and concepts and provide students with a self-paced, interactive learning tool that reinforces the text. The sessions provide step-by-step explanations that are easy to follow and understand.

Each session includes practice tasks, sample answers, and challenge tasks to keep students interested and engaged as they learn.

- Topics include DFDs, object-oriented analysis, functional decomposition diagrams, structure charts, data normalization, entity-relationship diagrams, decision tables, financial tools, and project management.
- A **Your Turn** feature in every Video Learning Session challenges students to apply their skills and check their work against sample answers. This hands-on practice can help students better handle actual assignments and tasks.
- The Video Learning Sessions offer a self-paced multimedia format that students can review at their convenience.
- Instructors may use the Video Learning Sessions as classroom presentations, distance-education support, student review tools, and exam preparation.

Other New and Updated Features in This Text

Systems Analysis and Design, Ninth Edition offers these exciting new and expanded features:

- New Ready for a Challenge end-of-chapter assignment allows students to practice critical thinking skills, first by trying *Practice Tasks* and viewing sample answers, and then by completing the *Challenge Tasks*. These tasks can help students develop perception, organization, analysis, problem-solving, and decision-making skills that they can take to the workplace.
- Increased emphasis on project management skills and techniques, with one or more Gantt charts in each chapter, work breakdown structures, and realistic project examples. A link to Open Workbench connects students to open-source project management software that they can download and install.
- Question of Ethics mini-case in each chapter challenges students with real-life ethical issues in an IT environment.
- Multi-method coverage provides comparison of structured, object-oriented, and agile development methods, starting in Chapter 1. New material on agile methods includes examples of extreme programming, scrum, spiral models, and related topics.
- New coverage of risk management, both in a project management context and as a key element of IT security planning.
- Extensive update of networking coverage, including new material on switches, routers, and multistation access units. New coverage of wireless networks, including wireless standards, topologies, and trends.
- Expansion of IT security material, including risk management, fault management, backup and recovery, wireless security issues, and a six-level security framework.

- Expanded coverage of IT trends, including cloud computing, Web 2.0, social networking, RFID, wireless networks, mobile computing, offshore outsourcing, e-business, ERP, Web hosting, client/server architecture, network concepts, Webinars, podcasts, RSS feeds, Web-based applications, and others.
- Updated Systems Analyst's Toolkit teaches students IT support skills in four cross-functional areas, including Communication Tools, CASE Tools, Financial Analysis Tools, and Internet Resource Tools.
- New Management Information Systems CourseMate Web site for *Systems Analysis and Design, Ninth Edition* available for a fully digital course solution. CourseMate provides one location for all interactive activities, Video Learning Sessions, and an interactive e-book. EngagementTracker provides the ability to assess student understanding of concepts through the interactive activities.

Organization of This Textbook

Systems Analysis and Design, *Ninth Edition*, contains 16 learning units in twelve chapters and a four-part Systems Analyst's Toolkit that teaches valuable cross-functional skills.

Chapter 1 – Introduction to Systems Analysis and Design Chapter 1 provides an up-to-date overview of IT issues, major trends, and various systems development approaches, including structured, object-oriented, and agile methods. The chapter emphasizes the important role of systems analysis and design in supporting business objectives.

Chapter 2 – Analyzing the Business Case Chapter 2 offers a business-related starting point for successful systems analysis. Topics include strategic planning, review of systems requests, how to conduct a feasibility study, and the steps in a preliminary investigation.

Chapter 3 – Managing Systems Projects Chapter 3 explains project management, cost estimating, and change control for information systems. This chapter includes hands-on skills that systems analysts can use to create Gantt charts and PERT charts.

Chapter 4 – Requirements Modeling Chapter 4 describes fact-finding techniques and team-based modeling methods, including JAD and RAD, that systems analysts use to model and document a new system.

Chapter 5 – Data and Process Modeling Chapter 5 explains how systems analysts create a logical model for the new system by using data flow diagrams and process description tools, including structured English, decision tables, and decision trees.

Chapter 6 – Object Modeling Chapter 6 explains object-oriented tools and techniques, including use case diagrams, class diagrams, sequence diagrams, state-transition diagrams, activity diagrams, and the Unified Modeling Language.

Chapter 7 – Development Strategies Chapter 7 focuses on software acquisition options, including outsourcing and offshore outsourcing options, application service providers, and other trends that view software as a service rather than a product.

Chapter 8 – User Interface Design Chapter 8 highlights output and report design, the interaction between humans and computers, including usability issues, graphical screen design, input issues, and data entry guidelines.

Chapter 9 – Data Design Chapter 9 describes data design terms, concepts, and skills including entity-relationship diagrams, cardinality, data normalization rules, data warehousing, data mining, a comparison of logical and physical records, and data control measures.

Chapter 10 – System Architecture Chapter 10 explains the elements of system architecture, with emphasis on RFID, ERP, supply chain management, client/server architecture, and network topology, including wireless networking standards and trends.

Chapter 11 – Managing Systems Implementation Chapter 11 includes coverage of application development and implementation topics, including structure charts, documentation techniques, system testing, user training, data conversion, changeover methods, and post-implementation evaluation.

Chapter 12 – Managing Systems Support and Security Chapter 12 describes user support, maintenance techniques, and factors that indicate the end of a system's useful life. This chapter explains IT security concepts, techniques, and tools, and specifically addresses six security levels: physical, network, application, file, user, and procedural security. Chapter 12 also describes risk management, data backup and disaster recovery, and explains future challenges and opportunities that IT professionals will face in a dynamic workplace.

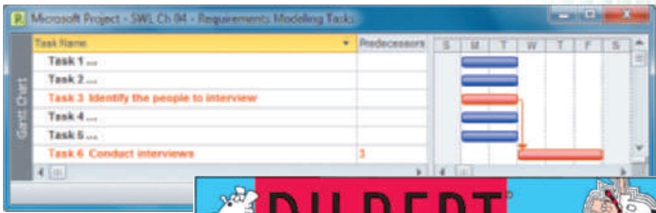
Toolkit Part A – Communication Tools Part A of the Toolkit describes oral and written communication tools that can make a systems analyst more effective. Topics include guidelines for successful communications, tips for better readability, how to organize and plan a presentation, effective speaking techniques, and managing communication skills.

Toolkit Part B – CASE Tools Part B of the Toolkit focuses on computer-aided software engineering (CASE) tools that systems analysts use to document, model, and develop information systems. Examples of several popular CASE tools are provided, along with sample screens that show CASE tool features.

Toolkit Part C – Financial Analysis Tools Part C of the Toolkit explains various tools that systems analysts use to determine feasibility and evaluate the costs and benefits of an information system. Specific tools include payback analysis, return on investment (ROI), and net present value (NPV).

Toolkit Part D – Internet Resource Tools Part D of the Toolkit explains Internet-based information gathering strategies. Topics include search engines, subject directories, the invisible Web, advanced search techniques, Boolean logic and Venn diagrams. This Toolkit Part also discusses social networking, newsgroups, newsletters, blogs, podcasts, RSS feeds, Webinars, mailing lists, Web-based discussion groups, chat rooms, instant messaging, and online learning opportunities.

FOR THE STUDENT



The Shelly Cashman Series wants you to have a valuable learning experience that will provide the knowledge and skills you need to be successful. With that goal in mind, we have included many activities, games, and learning tools, that we hope you will find interesting, challenging, and enjoyable. For example, because a picture is worth a thousand words, each systems development phase begins with an eye-catching Dilbert© cartoon and a multi-color Gantt chart that provides a "*You are Here*" roadmap.

Chapter Opening Features

Each chapter contains the following features to help you get started:

- **Chapter Introduction** Read the Chapter Introduction for a brief overview of the chapter.
- **Chapter Objectives** The Chapter Objectives lists the main skills and knowledge you will have when you finish the chapter.
- **Chapter Introduction Case: Mountain View College Bookstore** The Mountain View College Bookstore case is a continuing case study that provides a real-world overview of the topics that will be covered in each chapter. As you work through the textbook, you will see how the Mountain View IT team discusses the issues, identifies the key points, and creates specific task lists.

Learning Tools within the Chapter

As you work through each chapter, you will find these helpful tools and features:

- **Video Learning Sessions** An online library of 18 self-paced multimedia sessions is available to you and the text includes reminders about them. Depending on your personal learning style, you might use the videos in various ways. For example, one approach might be to review the chapter, watch the VLS, try the Your Turn tasks, and then check your answers.

- **A Question of Ethics** A mini-case in each chapter will challenge you with real-life ethical issues in an IT environment.
- **Case in Point** This exciting feature provides four embedded mini-case opportunities for you to analyze and apply the skills and concepts you are learning in the chapter.
- **Toolkit Time** The Systems Analyst's Toolkit explains skills that you can apply at any point in the textbook. Toolkit Time marginal notes remind you about the Toolkit, where to find it, and how it might help you address the issues or material in the chapter.
- **On the Web** Learn more about a topic by visiting the suggested Web sites and exploring the links we have provided.

End-of-Chapter Exercises

The following exercises are in every text chapter:

- **Learn It Online** Each chapter features a Learn It Online page that includes six exercises. These exercises utilize the Web to offer chapter-related reinforcement activities that will help you gain confidence in systems analysis and design. These exercises include True/False, Multiple Choice, Short Answer, Flash Cards, Practice Test, and several learning games.
- **CASE SIM: SCR Associates** This is an interactive Web-based case study, with a work session at the end of each chapter. Visit SCR's Web site and log on to the company's intranet to read e-mail messages addressed to you, listen to voice mail messages, and perform assigned tasks in a realistic corporate setting. In this simulation you report to Jesse Baker, but you e-mail your completed assignments to your instructor. Detailed instructions on how to use this case are available in the Management Information Systems CourseMate Web site for *Systems Analysis and Design, Ninth Edition* at www.cengagebrain.com. To log on to the SCR intranet, you must use the password ***sad9***. When you log on to the SCR intranet, you also will be asked to enter your first and last name so your e-mail can be addressed to you correctly.
- **Chapter Exercises** In this section, you will find 10 Review Questions, four Discussion Topics, and four Projects. These exercises allow you to apply your understanding of the material and will help to prepare you for tests and assessments.
- **Apply Your Knowledge** This section includes four mini-cases per chapter. Each mini-case requires you to use the knowledge and skills you learned in the chapter.
- **Case Studies** Case studies provide practical experience and allow you to practice specific skills learned in the chapter. Each chapter contains several case studies, two of which (New Century Health Clinic and Personal Trainer, Inc.) continue throughout the textbook. You can complete your assignments using Microsoft Word and Excel forms, available in the Management Information Systems CourseMate Web site for *Systems Analysis and Design, Ninth Edition* at www.cengagebrain.com.

- **Chapter Capstone Case: SoftWear, Limited** SoftWear, Limited (SWL) is a continuing case study where students act as members of the SWL systems development team and perform various assignments in each chapter, including a set of project management tasks and a sample Gantt chart.
- **Ready for a Challenge** This new end-of-chapter assignment stresses critical thinking skills, which many educators and employers believe are very important in the workplace. Perform the Practice Tasks first, view the sample answers, and apply your knowledge and skill to the Challenge Tasks. *Ready for a Challenge* can help you develop perception, organization, analysis, problem-solving, and decision-making skills that you can take to the workplace.

Additional Support Tools

These additional tools can enhance your learning experience:

GLOSSARY/INDEX This edition of the textbook includes a glossary/index feature to assist your understanding of key terms and phrases, or to use as a quick reference tool.

STUDENT STUDY TOOL This interactive study tool, accessible via the Management Information Systems CourseMate Web site for *Systems Analysis and Design, Ninth Edition* provides:

- Detailed outlines of every chapter that highlight key topics covered and can be used as a guide when reviewing for an exam
- Chapter glossaries that allow you to look up all key terms in one place, and provide page references where key terms can be found if you need more information
- Figures and Test Yourself questions that provide additional reinforcement of chapter concepts
- User guide for Open Workbench (a free, open-source project management program), and links to download and install a trial version of Microsoft Project and a full version of Open Workbench

MANAGEMENT INFORMATION SYSTEMS COURSEMATE Broaden your learning experience and enhance your understanding of the material in each chapter with the Management Information Systems CourseMate Web site. Visit www.cengagebrain.com for access to:

- Full, interactive digital e-book
- Video Learning Sessions and Your Turn exercises
- Ready for a Challenge Practice Tasks and Challenge Tasks
- On the Web links
- Learn It Online exercises, including True/False, Multiple Choice, Short Answer, Flash Cards, Practice Test, and several learning games
- SCR Associates Internet and intranet sites
- Forms Library
- Project Management Resources

FOR THE INSTRUCTOR

The Shelly Cashman Series is dedicated to providing you all of the tools you need to make your class a success. Information on all supplementary materials is available through your Course Technology representative or by calling one of the following telephone numbers: Colleges, Universities, Continuing Education Departments, Post-Secondary Vocational Schools, Career Colleges, Business, Industry, Government, Trade, Retailer, Wholesaler, Library, and Resellers, call Cengage Learning at 800-354-9706; K-12 Schools, Secondary and Vocational Schools, Adult Education, and School Districts, call Cengage Learning at 800-354-9706. In Canada, call Nelson Cengage Learning at 800-268-2222.

Instructor Resources Disc

The Instructor Resources disc (0-538-48163-3) for this textbook includes both teaching and testing aids. The contents of the disc are listed below:

- **Instructor's Manual** Includes lecture notes summarizing the chapter sections, figures and boxed elements found in every chapter, teacher tips, classroom activities, lab activities, and quick quizzes in Microsoft Word files.
- **Syllabus** Easily customizable sample syllabus that covers policies, assignments, exams, and other course information. Also included is a Microsoft Project file used to create the five Phase Opener Gantt charts. An instructor can use this project file to create a visual syllabus that could include additional tasks, quizzes, and projects. The project file also can be used to track class progress through the course. Instructors are welcome to distribute this file to students, and show them how to manage tasks, resources, and deadlines for team projects that might be assigned.
- **PowerPoint Presentations** A multimedia lecture presentation system provides slides for each chapter, based on chapter objectives.
- **Figure Files** Illustrations for every figure in the textbook in electronic form.
- **Solutions to Exercises** Includes solutions for end-of-chapter exercises, including Ready for a Challenge Practice and Challenge Task solutions, chapter reinforcement exercises, and extra case studies.
- **Test Bank & Test Engine** Test Banks include 112 questions for every chapter, and feature objective-based and critical thinking question types, page number references, and figure references when appropriate. The ExamView test engine is the ultimate tool for your testing needs.
- **Additional Activities for Students** The forms that students can use to complete the Case Studies are included. Two additional case studies are also provided for every chapter, to be assigned as homework, extra credit, or assessment tools. Chapter Reinforcement Exercises, which are true/false, multiple-choice, and short answer questions that help students gain confidence in the material learned are included, as are the Your Turn Practice Tasks and sample solutions.
- **Additional Faculty Files** A copy of the powerful CASE tool, Visible Analyst — Student Edition, is provided for your evaluation. Several sample solutions to case study tasks also are included. To install this program, you follow a simple registration process that entitles you to use the software and obtain support. Detailed instructions are provided on the Instructor Resources disc. Also included are Word document versions of the e-mail and voice mail messages posted for students on the SCR Web site and the Interview Summaries for the New Century Case Study.

Course Cartridge Content

Course Technology has partnered with the leading distance learning solution providers and class-management platforms today. To access this material, visit http://www.cengage.com/coursecare/cartridge/ and search for your title. Instructor resources include the following: additional case projects, sample syllabus, PowerPoint presentations, and more. For students to access this material, they must have purchased a Course Cartridge PIN-code specific to this title and your campus platform. The resources for students might include (based on instructor preferences): topic reviews, review questions, practice tests, and more. For additional information, please contact your sales representative.

SOFTWARE BUNDLING OPPORTUNITIES *Systems Analysis and Design, Ninth Edition* can be bundled with several popular software programs:

- **Visible Analyst Student Edition** Whether you are designing e-business applications, developing a data warehouse, or integrating legacy systems with new enterprise applications, Visible Analyst is a valuable software based learning tool that helps students become more marketable with its advanced, affordable, and easy to use modeling capabilities. Visible Analyst was recently awarded the "Best Systems Analysis & Design Modeling Tool" by the Indian Education Ministry. Key users include: Business Analysts who analyze the organization and design of businesses or government departments and assess business models and their integration with technology; other professions that use systems analysis and design methods and techniques include Systems Analysts, Database Engineers, Computer Scientists, and Software Engineers. Visible Analyst is a separate software tool available individually as a Student Edition or as a University Edition with concurrent floating licenses for college or university computer labs. For more information about Visible Analyst, please visit: www.visible.com/Modeler/index.htm or contact sales@visible.com.
- **Microsoft Visio** The advanced diagramming tools of Visio 2010 help you simplify complexity with dynamic, data-driven visuals and new ways to share on the Web in real time. Start by building your diagram with professional-looking templates and modern, pre-drawn shapes. Then, easily link your diagram to popular data sources (such as Excel). You'll see data automatically refresh right within your diagram, reflected in vibrant visuals such as icons, symbols, colors, and bar graphs. Finally, with just a few clicks, publish your data-linked diagram to SharePoint, and provide access to others on the Web, even if they don't have Visio. Together, simplicity, data-driven shapes, and Web sharing make Visio 2010 one of the most powerful ways to see and understand important information.
- **Microsoft Project** Microsoft® Project 2010 delivers powerful, visually enhanced ways to effectively manage a wide range of projects and programs. From meeting crucial deadlines to selecting the right resources and empowering your teams, Project 2010 offers easier and more intuitive experiences to help you simply be more productive and realize amazing results.

ACKNOWLEDGMENTS

First, special thanks to Deb Kaufmann, our development editor, who made an enormous contribution to this edition. Her insight and suggestions were extremely valuable. Thanks to Larry Brock, Andrew Page, and Ron Savilla. As former students at Central Piedmont Community College, their ideas were especially valuable, and helped shape the new edition. Thanks also to David Rosenblatt, who contributed to an earlier edition of *Systems Analysis and Design,* and returned to help with the Ninth Edition.

Finally, thanks to our students for their feedback and comments. They suggested that we add additional *Video Learning Sessions* and interactive content such as *Ready for a Challenge* and *Your Turn*. We hope they continue to offer suggestions, and we will certainly continue to listen to them.

ABOUT OUR COVERS

The Shelly Cashman Series is continually updating our approach and content to reflect the way today's students learn and experience new technology. This focus on student success is reflected on our covers, which feature real students from Bryant University using the Shelly Cashman Series in their courses, and reflect the varied ages and backgrounds of the students learning with our books. When you use the Shelly Cashman Series, you can be assured that you are learning computer skills using the most effective courseware available.

PHASE 1 SYSTEMS PLANNING

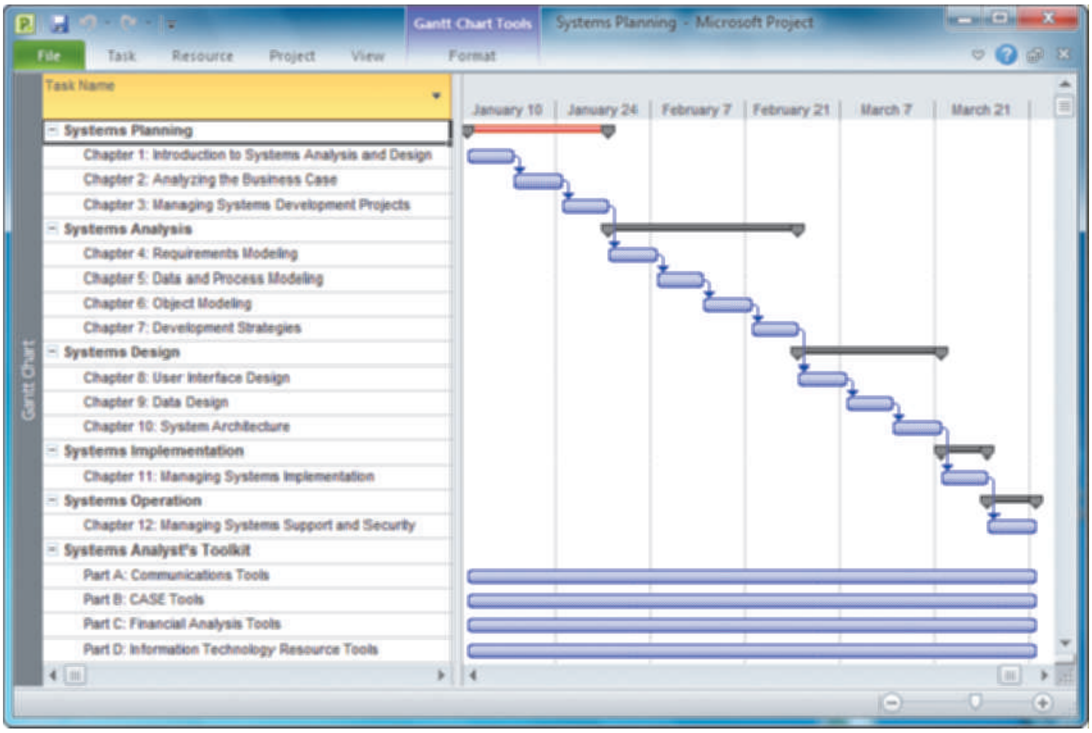
DELIVERABLE
Preliminary investigation report

TOOLKIT SUPPORT
Communications and financial analysis tools

VIDEO LEARNING SESSIONS
Project Management Concepts

As the Dilbert cartoon suggests, it is always a good idea to know whether a project fits the company's overall strategy. You will learn more about the relationship between systems projects and corporate strategies in the systems planning phase.

Systems planning is the first of five phases in the systems development life cycle. After an introduction to systems analysis and design, you will learn how systems projects get started, how to evaluate a project proposal to determine its feasibility, and how to use project management tools and techniques. The deliverable for this phase is the preliminary investigation report.



CHAPTER

Introduction to Systems Analysis and Design

Chapter 1 is the first of three chapters in the systems planning phase. This chapter describes the role of information technology in today's dynamic business environment. In this chapter, you will learn about the development of information systems, systems analysis and design concepts, and various systems development methods. This chapter also describes the role of the information technology department and its people.

INTRODUCTION

OBJECTIVES

When you finish this chapter, you will be able to:

- Describe the impact of information technology on business strategy and success
- Define an information system and describe its components
- Explain how profiles and models can represent business functions and operations
- Explain how the Internet has affected business strategies and relationships
- Identify various types of information systems and explain who uses them
- Distinguish between structured analysis, object-oriented analysis, and agile methods
- Compare the traditional waterfall model with agile methods and models
- Apply five basic guidelines for systems development
- Discuss the role of the information technology department and the systems analysts who work there

The headlines in Figure 1-1 offer dramatic examples of how information technology affects our society. Companies use information as a weapon in the battle to increase productivity, deliver quality products and services, maintain customer loyalty, and make sound decisions. In a global economy with intense competition, information technology can mean the difference between success and failure.

FIGURE 1-1 These headlines show the enormous impact of information technology in the twenty-first century.

CHAPTER INTRODUCTION CASE: Mountain View College Bookstore

Background: Mountain View College is located in New England. The school has grown rapidly and now has 8,000 students at three campuses, each with a branch bookstore. Wendy Lee, manager of college services, is responsible for all bookstore operations. Wendy wants a new information system that will increase efficiency and improve customer service.

As the case begins, Tina Allen, a systems analyst in the college's Information Technology department, is talking with David Conroe. David is majoring in information systems at Mountain View College and is earning credit toward his degree by working part-time as a student intern.

Participants: Tina and David
Location: Tina's office, 10 a.m. Monday morning, August 22, 2011
Project status: Initial discussion
Discussion topics: Basic systems development concepts

Tina: Welcome aboard, David.

David: *I'm glad to be here. What's on the agenda?*

Tina: Well, there's been some talk about a new bookstore information system. Wendy says nothing is definite yet, but she suggested that we should get ready.

David: *So we start by learning about the bookstore business?*

Tina: Yes, the best system in the world isn't worth much unless it supports business and information needs. But let's not get ahead of ourselves. First, we need to talk about business information systems in general. Then we'll build a business model so we can understand the specific operations and processes at the bookstore. We'll also discuss systems analysis and design tools and techniques. Let's start with an overview of information systems and their characteristics.

David: *That makes sense. What about the basic systems analysis techniques you mentioned? Can you tell me a bit more?*

Tina: On this project, we'll use what's called a structured method, which is based on the concept of a systems development life cycle, or SDLC for short. I'll also explain object-oriented and agile methods, and you'll learn about modeling tools and techniques. We'll follow a set of basic system development guidelines as we go along.

David: *How does the SDLC work?*

Tina: The SDLC is like constructing a building. First, you would list specific objectives for the project. Then, you might hire an architect to create drawings that show the finished building. Later, you'd need detailed blueprints for the construction workers. When the building is done, you would check everything, turn it over to the new owners, and make sure they're happy with the results.

David: *And that's how we'll develop new information systems?*

Tina: It sure is. We'll use a program called Microsoft Project to create a list of tasks we can work on.

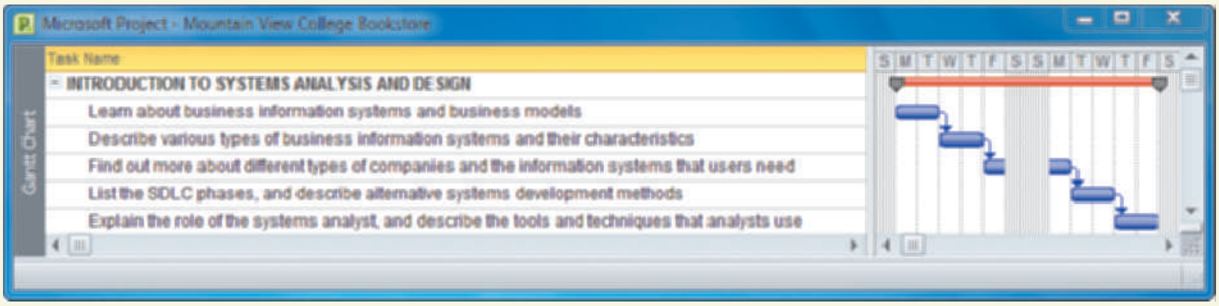


FIGURE 1-2 Typical introductory tasks for systems projects

The Impact of Information Technology

Information technology (IT) refers to the combination of hardware, software, and services that people use to manage, communicate, and share information. Although fictitious, the bold headlines in Figure 1-1 show the huge impact of IT on our society.

More than ever, business success depends on information technology. IT is driving a new digital economy, where advances in hardware, software, and connectivity can provide enormous benefits to businesses and individuals. Although economic trends affect IT spending levels, most firms give IT budgets a high priority, in good times or bad. The reason is simple — during periods of growth, companies cannot afford to lag behind the IT curve. Conversely, when the economy slows down, firms often use IT to reduce operating costs and improve efficiency.

The Future

ON THE WEB

To learn more about the future of IT, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate The Future of IT link.

If you ask a group of IT professionals to name a company that has been successful for over 100 years, the answer probably would be IBM. As its name suggests, International Business Machines was a major supplier of typewriters and data processing equipment long before the computer era. As a longtime IT leader, IBM's predictions deserve close attention.

The company that became IBM was founded in 1896 by Herman Hollerith, who invented the punched card system shown in Figure 1-3. Hollerith wanted to analyze the 1890 census data, and his idea was brilliant. First, letters and numbers were coded by being punched into specific locations on a card. His machines then used a simple

FIGURE 1-3 Several years before the Wright brothers made their first flight, Herman Hollerith devised a machine that could manage data stored on punched cards.

electrical circuit to detect the position of the holes and recognize the characters. This concept transformed a set of punched cards into what we would call a database, which could be sorted, queried, and printed. Punched card technology revolutionized data storage, and was the main form of information management well into the 1960s and beyond. The concept still is used with some types of voting forms and other documents.

Today, IBM is a global giant with a half-million employees, numerous patents, and more Nobel Prize winners than any other IT firm. In its 2009 Annual Report, IBM chairman Samuel J. Palmisano noted three issues that will shape the future of IT and guide the company's strategy. As shown in Figure 1-4, these include changes in the world, changes in technology, and changes in client demand. IT professionals should study these trends and prepare for the future. The table in Figure 1-5 on the next page summarizes IBM's vision and how it might affect the IT industry and the people who work in it.

1. Changes in the World: The lowering of trade barriers, the rise of the developing world and the emergence of the World Wide Web were unleashing the flow of work on a global scale. We believed these changes were powerful and irreversible, and that they would lead to new business models and a new form of the corporation itself—what we came to call the globally integrated enterprise.

2. Changes in Technology: At the same time, a new model of computing was replacing the PC-based, client/server approach. Computational capability was being put into things no one would recognize as computers: phones, cameras, cars, appliances, roadways, power lines, clothes—and even natural systems, such as agriculture and rivers. All of this was being connected through the Internet. And we now had the computing power, advanced analytics and new models (now known as "clouds") to turn mountains of data into insight. As a result, the economic, societal and physical systems of the world were becoming instrumented, interconnected and intelligent. Our planet was becoming smarter.

3. Changes in Client Demand: Compelled by the new opportunities and competitive demands of these first two shifts, enterprises and institutions were no longer content with cost savings from off-the-shelf technologies and solutions. They now sought to innovate—not just in their products and services, but also their business processes, management systems, policies and core business models. To accomplish that, they needed to integrate advanced technology far deeper into their operations.

FIGURE 1-4 The 2009 IBM Annual Report points out three key issues that affect the company's strategy and operational plans.

Systems Development

Business information systems are developed by people who are technically qualified, business-oriented, and highly motivated. Successful developers also must be good communicators with strong analytical and critical thinking skills.

TOPIC	IBM'S VISION	HOW WILL THIS AFFECT IT JOBS GENERALLY?	WHAT WILL BE THE IMPACT ON FUTURE SYSTEMS ANALYSTS?
Changes in the World	IBM foresees a new kind of corporation: a globally integrated enterprise, driven by the Internet and free of traditional trade barriers.	Language skills will be extremely important — the more the better. Diversity will open new opportunities, and developing countries will be able to compete more effectively. Where the physical work is done will be less important than how the virtual company deploys its assets.	Systems analysts will be affected by global trends. They will probably work for more firms in their careers, be exposed to more information, and see greater change than at any time in history.
Changes in Technology	Powered by an enormous increase in computing power, new IT models will include networks with smart, interconnected devices such as communication systems, automobiles, entertainment, highway infrastructure, and power grids.	Technical skills will be in demand, but so will the ability to "think outside the box." New technology will drive major changes in how personal and business services are provided. Firms will compete in a global marketplace that will reward innovation, creativity, and positive societal outcomes.	The systems analyst will need to have both business savvy and technical skills. He or she will have a unique opportunity to work at the intersection of business operations and information technology. The synergy between technology growth and globalization will create jobs and opportunities for people with the right skills.
Changes in Client Demand	In the face of worldwide competition and enormous technology change, firms will stress innovation, vision, and the ability to adapt rapidly. Every aspect of their business plans, processes, and operations will be affected.	Successful IT workers must be able to innovate, analyze, and communicate effectively. The winners will be those who can adapt to change and embrace new technology and new ways of doing business.	Students preparing for the workplace of tomorrow will need a strong skill set. Systems analysts will be expected to bring communications, modeling, problem-solving, decision-making, and critical thinking skills to the workplace — and to be aware of ethical issues that might affect them.

FIGURE 1-5 If IBM's vision is accurate, what will it mean to IT professionals?

Systems Analysis and Design

Systems analysis and design is a step-by-step process for developing high-quality information systems. An **information system** combines information technology, people, and data to support business requirements. For example, information systems handle daily business transactions, improve company productivity, and help managers make sound decisions. The IT department team includes **systems analysts** who plan, develop, and maintain information systems.

With increasing demand for talented people, employment experts predict a shortage of qualified applicants to fill IT positions. Many companies list employment opportunities on their Web sites, as shown in Figure 1-6.

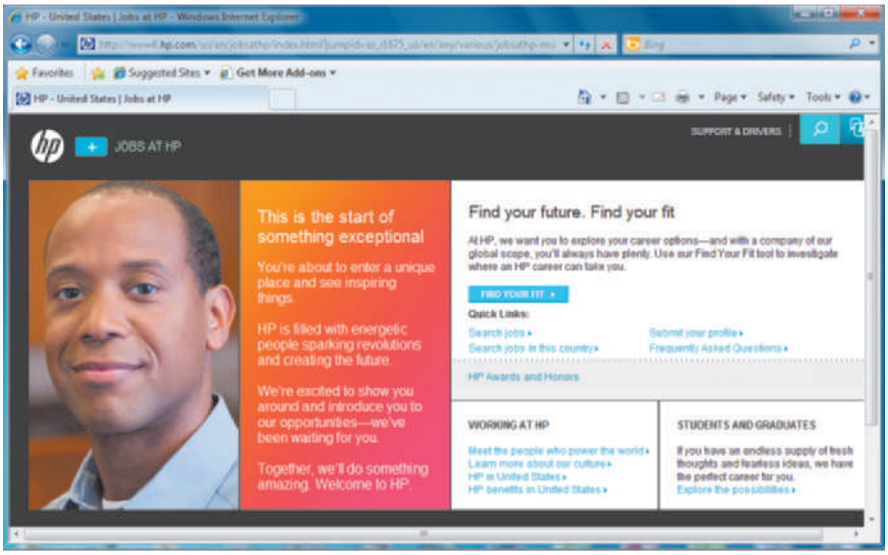


FIGURE 1-6 HP invites potential candidates to search for jobs on its Web site.

Who Develops Information Systems?

Traditionally, a company either developed its own information systems, called **in-house applications**, or purchased systems called **software packages** from outside vendors. Today, the choice is much more complex. Options include Internet-based application services, outsourcing, custom solutions from IT consultants, and enterprise-wide software strategies.

Regardless of the development method, launching a new information system involves risks as well as benefits. The greatest risk occurs when a company tries to decide *how* the system will be constructed before determining *what* the system needs to do. Instead of putting the cart before the horse, a company must begin by outlining its business needs and identifying possible IT solutions. Typically, this important work is performed by systems analysts and other IT professionals. A firm should not consider implementation options until it has a clear set of objectives. Later on, as the system is developed, a systems analyst's role will vary depending on the implementation option selected.

Information System Components

A **system** is a set of related components that produces specific results. For example, specialized systems route Internet traffic, manufacture microchips, and control complex entities like the Mars Rover shown in Figure 1-7. A **mission-critical system** is one that is vital to a company's operations. An order processing system, for example, is mission-critical because the company cannot do business without it.

Every system requires input data. For example, your computer receives data when you press a key or click a menu command. In an information system, **data** consists of basic facts that are the system's raw material. **Information** is data that has been transformed into output that is valuable to users.

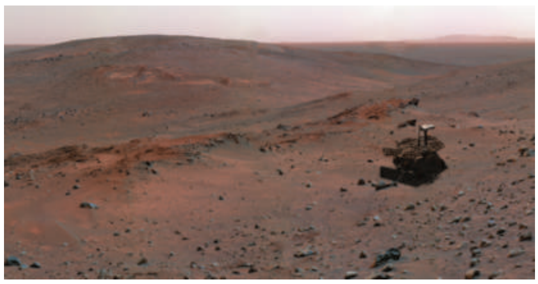


FIGURE 1-7 Imagine the complexity of the systems used to launch and operate the Mars Rover.

An information system has five key components, as shown in Figure 1-8: hardware, software, data, processes, and people.

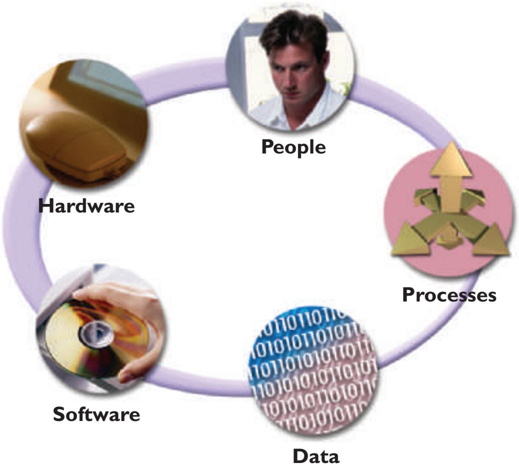


FIGURE 1-8 The five main components of an information system.

Hardware

Hardware consists of everything in the physical layer of the information system. For example, hardware can include servers, workstations, networks, telecommunications equipment, fiber-optic cables, mobile devices, scanners, digital capture devices, and other technology-based infrastructure. As new technologies emerge, manufacturers race to market the innovations and reap the rewards.

Hardware purchasers today face a wide array of technology choices and decisions. In 1965, Gordon Moore, a cofounder of Intel, predicted that the number of transistors on an integrated circuit would double about every 24 months. His concept, called **Moore's Law**, has remained valid for more than 50 years. Fortunately, as hardware became more powerful, it also became much less expensive. Large businesses with thousands or millions of sales transactions require company-wide information systems and powerful servers, such as those shown in Figure 1-9.

FIGURE 1-9 Multiple servers provide the power and speed that modern IT systems require.

Software

Software refers to the programs that control the hardware and produce the desired information or results. Software consists of system software and application software.

System software manages the hardware components, which can include a single workstation or a global network with many thousands of clients. Either the hardware manufacturer supplies the system software or a company purchases it from a vendor. Examples of system software include the operating system, security software that protects the computer from intrusion, device drivers that communicate with hardware such as printers, and utility programs that handle specific tasks such as data backup and disk management. System software also controls the flow of data, provides data security, and manages network operations. In today's interconnected business world, network software is vitally important.

Application software consists of programs that support day-to-day business functions and provide users with the information they require. Application software can serve one user or thousands of people throughout an organization. Examples of company-wide applications, called **enterprise applications**, include order processing systems, payroll systems, and company communications networks. On a smaller scale, individual users increase their productivity with tools such as spreadsheets, word processors, and database management systems.

Application software includes horizontal and vertical systems. A **horizontal system** is a system, such as an inventory or payroll application, that can be adapted for use in many different types of companies. A **vertical system** is designed to meet the unique requirements of a specific business or industry, such as a Web-based retailer, a medical practice, or a video chain.

ON THE WEB

To learn more about Moore's Law, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Moore's Law link.

Most companies use a combination of software that is acquired at various times. When planning an information system, a company must consider how a new system will interface with older systems, which are called **legacy systems**. For example, a new human resources system might need to exchange data with an older payroll application.

Data

Data is the raw material that an information system transforms into useful information. An information system can store data in various locations, called tables. By linking the tables, the system can extract specific information. Figure 1-10 shows a payroll system that stores data in four separate tables. Notice that the linked tables work together to supply 19 different data items to the screen form. Users, who would not know or care where the data is stored, see an integrated form, which is their window into the payroll system.

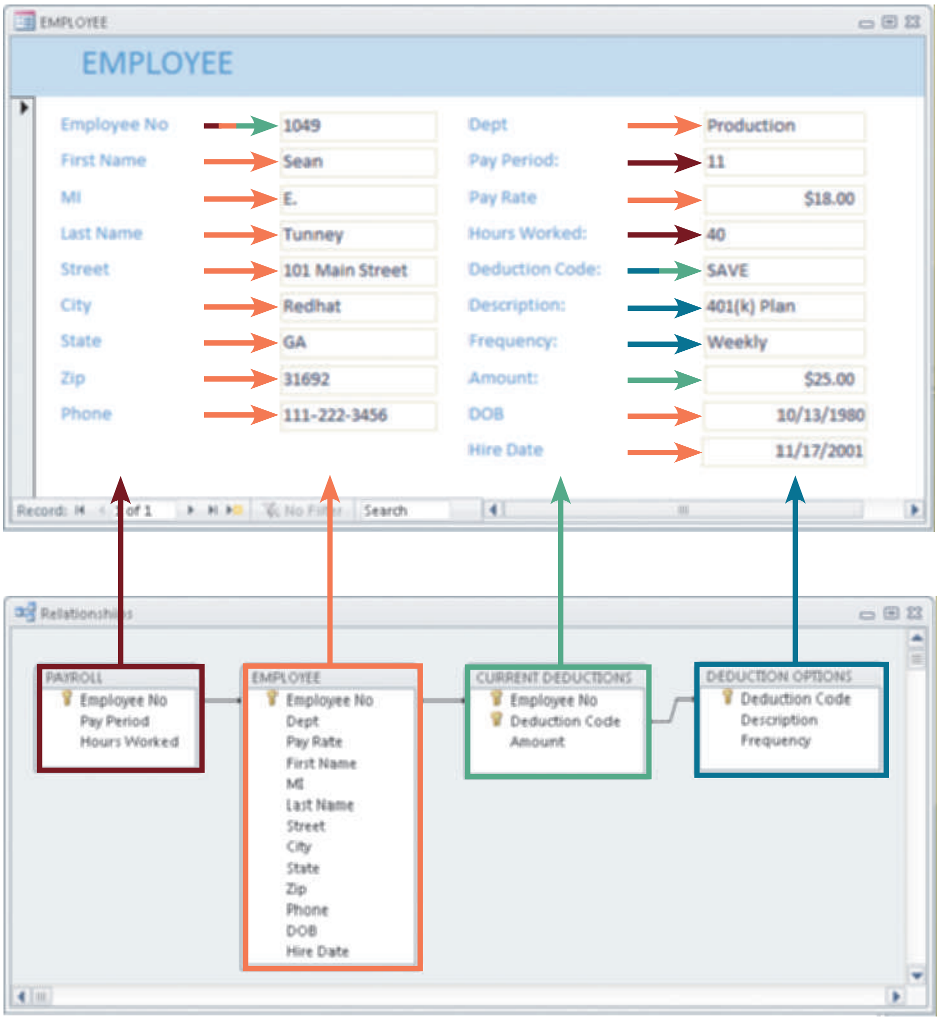


FIGURE 1-10 In a typical payroll system, data is stored in separate tables, which are joined together to form a database that contains all the information.

Processes

Processes describe the tasks and business functions that users, managers, and IT staff members perform to achieve specific results. Processes are the building blocks of an information system because they represent actual day-to-day business operations. To build a successful information system, analysts must understand business processes and document them carefully.

People

People who have an interest in an information system are called **stakeholders**. Stakeholders include the management group responsible for the system, the **users** (sometimes called **end users**) inside and outside the company who will interact with the system, and IT staff members, such as systems analysts, programmers, and network administrators who develop and support the system.

Each stakeholder group has a vital interest in the information system, but most experienced IT professionals agree that the success or failure of a system usually depends on whether it meets the needs of its users. For that reason, it is essential to understand user requirements and expectations throughout the development process.

TOOLKIT TIME

Business process modeling tools, which are described in Part B of the System's Analyst's Toolkit, can help you document and describe business operations. To learn more about these tools, turn to Part B of the four-part Toolkit that follows Chapter 12.

UNDERSTAND THE BUSINESS

IT professionals must understand a company's operations to design successful systems. Each business situation is different. For example, a retail store, a medical practice, and a hotel chain all have unique information systems requirements. Systems analysts use a process called **business process modeling** to represent company operations and information needs. Business process modeling requires a business profile and a series of models that document business processes.

As the business world changes, systems analysts can expect to work in new kinds of companies that require innovative IT solutions, including Web-based systems that serve customers and carry out online transactions with other businesses.

Business Profile

A **business profile** is an overview of a company's mission, functions, organization, products, services, customers, suppliers, competitors, constraints, and future direction. Although much of this information is readily available, a systems analyst usually needs to do additional research and fact-finding. A business profile is the starting point for the modeling process.

Business Process

A **business process** is a specific set of transactions, events, and results that can be described and documented. A **business process model** (**BPM**) graphically displays one or more business processes, such as handling an airline reservation, filling a product order, or updating a customer account. The example in Figure 1-11 shows a simple model that includes an event, three processes, and a result.

A rough sketch might be sufficient to document a simple business process. For complex operations, however, analysts apply computer-based modeling tools that use a standard language called **business process modeling notation** (**BPMN**). BPMN includes various shapes and symbols to represent events, processes, and workflows, as shown in the example in Figure 1-12. Modeling tools include multi-purpose graphical applications, such as Microsoft Visio, and computer-aided software engineering programs such as Visible Analyst. Business process modeling is described in more detail in Part B of the Systems Analyst's Toolkit that follows Chapter 12.

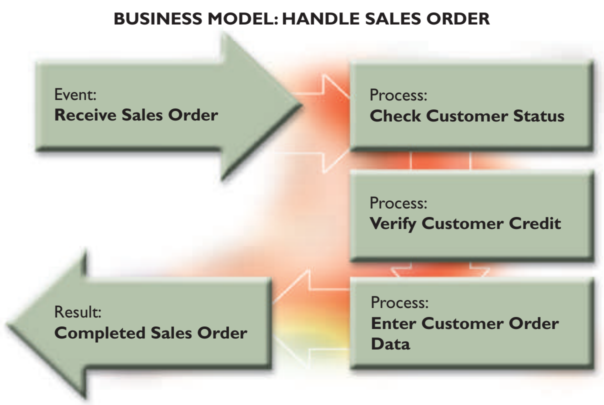


FIGURE 1-11 A simple business model might consist of an event, three processes, and a result.

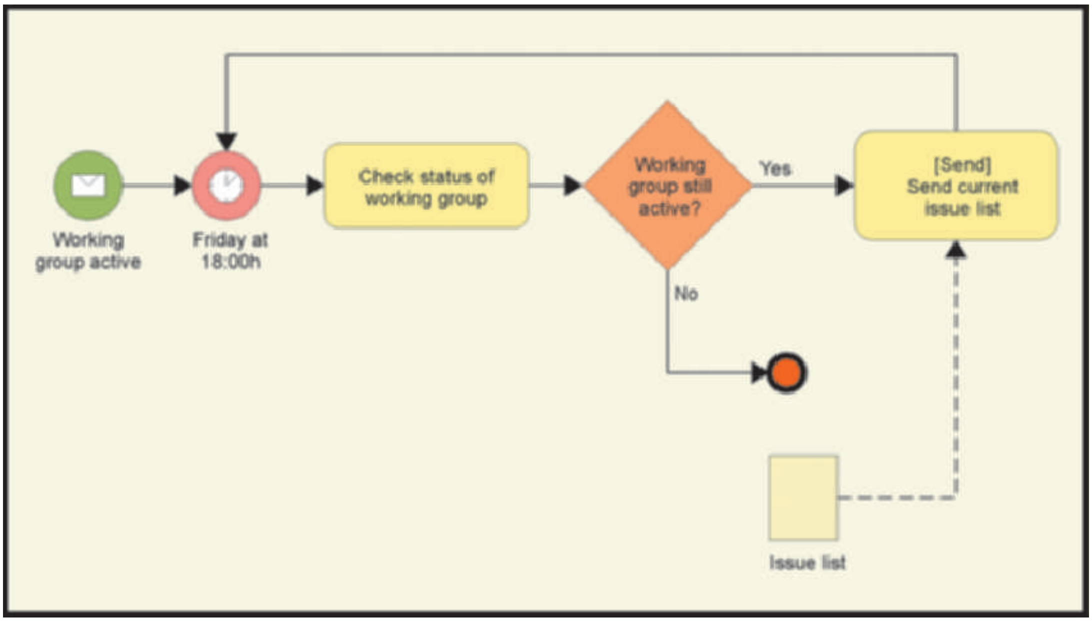


FIGURE 1-12 This sample uses business process modeling notation (BPMN) to represent events, processes, and workflow.

New Kinds of Companies

Traditionally, IT companies were identified as product-oriented or service-oriented. **Product-oriented** firms manufactured computers, routers, or the microchips shown in Figure 1-13, whereas **service-oriented** companies included vendors, software developers, and service providers.

Today, those distinctions are gone. Most successful IT companies offer a mix of products, services, and support. Value-added services such as consulting, financing, and technical support can be more profitable than hardware. In a striking example of this trend, IBM stated in its 2009 annual report that software, services, and financing produced 93 percent of pre-tax income, while hardware accounted for only 7 percent. Even more interesting is the contrast shown in Figure 1-14. Over a nine-year period, hardware profits declined by two-thirds, while software, services and financing income grew rapidly.

FIGURE 1-13 Intel is an example of a product-oriented company that manufactures technology products, such as the microchip shown here.

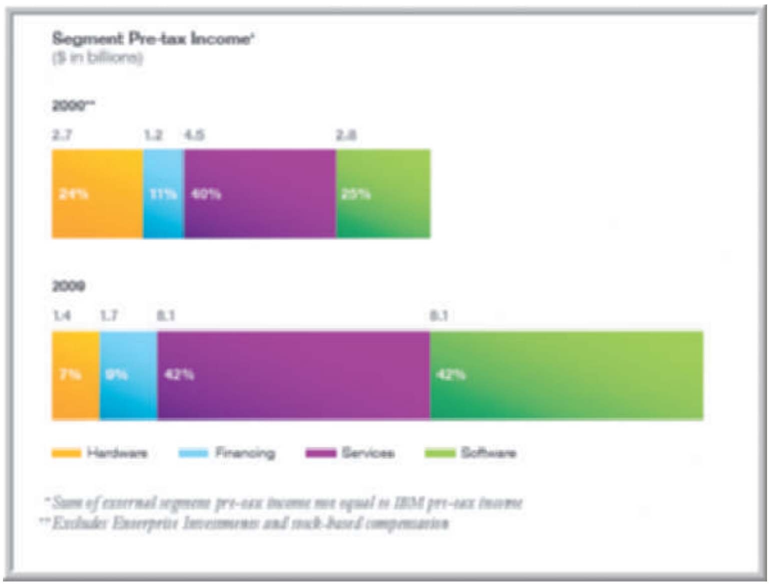


FIGURE 1-14 IBM's 2009 report shows a sharp increase in profitable business segments such as software and services, and a decline in profit from hardware sales.
Source: IBM Annual Report, 2009.

The newest kind of company is the **Internet-dependent** firm, often described as a **dot-com** (**.com**) company because its primary business depends on the Internet rather than a traditional business channel. Google, Yahoo!, Amazon, and eBay are examples of pure dot-com companies. At the other end of the scale are traditional firms, sometimes called **brick-and-mortar** companies because they conduct business primarily from physical locations. Most successful brick-and-mortar firms — such as Lowe's, Target, and Wal-Mart — have expanded their Web-based marketing channels to increase sales and serve customers better. This strategy combines the convenience of online shopping *and* the alternative of hands-on purchasing for customers who prefer that option.

Today, with rising fuel prices and an eye on expenses, shopping at home is more popular than ever. In addition to physical products, consumers also buy many types of digital content. As shown in Figure 1-15, Netflix is an example of a Web-based firm that has seen a sharp increase in sales. Some of the growth came at the expense of brick-and-mortar competitors. Perhaps more importantly, Netflix has been a leader in delivering streaming Internet content that can displayed on large screen TV sets. The company claims to provide maximum convenience and value to consumers of home-based entertainment.

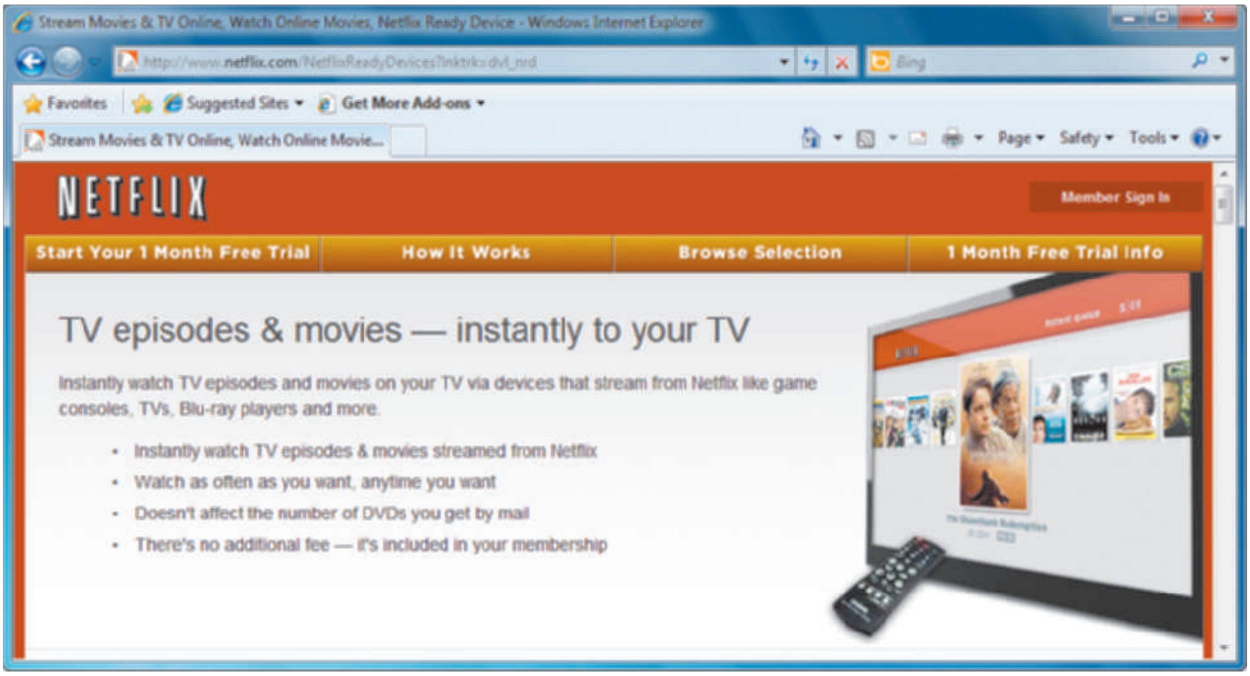


FIGURE 1-15 Netflix offers an Internet portal for digital content that can be viewed on large screen TVs.

CASE IN POINT 1.1: CLOUD NINE FINANCIAL ADVISORS

Cloud Nine provides its clients with a monthly newsletter that offers recommendations about stocks to buy or sell. Doug Layton, Cloud Nine's president, has asked your opinion on whether dot-com stocks might be good investments for the future. He specifically mentioned Google, eBay, Amazon.com, and Yahoo!, but he said you could suggest other companies. Doug wants you to do some Internet research to learn more about these Web-based companies and their future prospects. You can use a search engine, or start by visiting the Web sites of publications such as *Forbes*, *Fortune Magazine*, *Business Week*, or *The Wall Street Journal*, among others.

Impact of the Internet

Internet-based commerce is called **e-commerce** (**electronic commerce**) or **I-commerce** (**Internet commerce**).

Internet-based systems involve various hardware and software designs, but a typical model is a series of Web pages that provides a user interface, which communicates with database management software and a Web-based data server. As Internet-based commerce continues to grow, career opportunities will expand significantly for IT professionals such as Web designers, database developers, and systems analysts. The surge in demand will come from dot-com companies large and small, and from mainstream retailers.

E-commerce includes two main sectors: **B2C** (**business-to-consumer**) and **B2B** (**business-to-business**).

B2C (Business-to-Consumer)

Using the Internet, consumers can go online to purchase an enormous variety of products and services. This new shopping environment allows customers to do research, compare prices and features, check availability, arrange delivery, and choose payment methods in a single convenient session. Many companies, such as airlines, offer incentives for online transactions because Web-based processing costs are lower than traditional methods. By making flight information available online to last-minute travelers, some airlines also offer special discounts on seats that might otherwise go unfilled.

B2C commerce is changing traditional business models and creating new ones. For example, a common business model is a retail store that sells a product to a customer. To carry out that same transaction on the Internet, the company must develop an online store and deal with a totally different set of marketing, advertising, and profitability issues. Some companies have found new ways to use established business models. For example, eBay.com has transformed a traditional auction concept into a popular and successful method of selling goods and services.

In the e-commerce battles, the real winners will be online consumers, who will have access to more information, better choices, and an enhanced shopping experience. For example, in addition to the traditional offerings, the Lowe's Web site shown in Figure 1-16 includes a gift advisor, buying guides, how-to clinics, and interactive design tools.

ON THE WEB

To learn more about electronic commerce, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Electronic Commerce link.

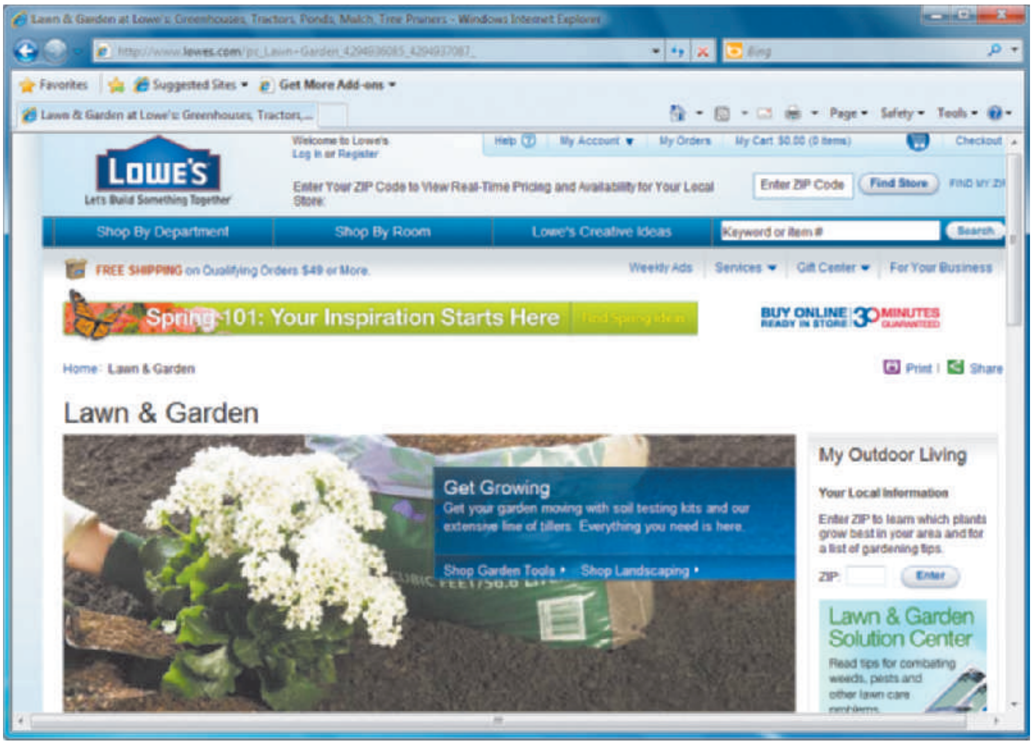


FIGURE 1-16 Lowe's is an example of a mainstream retailer that effectively combines traditional and online marketing strategies.

B2B (Business-to-Business)

Although the business-to-consumer (B2C) sector is more familiar to retail customers, the volume of business-to-business (B2B) transactions is many times greater. Industry observers predict that B2B sales will increase sharply as more firms seek to improve efficiency and reduce costs.

Initially, electronic commerce between two companies used a data sharing arrangement called **electronic data interchange** (**EDI**). EDI enabled computer-to-computer data transfer, usually over private telecommunications lines. Firms used EDI to plan production, adjust inventory levels, or stock up on raw materials using data from another company's information system. As B2B volume soared, the development of **extensible markup language** (**XML**) enabled company-to-company traffic to migrate to the Internet, which offered standard protocols, universal availability, and low communication costs. XML is a flexible data description language that allows Web-based communication between different hardware and software environments.

ON THE WEB

To learn more about XML, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Extensible Markup Language link.

Because it allows companies to reach the global marketplace, B2B is especially important to smaller suppliers and customers who need instant information about market prices and availability. On an industry-wide scale, many B2B sites exist where buyers, sellers, distributors, and manufacturers can offer products, submit specifications, and transact business. This popular form of online B2B interaction is called **supply chain management** (**SCM**), or **supplier relationship management** (**SRM**). Figure 1-17 shows a software vendor that offers SCM solutions designed to reduce supply chain costs.

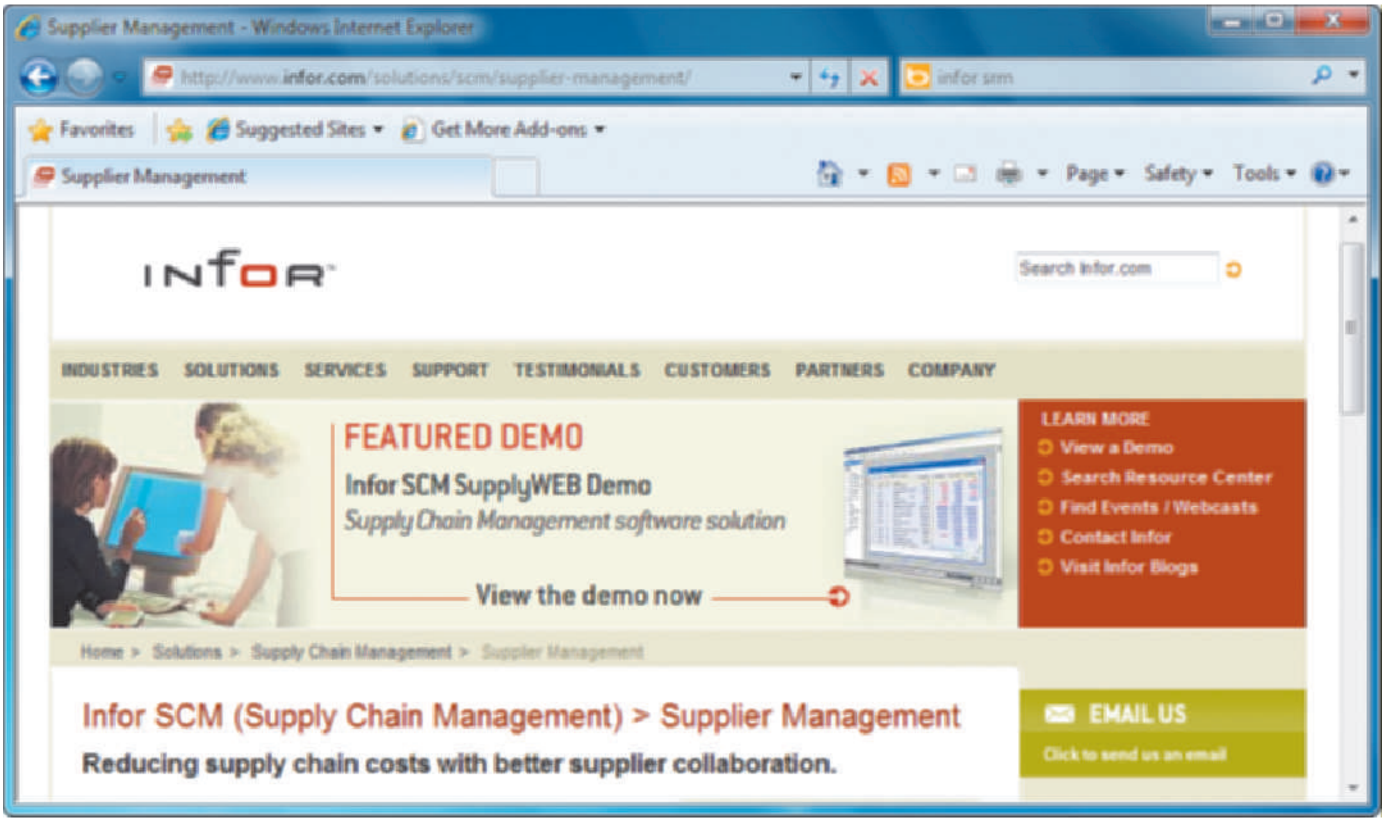


FIGURE 1-17 Infor is a software vendor that offers SCM solutions based on real-time supplier collaboration.

Business Information Systems

In the past, IT managers divided systems into categories based on the user group the system served. Categories and users included office systems (administrative staff), operational systems (operational personnel), decision support systems (middle-managers and knowledge workers), and executive information systems (top managers).

Today, traditional labels no longer apply. For example, all employees, including top managers, use office productivity systems. Similarly, operational users often require decision support systems. As business changes, information use also changes in most companies. Today, it makes more sense to identify a system by its functions and features, rather than by its users. A new set of system definitions includes enterprise computing systems, transaction processing systems, business support systems, knowledge management systems, and user productivity systems.

Enterprise Computing

Enterprise computing refers to information systems that support company-wide operations and data management requirements. Wal-Mart's inventory control system, Boeing's production control system, and Hilton Hotels' reservation system are examples of enterprise computing systems. The main objective of enterprise computing is to integrate a company's primary functions (such as production, sales, services, inventory control, and accounting) to improve efficiency, reduce costs, and help managers make key decisions. Enterprise computing also improves data security and reliability by imposing a company-wide framework for data access and storage.

In many large companies, applications called **enterprise resource planning (ERP)** systems provide cost-effective support for users and managers throughout the company. For example, a car rental company can use ERP to forecast customer demand for rental cars at hundreds of locations.

ON THE WEB

To learn more about enterprise resource planning, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Enterprise Resource Planning link.

By providing a company-wide computing environment, many firms have been able to achieve dramatic cost reductions. Other companies have been disappointed in the time, money, and commitment necessary to implement ERP successfully. A potential disadvantage of ERP is that ERP systems generally impose an overall structure that might or might not match the way a company operates. ERP is described in more detail in Chapter 7, which discusses system development strategies.

Because of its growth and potential, many hardware and software vendors target the enterprise computing market and offer a wide array of products and services. Figure 1-18 shows a Web site that is dedicated to marketing enterprise computing software and solutions.

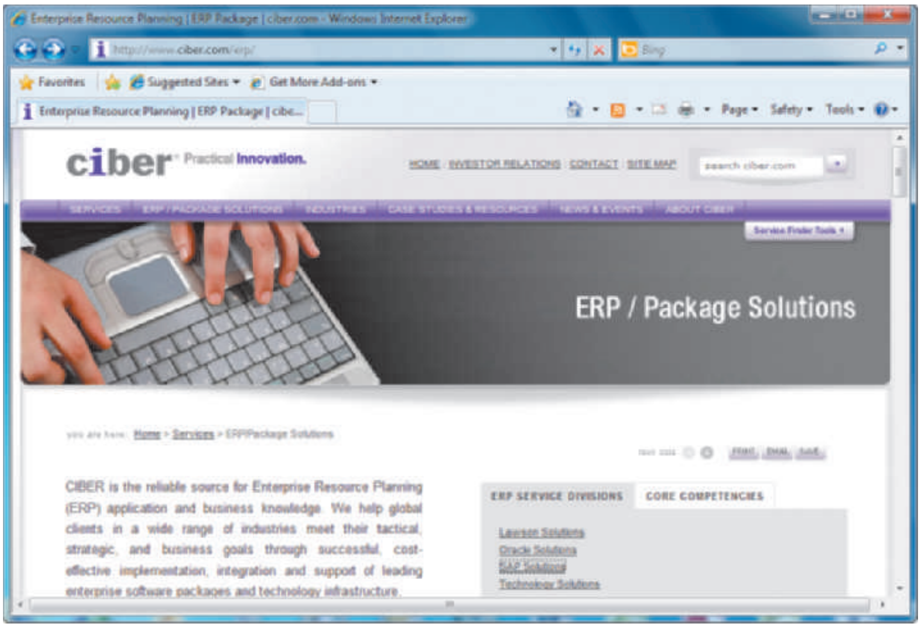


FIGURE 1-18 Ciber offers ERP applications and consulting, using software from leading software providers.

Transaction Processing

Transaction processing (TP) systems process data generated by day-to-day business operations. Examples of TP systems include customer order processing, accounts receivable, and warranty claim processing.

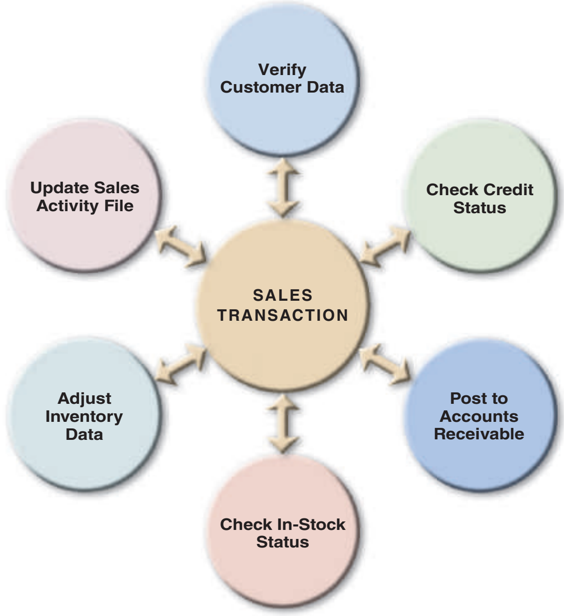


FIGURE 1-19 A single sales transaction consists of six separate tasks, which the TP system processes as a group.

TP systems perform a series of tasks whenever a specific transaction occurs. In the example shown in Figure 1-19, a TP system verifies customer data, checks the customer's credit status, posts the invoice to the accounts receivable system, checks to ensure that the item is in stock, adjusts inventory data to reflect a sale, and updates the sales activity file. TP systems typically involve large amounts of data and are mission-critical systems because the enterprise cannot function without them.

TP systems are efficient because they process a set of transaction-related commands as a group rather than individually. To protect data integrity, however, TP systems ensure that if any single element of a transaction fails, the system does not process the rest of the transaction.

Business Support

Business support systems provide job-related information support to users at all levels of a company. These systems can analyze transactional data, generate information needed to manage and control business processes, and provide information that leads to better decision-making.

The earliest business computer systems replaced manual tasks, such as payroll processing. Companies soon realized that computers also could produce valuable information. The new systems were called **management information systems** (**MIS**) because managers were the primary users. Today, employees at *all* levels need information to perform their jobs, and they rely on information systems for that support.

A business support system can work hand in hand with a TP system. For example, when a company sells merchandise to a customer, a TP system records the sale, updates the customer's balance, and makes a deduction from inventory. A related business support system highlights slow- or fast-moving items, customers with past due balances, and inventory levels that need adjustment.

ON THE WEB

To learn more about RFID, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the RFID link.

To compete effectively, firms must collect production, sales, and shipping data and update the company-wide business support system immediately. The newest development in data acquisition is called **radio frequency identification** (**RFID**) technology, which uses high-frequency radio waves to track physical objects, such as the item shown in Figure 1-20. RFID's dramatic growth has been fueled by companies like Wal-Mart, which requires its suppliers to add RFID tags to all items.

An important feature of a business support system is decision support capability. Decision support helps users make decisions by creating a computer model and applying a set of variables. For example, a truck fleet dispatcher might run a series of **what-if** scenarios to determine the impact of increased shipments or bad weather. Alternatively, a retailer might use what-if analysis to determine the price it must charge to increase profits by a specific amount while volume and costs remain unchanged.

FIGURE 1-20 Retailers use RFID tags for security and inventory control.

Knowledge Management

Knowledge management systems are called **expert systems** because they simulate human reasoning by combining a knowledge base and inference rules that determine how the knowledge is applied.

A **knowledge base** consists of a large database that allows users to find information by entering keywords or questions in normal English phrases. A knowledge management system uses **inference rules**, which are logical rules that identify data patterns and relationships.

Figure 1-21 shows a knowledge management system that Toshiba maintains for its customers and users. After a user enters a symptom, problem, or question, Toshiba's Knowledge Base searches for a solution and displays the results.

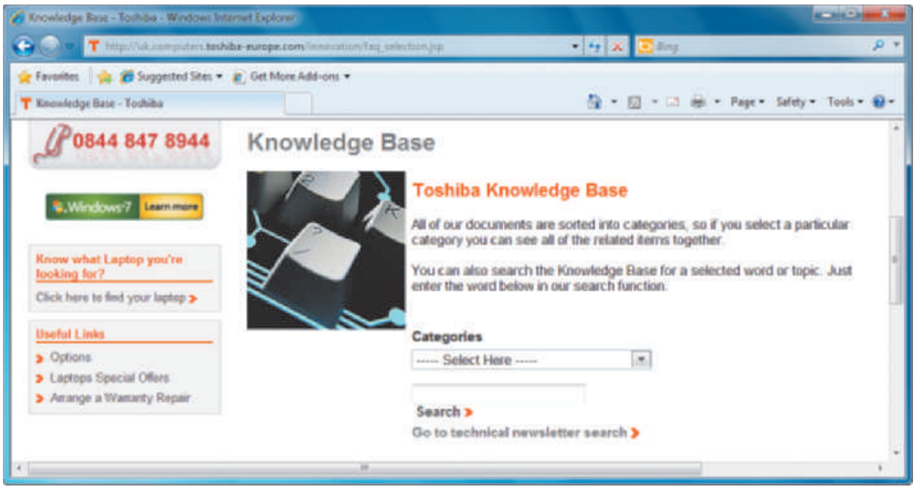


FIGURE 1-21 The interactive Toshiba Knowledge Base allows users to search for solutions.

Knowledge management systems do not use strict logical rules. Instead, many knowledge management systems use a technique called **fuzzy logic** that allows inferences to be drawn from imprecise relationships. Using fuzzy logic, values need not be black and white, like binary logic, but can be many shades of gray. The results of a fuzzy logic search will display in priority order, with the most relevant results at the top of the list.

User Productivity

Companies provide employees at all levels with technology that improves productivity. Examples of **user productivity systems** include e-mail, voice mail, fax, video and Web conferencing, word processing, automated calendars, database management, spreadsheets, desktop publishing, presentation graphics, company intranets, and high-speed Internet access. User productivity systems also include groupware. **Groupware** programs run on a company intranet and enable users to share data, collaborate on projects, and work in teams. GroupWise, offered by Novell, is a popular example of groupware.

When companies first installed word processing systems, managers expected to reduce the number of employees as office efficiency increased. That did not happen, primarily because the basic nature of clerical work changed. With computers performing most of the repetitive work, managers realized that office personnel could handle tasks that required more judgment, decision-making, and access to information.

Computer-based office work expanded rapidly as companies assigned more responsibility to employees at lower organizational levels. Relatively inexpensive hardware, powerful networks, corporate downsizing, and a move toward employee empowerment also contributed to this trend. Today, administrative assistants and company presidents alike are networked, use computer workstations, and need to share corporate data to perform their jobs.

Information Systems Integration

Most large companies require systems that combine transaction processing, business support, knowledge management, and user productivity features. For example, suppose an international customer has a problem with a product and makes a warranty claim. A customer service representative enters the claim into a TP system. The transaction updates two other systems: a knowledge management system that tracks product problems and warranty activity, and a quality control system with decision support capabilities. A quality control engineer uses what-if analysis to determine if it would be

ON THE WEB

To learn more about knowledge management systems, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Knowledge Management Systems link.

advantageous to make product design changes to reduce warranty claims. In this example, a TP system is integrated with a knowledge management system and a business support system with decision support features.

What Information Do Users Need?

Corporate organizational structure has changed considerably in recent years. As part of downsizing and business process reengineering, many companies reduced the number of management levels and delegated responsibility to operational personnel. Although modern organization charts tend to be flatter, an organizational hierarchy still exists in most companies.

A typical organizational model identifies business functions and organizational levels, as shown in Figure 1-22. Within the functional areas, operational personnel report to supervisors and team leaders. The next level includes middle managers and knowledge workers, who, in turn, report to top managers. In a corporate structure, the top managers report to a board of directors elected by the company's shareholders.

A systems analyst must understand the company's organizational model to recognize who is responsible for specific processes and decisions and to be aware of what information is required by whom.

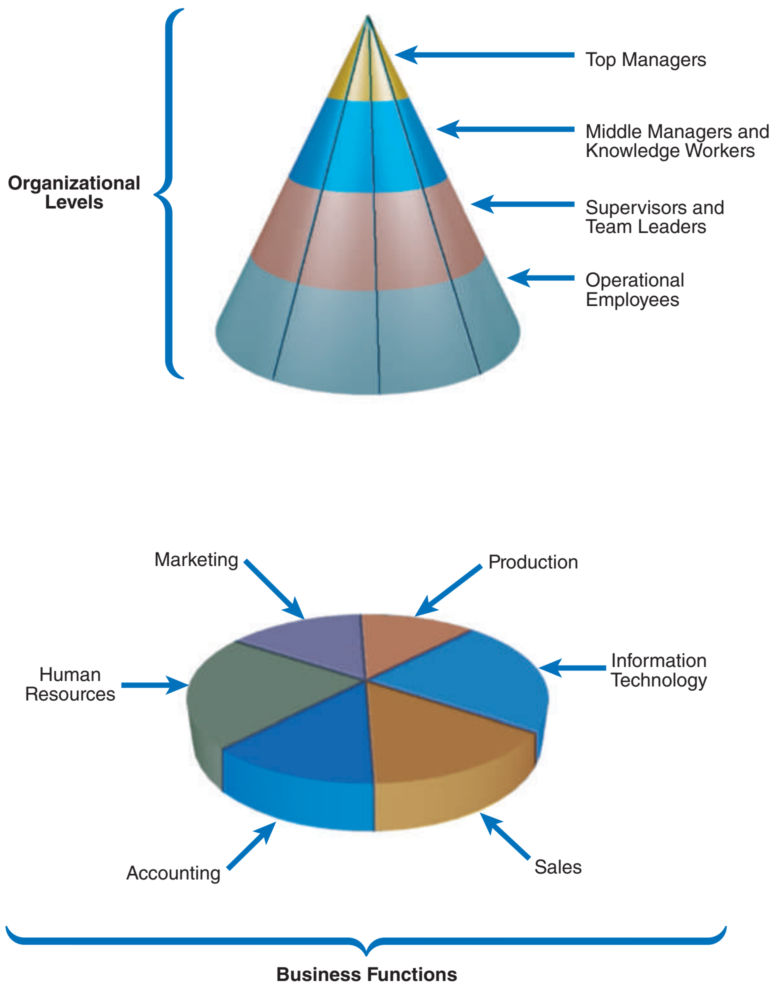


FIGURE 1-22 A typical organizational model identifies business functions and organizational levels.

Top Managers

Top managers develop long-range plans, called **strategic plans**, which define the company's overall mission and goals. To plot a future course, top managers ask questions such as "How much should the company invest in information technology?" or "How much will Internet sales grow in the next five years?" or "Should the company build new factories or contract out the production functions?"

Strategic planning affects the company's future survival and growth, including long-term IT plans. Top managers focus on the overall business enterprise and use IT to set the company's course and direction. To develop a strategic plan, top managers also need information from outside the company, such as economic forecasts, technology trends, competitive threats, and governmental issues.

Middle Managers and Knowledge Workers

Just below the top management level, most companies have a layer of middle managers and knowledge workers. Middle managers provide direction, necessary resources, and performance feedback to supervisors and team leaders. Because they focus on a somewhat shorter time frame, middle managers need more detailed information than top managers, but somewhat less than supervisors who oversee day-to-day operations. For example, a middle manager might review a weekly sales summary for a three-state area, whereas a local sales team leader would need a daily report on customer sales at a single location.

In addition to middle managers, every company has people called knowledge workers. **Knowledge workers** include professional staff members such as systems analysts, programmers, accountants, researchers, trainers, and human resource specialists. Knowledge workers also use business support systems, knowledge management systems, and user productivity systems. Knowledge workers provide support for the organization's basic functions. Just as a military unit requires logistical support, a successful company needs knowledge workers to carry out its mission.

Supervisors and Team Leaders

Supervisors, often called team leaders, oversee operational employees and carry out day-to-day functions. They coordinate operational tasks and people, make necessary decisions, and ensure that the right tools, materials, and training are available. Like other managers, supervisors and team leaders need decision support information, knowledge management systems, and user productivity systems to carry out their responsibilities.

Operational Employees

Operational employees include users who rely on TP systems to enter and receive data they need to perform their jobs. In many companies, operational users also need information to handle tasks and make decisions that were assigned previously to supervisors. This trend, called **empowerment**, gives employees more responsibility and accountability. Many companies find that empowerment improves employee motivation and increases customer satisfaction.

Systems Development Tools

In addition to understanding business operations, systems analysts must know how to use a variety of techniques, such as modeling, prototyping, and computer-aided systems engineering tools to plan, design, and implement information systems. Systems analysts work with these tools in a team environment, where input from users, managers, and IT staff contributes to the system design.

Modeling

Modeling produces a graphical representation of a concept or process that systems developers can analyze, test, and modify. A systems analyst can describe and simplify an information system by using a set of business, data, object, network, and process models.

A **business model**, or **requirements model**, describes the information that a system must provide. A **data model** describes data structures and design. An **object model** describes objects, which combine data and processes. A **network model** describes the design and protocols of telecommunications links. A **process model** describes the logic that programmers use to write code modules. Although the models might appear to overlap, they actually work together to describe the same environment from different points of view.

System developers often use multipurpose charting tools such as Microsoft Visio to display business-related models. Visio is a popular tool that systems analysts can use to create business process diagrams, flowcharts, organization charts, network diagrams, floor plans, project timelines, and work flow diagrams, among others.

TOOLKIT TIME

The CASE tools in Part B of the Systems Analyst's Toolkit can help you develop and maintain complex information systems. To learn more about these tools, turn to Part B of the four-part Toolkit that follows Chapter 12.

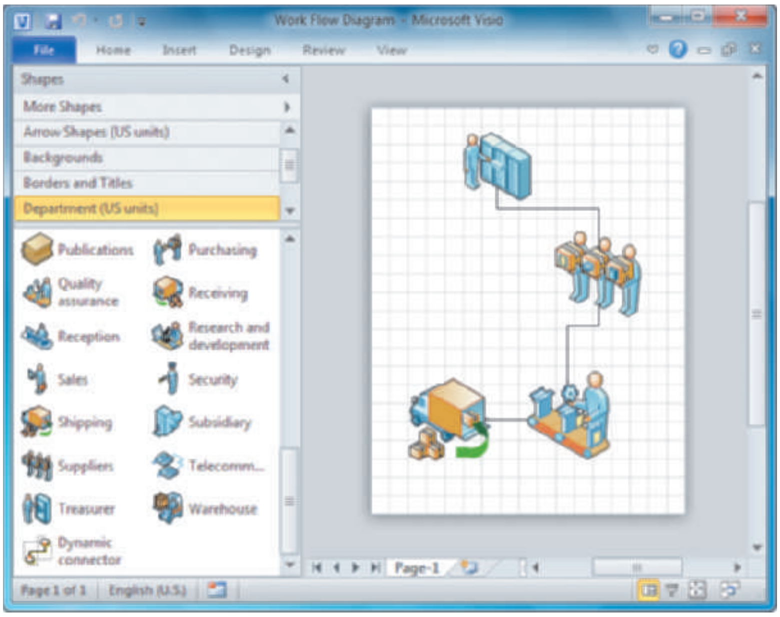


FIGURE 1-23 Microsoft Visio 2010 allows you to drag and drop various symbols and connect them to show a business process.

Figure 1-23 shows how you can drag and drop various symbols from the left pane into the drawing on the right, and connect them to show a business process.

You will learn about many types of models in this textbook, including data flow diagrams, object diagrams, and entity-relationship diagrams. Business process modeling is explained in more detail in Part B of the Systems Analyst's Toolkit.

Prototyping

Prototyping tests system concepts and provides an opportunity to examine input, output, and user interfaces before final decisions are made. A **prototype** is an early working version of an information system. Just as an aircraft manufacturer tests a new design in a wind tunnel, systems analysts construct and study information system prototypes. A prototype can serve as an initial model that is used as a benchmark to evaluate the finished system, or the prototype itself can develop into the final version of the system. Either way, prototyping speeds up the development process significantly.

A possible disadvantage of prototyping is that important decisions might be made too early, before business or IT issues are understood thoroughly. A prototype based on careful fact-finding and modeling techniques, however, can be an extremely valuable tool.

ON THE WEB

To learn more about CASE Tools, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the CASE Tools link.

Computer-Aided Systems Engineering (CASE) Tools

Computer-aided systems engineering (CASE), also called **computer-aided software engineering**, is a technique that uses powerful software, called **CASE tools**, to help systems analysts develop and maintain information systems. CASE tools provide an overall framework for systems development and support a wide variety of design methodologies, including structured analysis and object-oriented analysis.

Because CASE tools make it easier to build an information system, they boost IT productivity and improve the quality of the finished product. Part B of the Systems Analyst's Toolkit explains how analysts use CASE tools to create business profiles, build business models, and document complex processes. After developing a model, many CASE tools can generate program code, which speeds the implementation process. Figure 1-24 shows the Web site for Visible Systems Corporation, a leading vendor of CASE tools.

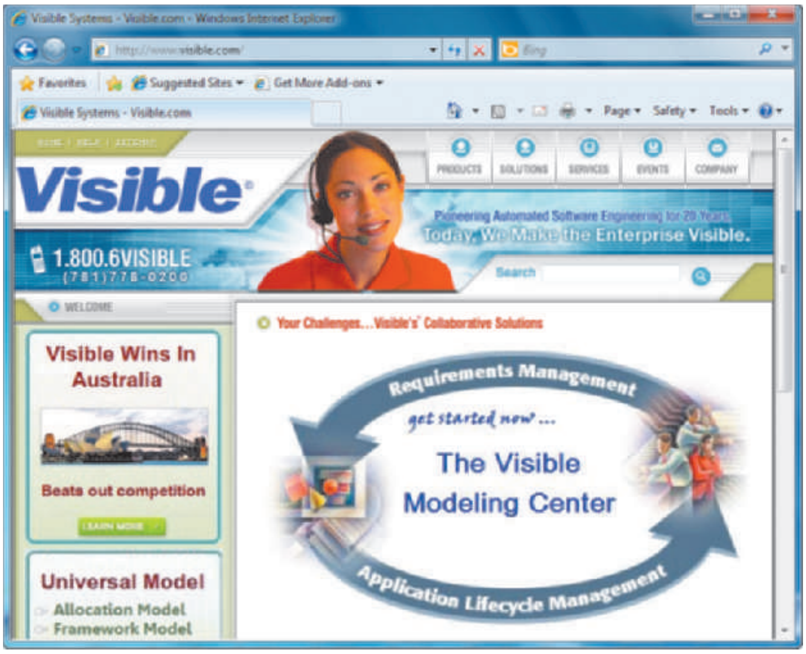


FIGURE 1-24 Visible Systems Corporation offers a wide array of software engineering tools, including Visible Analyst, a popular CASE tool.

Systems Development Methods

Many options exist for developing information systems, but the most popular alternatives are **structured analysis**, which is a traditional method that still is widely used, **object-oriented (O-O) analysis**, which is a more recent approach that many analysts prefer, and **agile methods**, also called **adaptive methods**, which include the latest trends in software development. Figure 1-25 provides an overview of the three methods, which are discussed in the following sections.

	STRUCTURED ANALYSIS	OBJECT-ORIENTED ANALYSIS	AGILE/ADAPTIVE METHODS
Description	Represents the system in terms of data and the processes that act upon that data. System development is organized into phases, with deliverables and milestones to measure progress. The SDLC waterfall model typically consists of five phases. Iteration is possible among the phases, as shown in Figure 1-27.	Views the system in terms of objects that combine data and processes. The objects represent actual people, things, transactions, and events, as shown in Figure 1-28. Compared to structured analysis, O-O phases tend to be more interactive. Can use the waterfall model or the model that stresses greater iteration, as shown in Figure 1-29.	Stresses intense team-based effort, as shown in Figures 1-30 and 1-31. Breaks development process down into cycles, or iterations that add functionality. Each iteration is designed, built, and tested in an ongoing process. Attempts to reduce major risks by incremental steps in short time intervals. Typically uses a spiral model, as shown in Figure 1-32.
Modeling tools	Data flow diagrams (DFDs) and process descriptions, which are described in Chapter 5. Also, business process modeling, which is explained in Part B of the Systems Analyst's Toolkit.	Various object-oriented diagrams depict system actors, methods, and messages, which are described in Chapter 6. Also, business process modeling, which is explained in Part B of the Systems Analyst's Toolkit.	Tools that enhance communication, such as collaborative software, brainstorming, and whiteboards. Business process modeling, which is explained in Part B of the Systems Analyst's Toolkit, works well with agile methods.
Pros	Traditional method, which has been very popular over time. Relies heavily on written documentation. Frequent phase iteration can provide flexibility comparable with other methods. Well-suited to project management tools and techniques.	Integrates easily with object-oriented programming languages. Code is modular and reusable, which can reduce cost and development time. Easy to maintain and expand as new objects can be cloned using inherited properties.	Very flexible and efficient in dealing with change. Stresses team interaction and reflects a set of community-based values. Frequent deliverables constantly validate the project and reduce risk.
Cons	Changes can be costly, especially in later phases. Requirements are defined early, and can change during development. Users might not be able to describe their needs until they can see examples of features and functions.	Somewhat newer method might be less familiar to development team members. Interaction of objects and classes can be complex in larger systems.	Team members need a high level of technical and communications skills. Lack of structure and documentation can introduce risk factors. Overall project might be subject to scope change as user requirements change.

FIGURE 1-25 Comparison of structured, object-oriented, and agile/adaptive development methods.

VIDEO LEARNING SESSIONS

To learn more about project management, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and navigate to the **Video Learning Sessions** for this book. These sessions can help you understand key concepts, practice your skills, and check your work.

VIDEO LEARNING SESSIONS

To learn more about DFD symbols and diagrams, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and navigate to the **Video Learning Sessions** for this book. These sessions can help you understand key concepts, practice your skills, and check your work.

Although most projects utilize one of these approaches, it is not unusual for system developers to mix and match methods to gain a better perspective. In addition to these three main development methods, some organizations choose to develop their own in-house approaches or use techniques offered by software suppliers, CASE tool vendors, or consultants. Many alternatives exist, and most IT experts agree that no one system development method is best in all cases. An approach that works well for one project might have major disadvantages or risks in another situation. The important thing is for a systems analyst to understand the various methods and the strengths and weaknesses of each approach.

Regardless of the development strategy, people, tasks, timetables, and costs must be managed effectively. Complex projects can involve dozens of people, hundreds of tasks, and many thousands of dollars. **Project management** is the process of planning, scheduling, monitoring, controlling, and reporting upon the development of an information system. Chapter 3 describes project management tools and techniques in detail.

Structured Analysis

Structured analysis is a traditional systems development technique that is time-tested and easy to understand. Structured analysis uses a series of phases, called the **systems development life cycle** (**SDLC**), to plan, analyze, design, implement, and support an information system. Although structured analysis evolved many years ago, it remains a popular systems development method. Structured analysis is based on an overall plan, similar to a blueprint for constructing a building, so it is called a **predictive** approach.

Structured analysis uses a set of process models to describe a system graphically. Because it focuses on processes that transform data into useful information, structured analysis is called a **process-centered** technique. In addition to modeling the processes, structured analysis also addresses data organization and structure, relational database design, and user interface issues.

A process model shows the data that flows in and out of system processes. Inside each process, input data is transformed by **business rules** that generate the output. Figure 1-26 shows a process model that was created with Visible Analyst, a popular software development tool. The model, which represents a school registration system, is a called a **data flow diagram** (**DFD**) because it uses various symbols and shapes to represent data flow, processing, and storage. You will learn more about DFDs in Chapter 5, and you can view a Video Learning Session that explains DFDs and how they are used as modeling tools.

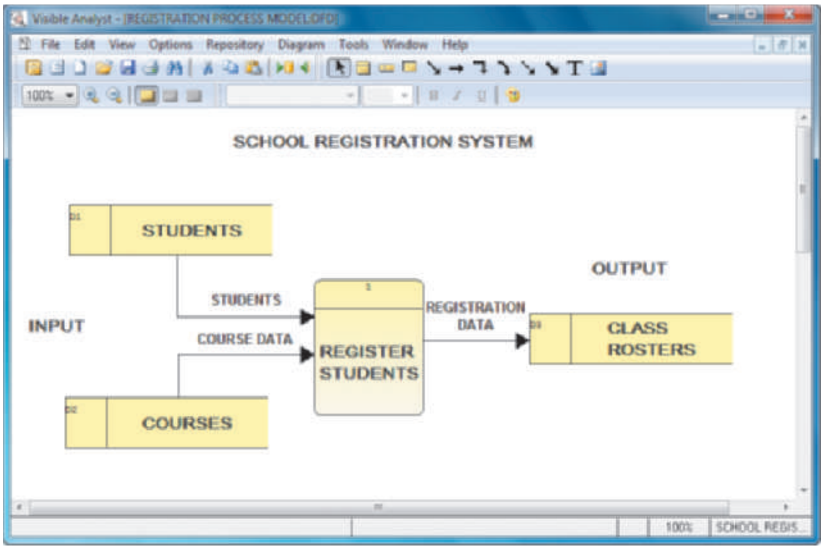


FIGURE 1-26 This Visible Analyst screen shows a process model for a school registration system. The REGISTER STUDENTS process accepts input data from two sources and transforms it into output data.

Structured analysis uses the SDLC to plan and manage the systems development process. The SDLC describes activities and functions that all systems developers perform, regardless of which approach they use. In the **waterfall model**, the result of each phase is called a **deliverable**, or **end product**, which flows into the next phase.

Some analysts see a disadvantage in the built-in structure of the SDLC, because the waterfall model does not emphasize interactivity among the phases. This criticism can be valid if the SDLC phases are followed too rigidly. However, adjacent phases usually interact, as shown by the dotted lines in Figure 1-27 and interaction among several phases is not uncommon. Other analysts regard the waterfall model as a two-way *water flow* model, with emphasis on iteration and user input. Used in this manner, the traditional model is not as different from agile methods as it might appear to be.

The SDLC model usually includes five steps, which are described in the following sections: systems planning, systems analysis, systems design, systems implementation, and systems support and security.

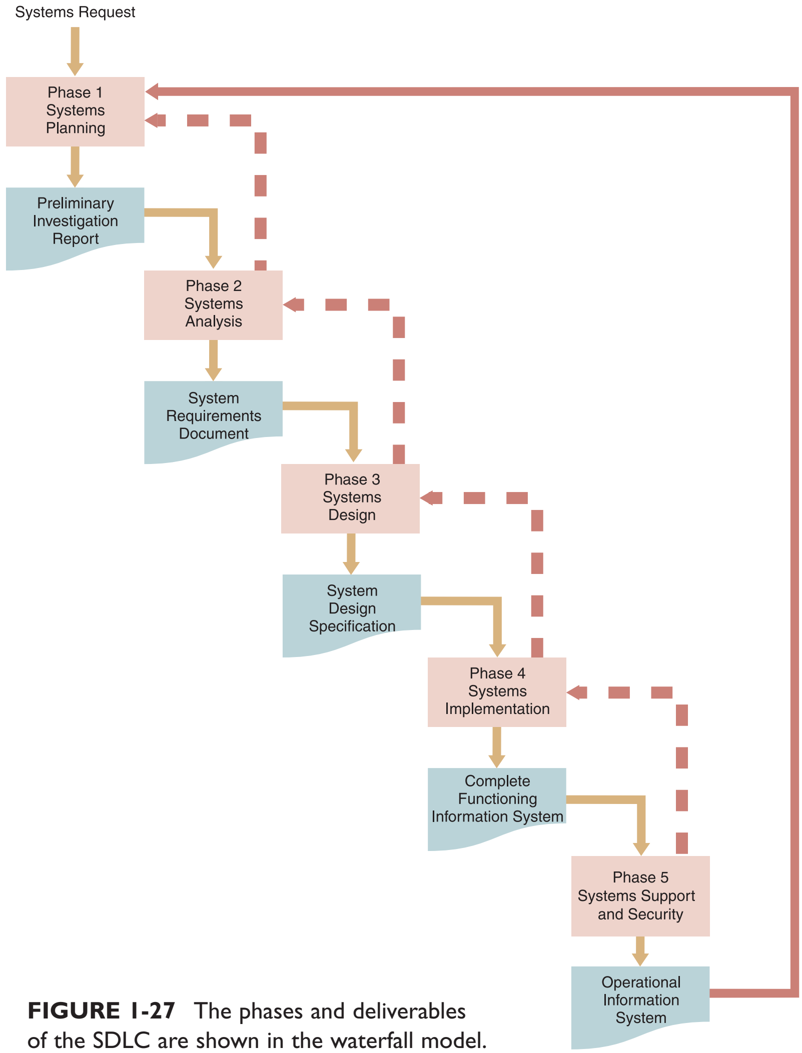


FIGURE 1-27 The phases and deliverables of the SDLC are shown in the waterfall model.

SYSTEMS PLANNING The **systems planning phase** usually begins with a formal request to the IT department, called a **systems request**, which describes problems or desired changes in an information system or a business process. In many companies, IT systems planning is an integral part of overall business planning. When managers and users develop their business plans, they usually include IT requirements that generate systems requests. A systems request can come from a top manager, a planning team, a department head, or the IT department itself. The request can be very significant or relatively minor. A major request might involve a new information system or the upgrading of an existing system. In contrast, a minor request might ask for a new feature or a change to the user interface.

The purpose of this phase is to perform a **preliminary investigation** to evaluate an IT-related business opportunity or problem. The preliminary investigation is a critical step because the outcome will affect the entire development process. A key part of the preliminary investigation is a **feasibility study** that reviews anticipated costs and benefits and recommends a course of action based on operational, technical, economic, and time factors.

Suppose you are a systems analyst and you receive a request for a system change or improvement. Your first step is to determine whether it makes sense to launch a preliminary investigation at all. Often you will need to learn more about business operations before you can reach a conclusion. After an investigation, you might find that the information system functions properly, but users need more training. In some situations, you might recommend a business process review, rather than an IT solution. In other cases, you might conclude that a full-scale systems review is necessary. If the development process continues, the next step is the systems analysis phase.

SYSTEMS ANALYSIS The purpose of the **systems analysis phase** is to build a logical model of the new system. The first step is **requirements modeling**, where you investigate business processes and document what the new system must do to satisfy users. Requirements modeling continues the investigation that began during the systems planning phase. To understand the system, you perform fact-finding using techniques such as interviews, surveys, document review, observation, and sampling. You use the fact-finding results to build business models, data and process models, and object models.

The deliverable for the systems analysis phase is the **system requirements document**. The system requirements document describes management and user requirements, costs and benefits, and outlines alternative development strategies.

SYSTEMS DESIGN The purpose of the **systems design phase** is to create a physical model that will satisfy all documented requirements for the system. At this stage, you design the user interface and identify necessary outputs, inputs, and processes. In addition, you design internal and external controls, including computer-based and manual features to guarantee that the system will be reliable, accurate, maintainable, and secure. During the systems design phase, you also determine the application architecture, which programmers will use to transform the logical design into program modules and code.

The deliverable for this phase is the **system design specification**, which is presented to management and users for review and approval. Management and user involvement is critical to avoid any misunderstanding about what the new system will do, how it will do it, and what it will cost.

SYSTEMS IMPLEMENTATION During the **systems implementation phase,** the new system is constructed. Whether the developers use structured analysis or O-O methods, the procedure is the same — programs are written, tested, and documented, and the system is installed. If the system was purchased as a package, systems analysts configure the software and perform any necessary modifications. The objective of the systems implementation phase is to deliver a completely functioning and documented information system. At the conclusion of this phase, the system is ready for use. Final preparations include converting data to the new system's files, training users, and performing the actual transition to the new system.

The systems implementation phase also includes an assessment, called a **systems evaluation**, to determine whether the system operates properly and if costs and benefits are within expectations.

SYSTEMS SUPPORT AND SECURITY During the **systems support and security phase**, the IT staff maintains, enhances, and protects the system. Maintenance changes correct errors and adapt to changes in the environment, such as new tax rates. Enhancements provide new features and benefits. The objective during this phase is to maximize return on the IT investment. Security controls safeguard the system from both external and internal threats. A well-designed system must be secure, reliable, maintainable, and scalable. A **scalable** design can expand to meet new business requirements and volumes. Information systems development is always a work in progress. Business processes change rapidly, and most information systems need to be updated significantly or replaced after several years of operation.

VIDEO LEARNING SESSIONS

To learn more about object modeling, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and navigate to the **Video Learning Sessions** for this book. These sessions can help you understand key concepts, practice your skills, and check your work.

Object-Oriented Analysis

Whereas structured analysis treats processes and data as separate components, object-oriented analysis combines data and the processes that act on the data into things called **objects**. Systems analysts use O-O to model real-world business processes and operations. The result is a set of software objects that represent actual people, things, transactions, and events. Using an O-O programming language, a programmer then writes the code that creates the objects.

An object is a member of a **class**, which is a collection of similar objects. Objects possess characteristics called **properties**, which the object inherits from its class or possesses on its own. As shown in Figure 1-28, the class called PERSON includes INSTRUCTOR and STUDENT. Because the PERSON class has a property called Address, a STUDENT inherits the Address property. A STUDENT also has a property called Major that is not shared by other members of the PERSON class.

In O-O design, built-in processes called **methods** can change an object's properties. For example, in a Web-based catalog store, an ORDER object might have a property called STATUS that changes when a CUSTOMER object clicks to place, confirm, or cancel the order.

One object can send information to another object by using a message. A **message** requests specific behavior or information from another object. For example, an ORDER object might send a message to a CUSTOMER object that requests a shipping address. When it receives the message, the CUSTOMER object supplies the information. The ORDER object has the capability to send the message, and the CUSTOMER object knows what actions to perform when it receives the message. O-O analysis uses object models to represent data and behavior, and to show how objects affect other objects. By describing the objects and methods needed to support a business operation, a system developer can design reusable components that speed up system implementation and reduce development cost.

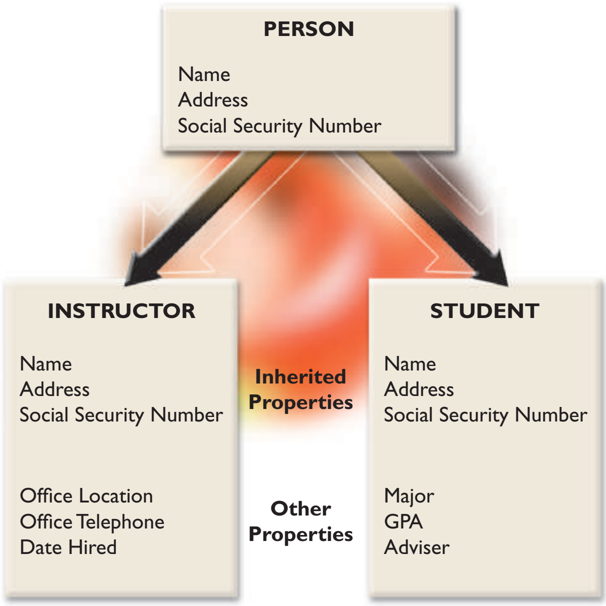


FIGURE 1-28 The PERSON class includes INSTRUCTOR and STUDENT objects, which have their own properties and inherited properties.

Object-oriented methods usually follow a series of analysis and design phases that are similar to the SDLC, although there is less agreement on the number of phases and their names. In an O-O model, the phases tend to be more interactive. Figure 1-29 shows a system development model where planning, analysis, and design tasks interact continuously to produce prototypes that can be tested and implemented. The result is an **interactive model** that can accurately depict real-world business processes.

O-O methodology is popular because it provides an easy transition to O-O programming languages such as Java, Smalltalk, C++, Python, and Perl. Chapter 6 covers O-O analysis and design, with a detailed description of O-O terms, concepts, tools, and techniques.

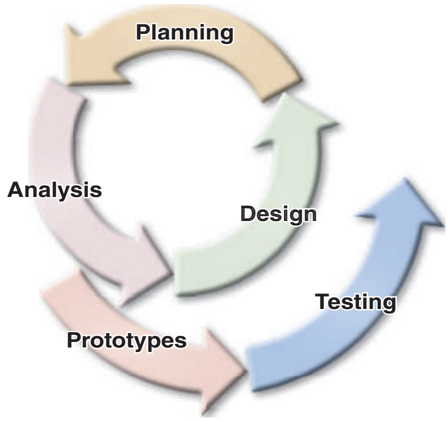


FIGURE 1-29 An interactive model often is used with O-O development methods. In this model, planning, analysis, and design tasks interact continuously.

Agile Methods

Development techniques change over time. For example, structured analysis is a traditional approach, and agile methods are the newest development. Structured analysis builds an overall plan for the information system, just as a contractor might use a blueprint for constructing a building. Agile methods, in contrast, attempt to develop a system incrementally, by building a series of prototypes and constantly adjusting them to user requirements. As the agile process continues, developers revise, extend, and merge earlier versions into the final product. An agile approach emphasizes continuous feedback, and each incremental step is affected by what was learned in the prior steps.

Although relatively new to software development, the notion of **iterative** development can be traced back to Japanese auto firms that were able to boost productivity by using a flexible manufacturing system, where team-based effort and short-term milestones helped keep quality up and costs down. Agile methods have attracted a wide following and an entire community of users, as shown in Figure 1-30 on the next page.

ON THE WEB

To learn more about agile systems development methods, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Agile Methods link.

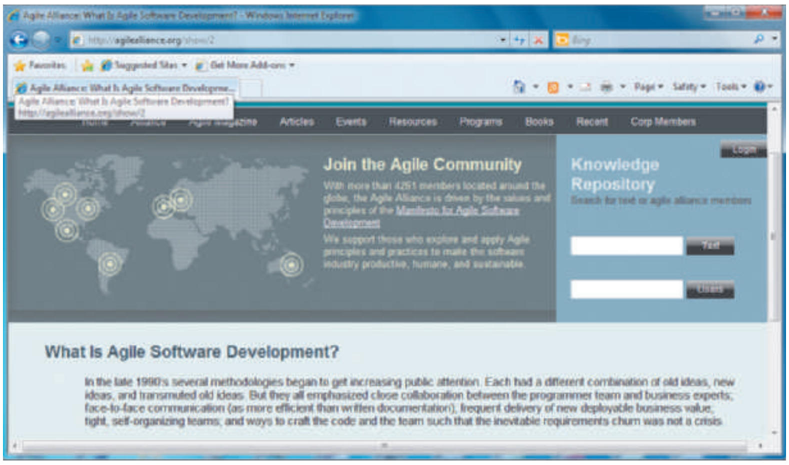


FIGURE 1-30 Agile methods have attracted a wide following and an entire community of users.

Agile methods typically use a **spiral model**, which represents a series of iterations, or revisions, based on user feedback. As the process continues, the final product gradually evolves. An agile approach requires intense interactivity between developers and individual users, and does not begin with an overall objective. Instead, the agile process determines the end result. Proponents of the spiral model believe that this approach reduces risks and speeds up software development.

Spiral models initially were suggested in the 1990s by Barry Boehm, a noted software engineering professor. He stated that each iteration, or phase, of the model must have a specific goal that is accepted, rejected, or changed by the user, or client. Thus, each iteration produces feedback and enhancements, which enable the team to reach the overall project goal. Typically, each iteration in a spiral model includes planning, risk analysis, engineering, and evaluation, as shown in the table in Figure 1-31. The repeated iterations produce a series of prototypes, which evolve into the finished system. Notice that these phases resemble SDLC tasks, which also can be iterative.

PHASE	TASKS
Planning	Define objectives, constraints, and deliverables
Risk analysis	Identify risks and develop acceptable resolutions
Engineering	Develop a prototype that includes all deliverables
Evaluation	Perform assessment and testing to develop objectives for next iteration

FIGURE 1-31 Typical phases and tasks in a spiral model.

Numerous other adaptive variations and related methods exist, and most IT developers expect this trend to continue in the future. Two examples are Scrum and Extreme Programming (XP), which are discussed in detail in Chapter 4, Requirements Modeling, and in Chapter 11, Application Development.

Although agile methods are becoming popular, analysts should recognize that these approaches have advantages and disadvantages. By their nature, agile methods can allow developers to be much more flexible and responsive, but can be riskier than more traditional methods. For example, without a detailed set of system requirements, certain features requested by some users might not be consistent with the company's larger game plan.

Other potential disadvantages of agile methods can include weak documentation, blurred lines of accountability, and too little emphasis on the larger business picture. Also, unless properly implemented, a long series of iterations might actually add to project cost and development time. The bottom line is that systems analysts should understand the pros and cons of any approach before selecting a development method for a specific project.

Other Development Methods

IT professionals know that the key to success is user input — before, during, and after a system is developed. Over time, many companies discovered that systems development teams composed of IT staff, users, and managers could complete their work more rapidly and produce better results. Two methodologies became popular: **joint application development** (JAD) and **rapid application development** (RAD).

Both JAD and RAD use teams composed of users, managers, and IT staff. The difference is that JAD focuses on team-based fact-finding, which is only one phase of the development process, whereas RAD is more like a compressed version of the entire process. JAD, RAD, and agile methods are described in more detail in Chapter 4.

In addition to the methods described in this chapter, you might encounter other systems development techniques. If a systems analyst wants additional choices, he or she can choose from an entire industry of IT software companies and consulting firms. For example, a popular approach offered by the Rational group at IBM is called the **Rational Unified Process** (**RUP®**). According to IBM, RUP® offers a flexible, iterative process for managing software development projects that can minimize risk, ensure predictable results, and deliver high-quality software on time.

Another option is what Microsoft calls **Microsoft Solutions Framework** (**MSF**), which documents the experience of its own software development teams. Although the Microsoft process differs from the SDLC phase-oriented approach, MSF developers perform the same kind of planning, ask the same kinds of fact-finding questions, deal with the same kinds of design and implementation issues, and resolve the same kinds of problems. Using this approach, MSF examines a broader business and organizational context that surrounds the development of an information system.

Companies often choose to follow their own methodology. Using CASE tools, an IT team can apply a variety of techniques rather than being bound to a single, rigid methodology. As shown in Part B of the Systems Analyst's Toolkit, many CASE tools offer a complete set of analysis and modeling tools that support various methods and strategies. Regardless of the development model, it will be necessary to manage people, tasks, timetables, and expenses by using various project management tools and techniques.

ON THE WEB

To learn more about Microsoft Solutions Framework, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Microsoft Solutions Framework link.

Systems Development Guidelines

The basic principles in Figure 1-32 apply to any IT project, large or small. Although you will develop your own methods and techniques, these guidelines will help you achieve success as a systems analyst.

FIVE BASIC SYSTEMS DEVELOPMENT GUIDELINES	
Develop a Plan	Prepare an overall project plan and stick to it. Complete the tasks in a logical sequence. Develop a clear set of ground rules and be sure that everyone on the team understands them clearly.
Involve Users and Listen Carefully to Them	Ensure that users are involved in the development process, especially when identifying and modeling system requirements. When you interact with users, listen closely to what they are saying. Chapter 4 describes fact-finding and how to get the most out of face-to-face communication.
Use Project Management Tools and Techniques	Try to keep the project on track and avoid surprises. Create a reasonable number of checkpoints — too many can be burdensome, but too few will not provide adequate control. In Chapter 3, you will learn how to use Microsoft Project to help you manage tasks, allocate resources, and monitor progress.
Develop Accurate Cost and Benefit Information	Managers need to know the cost of developing and operating a system, and the value of the benefits it will provide. You must provide accurate, realistic cost and benefit estimates, and update them as necessary.
Remain Flexible	Be flexible within the framework of your plan. Systems development is a dynamic process, and overlap often exists among tasks. The ability to react quickly is especially important when you are working on a system that must be developed rapidly.

FIGURE 1-32 These basic guidelines apply throughout the systems development process.

THE INFORMATION TECHNOLOGY DEPARTMENT

The IT department develops and maintains information systems. The structure of the IT department varies among companies, as does its name and placement within the organization. In a small firm, one person might handle all computer support activities and services, whereas a large corporation might require many people with specialized skills to provide information systems support. Figure 1-33 shows a typical IT organization in a company that has networked PCs, enterprise-wide databases, centralized processing, and Web-based operations.

The IT group provides **technical support**, which includes six main functions: application development, systems support and security, user support, database administration, network administration, and Web support. These functions overlap considerably and often have different names in different companies.

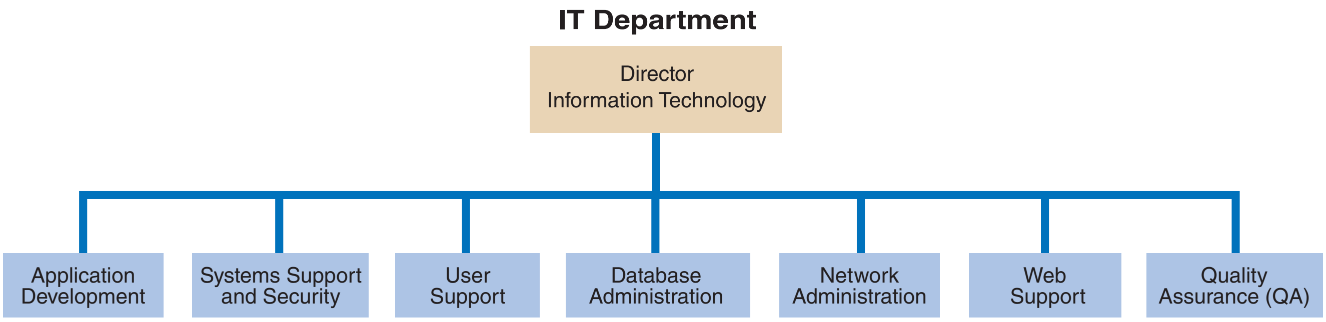


FIGURE 1-33 Depending on its size, an IT department might have separate organizational units for these functions, or they might be combined into a smaller number of teams.

Application Development

The IT **application development** group typically provides leadership and overall guidance, but the systems themselves are developed by teams consisting of users, managers, and IT staff members. A popular model for information systems development is a project-oriented team using RAD or JAD, with IT professionals providing overall coordination, guidance, and technical support.

CASE IN POINT 1.2: GLOBAL HOTELS AND MOMMA'S MOTELS

Suppose you work in the IT department of Global Hotels, a multinational hotel chain. Global Hotels runs several specialized business support systems, including a guest reservations system that was developed in-house to meet the requirements of a large company with worldwide operations. Guests can make one-stop online reservations by visiting Global's Web site, which has links to all major travel industry sites.

Global Hotels just acquired Momma's, a regional chain of 20 motels in western Canada. Momma's uses a vertical reservations package suitable for small- to medium-sized businesses, and a generic accounting and finance package. Should Momma's use Global Hotels' information systems or continue with its own? In your answer, consider issues such as business profiles, business processes, system interactivity, EDI, XML, e-commerce, and the characteristics of both information systems. What additional information would be helpful to you in making a recommendation?

Systems Support and Security

Systems support and security provides vital protection and maintenance services for system hardware and software, including enterprise computing systems, networks, transaction processing systems, and corporate IT infrastructure. The systems support and security group implements and monitors physical and electronic security hardware, software, and procedures. This group also installs and supports operating systems, telecommunications software, and centralized database management systems. In addition, systems support and security technicians provide technical assistance to other groups in the IT department. If a site has a large number of remote clients, the systems support group often includes a **deployment team** that installs and configures the workstations.

User Support

User support provides users with technical information, training, and productivity support. The user support function usually is called a **help desk** or **information center** (**IC**). A help desk's staff trains users and managers on application software such as e-mail, word processing spreadsheets, and graphics packages. User support specialists answer questions, troubleshoot problems, and serve as a clearinghouse for user problems and solutions.

Database Administration

Database administration involves data design, management, security, backup, and access. In small- and medium-sized companies, an IT support person performs those roles in addition to other duties. Regardless of company size, mission-critical database applications require continuous attention and technical support.

Network Administration

Business operations depend on networks that enable company-wide information systems. **Network administration** includes hardware and software maintenance, support, and security. In addition to controlling user access, network administrators install, configure, manage, monitor, and maintain network applications. Network administration is discussed in more detail in Chapter 10.

Web Support

Web support is a vital technical support function. Web support specialists design and construct Web pages, monitor traffic, manage hardware and software, and link Web-based applications to the company's information systems. Reliable, high-quality Web support is especially critical for companies engaged in e-commerce.

Quality Assurance (QA)

Many large IT departments also use a **quality assurance** (**QA**) team that reviews and tests all applications and systems changes to verify specifications and software quality standards. The QA team usually is a separate unit that reports directly to IT management.

CASE IN POINT 1.3: What Should Lisa Do?

Lisa Jameson has two job offers. One is from Pembroke Boats, a boat manufacturer that employs 200 people in a small Ohio town. Pembroke does not have an IT department and wants her to create one. The job position is called information coordinator, but she would be the only IT person.

The other offer, which pays about $7,500 more annually, is from Albemarle Express, a nationwide trucking firm located in Detroit. At Albemarle Express, Lisa would be a programmer-analyst, with the promise that if she does well in her position, she eventually will move into a systems analyst position and work on new systems development. Lisa has heard a rumor that another company might acquire Albemarle Express, but that rumor has occurred before and nothing has ever happened. What should Lisa do, and why?

The Systems Analyst

A systems analyst investigates, analyzes, designs, develops, installs, evaluates, and maintains a company's information systems. To perform those tasks, a systems analyst constantly interacts with users and managers within and outside the company. On large projects, the analyst works as a member of an IT department team; on smaller assignments, he or she might work alone.

Most companies assign systems analysts to the IT department, but analysts also can report to a specific user area such as marketing, sales, or accounting. As a member of a functional team, an analyst is better able to understand the needs of that group and how information systems support the department's mission. Smaller companies often use consultants to perform systems analysis work on an as-needed basis.

TOOLKIT TIME

The communications tools in Part A of the Systems Analyst's Toolkit can help you develop better reports and presentations. To learn more about these tools, turn to Part A of the four-part Toolkit that follows Chapter 12.

Responsibilities

The systems analyst's job overlaps business and technical issues. Analysts help translate business requirements into IT projects. When assigned to a systems development team, an analyst might help document business profiles, review business processes, select hardware and software packages, design information systems, train users, and plan e-commerce Web sites.

A systems analyst plans projects, develops schedules, and estimates costs. To keep managers and users informed, the analyst conducts meetings, delivers presentations, and writes memos, reports, and documentation. The Systems Analyst's Toolkit that follows Chapter 12 includes various tools to help you with each of those important skills.

Knowledge, Skills, and Education

A successful systems analyst needs technical knowledge, oral and written communication skills, an understanding of business operations, and critical thinking skills. Educational requirements vary widely depending on the company and the position. In a rapidly changing IT marketplace, a systems analyst must manage his or her own career and have a plan for professional development.

TECHNICAL KNOWLEDGE State-of-the-art knowledge is extremely important in a rapidly changing business and technical environment. The Internet offers numerous opportunities to update technical knowledge and skills. Many sites, such as the one shown in Figure 1-34 offer a convenient way for IT professionals to learn about technical developments, exchange experiences, and get answers to questions. Analysts also maintain their skills by attending training courses, both on-site and online. Networking with colleagues is another way to keep up with new developments, and membership in professional associations also is important.

COMMUNICATION SKILLS A systems analyst needs strong oral and written communication skills, and the ability to interact with people at all levels, from operational staff to senior executives. Often, the analyst must work with people outside the company, such as software and hardware vendors, customers, and government officials. Analysts often coordinate IT project teams, where they use communication skills to guide and motivate team members.

FIGURE 1-34 The TechRepublic Web site offers support for IT professionals. Features include newsletters, forums, product information, and a searchable knowledge base.

BUSINESS SKILLS A systems analyst works closely with managers, supervisors, and operational employees. To be effective, he or she must understand business operations and processes, communicate clearly, and translate business needs into requirements that can be understood by programmers and systems developers. A successful analyst is business-oriented, curious, comfortable with financial tools, and able to see the big picture. Chapter 2, Analyzing the Business Case, describes some basic concepts, including strategic planning, SWOT analysis, and feasibility tests. In addition, the Systems Analyst's Toolkit, which follows Chapter 12, explains communication and financial tools that can help analysts handle business-related tasks.

FIGURE 1-35 The Critical Thinking Community is a nonprofit organization that provides encouragement and resources for critical thinkers.

CRITICAL THINKING SKILLS Although no standard definition exists, most educators agree that **critical thinking skills** include the ability to compare, classify, evaluate, recognize patterns, analyze cause-and-effect, and apply logic. Critical thinkers often use a *what-if* approach, and they have the ability to evaluate their own thinking and reasoning.

Critical thinking skills are valuable in the IT industry, where employers seek job candidates who can demonstrate these skills and bring them to the workplace. Figure 1-35 shows the Foundation for Critical Thinking site, which offers many resources to support the critical thinking community.

EDUCATION Companies typically require systems analysts to have a college degree in information systems, computer science, or business, and some IT experience usually is required. For higher-level positions, many companies require an advanced degree. Sometimes, educational requirements can be waived if a candidate has significant experience, skills, or professional certifications. Part D of the Systems Analyst's Toolkit describes many valuable IT resources for personal and professional development.

Certification

Many hardware and software companies offer certification for IT professionals. **Certification** verifies that an individual demonstrated a certain level of knowledge and skill on a standardized test. Certification is an excellent way for IT professionals to learn new skills and gain recognition for their efforts. Although certification does not guarantee competence or ability, many companies regard certification as an important credential for hiring or promotion. You can learn more about certification in Chapter 12, Managing Systems Support and Security, and by visiting the Web sites of individual companies such as Microsoft, Cisco Systems, Sun Microsystems, and Novell.

In addition to traditional hardware and software certifications, some firms are exploring ways to assess critical thinking skills, as shown in Figure 1-3. These skills include perception, organization, analysis, problem-solving, and decision-making. Whether or not formal certification is involved, these skills are extremely valuable to IT professionals and the employers who hire them.

FIGURE 1-36 Employers want to hire people who think logically and effectively. Some firms are working on certification exams designed to measure these skills.

Career Opportunities

The demand for systems analysts is expected to remain strong. Companies will need systems analysts to apply new information technology, and the explosion in e-commerce will fuel IT job growth. The systems analyst position is a challenging and rewarding one that can lead to a top management position. With an understanding of technical and business issues, a systems analyst has an unlimited horizon. Many companies have presidents and senior managers who started in IT departments as systems analysts.

The responsibilities of a systems analyst at a small firm are different from those at a large corporation. Would you be better off at a small or large company? Where will you find the best opportunity for experience and professional growth? Each person looks for different rewards in a job. What will be important to you?

JOB TITLES First, do not rely on job titles alone. Some positions are called systems analysts, but involve only programming or technical support. In other cases, systems analyst responsibilities are found in positions titled computer specialist, programmer, programmer/analyst, systems designer, software engineer, and various others. Be sure the responsibilities of the job are stated clearly when you consider a position.

COMPANY ORGANIZATION Find out all you can about the company and where the IT department fits in the organization chart. Where are IT functions performed, and by whom? A firm might have a central IT group, but decentralize the systems development function. This situation sometimes occurs in large conglomerates, where the parent company consolidates information that actually is developed and managed at the subsidiary level. Where would you rather work?

COMPANY SIZE If you like more variety, a smaller firm might suit you best. If you want to specialize, however, then consider a larger company with state-of-the-art systems. Although you might have more responsibility in a smaller company, the promotional opportunities and financial rewards often are greater in larger companies. You also might want to consider working as an independent consultant, either on your own or with others. Many consulting firms have been successful in offering their services to smaller business enterprises that do not have the expertise to handle systems development on their own.

TOOLKIT TIME

The information technology resource tools in Part D of the Systems Analyst's Toolkit can help you obtain technical data, advance your career, and network with other IT professionals. To learn more about these tools, turn to Part D of the four-part Toolkit that follows Chapter 12.

CORPORATE CULTURE In addition to having goals, methods, and information systems requirements, every firm has an underlying **corporate culture**. A corporate culture is the set of beliefs, rules, traditions, values, and attitudes that define a company and influence its way of doing business. To be successful, a systems analyst must understand the corporate culture and how it affects the way information is managed. Companies sometimes include statements about corporate culture in their mission statements, which are explained in Chapter 2.

SALARY, LOCATION, AND FUTURE GROWTH Finally, consider salary, location, and the company's prospects for future growth and success. Think about your impressions of the company and the people you met during your interviews. Most important, review your short- and long-term goals very carefully before deciding which position is best for you.

CASE IN POINT 1.4: JUST-IN-TIME AIRFREIGHT, INC.

Suppose you are the IT director at Just-in-Time Airfreight, and you have received authorization to hire another systems analyst. This will be an entry-level position, and the person will assist senior systems analysts on various projects involving the reservations and the human resources systems. Using the information in this chapter, draft an ad that would appear in *The Wall Street Journal*, local newspapers, and online. You can get some ideas by visiting monster.com, or a similar site. In your ad, be sure to list desired skills, experience, and educational requirements.

A QUESTION OF ETHICS

You are enjoying your job as a summer intern in the IT department of a local company. At lunch yesterday, several people were discussing ethical issues. You learned that some of them belong to IT organizations that have ethical codes to guide members and set professional standards. For example, Ann, your supervisor, belongs to the Association for Computing Machinery (ACM), which has over 97,000 members and a Web site at **acm.org.** Ann said that the ACM code of ethics is important to her, and would definitely influence her views. On the other hand, Jack, a senior programmer, believes that his own personal standards would be sufficient to guide him if ethical questions were to arise.

Because you are excited about your career as an IT professional, you decide to examine the ACM code of ethics and make up your own mind. After you do so, would you tend to agree more with Ann or with Jack?

CHAPTER SUMMARY

In this chapter, you learned that information technology (IT) refers to the combination of hardware, software, and services that people use to manage, communicate, and share information. Technology is changing rapidly, and IT professionals must prepare for the future. IT supports business operations, improves productivity, and helps managers make decisions. Systems analysis and design is the process of developing information systems that transform data into useful information.

Traditionally, companies either developed in-house applications or purchased software packages from vendors. Today, the choice is much more complex, but it is always important for companies to plan the system carefully before considering implementation options.

The essential components of an information system are hardware, software, data, processes, and people. Hardware consists of everything in the physical layer of the information system. Software consists of system software, which manages the hardware components, and application software, which supports day-to-day business operations. Data is the raw material that an information system transforms into useful information. Processes describe the tasks and functions that users, managers, and IT staff members perform. People who interact with a system include users, from both within and outside the company.

A systems analyst starts with a business profile, which is an overview of company functions, and then he or she creates a series of business models that represent business processes, which describe specific transactions, events, tasks, and results. Analysts use business process modeling tools to document complex operations.

Most successful companies offer a mix of products, technical and financial services, consulting, and customer support. A rapidly growing business category is the Internet-dependent (dot-com) firm, which relies solely on Internet-based operations. E-commerce includes business-to-consumer (B2C) sales, and business-to-business (B2B) transactions that use Internet-based digital marketplaces or private electronic data interchange (EDI) systems.

Based on their functions and features, business information systems are identified as enterprise computing systems, transaction processing systems, business support systems, knowledge management systems, or user productivity systems. In most companies, significant overlap and integration exists among the various types of information systems.

A typical organization structure includes top managers, middle managers and knowledge workers, supervisors and team leaders, and operational employees. Top managers develop strategic plans, which define an overall mission and goals. Middle managers provide direction, resources, and feedback to supervisors and team leaders.

Knowledge workers include various professionals who function as support staff. Supervisors and team leaders oversee operational employees. Each organizational level has a different set of responsibilities and information needs.

Systems analysts use modeling, prototyping, and computer-aided systems engineering (CASE) tools. Modeling produces a graphical representation of a concept or process, whereas prototyping involves the creation of an early working model of the information or its components. A systems analyst uses CASE tools to perform various systems development tasks.

Three popular system development approaches are structured analysis, which is a traditional method that still is widely used, object-oriented analysis (O-O), which is a more recent approach that many analysts prefer, and agile methods, also called adaptive methods, which include the latest trends in software development.

Structured analysis uses a series of phases, called the systems development life cycle (SDLC) that usually is shown as a waterfall model. Structured analysis uses an overall plan, similar to a blueprint for constructing a building, so it is called a predictive approach. This method uses a set of process models to describe a system graphically, and also addresses data organization and structure, relational database design, and user interface issues.

Object-oriented analysis combines data and the processes that act on the data into things called objects that represent people, things, transactions, and events. Objects have characteristics called properties, built-in processes called methods, and can send information to other objects by using messages. Using an O-O programming language, a programmer then writes the code that creates the objects. Object-oriented methods usually follow a series of analysis and design phases similar to the SDLC, but the phases are more interactive.

Agile methods are the newest development approach, and attempt to develop a system incrementally by building a series of prototypes and constantly adjusting them to user requirements. Agile methods typically use a spiral model, which represents a series of iterations, or revisions, based on user feedback. The repeated iterations produce a series of prototypes, which evolve into the finished system.

Regardless of the development strategy, people, tasks, timetables, and costs must be managed effectively using project management tools and techniques, which are described in detail in Chapter 3.

Some firms choose to develop their own in-house methods or adopt techniques offered by software suppliers, CASE tool vendors, or consultants. Companies also use team-based strategies called joint application development (JAD) and rapid application development (RAD). JAD focuses on team-based fact-finding, whereas RAD is more like a compressed version of the entire process. JAD and RAD are described in more detail in Chapter 4.

The IT department develops, maintains, and operates a company's information systems. IT staff members provide technical support, including application development, systems support, user support, database administration, network administration, and Web support. These functions overlap considerably and often have different names in different companies.

In addition to technical knowledge, a systems analyst must understand the business, think critically, and communicate effectively. Valuable credentials such as certifications are available to systems analysts. A systems analyst's responsibilities depend on a company's organization, size, and culture. Systems analysts need to consider salary, location, and future growth potential when making a career decision.

Key Terms and Phrases

adaptive methods *21*
agile methods *21*
application development *28*
application software *8*
B2B (business-to-business) *13*
B2C (business-to-consumer) *13*
brick-and-mortar *12*
business model *19*
business process *10*
business process model (BPM) *10*
business process modeling *10*
business process modeling notation (BPMN) *10*
business process reengineering (BPR) *18*
business profile *10*
business rules *22*
business support systems *16*
CASE tools *20*
certification *32*
class *24*
computer-aided software engineering (CASE) *20*
computer-aided systems engineering (CASE) *20*
corporate culture *33*
critical thinking skills *31*
data *7*
data flow diagram (DFD) *22*
data model *19*
database administration *29*
deliverable *22*
deployment team *29*
dot-com (.com) *12*
e-commerce (electronic commerce) *13*
electronic data interchange (EDI) *14*
empowerment *19*
end product *22*
end users *10*
enterprise applications *8*
enterprise computing *15*
enterprise resource planning (ERP) *15*
expert systems *16*
extensible markup language (XML) *14*
feasibility study *23*
fuzzy logic *17*
groupware *17*
hardware *8*
help desk *29*
horizontal system *8*
I-commerce (Internet commerce) *13*
inference rules *17*
information *7*
information center (IC) *29*
information system *7*
information technology (IT) *4*
in-house applications *7*
interactive model *25*
Internet-dependent *12*
iterative *25*
joint application development (JAD) *26*
knowledge base *17*
knowledge management systems *16*
knowledge workers *19*
legacy systems *9*
management information systems (MIS) *16*
message *25*
methods *25*
Microsoft Solutions Framework (MSF) *27*
mission-critical system *7*
modeling *19*
Moore's Law *8*
network administration *29*
network model *19*
object-oriented (O-O) analysis *21*
object model *19*
objects *24*
predictive *22*
preliminary investigation *23*
process model *19*
process-centered *22*
processes *9*
product-oriented *11*
project management *22*
properties *24*
prototype *20*
quality assurance (QA) *29*
radio frequency identification (RFID) *16*
rapid application development (RAD) *26*
Rational Unified Process (RUP®) *27*
requirements model *19*
requirements modeling *23*
scalable *24*
service-oriented *11*
software *8*
software packages *7*
spiral model *26*
stakeholders *10*
strategic plans *18*
structured analysis *21*
supplier relationship management (SRM) *14*
supply chain management (SCM) *14*
system *7*
system design specification *24*
system requirements document *24*
system software *8*
systems analysis and design *7*
systems analysis phase *23*
systems analysts *7*
systems design phase *24*
systems development life cycle (SDLC) *22*
systems evaluation *24*
systems implementation phase *24*
systems planning phase *23*
systems request *23*
systems support and security *29*
systems support and security phase *24*
technical support *28*
transaction processing (TP) systems *15*
user productivity systems *17*
user support *29*
users *10*
vertical system *8*
waterfall model *22*
Web support *29*
what-if *16*

Learn It Online

Instructions: To complete the Learn It Online exercises, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to the resources for this chapter, and click the link for the exercise you want to complete.

1 Chapter Reinforcement

TF, MC, and SA

Click the Chapter Reinforcement link. Print the quiz by clicking Print on the File menu for each page. Answer each question.

2 Flash Cards

Click the Flash Cards link and read the instructions. Type 20 (or a number specified by your instructor) in the Number of playing cards text box, type your name in the Enter your Name text box, and then click the Flip Card button. When the flash card is displayed, read the question and then click the ANSWER box arrow to select an answer. Flip through the Flash Cards. If your score is 15 (75%) correct or greater, click Print on the File menu to print your results. If your score is less than 15 (75%) correct, then redo this exercise by clicking the Replay button.

3 Practice Test

Click the Practice Test link. Answer each question, enter your first and last name at the bottom of the page, and then click the Grade Test button. When the graded practice test is displayed on your screen, click Print on the File menu to print a hard copy. Continue to take practice tests until you score 80% or better.

4 Who Wants To Be a Computer Genius?

Click the Computer Genius link. Read the instructions, enter your first and last name at the bottom of the page, and then click the Play button. When your score is displayed, click the PRINT RESULTS link to print a hard copy.

5 Wheel of Terms

Click the Wheel of Terms link. Read the instructions, and then enter your first and last name and your school name. Click the PLAY button. When your score is displayed on the screen, right-click the score and then click Print on the shortcut menu to print a hard copy.

6 Crossword Puzzle Challenge

Click the Crossword Puzzle Challenge link. Read the instructions, and then enter your first and last name. Click the SUBMIT button. Work the crossword puzzle. When you are finished, click the Submit button. When the crossword puzzle is redisplayed, click the Print Puzzle button to print a hard copy.

SCR Associates Case Simulation **Session 1: Introduction**

Overview

The SCR Associates case study is a Web-based simulation that allows you to practice your skills in a real-world environment. The case study transports you to SCR's intranet, where you complete 12 work sessions, each aligning with a chapter. As you work on the case, you will receive e-mail and voice mail messages, obtain information from SCR's online libraries, and perform various tasks.

Background

SCR Associates is an IT consulting firm that offers solutions and training. SCR has decided to expand by opening a high-tech training center. The company needs to develop an information system to support operations at the new training center. The new system will be called TIMS (Training Information Management System).

As a newly hired systems analyst, you report to Jesse Baker, systems group manager. She will expect you to apply the knowledge and skills you learn in each chapter. Your work should be accurate, thorough, and have a professional appearance.

What's in the textbook?

- A preview of each work session, to set the stage for your tasks.
- A list of the session tasks for easy reference.

What's on the Web?

- A realistic simulation that includes the SCR Web site and a link to the company intranet. The Web site contains public information about the company, including its organization, mission, values, goals, and services.
- An intranet portal that asks you to enter your name and a password.
- Links for e-mail and voice mail that explain your tasks and add realism.
- A task list, with four specific tasks per session.
- Background information you will need, including a data library, a forms library, and resources list.

How do I use the case?

- Review the background material.
- Read the preview for this session and study the Task List. You might need to review the chapter material in order to perform the tasks.
- Visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to the **SCR Case Simulation**, and locate the intranet link.
- Enter your name and the password **sad9e.** An opening screen will display the work sessions.
- Select a work session.
- Check your e-mail and voice mail messages carefully, and then begin working on the task list.

SCR Associates Case Simulation Session 1: Introduction (Continued)

Preview: Session 1

This is your second day on the job as a systems analyst at SCR Associates. You spent most of yesterday filling out personnel forms and learning your way around the office. This morning, you sit at your desk and examine SCR's Internet site. You explore the entire site, which reflects SCR's history, purpose, and values. You especially are impressed with the emphasis SCR puts on its relationships with clients. When you finish examining the site, you are more convinced than ever that SCR will be a great career opportunity. You are excited about your new job, and eager to get started.

Task List

1. *Investigate SCR's Internet site and learn about the company's history, purpose, and values. Send Jesse a brief memo with suggestions to expand or improve these sections.*
2. *On the SCR intranet, visit the data, forms, and resource libraries and review a sample of the information in each library.*
3. *Using the SCR functions and organization listed in the data library, create an organization chart using Microsoft Word, Visio, or a drawing program.*
4. *Jesse says that SCR has plenty of competition in the IT consulting field. Get on the Internet and find three other IT consulting firms. She wants a brief description of each firm and the services it offers.*

FIGURE 1-37 Task list: Session 1.

Chapter Exercises

Review Questions

1. What is information technology, and why is it important to a business?
2. Define business profiles, processes, and modeling.
3. Identify the main components of an information system, and describe the system's stakeholders.
4. Explain the difference between vertical and horizontal systems packages.
5. How do companies use EDI? What are some advantages of using XML?
6. Describe five types of information systems, and give an example of each.
7. Describe four organizational levels of a typical business and their information requirements.
8. Describe the phases of the systems development life cycle, and compare the SDLC waterfall model with the spiral model.
9. Explain the use of models, prototypes, and CASE tools in the systems development process. Also explain the pros and cons of agile development methods.
10. What is object-oriented analysis, and how does it differ from structured analysis?

Discussion Topics

1. Some experts believe that the growth in e-commerce will cause states and local governments to lose a significant amount of sales tax revenue, unless Internet transactions are subject to sales tax. Do you agree? Why or why not?
2. Present an argument for and against the following proposition: Because IT managers must understand all phases of the business, a company should fill top management vacancies by promoting IT managers.
3. The head of the IT group in a company often is called the chief information officer (CIO) or chief technology officer (CTO). Should the CIO or CTO report to the company president, to the finance department, where many of the information systems are used, or to someone or somewhere else? Why would it matter?
4. Computers perform many jobs that previously were performed by people. Will computer-based transactions and expanded e-commerce eventually replace person-to-person contact? From a customer's point of view, is this better? Why or why not?

Projects

1. Contact at least three people at your school or a nearby company who use information systems. List the systems, the position titles of the users, and the business functions that the systems support.
2. Research newspaper, business magazine articles, or the Web to find computer companies whose stock is traded publicly. Choose a company and pretend to buy $1,000 of its stock. What is the current price per share? Why did you choose that company? Report each week to your class on how your stock is doing.
3. Do a search on the Web to learn more about agile system development approaches and spiral models. Prepare a summary of the results and a list of the sites you visited.
4. Is it really possible to measure thinking skills? Before you decide, visit the Critical Thinking Community site shown in Figure 1-35 on page 31. Prepare a brief memo with your conclusion and reasons.

Apply Your Knowledge

The Apply Your Knowledge section contains four mini-cases. Each case describes a situation, explains your role in the case, and asks you to respond to questions. You can answer the questions by applying knowledge you learned in the chapter.

1 Low-Voltage Components

Situation:

You are the IT manager at Low-Voltage Components, a medium-sized firm that makes specialized circuit boards. Low-Voltage's largest customer, TX Industries, recently installed a computerized purchasing system. If Low-Voltage connects to the TX system, TX will be able to submit purchase orders electronically. Although Low-Voltage has a computerized accounting system, that system is not capable of handling EDI.

1. Should Low-Voltage develop a system to connect with TX Industries' purchasing system? Why or why not?
2. What terms or concepts describe the proposed computer-to-computer relationship between Low-Voltage and TX Industries?
3. Is Low-Voltage's proposed new system a transaction processing system? Why or why not?
4. Before Low-Voltage makes a final decision, should the company consider an ERP system? Why or why not?

2 Systems Analyst Salaries

Situation:

As part of your job search, you decide to find out more about salaries and qualifications for systems analysts in the area where you would like to work. To increase your knowledge, search the Internet to perform the following research:

1. Find information about a career as a systems analyst.
2. Using the Internet, determine whether the Federal Bureau of Labor Statistics lists salary information for systems analysts. If so, summarize the information you find.
3. Find at least two online ads for systems analysts and list the employers, the qualifications, and the salaries, if mentioned.
4. Find at least one ad for an IT position that specifically mentions e-commerce.

3 MultiTech Interview

Situation:

You have an interview for an IT position with MultiTech, a large telecommunications company, and you want to learn more about the firm and its organizational structure. To prepare for the interview, you decide to review your knowledge about corporations, including the following questions:

1. What are the four organizational levels in a typical company?
2. How can you classify companies based on their mix of products and services?
3. What is empowerment?
4. What types of information systems might a large company use?

4 Rainbow's End Interview

Situation:

Your MultiTech interview seemed to go well, but you did not get the job. During the meeting, the interviewer mentioned that MultiTech uses structured analysis and relies heavily on modeling, prototyping, and CASE tools. Thinking back, you realize that you did not fully understand those terms. As you prepare for an interview with Rainbow's End, a large retail chain, you decide to review some IT terms and concepts. You want to be ready for the following questions:

1. What are the main differences between structured, O-O, and agile development methods?
2. What is a CASE tool and what does it do?
3. What is business process modeling and how is it done?
4. What is prototyping and why is it important?

Case Studies

Case studies allow you to practice specific skills learned in the chapter. Each chapter contains several case studies that continue throughout the textbook, and a chapter capstone case.

New Century Health Clinic

New Century Health Clinic offers preventive medicine and traditional medical care. In your role as an IT consultant, you will help New Century develop a new information system.

Background

Five years ago, cardiologists Timothy Jones and Dolores Garcia decided to combine their individual practices in Brea, California, to form New Century Health Clinic. They wanted to concentrate on preventive medicine by helping patients maintain health and fitness and by providing traditional medical care. Dr. Jones recently asked you to work with him as an IT consultant. He wants you to help New Century develop an information system that will support the clinic's operations and future growth. At your initial meeting, he provided you with some background information and asked for your suggestions about how to begin.

At your desk, you begin to review New Century's situation. The clinic is located near a new shopping mall in a busy section of the city. New Century's staff includes four doctors, three registered nurses, four physical therapists, and six office staff workers. The clinic currently has a patient base of 3,500 patients from 275 different employers, many of which provide insurance coverage for employee wellness and health maintenance. Currently, New Century accepts 34 different insurance policies.

Anita Davenport, who has been with New Century since its inception, is the office manager. She supervises the staff, including Fred Brown, Susan Gifford, Tom Capaletti, Lisa Sung, and Carla Herrera. Fred Brown handles office payroll, tax reporting, and profit distribution among the associates. Susan Gifford is responsible for the maintenance of patient records. Tom Capaletti handles most of the paperwork concerning insurance reporting and accounting. Lisa Sung has the primary responsibility for the appointment book, and her duties include making reminder calls to patients and preparing daily appointment lists. Carla Herrera is concerned primarily with ordering and organizing office and clinic supplies.

Each of the six office staff people has one or more primary responsibilities; however, all members of the staff help out whenever necessary with patient records, insurance processing, and appointment processing. In addition to their regular responsibilities, all six office workers are involved in the preparation of patient statements at the end of each month.

Using this information, you begin to prepare for your next meeting with Dr. Jones.

Assignments

1. Create an organization chart of the office staff using Microsoft Word or a similar program, or you can draw it by hand. In Word 2010 and Word 2007, click the Insert tab on the Ribbon, then Smart Art, then Hierarchy.
2. Identify at least three business processes that New Century performs, and explain who is responsible for the specific tasks.
3. Explain how New Century might use a transaction processing system, a business support system, and a user productivity system. For each type of system, provide a specific example, and explain how the system would benefit the clinic.
4. During the systems development process, should New Century consider any of the following: B2B, vertical and horizontal system packages, or Internet-based solutions? Explain your answers.

PERSONAL TRAINER, INC.

Personal Trainer, Inc. owns and operates fitness centers in a dozen midwestern cities. The centers have done well, and the company is planning an international expansion by opening a new "supercenter" in the Toronto area.

Background

Cassia Umi, president, heads Personal Trainer's management team. Three managers report to her at the firm's Chicago headquarters: Janet McDonald, manager, finance; Tai Tranh, manager, sales and marketing; and Reed Cotter, manager, operations. The managers who run the 12 existing centers all report to Reed.

Cassia wants the new supercenter to emphasize a wide variety of personal services and special programs for members. If the supercenter approach is successful, it will become the model for Personal Trainer's future growth. Cassia personally selected Gray Lewis, a manager with three years of fitness center experience, to run the new facility.

The new supercenter will feature a large exercise area with state-of-the-art equipment, a swimming pool, a sporting goods shop, a health food store, and a snack bar. In addition, the center will offer child care with special programs for various ages, a teen center, and a computer café. Cassia also wants members to have online access to customized training programs and progress reports.

Personal Trainer currently uses BumbleBee, a popular accounting package, to manage its receivables, payables, and general ledger. Membership lists and word processing are handled with Microsoft Office products.

Cassia believes the new supercenter will require additional data management capability, and she decided to hire Patterson and Wilder, an IT consulting firm, to help Personal Trainer develop an information system for the new operation. The firm assigned Susan Park, an experienced consultant, to work with the Personal Trainer team.

Susan's first task was to learn more about business operations at the new center, so she requested a meeting with Gray. After some small talk, the discussion went like this:

Susan: Tell me about your plans for the new operation. I'm especially interested in what kind of information management you'll need.

Gray: *Cassia thinks that we'll need more information support because of the size and complexity of the new operation. To tell the truth, I'm not so sure. We've had no problem with BumbleBee at the other centers, and I don't really want to reinvent the wheel.*

Susan: Maybe we should start by looking at the similarities — and the differences — between the new center and the existing ones.

Gray: *Okay, let's do that. First of all, we offer the same basic services everywhere. That includes the exercise equipment, a pool, and, in most centers, a snack bar. Some centers also sell sporting goods, and one offers child care — but not child-fitness programs. It is true that we've never put all this together under one roof. And, I admit, we've never offered online access. To be honest, I'm not absolutely sure what Cassia has in mind when she talks about 24/7 Web-based access. One more feature — we plan to set up two levels of membership — let's call them silver and gold for now. Silver members can use all the basic services, but will pay additional fees for some special programs, such as child fitness. Gold members will have unlimited use of all services.*

Susan: So, with all this going on, wouldn't an overall system make your job easier?

Gray: *Yes, but I don't know where to start.*

(continued)

Susan: Gray, that's why I'm here. I'll work with you and the rest of the team to come up with a solution that supports your business.

Gray: *Sounds good to me. When can we start?*

Susan: Let's get together first thing tomorrow. Bring along an organization chart and think about how you plan to run the new facility. We'll try to build a model of the new operation so we can identify the business functions. When we know what the functions are, we can figure out what kind of information is needed or generated by each function. That will be our starting point.

Assignments

1. Develop a business profile for Personal Trainer, based on the facts provided. List at least three of Personal Trainer's business processes.
2. Create an organization chart for Personal Trainer using Microsoft Word or a similar program, or you can draw it by hand. In Word 2010 and Word 2007, click the Insert tab on the Ribbon, then Smart Art, then Hierarchy.
3. Review the conversation between Susan and Gray. In your opinion, is Gray totally supportive of the new system? Why or why not? Do you agree with the way that Susan responds to Gray's comments? Why or why not?
4. Should Personal Trainer consider any of the following systems: enterprise computing, transaction processing, business support, knowledge management, or user productivity? Why or why not? What opportunities might Personal Trainer have for Web-based B2C transactions in the future? What about B2B?

Original Kayak Adventures

Original Kayak Adventures (OKA) offers guided eco-tours and kayak rentals along the Hudson River.

Background

John and Edie Caputo, who are avid kayakers and amateur naturalists, founded OKA two years ago. The Caputos spent many weekends and vacations exploring the Hudson's numerous creeks and tributaries. John was a sales representative and Edie worked for a Web design firm. Two years ago, John's division was purchased by a rival company, which announced plans to move operations to another state. Rather than relocate, the Caputos decided to launch OKA. They reasoned that Edie could leave her job and work as a freelance Web designer, which would provide some income while John tried to build OKA into a profitable business. John and Edie are convinced that the ecotourism market will expand greatly, and they look forward to sharing their experience and knowledge with others who enjoy nature and kayaking.

Original Kayak Adventures advertises in regional magazines and maintains a Web site, which Edie designed. Customers say that the site is attractive and informative, but the Caputos are not sure of its effectiveness in attracting new business. At this time, no other kayak rental firms operate within 20 miles of OKA's location.

So far, the Caputos' plan is working out well. OKA rents space at a nearby marina, where Edie runs the office and operates her Web design business. She also handles rentals when John is giving lessons or busy with a tour group. On summer weekends and holidays, Janet Jacobs, a local college student, handles telephone inquiries and reservations.

OKA's inventory includes 16 rental kayaks of various types, eight car-top carriers, and a large assortment of accessories and safety equipment. Based on customer requests, Edie is considering adding a selection of books and videos about kayaking and ecotourism.

OKA has three main business segments: rentals, instruction, and guided tours. Most customers make advance reservations for scheduled tours and instruction sessions, but sometimes space is available for last-minute customers. Rentals are split evenly between reservations and walk-in customers.

Reservations are entered in a loose-leaf binder, with separate tabs for each business activity. Edie also created a Microsoft Access database to record reservations. When she has time, she enters the reservation date, the reservation details and kayak type, and the customer information into a table, which is sorted by reservation date. Each day, she prints a reservation list. For quick reference, Edie also displays kayak availability on a wall-mounted board with color-coded magnets that show the available or reserved status of each rental kayak. In addition to the database, Edie uses an inexpensive accounting package to keep OKA's books.

Although the OKA database handles the basic information, the Caputos have noticed some drawbacks. For example, reservations for guided tours or instruction sessions sometimes conflict with John's or Edie's availability. The Caputos also would like to get more information about rental patterns, customer profiles, advertising effectiveness, and future business opportunities. John and Edie have talked about updating the system, but they have been too busy to do so.

Assignments

1. Develop a business profile for Original Kayak Adventures. The profile should include information about OKA's business activities, organization, resources, customers, and potential opportunity to engage in e-commerce.
2. List OKA's main functions and business processes. Draw a model of an OKA business process, including possible events, processes, and results.
3. What types of information systems does OKA use? Do these systems support its current and future business objectives? Why or why not?
4. From an object-oriented viewpoint, OKA treats reservations as a class. Based on the background information provided, what are some properties of reservation objects?

CHAPTER CAPSTONE CASE: SoftWear, Limited

SoftWear, Limited (SWL) is a continuing case study that illustrates the knowledge and skills described in each chapter. In this case study, the student acts as a member of the SWL systems development team and performs various tasks.

Background

SoftWear, Limited, manufactures and sells casual and recreational clothing for men and women. SWL was formed about 10 years ago when a national firm sold the division during a corporate downsizing. A group of managers obtained financing and became owners of the company. With clever marketing, competitive pricing, and efficient production, SWL has grown to more than 450 employees, including the corporate headquarters and manufacturing plants. Last year, SWL had sales of $700 million.

The company employs 90 people at its Raleigh, North Carolina, headquarters, including officers, managers, and support staff. Another 30 salaried and 340 hourly people are employed at production facilities in Haskell, California, and Florence, Texas. The company also is considering new factories in Canada and Australia.

SWL maintains a Web site with information about the company and its products. SWL's Web site features text, graphics, and audio and allows customers to send e-mail, order products from the SoftWear catalog, and request special promotional items, including beach umbrellas, hats, and T-shirts customized with the purchaser's logo. SWL also is studying other ways to use the Internet to boost product sales and expand its marketing efforts, including a special European promotion designed to increase awareness of SWL's Web site.

Organization

SWL's headquarters includes the executive, operations, marketing, finance, and human resources departments. Figure 1-38 shows the organization chart of the management positions within SWL. Notice that the director of information technology, Ann Hon, reports to Michael Jeremy, vice president of finance. The director of the payroll department, Amy Calico, also reports to Mr. Jeremy.

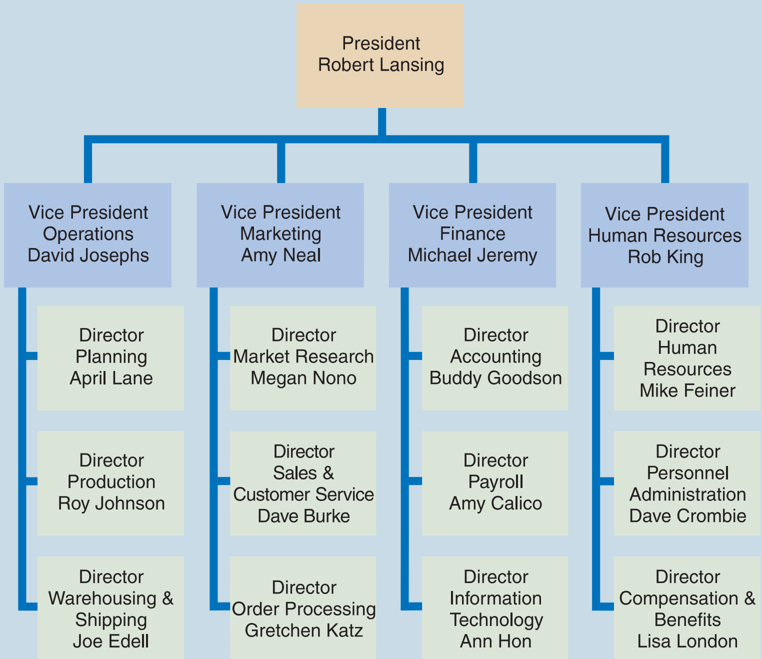


FIGURE 1-38 Organization chart of SoftWear, Limited.

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

The IT department includes Ann Hon, the director; Jane Rossman, the systems support manager; Zachary Ridgefield, the user support manager; and Ella Trung, the Web support manager. Figure 1-39 shows the organization of the IT department. At SWL, the systems support group also handles new systems development, network administration, and database administration.

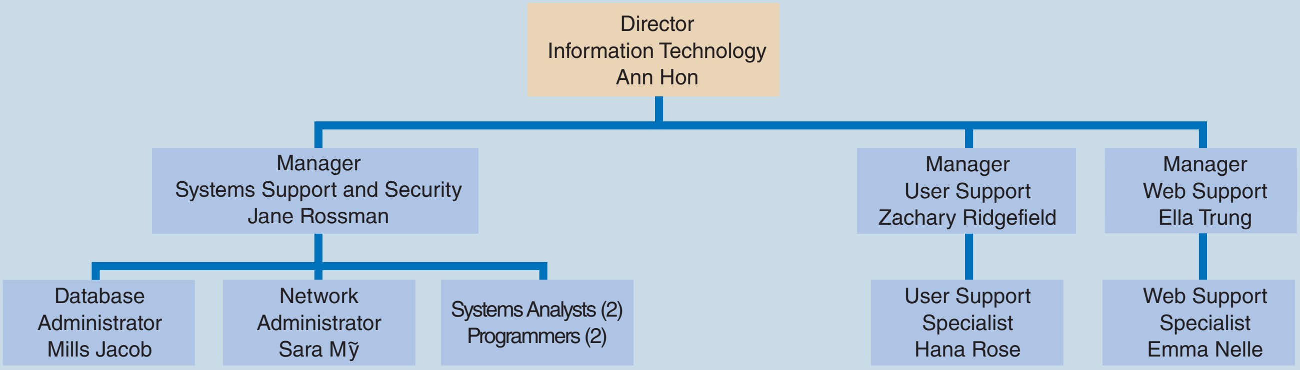


FIGURE 1-39 Organization chart of the IT department of SoftWear, Limited.

Systems analysts and programmers report to Jane Rossman, systems support manager. Systems analysts primarily analyze and design information systems. Programmers primarily develop, test, and implement code necessary for systems development, enhancements, and maintenance. In addition to the current staff, SWL is planning to hire a programmer-analyst who will divide his or her time between systems analysis and programming duties.

The technical support staff members are responsible for the system software on all SWL computers. They also provide technical advice and guidance to the other groups within the IT department.

The operations staff is responsible for centralized IT functions, including SWL's mainframe computer, and provides network and database administration.

Current Systems

SWL uses a manufacturing and inventory control system at its factories, but the system does not exchange data with SWL's suppliers at this time. The company's sales processing system handles online and catalog transactions, and produces sales reports. The marketing staff, however, wants even more information about sales trends and marketing analysis data. A company intranet connects employees at all locations, and provides e-mail, shared calendars, and a document library. Most administrative employees have workstations with Microsoft Office applications, but SWL has not provided company-wide training or help desk support.

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

SWL Team Tasks

1. Write an employment advertisement for a new systems analyst position at SWL. Perform Internet research to locate examples of advertisements for systems analysts, and consider SWL's business profile when you write the advertisement.
2. Should SWL consider any of the following systems: ERP, business support, or knowledge management? Why or why not?
3. What opportunities might SWL have for Web-based B2B transactions in the future?
4. Should SWL consider ways to increase a sense of empowerment among its employees? Why or why not? Could user productivity software play a role in that effort? How?

Manage the SWL Project

You have been asked to manage SWL's new information system project. One of your most important activities will be to identify project tasks and determine when they will be performed. Before you begin, you should review the SWL case in this chapter. Then list and analyze the tasks, as follows:

LIST THE TASKS Start by listing and numbering at least 10 tasks that the SWL team needs to perform to fulfill the objectives of this chapter. Your list can include SWL Team Tasks and any other tasks that are described in this chapter. For example, Task 3 might be to Draw an SWL organization chart, and Task 6 might be to Identify the various levels of SWL management.

ANALYZE THE TASKS Now study the tasks to determine the order in which they should be performed. First identify all concurrent tasks, which are not dependent on other tasks. In the example shown in Figure 1-40, Tasks 1, 2, 3, 4, and 5 are concurrent tasks, and could begin at the same time if resources were available.

Other tasks are called dependent tasks, because they cannot be performed until one or more earlier tasks have been completed. For each dependent task, you must identify specific tasks that need to be completed before this task can begin. For example, you would want an organization chart to help you identify the management levels, so Task 6 cannot begin until Task 3 is completed, as Figure 1-40 shows.

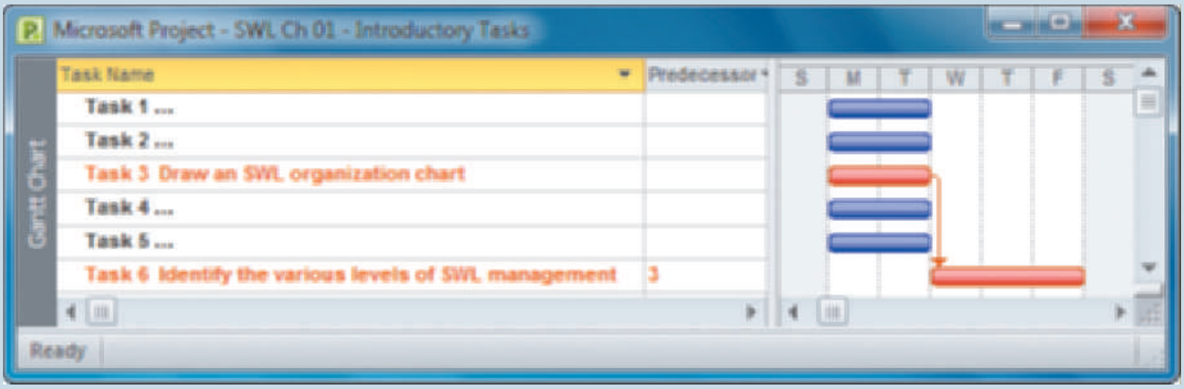


FIGURE 1-40 Tasks 1, 2, 3, 4, and 5 are concurrent tasks that could be performed at the same time. Task 6 is a dependent task that cannot be performed until Task 3 has been completed.

Chapter 3 describes project management tools, techniques, and software. To learn more, you can use the Features section on your Student Study Tool CD-ROM, or visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and locate the project management resources library for this book. On the Web, Microsoft offers demo versions, training, and tips for using Project 2010. You also can visit the OpenWorkbench.org site to learn more about this free, open-source software.

Ready for a Challenge?

In additional to technical skills, IT professionals need critical thinking skills such as perception, organization, analysis, problem-solving, and decision-making. The Ready for a Challenge feature can help you learn, practice, and apply critical thinking skills that you can take to the workplace.

You are a summer intern in the IT department at Game Technology. You report to the IT director, Mike McGee. Mike assigned you to work with two systems analysts: Lauren Jacksina and Cathy Ross. Lauren and Cathy both report to Felesia Stukes, manager — IT development, who reports to Mike. Joe Turner, manager — IT operations also reports to Mike. Dawn Rountree, database administrator, and Greg Wade, network administrator, report to Joe.

As an intern, you are expected to keep a journal to record your day-to-day experiences and things you learn. This week, you need to list some characteristics of various systems development methods used by the IT team. Your journal should include the following topics:

- Which development method relies heavily on written documentation?
- Which development method depicts system actors using diagrams?
- Which development method uses a spiral model?

Practice Tasks

A. Draw an organization chart showing all IT department positions, You can use Microsoft Word or a similar program, or you can draw it by hand. In Word 2010 and Word 2007, click the Insert tab on the Ribbon, then click Smart Art, then Hierarchy.

B. Write a journal entry that answers the questions about development methods.

After you complete the Practice Tasks, to check your work and view sample answers, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to the resources for this chapter, and locate Ready for a Challenge?.

The Challenge

Three weeks ago, Mike McGee left to join a larger company, and management decided to reorganize the IT department. Felesia Stukes was promoted to IT director, and to save money, her old job was eliminated. You, Lauren, Cathy, and Joe will report to her. The rest of the team is unchanged, except that a new programmer, Annie Edenton, has been hired, reporting to Dawn Rountree. Dawn's title has been changed to data design specialist.

To update your journal, you need to add three more pieces of information:

- Which development method uses a five-phase model?
- Which development method stresses intense team-based efforts?
- Which development method uses a waterfall model?

Challenge Tasks

A. Draw a new organization chart showing the changes, with full names and titles.

B. Write a journal entry that answers the new questions about development methods.

This page intentionally left blank

CHAPTER 2 Analyzing the Business Case

Chapter 2 explains how to analyze a business case. This chapter also explains why it is important to understand business operations and requirements, how IT projects support a company's overall strategic plan, how systems projects get started, and how systems analysts conduct a preliminary investigation and feasibility study.

INTRODUCTION

OBJECTIVES

When you finish this chapter, you will be able to:

- Explain the concept of a business case and how a business case affects an IT project
- Describe the strategic planning process and why it is important to the IT team
- Conduct a SWOT analysis and describe the four factors involved
- Explain the purpose of a mission statement
- Explain how the SDLC serves as a framework for systems development
- List reasons for systems projects and factors that affect such projects
- Describe systems requests and the role of the systems review committee
- Define operational, technical, economic, and schedule feasibility
- Describe the steps and the end product of a preliminary investigation

During the systems planning phase, the IT team reviews a proposal to determine if it presents a strong business case. The term **business case** refers to the reasons, or justification, for a proposal. A strong business case suggests that the company should pursue the alternative, above other options, because it would be in the firm's best interest to do so. To analyze the business case for a specific proposal, the analyst must consider the company's overall mission, objectives, and IT needs.

This chapter begins with a discussion of strategic planning, because the IT team must understand, support, and help plan long-term strategic goals. Along with financial, marketing, and human resources, companies need information technology to achieve growth and success.

Systems development typically starts with a systems request, followed by a preliminary investigation, which includes a feasibility study. You will learn how systems requests originate, how they are evaluated, and how to conduct a preliminary investigation. You also will learn about fact-finding techniques that begin at this point and carry over into later development phases. Finally, you will examine the report to management, which concludes the systems planning phase.

Chapter Introduction Case: Mountain View College Bookstore

Background: Wendy Lee, manager of college services at Mountain View College, wants a new information system that will improve efficiency and service at the three college bookstores.

In this part of the case, Tina Allen (systems analyst) and David Conroe (student intern) are talking about justification for the new system and the project's feasibility.

Participants: Tina and David
Location: Mountain View College cafeteria, Tuesday afternoon, September 6, 2011.
Project status: Tina has received a systems request from Wendy Lee for a new bookstore information system.
Discussion topics: Analysis of business justification and project feasibility

Tina: Hi, David. Are you ready to get started?

David: *Sure. What's our next step?*

Tina: Well, when we analyze a specific systems request, we need to see how the proposal fits into the overall picture at the college. In other words, we have to analyze the business case for the request.

David: *What's a business case?*

Tina: A business case is the justification for a project. A strong business case means that a proposal will add substantial value to the organization and support our strategic plan.

David: *What's a strategic plan?*

Tina: A strategic plan is like a road map for the future. Without a long-range plan, it's hard to know if you're heading in the right direction. Our plan starts with a mission statement, which reflects our purpose, our vision, and our values.

David: *I see what you mean. I read the mission statement this morning. It says that we will strive to be an efficient, customer-friendly bookstore that uses a mix of interpersonal skills and technology to serve our students and support the overall objectives of the college. That says a lot in just one sentence.*

Tina: It sure does. Now, let's get to the specifics. I just received a systems request from the college business manager. She wants us to develop a new information system for the bookstore.

David: *Do we have a green light to get started?*

Tina: Yes and no. Mountain View College doesn't have a formal procedure for evaluating IT requests, and we don't have a systems review committee. Maybe that's something we should consider for the future. Meanwhile, we need to conduct a preliminary investigation to see whether this request is feasible.

David: *What do you mean by "feasible"?*

Tina: To see if a systems request is feasible, we have to look at four separate yardsticks: operational feasibility, technical feasibility, economic feasibility, and schedule feasibility. If the request passes all the tests, we continue working on the system. If not, we stop.

David: *How will we know if the request passes the tests?*

Tina: That's our next step. Here's a task list to get us started:

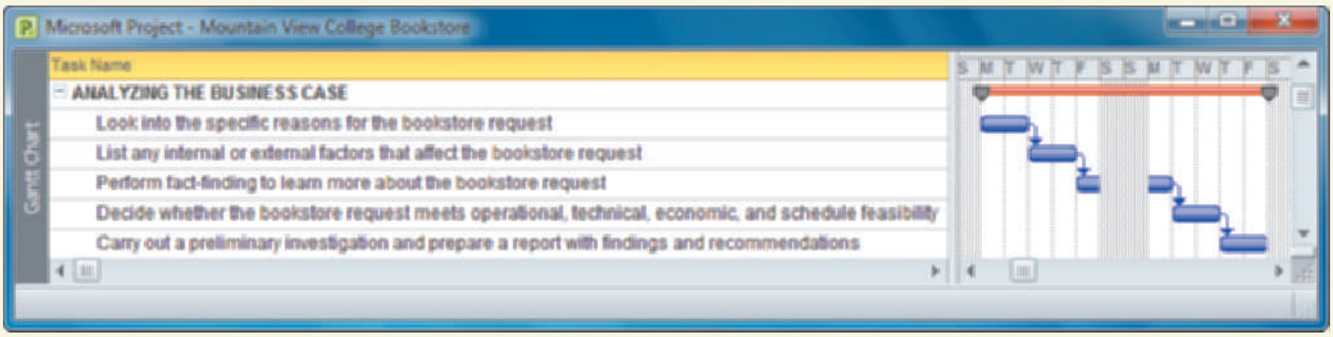


FIGURE 2-1 Typical business case analysis task list.

Strategic Planning — A Framework for IT Systems Development

ON THE WEB

To learn more about strategic planning, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Strategic Planning link.

Companies develop and maintain IT systems to support their current and future business operations. Some IT needs are immediate, such as fixing a logic problem in a payroll system. Other needs might be on the horizon, such as planning IT support for a new factory, a future merger, or a corporate restructuring. In most companies, the IT team reviews each IT-related proposal, project, and systems request to determine if it presents a strong business case, or justification.

Most successful IT managers engage in long-range planning, even as they handle day-to-day maintenance and support. To carry out this task effectively, they must understand and participate in the firm's strategic planning process. **Strategic planning** is the process of identifying long-term organizational goals, strategies, and resources. Strategic planning looks beyond day-to-day activities and focuses on a horizon that is 3, 5, 10, or more years in the future.

Strategic Planning Overview

Why does a systems analyst need to know about strategic planning? The answer might be found in an old story about two stonecutters who were hard at work when a passerby asked them what they were doing. "I am cutting stones," said the first worker. The second worker replied, "I am building a cathedral." So it is with information technology: One analyst might say, "I am using a CASE tool," whereas another might say, "I am helping the company succeed in a major new business venture." Systems analysts should focus on the larger, strategic role of IT as they carry out their day-to-day responsibilities.

Strategic planning starts with a management review called a **SWOT analysis**. The letters stand for strengths, weaknesses, opportunities, and threats. A SWOT analysis usually starts with a broad overview. The first step is for top management to respond to questions like these:

- What are our strengths, and how can we use them to achieve our business goals?
- What are our weaknesses, and how can we reduce or eliminate them?
- What are our opportunities, and how do we plan to take advantage of them?
- What are our threats, and how can we assess, manage, and respond to the possible risks?

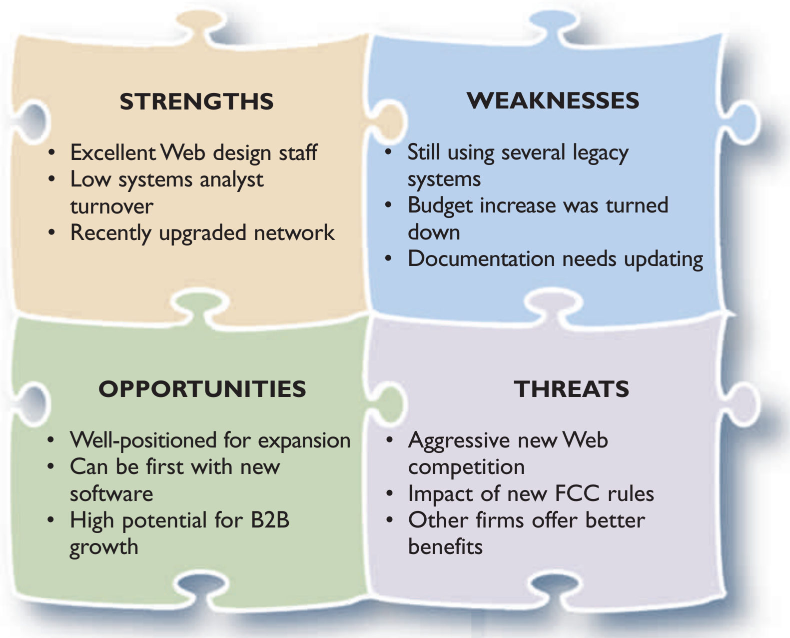


FIGURE 2-2 A SWOT analysis might produce results similar to those shown here.

A SWOT analysis is a solid foundation for the strategic planning process, because it examines a firm's technical, human, and financial resources. In Figure 2-2, the bulleted lists show samples of typical strengths, weaknesses, opportunities, and threats.

As the SWOT process continues, management reviews specific resources and business operations. For example, suppose that during a SWOT analysis, a firm studies an important patent that the company owns. The patent review might generate input like the examples shown in Figure 2-3.

SWOT ANALYSIS

Strength	Our patent covers valuable technology that we can use in many popular products.
Weakness	Our patent, like every patent, has a limited life, and when it expires, the technology will no longer be protected.
Opportunity	We might be able to use the patented technology in more products, license it to other firms, or expand our research to develop new patents based on the original technology.
Threat	There is a chance that a competitor will develop technology that is similar to ours, but does not technically infringe upon our patent.

FIGURE 2-3 Sample SWOT analysis of a specific asset, such as a patent.

There is no standard approach to strategic planning. Some managers believe that a firm's **mission statement** should contain an inspirational message to its stakeholders. Others feel that unless a firm starts with a realistic SWOT assessment, it might develop a mission statement that is unachievable. The majority of companies view the strategic planning process as a dynamic interaction, similar to the diagram in Figure 2-4, where the company's mission statement reflects a long-term horizon, but sets forth goals that are achievable and consistent with real-world conditions. Figure 2-5 on the next page shows how purpose, vision, and values are revealed in one company's Web site.

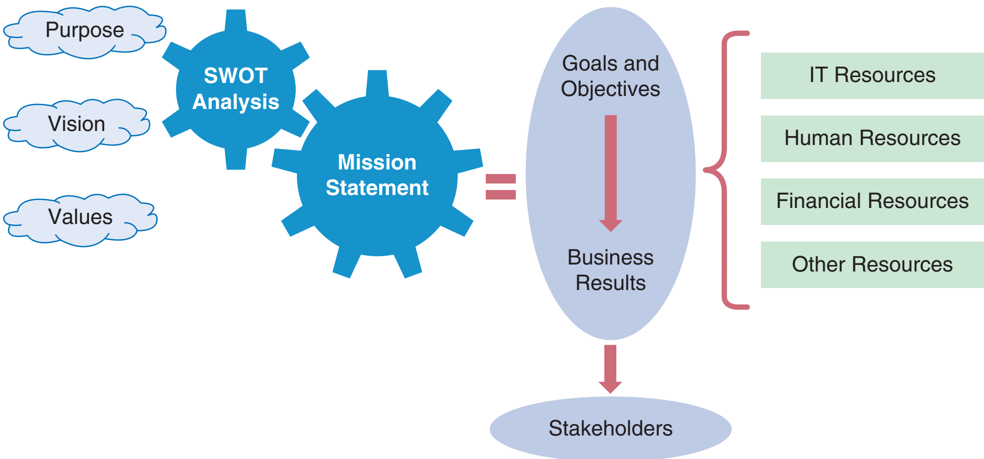


FIGURE 2-4 Strategic planning is a dynamic process that identifies specific goals and objectives that support the company's mission.

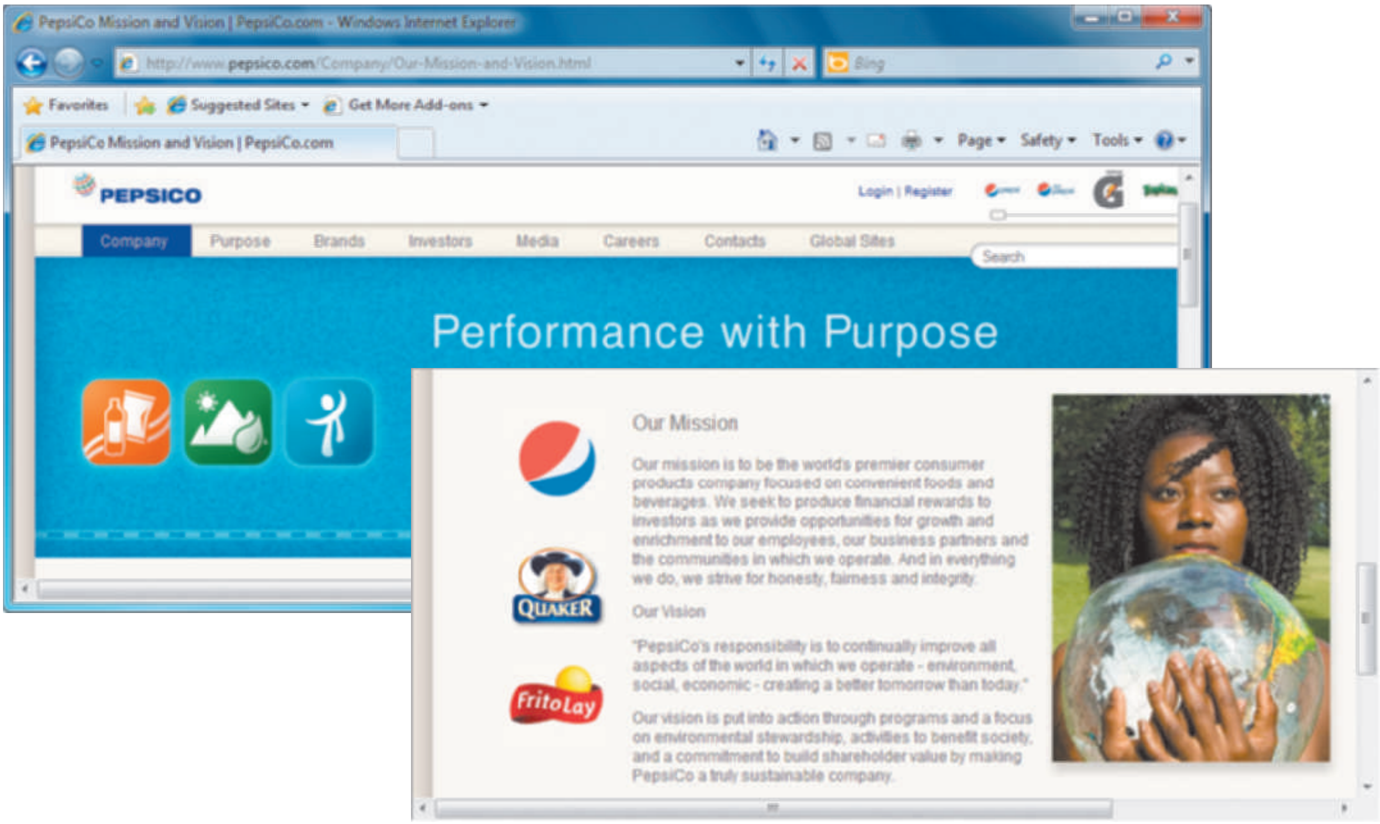


FIGURE 2-5 PepsiCo's Web site presents a combination of values, mission, and vision.

ON THE WEB

To learn more about mission statements, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Mission Statements link.

A mission statement is just the starting point. Next, the company identifies a set of goals that will accomplish the mission. For example, the company might establish one-year, three-year, and five-year goals for expanding market share. To achieve those goals, the company develops a list of shorter-term objectives. For example, if a goal is to increase Web-based orders by 30% next year, a company might set quarterly objectives with monthly milestones. Objectives also might include tactical plans, such as creating a new Web site and training a special customer support group to answer e-mail inquiries. Finally, the objectives translate into day-to-day business operations, supported by IT and other corporate resources. The outcome is a set of business results that affect company stakeholders.

CASE IN POINT 2.1: Lo Carb Meals

Lo Carb is a successful new company that has published several cookbooks, and marketed its own line of low-carbohydrate meals. Joe Turner, Lo Carb's president, has asked your opinion. He wants to know whether a mission statement really is necessary. After you review the chapter material, write a brief memo with your views. Be sure to include good (and not-so-good) examples of actual mission statements that you find on the Web.

A CASE Tool Example

You are a systems analyst working for Sally, the IT manager for a large hotel chain. Sally is working with top management to develop a strategic plan, and she asked you to assist her. The plan will guide future company goals and objectives, including IT projects.

Sally has experience with the Visible Analyst CASE tool, but she has never used it for strategic planning, so she asked you to do some research. First, you navigate to the strategic planning section, where you can enter planning statements such as assumptions, goals, objectives, critical success factors, and others. Planning statements also can document strengths, weaknesses, opportunities, and threats, as shown in Figure 2-6. After you visit the Help section to learn more about the strategic planning features, you feel confident that you can work effectively with this powerful tool.

When you present your results to Sally, she seems pleased. Because the term is new to you, you ask her what critical success factors are, and she replies that **critical success factors** are vital objectives that must be achieved for the company to fulfill its mission.

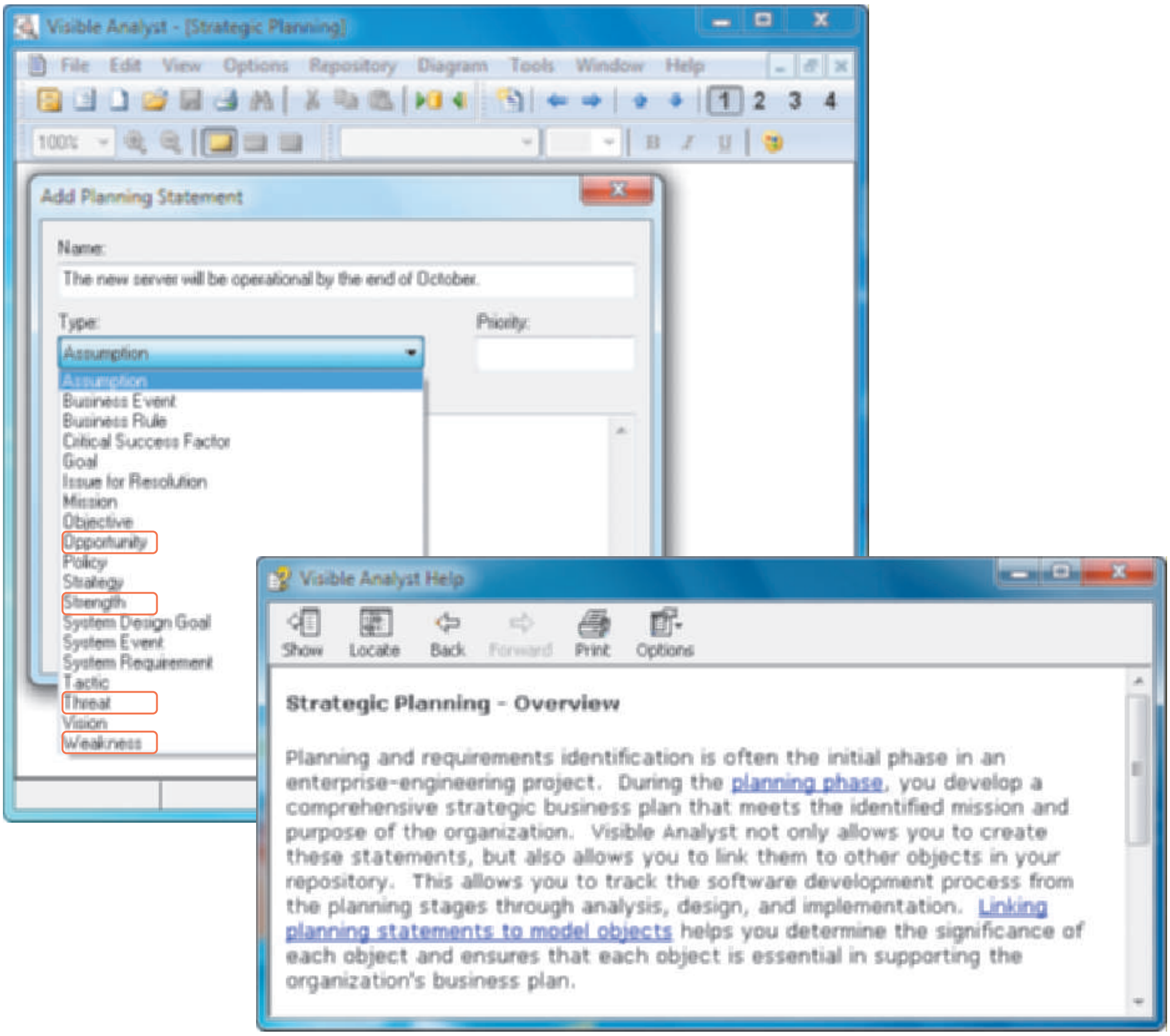


FIGURE 2-6 The Visible Analyst CASE tool supports strategic planning and allows a user to enter many kinds of information. Notice the four SWOT categories in the list.

The Role of the IT Department in Project Evaluation

Management leadership and information technology are linked closely, and remarkable changes have occurred in both areas. Ten years ago, a typical IT department handled all aspects of systems development and consulted users only when, and if, the department wanted user input. Today, systems development is much more team-oriented. New approaches to systems development, such as joint application development (JAD) and rapid application development (RAD), typically involve groups of users, managers, and IT staff working together right from the start.

Although team-oriented development is the norm, some companies see the role of the IT department as a gatekeeper, responsible for screening and evaluating systems requests. Should the IT department perform the initial evaluation, or should a cross-functional team do it? The answer depends on the company's size and organization, and whether IT is tightly integrated into business operations. In smaller companies or firms where only one person has IT skills, that person acts as a coordinator and consults closely with users and managers to evaluate systems requests. Larger firms are more likely to use an evaluation team or systems review committee.

The Future

If you could look into the future, here is what you might see: new industries, products, and services emerging from amazing advances in information technology, customers who expect world-class IT support, a surge in Internet-based commerce, and a global business environment that is dynamic and incredibly challenging. To some firms, these changes will be threatening; other companies will see opportunities and take advantage of them by creating and following a strategic plan.

CASE IN POINT 2.2: ATTAWAY AIRLINES, PART ONE

You are the IT director at Attaway Airlines, a small regional air carrier. You chair the company's systems review committee, and you currently are dealing with strong disagreements about two key projects. Dan Esposito, the marketing manager, says it is vital to have a new computerized reservation system that can provide better customer service and reduce operational costs. Molly Kinnon, vice president of finance, is equally adamant that a new accounting system is needed immediately, because it will be very expensive to adjust the current system to new federal reporting requirements. Molly outranks Dan, and she is your boss. The next meeting, which promises to be a real showdown, is set for 9:00 a.m. tomorrow. How will you prepare for the meeting? What questions and issues should be discussed?

WHAT IS A BUSINESS CASE?

As mentioned earlier, the term *business case* refers to the reasons, or justification, for a proposal. A business case should be comprehensive, yet easy to understand. It should describe the project clearly, provide the justification to proceed, and estimate the project's financial impact. ProSci's BPR Online Learning Center, as shown in Figure 2-7, offers a Business Case Tutorial Series. According to ProSci, the business case should answer questions such as the following:

- Why are we doing this project?
- What is the project about?
- How does this solution address key business issues?

- How much will it cost and how long will it take?
- Will we suffer a productivity loss during the transition?
- What is the return on investment and payback period?
- What are the risks of doing the project? What are the risks of *not* doing the project?
- How will we measure success?
- What alternatives exist?

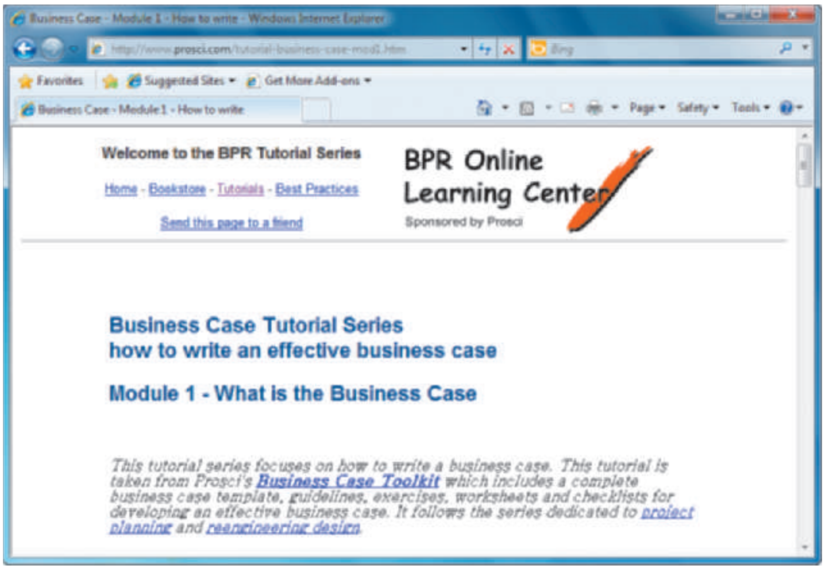


FIGURE 2-7 ProSci's BPR Online Learning Center offers a Business Case Tutorial Series that focuses on how to write a business case.

Information Systems Projects

This section discusses reasons for systems projects and the internal and external factors that affect systems projects. The section also includes a preview of project management, which is discussed in detail in Chapter 3.

Main Reasons for Systems Projects

The starting point for most projects is called a **systems request**, which is a formal way of asking for IT support. A systems request might propose enhancements for an existing system, the correction of problems, the replacement of an older system, or the development of an entirely new information system that is needed to support a company's current and future business needs.

As Figure 2-8 shows, the main reasons for systems requests are improved service to customers, better performance, support for new products and services, more information, stronger controls, and reduced cost.

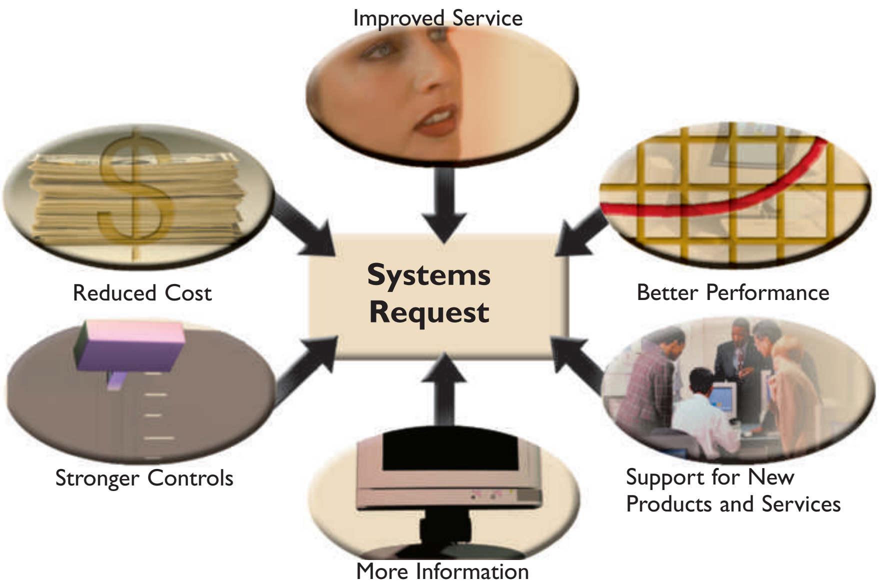


FIGURE 2-8 Six main reasons for systems requests.

IMPROVED SERVICE Systems requests often are aimed at improving service to customers or users within the company. Allowing mutual fund investors to check their account balances on a Web site, storing data on rental car customer preferences, or creating an online college registration system are examples that provide valuable services and increased customer satisfaction.

SUPPORT FOR NEW PRODUCTS AND SERVICES New products and services often require new types or levels of IT support. For example, a software vendor might offer an automatic upgrade service for subscribers; or a package delivery company might add a special service for RFID-tagged shipments. In situations like these, it is most likely that additional IT support will be required. At the other end of the spectrum, product obsolescence can also be an important factor in IT planning. As new products enter the marketplace, vendors often announce that they will no longer provide support for older versions. A lack of vendor support would be an important consideration in deciding whether or not to upgrade.

ON THE WEB

To learn more about biometric devices, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Biometric Devices link.

BETTER PERFORMANCE The current system might not meet performance requirements. For example, it might respond slowly to data inquiries at certain times, or it might be unable to support company growth. Performance limitations also result when a system that was designed for a specific hardware configuration becomes obsolete when new hardware is introduced.

MORE INFORMATION The system might produce information that is insufficient, incomplete, or unable to support the company's changing information needs. For example, a system that tracks customer orders might not be capable of analyzing and predicting marketing trends. In the face of intense competition and rapid product development cycles, managers need the best possible information to make major decisions on planning, designing, and marketing new products and services.

FIGURE 2-9 Students at West Virginia University use a hand scanning device to identify themselves.

STRONGER CONTROLS A system must have effective controls to ensure that data is secure and accurate. Some common security controls include passwords, various levels of user access, and **encryption**, or coding of data to keep it safe from unauthorized users. Hardware-based security controls include **biometric devices** that can identify a person by a retina scan or by mapping a facial pattern. A new biometric tool scans hands, rather than faces. The technology uses infrared scanners that create images with thousands of measurements of hand and finger characteristics, as shown in Figure 2-9.

In addition to being secure, data also must be accurate. Controls should minimize data entry errors whenever possible. For example, if a user enters an invalid customer number, the order processing system should reject the entry immediately and prompt the user to enter a valid number. Data entry controls must be effective without being excessive. If a system requires users to confirm every item with an "Are you sure? Y/N" message, internal users and customers might complain that the system is not user-friendly.

REDUCED COST The current system could be expensive to operate or maintain as a result of technical problems, design weaknesses, or the changing demands of the business. It might be possible to adapt the system to newer technology or upgrade it. On the other hand, cost-benefit analysis might show that a new system would be more cost effective and provide better support for long-term objectives.

CASE IN POINT 2.3: TRENT COLLEGE

Trent College is a private school in a small Maryland town. The college has outgrown its computerized registration system and is considering a new system. Althea Riddick, the college president, has asked you to list the reasons for systems projects, which are described on pages 59–61, and assign a relative weight to each reason, using a scale of 1 – 10, low to high. She said to use your best judgment, and support your conclusions in a brief memo to her. She also wants you to create a Microsoft Excel spreadsheet that will calculate the weighted values automatically for each reason.

Factors that Affect Systems Projects

Internal and external factors affect every business decision that a company makes, and IT systems projects are no exception. Figure 2-10 shows the main internal and external factors.

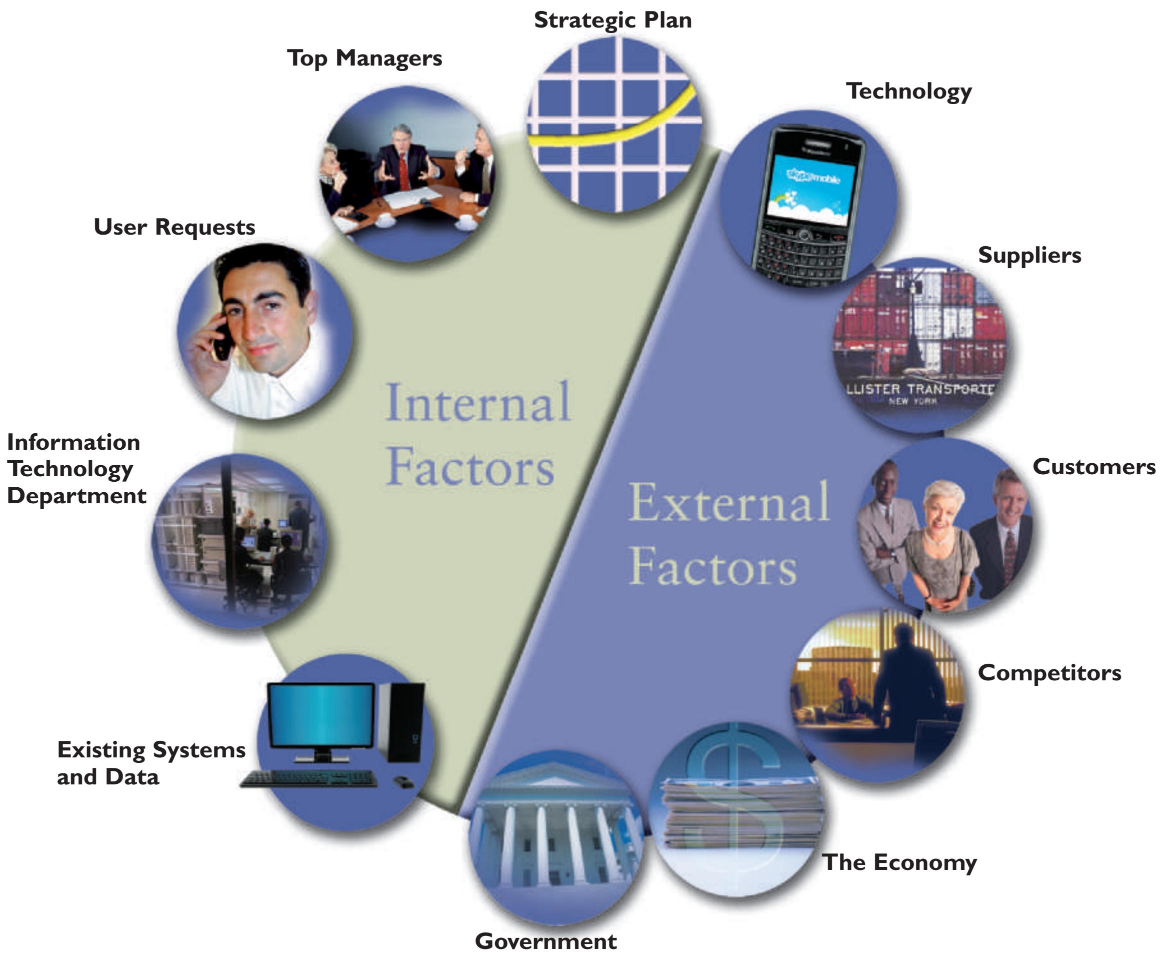


FIGURE 2-10 Internal and external factors that affect IT systems projects.

Internal Factors

Internal factors include the strategic plan, top managers, user requests, information technology department, and existing systems and data.

STRATEGIC PLAN A company's strategic plan sets the overall direction for the firm and has an important impact on IT projects. Company goals and objectives that need IT support will generate systems requests and influence IT priorities. A strategic plan that stresses technology tends to create a favorable climate for IT projects that extends throughout the organization.

TOP MANAGERS Directives from top managers are a prime source of large-scale systems projects. Those directives often result from strategic business decisions that require new IT systems, more information for decision making, or better support for mission-critical information systems.

USER REQUESTS As users rely more heavily on information systems to perform their jobs, they are likely to request even more IT services and support. For example, sales reps might request improvements to the company's Web site, a more powerful sales analysis report, a network to link all sales locations, or an online system that allows customers to obtain the status of their orders instantly. Or, users might not be satisfied with the current system because it is difficult to learn or lacks flexibility. They might want information systems support for business requirements that did not even exist when the system was developed.

INFORMATION TECHNOLOGY DEPARTMENT Many systems project requests come from the IT department. IT staff members often make recommendations based on their knowledge of business operations and technology trends. IT proposals might be strictly technical matters, such as replacement of certain network components, or suggestions might be more business oriented, such as proposing a new reporting or data collection system.

EXISTING SYSTEMS AND DATA Errors or problems in existing systems can trigger requests for systems projects. When dealing with older systems, analysts sometimes spend too much time reacting to day-to-day problems without looking at underlying causes. This approach can turn an information system into a patchwork of corrections and changes that cannot support the company's overall business needs. This problem typically occurs with legacy systems, which are older systems that are less technologically advanced. When migrating to a new system, IT planners must plan the conversion of existing data, which is described in detail in Chapter 11, Managing Systems Implementation.

External Factors

ON THE WEB

To learn more about JIT systems, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the JIT Systems link.

External factors include technology, suppliers, customers, competitors, the economy, and government.

TECHNOLOGY Changing technology is a major force affecting business and society in general. For example, the rapid growth of telecommunications has created entire new industries and technologies. Technology also dramatically reshapes existing business operations. The success of scanner technology resulted in universal bar coding that now affects virtually all products.

Some industry experts predict that bar code technology will be overshadowed in the future by **electronic product code** (EPC) technology that uses RFID tags to identify and monitor the movement of each individual product, from the factory floor to the retail checkout counter.

SUPPLIERS With the growth of electronic data interchange (EDI), relationships with suppliers are critically important. For example, an automobile company might require that suppliers code their parts in a certain manner to match the auto company's inventory control system. EDI also enables **just-in-time** (**JIT**) inventory systems, as shown in Figure 2-11, which rely on computer-to-computer data exchange to minimize unnecessary inventory. The purpose of a JIT system is to provide the right product at the right place at the right time.

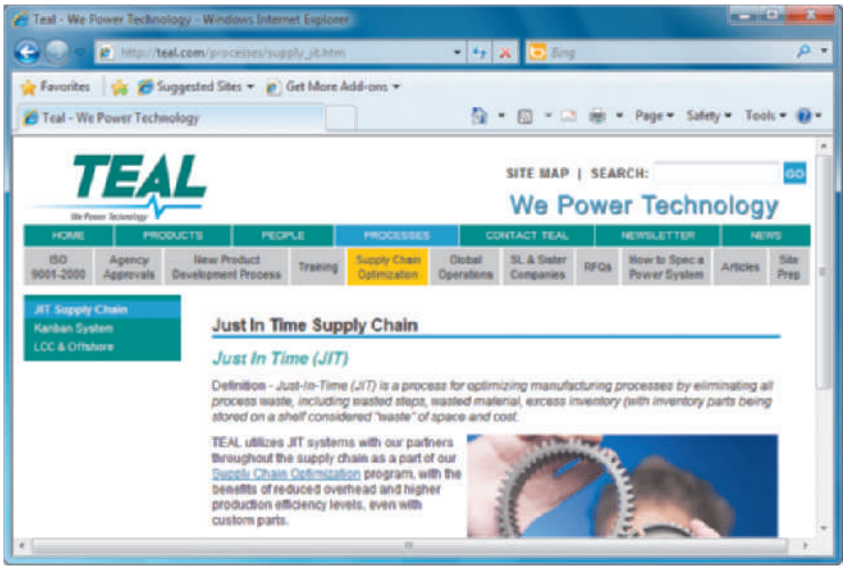


FIGURE 2-11 Just-in-time (JIT) inventory systems rely on computer-to-computer data exchange to minimize unnecessary inventory.

CUSTOMERS Customers are vitally important to any business. Information systems that interact with customers usually receive top priority. Many companies implement **customer relationship management** (**CRM**) systems that integrate all customer-related events and transactions, including marketing, sales, and customer service activities. Vendor-oriented CRM systems often interconnect with supplier relationship management (SRM) systems, which were discussed in Chapter 1. CRM components can provide automated responses to sales inquiries, Web-based order processing, and online inventory tracking. Because an efficient warehouse is just as important as a successful Web site, suppliers use *smart* forklifts that can read RFID tags or UPC numbers and transmit data to a CRM system, as shown in Figure 2-12.

One of the newest RFID applications is called **electronic proof of delivery** (**EPOD**). Using EPOD, a supplier uses RFID tags on each crate, case, or shipping unit to create a digital shipping list. The customer receives the list and scans the incoming shipment. If a discrepancy is detected, it is reported and adjusted automatically. Because they would be expensive to investigate manually, small shipping inconsistencies might not otherwise be traced. This is an example of technology-related cost control.

FIGURE 2-12 In an efficient warehouse, *smart* forklifts can read RFID tags or UPC numbers and transmit data to a CRM system.

COMPETITORS Competition drives many information systems decisions. For example, if one cellular telephone provider offers a new type of digital service, other firms must match the plan in order to remain competitive. New product research and development, marketing, sales, and service all require IT support.

ON THE WEB

To learn more about CRM systems, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the CRM Systems link.

VIDEO LEARNING SESSIONS

To learn more about project management, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and navigate to the **Video Learning Sessions** for this book. These sessions can help you understand key concepts, practice your skills, and check your work.

THE ECONOMY Economic activity has a powerful influence on corporate information management. In a period of economic expansion, firms need to be ready with scalable systems that can handle additional volume and growth. Predicting the business cycle is not an exact science, and careful research and planning is critically important.

GOVERNMENT Federal, state, and local government regulations affect the design of corporate information systems. For example, income tax reporting requirements must be designed into a payroll package. The debate about Internet sales tax issues could profoundly affect e-commerce, as well as traditional retail businesses.

Project Management

As mentioned earlier, business case analysis involves consideration of project reasons, costs, benefits, and risks. At the end of the preliminary investigation, if the project is approved, it can be planned, scheduled, monitored and controlled, and reported upon. Individual analysts or IT staff members often handle small projects, but companies usually designate a project manager to coordinate the overall effort for complex projects.

In Chapter 3, you will study project management concepts, skills, tools, and techniques. You also will learn about project risk management, and how to perform the following tasks:

- Develop a project risk management plan
- Identify the risks
- Analyze the risks
- Create a risk response plan
- Monitor and respond to risks

Figure 2-13 shows the latest version of Microsoft Project, a popular project management tool. Using this program, a project manager can define project tasks, list activities and participants, plan the sequence of work, estimate milestone dates, and track costs.

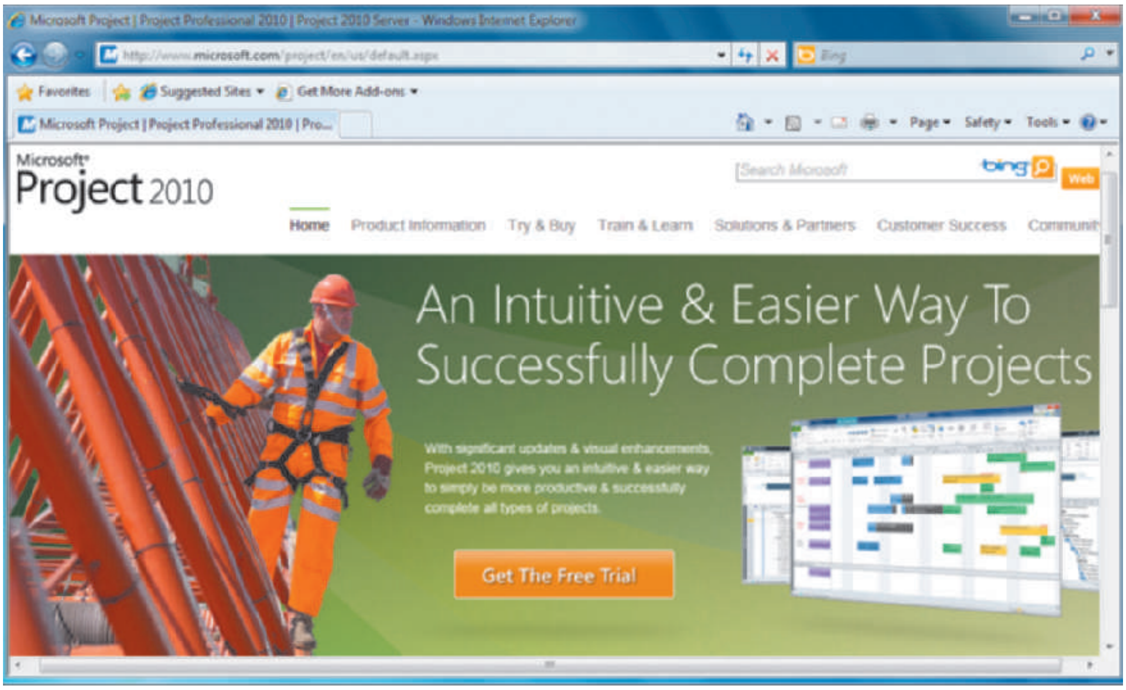


FIGURE 2-13 Microsoft Project is a powerful, popular software tool for project management.

Evaluation of Systems Requests

In most organizations, the IT department receives more systems requests than it can handle. Many organizations assign responsibility for evaluating systems requests to a group of key managers and users. Many companies call this group a **systems review committee** or a **computer resources committee**. Regardless of the name, the objective is to use the combined judgment and experience of several managers to evaluate systems projects.

Systems Request Forms

Many organizations use a special form for systems requests, similar to the online sample shown in Figure 2-14. A properly designed form streamlines the request process and ensures consistency. The form must be easy to understand and include clear instructions. It should include enough space for all required information and should indicate what supporting documents are needed. Many companies use online systems request forms that can be filled in and submitted electronically.

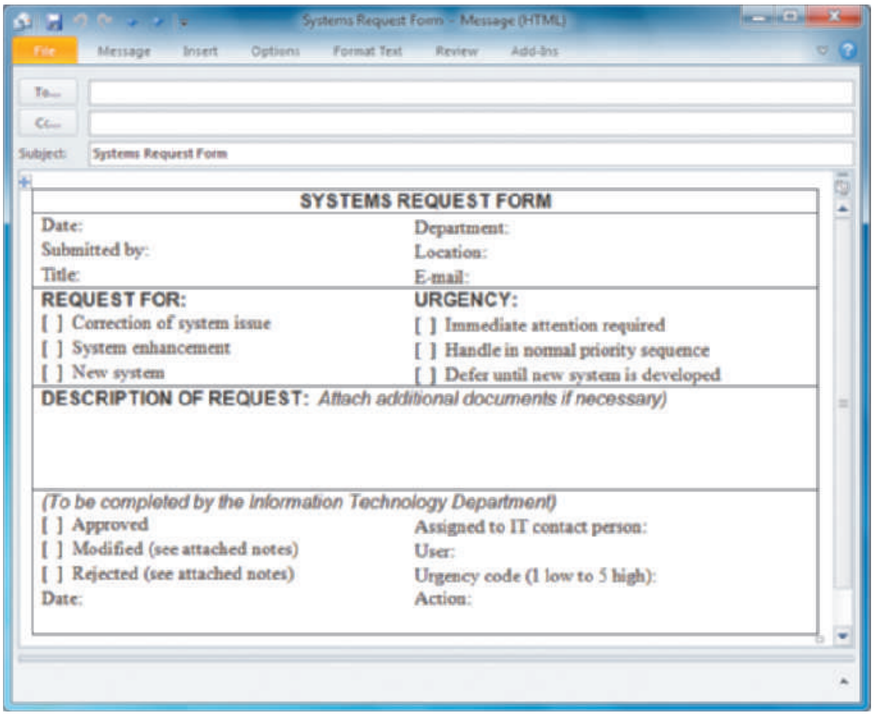


FIGURE 2-14 Example of an online systems request form.

When a systems request form is received, a systems analyst or IT manager examines it to determine what IT resources are required for the preliminary investigation. A designated person or a committee then decides whether to proceed with a preliminary investigation. Occasionally a situation will arise that requires an immediate response. For example, if the problem involves a mission-critical system, an IT maintenance team would attempt to restore normal operations. When the system is functioning properly, the team conducts a review and prepares a systems request to cover the work that was performed.

Systems Review Committee

Most large companies use a systems review committee to evaluate systems requests. Instead of relying on a single individual, a committee approach provides a variety of experience and knowledge. With a broader viewpoint, a committee can establish priorities more effectively than an individual, and one person's bias is less likely to affect the decisions. A typical committee consists of the IT director and several managers from other departments. The IT director usually serves as a technical consultant to ensure that committee members are aware of crucial issues, problems, and opportunities.

Although a committee offers many advantages, some disadvantages exist. For example, action on requests must wait until the committee meets. To avoid delay, committee members typically use e-mail and teleconferencing to communicate. Another potential disadvantage of a committee is that members might favor projects requested by their own departments, and internal political differences could delay important decisions.

Many smaller companies rely on one person to evaluate system requests instead of a committee. If only one person has the necessary IT skills and experience, that person must consult closely with users and managers throughout the company to ensure that business and operational needs are considered carefully.

Whether one person or a committee is responsible, the goal is to evaluate the requests and set priorities. Suppose four requests must be reviewed: the marketing group wants to analyze current customer spending habits and forecast future trends; the technical support group wants a cellular link so service representatives can download technical data instantly; the accounting department wants to redesign customer statements and allow Internet access; and the production staff wants an inventory control system that can exchange data with major suppliers. Which projects should the firm pursue? What criteria should be applied? How should priorities be determined? To answer those questions, the individual or the committee must assess the feasibility of each request.

OVERVIEW OF FEASIBILITY

As you learned in Chapter 1, a systems request must pass several tests, called a feasibility study, to see whether it is worthwhile to proceed further. As shown in Figure 2-15, a feasibility study uses four main yardsticks to measure a proposal: operational feasibility, technical feasibility, economic feasibility, and schedule feasibility.

Sometimes a feasibility study is quite simple and can be done in a few hours. If the request involves a new system or a major change, however, extensive fact-finding and investigation is required.

How much effort needs to go into a feasibility study? That depends on the

FIGURE 2-15 A feasibility study includes tests for operational, technical, economic, and schedule feasibility.

request. For example, if a department wants an existing report sorted in a different order, the analyst can decide quickly whether the request is feasible. On the other hand, a proposal by the marketing department for a new market research system to predict sales trends requires more effort. In both cases, the systems analyst asks these important questions:

- Is the proposal desirable in an operational sense? Is it a practical approach that will solve a problem or take advantage of an opportunity to achieve company goals?
- Is the proposal technically feasible? Are the necessary technical resources and people available for the project?
- Is the proposal economically desirable? What are the projected savings and costs? Are other intangible factors involved, such as customer satisfaction or company image? Is the problem worth solving, and will the request result in a sound business investment?
- Can the proposal be accomplished within an acceptable time frame?

To obtain more information about a systems request, you might perform initial fact-finding by studying organization charts, performing interviews, reviewing current documentation, observing operations, and surveying users. If the systems request is approved, more intensive fact-finding will continue during the systems analysis phase.

Operational Feasibility

Operational feasibility means that a proposed system will be used effectively after it has been developed. If users have difficulty with a new system, it will not produce the expected benefits. Operational feasibility depends on several vital issues. For example, consider the following questions:

- Does management support the project? Do users support the project? Is the current system well liked and effectively used? Do users see the need for change?
- Will the new system result in a workforce reduction? If so, what will happen to affected employees?
- Will the new system require training for users? If so, is the company prepared to provide the necessary resources for training current employees?
- Will users be involved in planning the new system right from the start?
- Will the new system place any new demands on users or require any operating changes? For example, will any information be less accessible or produced less frequently? Will performance decline in any way? If so, will an overall gain to the organization outweigh individual losses?
- Will customers experience adverse effects in any way, either temporarily or permanently?
- Will any risk to the company's image or goodwill result?
- Does the development schedule conflict with other company priorities?
- Do legal or ethical issues need to be considered?

Technical Feasibility

Technical feasibility refers to the technical resources needed to develop, purchase, install, or operate the system. When assessing technical feasibility, an analyst must consider the following points:

- Does the company have the necessary hardware, software, and network resources? If not, can those resources be acquired without difficulty?

- Does the company have the needed technical expertise? If not, can it be acquired?
- Does the proposed platform have sufficient capacity for future needs? If not, can it be expanded?
- Will a prototype be required?
- Will the hardware and software environment be reliable? Will it integrate with other company information systems, both now and in the future? Will it interface properly with external systems operated by customers and suppliers?
- Will the combination of hardware and software supply adequate performance? Do clear expectations and performance specifications exist?
- Will the system be able to handle future transaction volume and company growth?

ON THE WEB

To learn more about TCO, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the TCO link.

Economic Feasibility

Economic feasibility means that the projected benefits of the proposed system outweigh the estimated costs usually considered the **total cost of ownership** (**TCO**), which includes ongoing support and maintenance costs, as well as acquisition costs. To determine TCO, the analyst must estimate costs in each of the following areas:

- People, including IT staff and users
- Hardware and equipment
- Software, including in-house development as well as purchases from vendors
- Formal and informal training
- Licenses and fees
- Consulting expenses
- Facility costs
- The estimated cost of not developing the system or postponing the project

VIDEO LEARNING SESSIONS

To learn more about financial tools, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and navigate to the **Video Learning Sessions** for this book. These sessions can help you understand key concepts, practice your skills, and check your work.

In addition to costs, you need to assess tangible and intangible benefits to the company. The systems review committee will use those figures, along with your cost estimates, to decide whether to pursue the project beyond the preliminary investigation phase.

Tangible benefits are benefits that can be measured in dollars. Tangible benefits result from a decrease in expenses, an increase in revenues, or both. Examples of tangible benefits include the following:

- A new scheduling system that reduces overtime
- An online package tracking system that improves service and decreases the need for clerical staff
- A sophisticated inventory control system that cuts excess inventory and eliminates production delays

TOOLKIT TIME

The Financial Analysis tools in Part C of the Systems Analyst's Toolkit can help you analyze project costs, benefits, and economic feasibility. To learn more about these tools, turn to Part C of the four-part Toolkit that follows Chapter 12.

Intangible benefits are advantages that are difficult to measure in dollars but are important to the company. Examples of intangible benefits include the following:

- A user-friendly system that improves employee job satisfaction
- A sales tracking system that supplies better information for marketing decisions
- A new Web site that enhances the company's image

You also must consider the development timetable, because some benefits might occur as soon as the system is operational, but others might not take place until later.

Schedule Feasibility

Schedule feasibility means that a project can be implemented in an acceptable time frame. When assessing schedule feasibility, a systems analyst must consider the interaction between time and costs. For example, speeding up a project schedule might make a project feasible, but much more expensive.

Other issues that relate to schedule feasibility include the following:

- Can the company or the IT team control the factors that affect schedule feasibility?
- Has management established a firm timetable for the project?
- What conditions must be satisfied during the development of the system?
- Will an accelerated schedule pose any risks? If so, are the risks acceptable?
- Will project management techniques be available to coordinate and control the project?
- Will a project manager be appointed?

Chapter 3 describes various project management tools and techniques.

EVALUATING FEASIBILITY

The first step in evaluating feasibility is to identify and weed out systems requests that are not feasible. For example, a request would not be feasible if it required hardware or software that the company already had rejected.

Even if the request is feasible, it might not be necessary. For example, a request for multiple versions of a report could require considerable design and programming effort. A better alternative might be to download the server data to a personal computer-based software package and show users how to produce their own reports. In this case, training users would be a better investment than producing reports for them.

Also keep in mind that systems requests that are not currently feasible can be resubmitted as new hardware, software, or expertise becomes available. Development costs might decrease, or the value of benefits might increase enough that a systems request eventually becomes feasible. Conversely, an initially feasible project can be rejected later. As the project progresses, conditions often change. Acquisition costs might increase, and the project might become more expensive than anticipated. In addition, managers and users sometimes lose confidence in a project. For all those reasons, feasibility analysis is an ongoing task that must be performed throughout the systems development process.

SETTING PRIORITIES

After rejecting systems requests that are not feasible, the systems review committee must establish priorities for the remaining items. The highest priority goes to projects that provide the greatest benefit, at the lowest cost, in the shortest period of time. Many factors, however, influence project evaluation.

Factors that Affect Priority

When assessing a project's priority, a systems analyst should consider the following:

- Will the proposed system reduce costs? Where? When? How? How much?
- Will the system increase revenue for the company? Where? When? How? How much?
- Will the systems project result in more information or produce better results? How? Are the results measurable?
- Will the system serve customers better?
- Will the system serve the organization better?
- Can the project be implemented in a reasonable time period? How long will the results last?
- Are the necessary financial, human, and technical resources available?

Very few projects will score high in all areas. Some proposed systems might not reduce costs but will provide important new features. Other systems might reduce operating costs substantially but require the purchase or lease of additional hardware. Some systems might be very desirable but require several years of development before producing significant benefits.

Whenever possible, the analyst should evaluate a proposed project based on tangible costs and benefits that represent actual (or approximate) dollar values. For example, a reduction of $8,000 in network maintenance is an example of a tangible benefit.

Often, the evaluation involves intangible costs or benefits, as described in the section on economic feasibility. In contrast to tangible benefits, such as the network cost reduction example, it is more difficult to assign dollar values to intangible benefits such as enhancing the organization's image, raising employee morale, or improving customer service. Intangible costs and benefits often influence systems decisions and priorities and must be considered carefully.

Discretionary and Nondiscretionary Projects

Is the project absolutely necessary? Projects where management has a choice in implementing them are called **discretionary projects**. Projects where no choice exists are called **nondiscretionary projects**. Creating a new report for a user is an example of a discretionary project; adding a report required by a new federal law is an example of a nondiscretionary project.

If a particular project is not discretionary, is it really necessary for the systems review committee to evaluate it? Some people believe that waiting for committee approval delays critical nondiscretionary projects unnecessarily. Others believe that by submitting all systems requests to the systems review committee, the committee is kept aware of all projects that compete for the resources of the IT department. As a result, the committee assesses the priority of discretionary projects and can schedule them more realistically. Additionally, the committee might need to prioritize nondiscretionary projects when funds or staff are limited.

Many nondiscretionary projects are predictable. Examples include annual updates to payroll, tax percentages, or quarterly changes in reporting requirements for an insurance processing system. By planning ahead for predictable projects, the IT department manages its resources better and keeps the systems review committee fully informed without needing prior approval in every case.

CASE IN POINT 2.4: Attaway Airlines, Part Two

Back at Attaway Airlines, the morning meeting ended with no agreement between Dan Esposito and Molly Kinnon. In fact, a new issue arose. Molly now says that the new accounting system is entitled to the highest priority because the federal government soon will require the reporting of certain types of company-paid health insurance premiums. Because the current system will not handle this report, she insists that the entire accounting system is a nondiscretionary project. As you might expect, Dan is upset. Can part of a project be nondiscretionary? What issues need to be discussed? The committee meets again tomorrow, and the members will look to you, as the IT director, for guidance.

Preliminary Investigation Overview

A systems analyst conducts a **preliminary investigation** to study the systems request and recommend specific action. After obtaining an authorization to proceed, the analyst interacts with managers and users, as shown in the model in Figure 2-16. The analyst gathers facts about the problem or opportunity, project scope and constraints, project benefits, and estimated development time and costs. The end product of the preliminary investigation is a report to management.

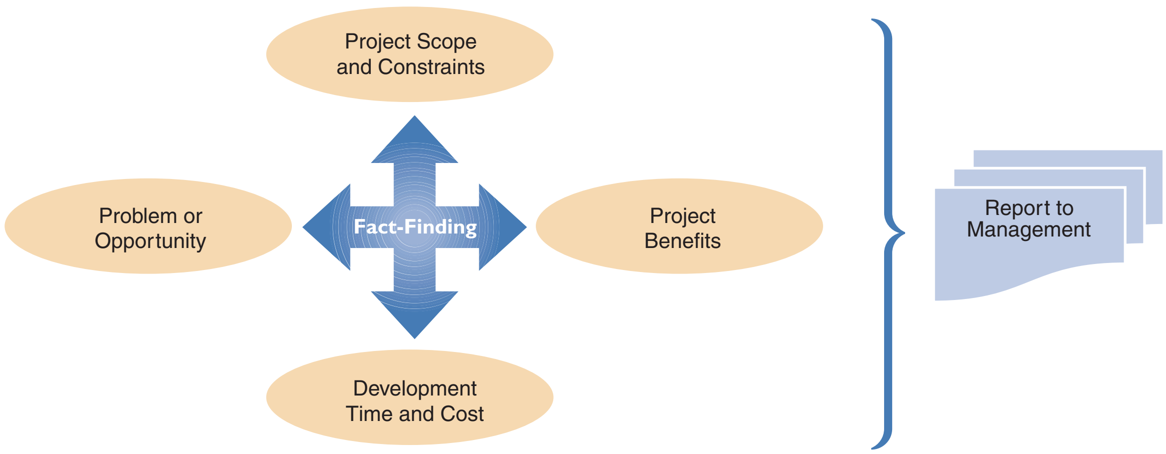


FIGURE 2-16 Model of a preliminary investigation.

Interaction with Managers and Users

Before beginning a preliminary investigation, a memo or an e-mail message should let people know about the investigation and explain your role. You should meet with key managers, users, and IT staff to describe the project, explain your responsibilities, answer questions, and invite comments. This starts an important dialogue with users that will continue throughout the entire development process.

A systems project often produces significant changes in company operations. Employees may be curious, concerned, or even opposed to those changes. It is not surprising to encounter some user resistance during a preliminary investigation. Employee attitudes and reactions are important and must be considered.

When interacting with users, you should be careful in your use of the word *problem*, because generally it has a negative meaning. When you ask users about *problems*, some will stress current system limitations rather than desirable new features or enhancements. Instead of focusing on difficulties, you should question users about additional capability they would like to have. Using this approach, you highlight ways to improve the user's job, you get a better understanding of operations, and you build better, more positive relationships with users.

Planning the Preliminary Investigation

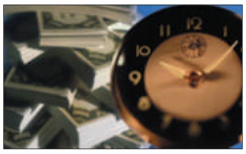
During a preliminary investigation, a systems analyst typically follows a series of steps, as shown in Figure 2-17. The exact procedure depends on the nature of the request, the size of the project, and the degree of urgency.

Step 1 **Understand the problem or opportunity.**

Step 2 **Define the project scope and constraints.**

Step 3 **Perform fact-finding.**
- Analyze organizational charts.
- Conduct interviews.
- Review documentation.
- Observe operations.
- Conduct a user survey.

Step 4 **Analyze project usability, cost, benefit, and schedule data.**



Step 5 **Evaluate feasibility**
- Operational
- Technical
- Economic
- Schedule

Step 6 **Present results and recommendations to management.**

FIGURE 2-17 Six steps in a preliminary investigation.

Figure 2-18 shows how a systems analyst might use Microsoft Project to plan and manage the preliminary investigation. Notice that the analyst has listed the tasks, estimated the duration of each task, and designated a specific order in which the tasks must be performed.

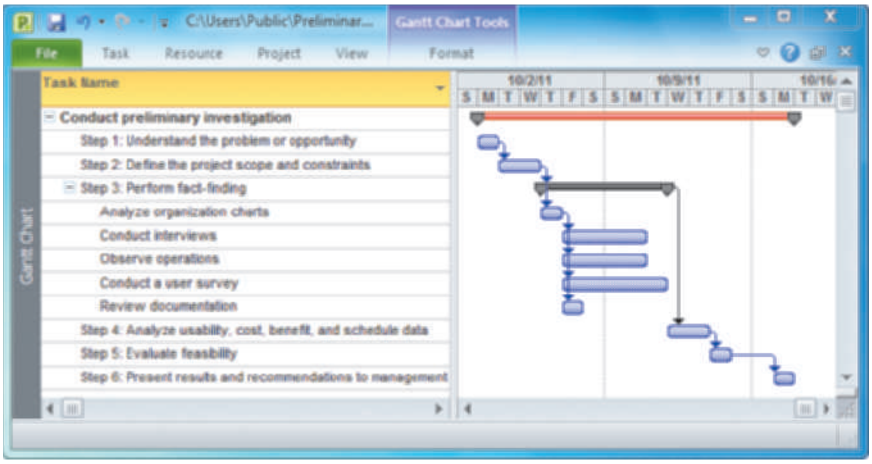


FIGURE 2-18 An analyst could use Microsoft Project to plan and manage a preliminary investigation.

Step 1: Understand the Problem or Opportunity

If the systems request involves a new information system or a substantial change in an existing system, systems analysts might need to develop a business profile that describes business processes and functions, as explained in Chapter 1. Even where the request involves relatively minor changes or enhancements, you need to understand how those modifications will affect business operations and other information systems. Often a change in one system has an unexpected effect on another system. When you analyze a systems request, you need to determine which departments, users, and business processes are involved.

In many cases, the systems request does not reveal the underlying problem, but only a symptom. For example, a request to investigate mainframe processing delays might reveal improper scheduling practices rather than hardware problems. Similarly, a request for analysis of customer complaints might disclose a lack of sales representative training, rather than problems with the product.

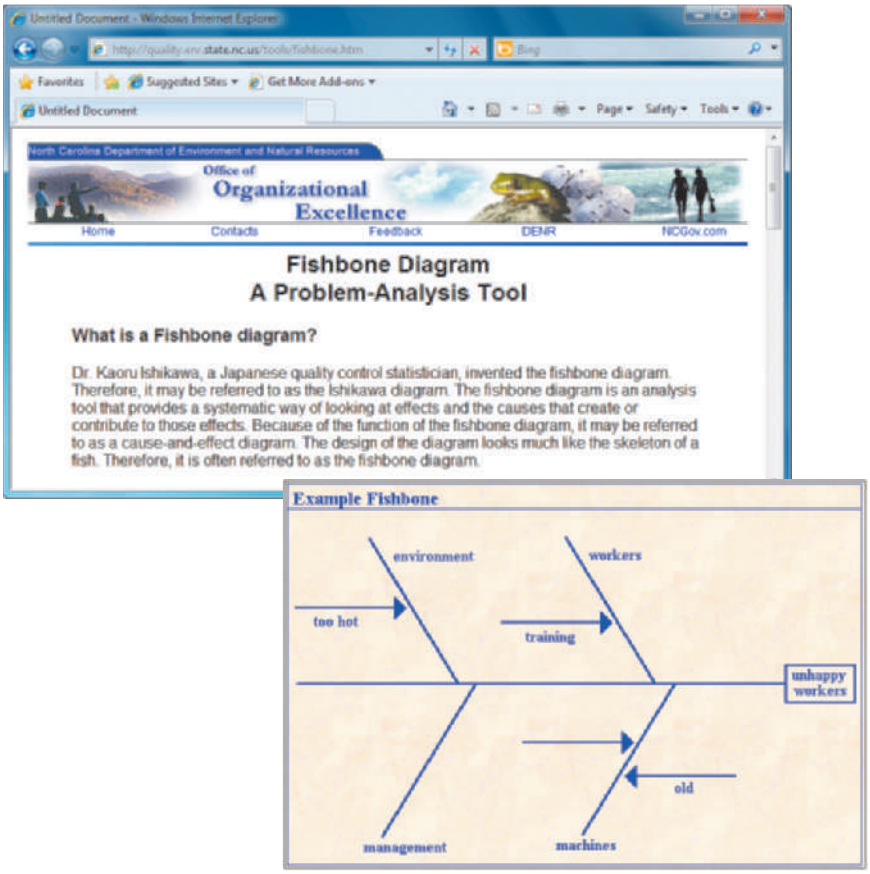


FIGURE 2-19 A fishbone diagram displays the causes of a problem. Typically, you must dig deeper to identify actual *causes* rather than just symptoms.

A popular technique for investigating causes and effects is called a **fishbone diagram**, or **Ishikawa diagram**, as shown in Figure 2-19. A fishbone diagram is an analysis tool that represents the possible causes of a problem as a graphical outline. When using a fishbone diagram, an analyst first states the problem and draws a main bone with sub-bones that represent possible causes of the problem. In the example shown in Figure 2-19, the problem is *unhappy workers*, and the analyst has identified four areas to investigate: *environment*, *workers*, *management*, and *machines*. In each area, the analyst identifies possible causes and draws them as horizontal sub-bones. For example, *too hot* is a possible cause in the *environment* bone. For each cause, the analyst must dig deeper and ask the question: What could be causing *this* symptom to occur? For example, *why* is it too hot? If the answer is

ON THE WEB

To learn more about fishbone diagrams, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Fishbone Diagram link.

insufficient air conditioning capacity, the analyst indicates this as a sub-bone to the *too hot* cause. In this manner, the analyst adds additional sub-bones to the diagram, until he or she uncovers root causes of a problem, rather than just the symptoms.

Step 2: Define the Project Scope and Constraints

Determining the **project scope** means defining the specific boundaries, or extent, of the project. For example, a statement that, *payroll is not being produced accurately* is very general, compared with the statement *overtime pay is not being calculated correctly for production workers on the second shift at the Yorktown plant*. Similarly, the statement, *the project scope is to modify the accounts receivable system*, is not as specific as the statement, *the project scope is to allow customers to inquire online about account balances and recent transactions*.

Some analysts find it helpful to define project scope by creating a list with sections called *Must Do*, *Should Do*, *Could Do,* and *Won't Do*. This list can be reviewed later, during the systems analysis phase, when the systems requirements document is developed.

Projects with very general scope definitions are at risk of expanding gradually, without specific authorization, in a process called **project creep**. To avoid this problem, you should define project scope as clearly as possible. You might want to use a graphical model that shows the systems, people, and business processes that will be affected. The scope of the project also establishes the boundaries of the preliminary investigation itself. A systems analyst should limit the focus to the problem at hand and avoid unnecessary expenditure of time and money.

Along with defining the scope of the project, you need to identify any constraints on the system. A **constraint** is a requirement or condition that the system must satisfy or an outcome that the system must achieve. A constraint can involve hardware, software, time, policy, law, or cost. System constraints also define project scope. For example, if the system must operate with existing hardware, that is a constraint that affects potential solutions. Other examples of constraints are: The order entry system must accept input from 15 remote sites; the human resources information system must produce statistics on hiring practices; and the new Web site must be operational by March 1. When examining constraints, you should identify their characteristics.

PRESENT VERSUS FUTURE Is the constraint something that must be met as soon as the system is developed or modified, or is the constraint necessary at some future time?

INTERNAL VERSUS EXTERNAL Is the constraint due to a requirement within the organization or does some external force, such as government regulation, impose it?

MANDATORY VERSUS DESIRABLE Is the constraint mandatory? Is it absolutely essential to meet the constraint, or is it merely desirable?

Figure 2-20 shows five examples of constraints. Notice that each constraint has three characteristics, which are indicated by its position in the figure and by the symbol that represents the constraint. The constraint in Example A is present, external, and mandatory. The constraint in Example B is future, external, and mandatory. The constraint in Example C is present, internal, and desirable. The constraint in Example D is present, internal, and mandatory. The constraint in Example E is future, internal, and desirable.

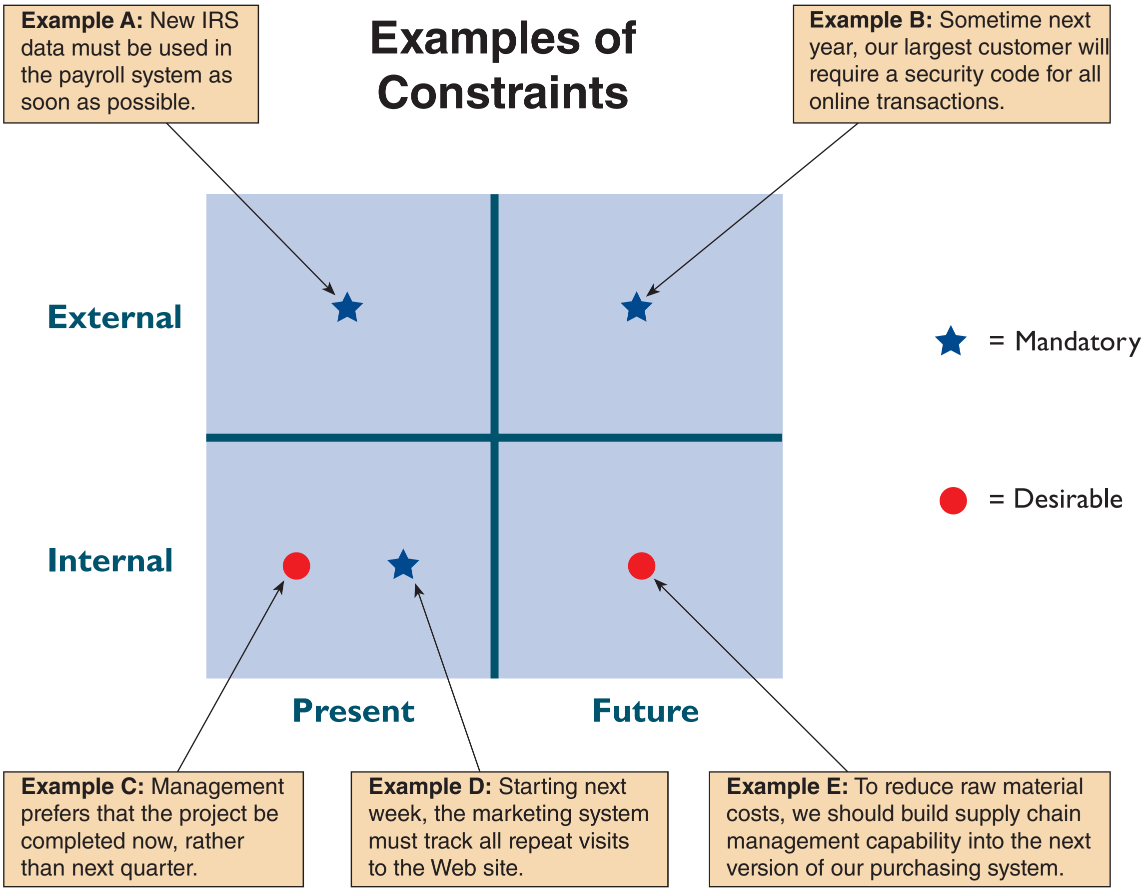


FIGURE 2-20 Examples of various types of constraints. The constraint in Example A is present, external, and mandatory. The constraint in Example B is future, external, and mandatory. The constraint in Example C is present, internal, and desirable. The constraint in Example D is present, internal, and mandatory. The constraint in Example E is future, internal, and desirable.

Regardless of the type, all constraints should be identified as early as possible to avoid future problems and surprises. A clear definition of project scope and constraints avoids misunderstandings that arise when managers assume that the system will have a certain feature or support for a project, but later find that the feature is not included.

Step 3: Perform Fact-Finding

The objective of fact-finding is to gather data about project usability, costs, benefits, and schedules. Fact-finding involves various techniques, which are described below. Depending on what information is needed to investigate the systems request, fact-finding might consume several hours, days, or weeks. For example, a change in a report format or data entry screen might require a single telephone call or e-mail message to a user, whereas a new inventory system would involve a series of interviews. During fact-finding, you might analyze organization charts, conduct interviews, review current documentation, observe operations, and carry out a user survey.

ANALYZE ORGANIZATION CHARTS In many instances, you will not know the organizational structure of departments involved in the study. You should obtain organization charts to understand how the department functions and identify individuals you might want to interview. Organization charts often can be obtained from the company's human resources department. If such charts are unavailable, you should obtain the necessary information directly from department personnel and then construct your own charts, as shown in Figure 2-21.

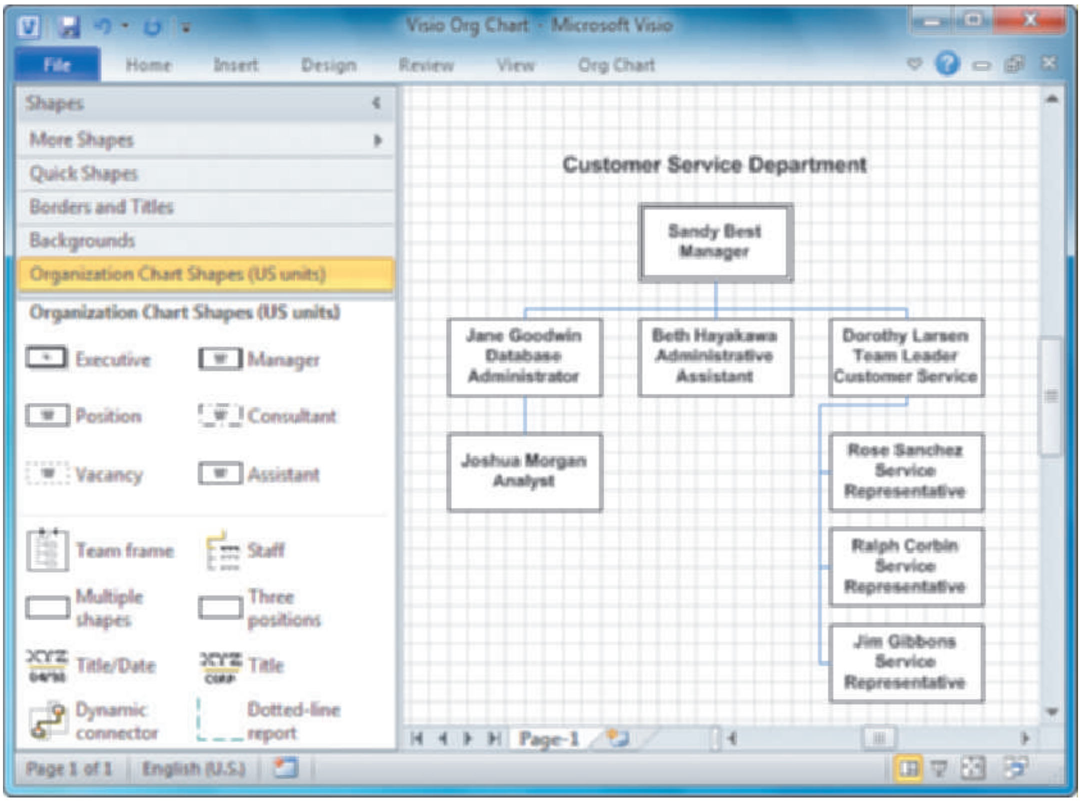


FIGURE 2-21 Microsoft Visio includes an organization chart drawing tool that is powerful and easy to use.

When organization charts are available, you should verify their accuracy. Keep in mind that organization charts show formal reporting relationships but not the informal alignment of a group, which also is important.

CONDUCT INTERVIEWS The primary method of obtaining information during the preliminary investigation is the interview. The interviewing process involves a series of steps:

1. Determine the people to interview.
2. Establish objectives for the interview.
3. Develop interview questions.
4. Prepare for the interview.
5. Conduct the interview.
6. Document the interview.
7. Evaluate the interview.

These seven steps are discussed in detail in Chapter 4, which describes fact-finding techniques that occur during the systems analysis phase of the SDLC.

Remember that the purpose of the interview, and of the preliminary investigation itself, is to uncover facts, not to convince others that the project is justified. Your primary role in an interview is to ask effective questions and listen carefully. If you plan to talk to several people about the same topic, you should prepare a standard set of questions for all the interviews. Also be sure to include open-ended questions, such as "What else do you think I should know about the system?" or "Is there any other relevant information that we have not discussed?"

When conducting interviews during the preliminary investigation, you should interview managers and supervisors who have a broad knowledge of the system and can give you an overview of the business processes involved. Depending on the situation, you might talk to operational personnel to learn how the system functions on a day-to-day basis.

REVIEW DOCUMENTATION Although interviews are an extremely important method of obtaining information, you also might want to investigate the current system documentation. The documentation might not be up to date, so you should check with users to confirm that you are receiving accurate and complete information.

OBSERVE OPERATIONS Another fact-finding method is to observe the current system in operation, as shown in Figure 2-22. You might see how workers carry out typical tasks. You might choose to trace or follow the actual paths taken by input source documents or output reports. In addition to observing operations, you might want to sample the inputs or outputs of the system. Using sampling techniques described in Chapter 4, you can obtain valuable information about the nature and frequency of the problem.

FIGURE 2-22 Sometimes, an analyst can get a better understanding of a system by watching actual operations.

CONDUCT A USER SURVEY Interviews can be time consuming. Sometimes you can obtain information from a larger group by conducting a user survey. In this case, you design a form that users complete and return to you for tabulation. A survey is not as flexible as a series of interviews, but it is less expensive, generally takes less time, and can involve a broad cross-section of people.

ANALYZE THE DATA Systems analysts use many techniques to locate the source of a problem. For example, the **Pareto chart** is a widely used tool for visualizing issues that need attention. Named for a nineteenth century economist, a Pareto chart is drawn as a vertical bar graph, as shown in Figure 2-23. The bars, which represent various causes of a problem, are arranged in descending order, so the team can focus on the most important causes. In the example shown, a systems analyst might use a Pareto chart to learn more about the causes of inventory system problems, so that necessary improvements can be made. Creating Pareto charts with Excel is a simple process.

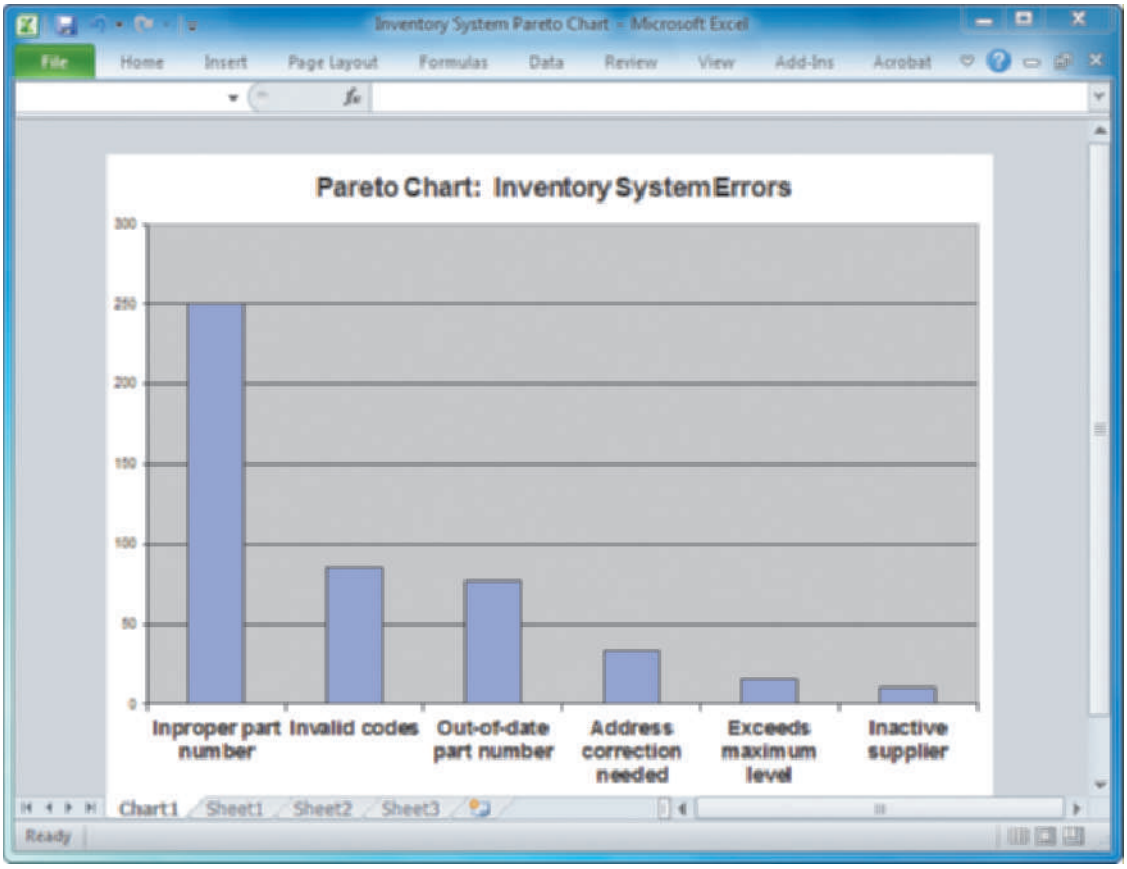


FIGURE 2-23 A Pareto chart can display and prioritize the causes of a problem, enabling an analyst to focus on the most important issues.

The **XY chart**, sometimes called a **scatter diagram**, is another problem-solving tool. Often, an analyst looks for a correlation between two variables. For example, suppose you are getting complaints about network response time, and you want to determine the cause. You would try to identify variables, such as the number of users, to see whether there is a correlation, or pattern. Figure 2-24 shows two XY charts with data samples. The first chart sample would suggest that there is no correlation between the delays and the number of users, and you would look elsewhere for the source of the problem. However, if the data resembles the second XY sample, it indicates a strong relationship between the number of users and the longer response times. That information would be extremely valuable in the problem-solving process.

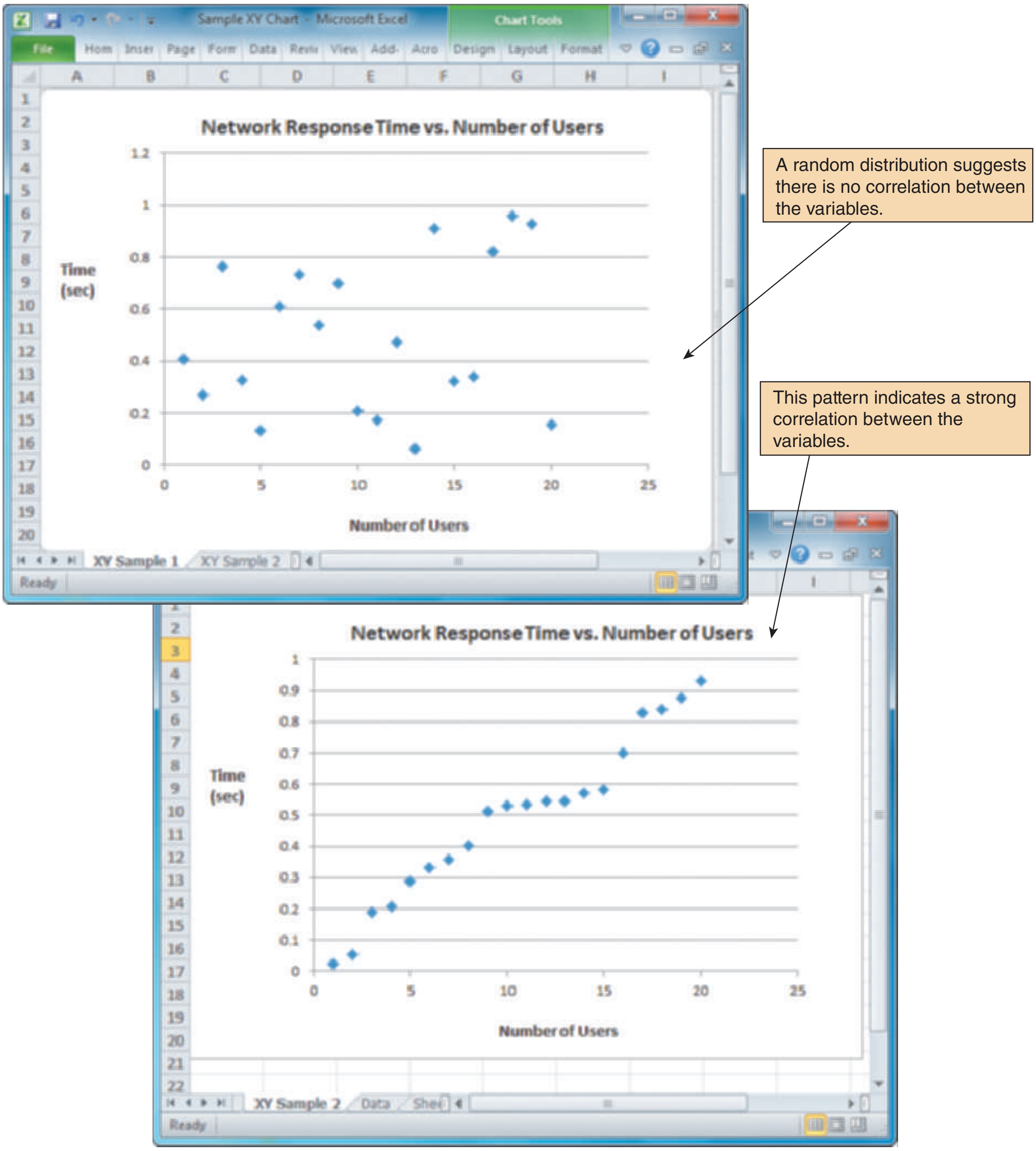


FIGURE 2-24 An XY chart shows correlation between variables, which is very important in problem-solving. Conversely, a *lack* of correlation suggests that the variables are independent, and that you should look elsewhere for the cause.

Step 4: Analyze Project Usability, Cost, Benefit, and Schedule Data

During fact-finding, you gathered data about the project's predicted costs, anticipated benefits, and schedule issues that could affect implementation. Before you can evaluate feasibility, you must analyze this data carefully. If you conducted interviews or used surveys, you should tabulate the data to make it easier to understand. If you observed current operations, you should review the results and highlight key facts that will be useful in the feasibility analysis. If you gathered cost and benefit data, you should be able to prepare financial analysis and impact statements using spreadsheets and other decision support tools.

Also, you should develop time and cost estimates for the requirements modeling tasks for the next SDLC phase, systems analysis. Specifically, you should consider the following:

- What information must you obtain, and how will you gather and analyze the information?
- Will you conduct interviews? How many people will you interview, and how much time will you need to meet with the people and summarize their responses?
- Will you conduct a survey? Who will be involved? How much time will it take people to complete it? How much time will it take to tabulate the results?
- How much will it cost to analyze the information and prepare a report with findings and recommendations?

Step 5: Evaluate Feasibility

You have analyzed the problem or opportunity, defined the project scope and constraints, and performed fact-finding to evaluate project usability, costs, benefits, and time constraints. Now you are ready to evaluate the project's feasibility. You should start by reviewing the answers to the questions listed on pages 67–69. Also consider the following guidelines:

OPERATIONAL FEASIBILITY Your fact-finding should have included a review of user needs, requirements, and expectations. When you analyze this data, you should look for areas that might present problems for system users and how they might be resolved. Because operational feasibility means that a system will be used effectively, this is a vital area of concern.

TECHNICAL FEASIBILITY The fact-finding data should identify the hardware, software, and network resources needed to develop, install, and operate the system. With this data, you can develop a checklist that will highlight technical costs and concerns, if any.

ECONOMIC FEASIBILITY Using the fact-finding data, you can apply the financial analysis tools described in Part C of the Systems Analyst's Toolkit to assess feasibility. The cost-benefit data will be an important factor for management to consider. Also, a cost estimate for the project development team will be built into the project management plan.

SCHEDULE FEASIBILITY The fact-finding data should include stakeholder expectations regarding acceptable timing and completion dates. As mentioned previously, often a trade-off exists between a project's schedule and its costs. For example, compressing a project schedule might be possible, but only if the budget is increased accordingly. The schedule data will be incorporated into the project plan in the form of task durations and milestones.

Step 6: Present Results and Recommendations to Management

TOOLKIT TIME

The Communication Tools in Part A of the Systems Analyst's Toolkit can help you develop better reports and presentations. To learn more about these tools, turn to Part A of the four-part Toolkit that follows Chapter 12.

At this stage, you have several alternatives. You might find that no action is necessary or that some other strategy, such as additional training, is needed. To solve a minor problem, you might implement a simple solution without performing further analysis. In other situations, you will recommend that the project proceed to the next development phase, which is systems analysis.

The final task in the preliminary investigation is to prepare a report to management, and possibly deliver a presentation, as shown in Figure 2-25. The report includes an evaluation of the systems request, an estimate of costs and benefits, and a **case for action**, which is a summary of the project request and a specific recommendation.

The format of a preliminary investigation report varies from one company to another. A typical report might consist of the following sections:

FIGURE 2-25 Oral presentations often are required during systems development, and systems analysts need to develop strong presentation skills.

- *Introduction* — the first section is an overview of the report. The introduction contains a brief description of the system, the name of the person or group who performed the investigation, and the name of the person or group who initiated the investigation.
- *Systems Request Summary* — the summary describes the basis of the systems request.
- *Findings* — the findings section contains the results of your preliminary investigation, including a description of the project's scope, constraints, and feasibility.
- *Case for Action* — a summary of the project request and a specific recommendation. Management will make the final decision, but the IT department's input is an important factor.
- *Project Roles* — this section lists the people who will participate in the project, and describes each person's role.
- *Time and Cost Estimates* — this section describes the cost of acquiring and installing the system, and the total cost of ownership during the system's useful life.
- *Expected Benefits* — this section includes anticipated tangible and intangible benefits and a timetable that shows when they are to occur.
- *Appendix* — an appendix is included in the report if you need to attach supporting information. For example, you might list the interviews you conducted, the documentation you reviewed, and other sources for the information you obtained. You do not need detailed reports of the interviews or other lengthy documentation. It is critical that you retain those documents to support your findings and for future reference.

A QUESTION OF ETHICS

As a new systems analyst at Premier Financial Services, you are getting quite an education. You report to Mary, the IT manager, who also chairs the systems review committee. Several months ago, the committee rejected a request from Jack, the finance director, for an expensive new accounts payable system, because the benefits did not appear to outweigh the costs.

Yesterday, Mary's boss called her in and asked her to reconsider Jack's request, and to persuade the other members to approve it. Mary wanted to discuss the merits of the request, but he cut her off rather abruptly. Mary happens to know that Jack and her boss are longtime friends.

Mary has confided in you. She is very uncomfortable about the meeting with her boss, and she believes that his request would undermine the integrity of the systems review process. Mary feels it would be unethical to grant preferred treatment just because a friendship is involved. She is thinking of submitting a request to step down as review committee chair, even though that might harm her career at the company.

Is this an ethical question, or just a matter of office politics? What would you say to Mary?

CHAPTER SUMMARY

Systems planning is the first phase of the systems development life cycle. Effective information systems help an organization support its business processes, carry out its mission, and serve its stakeholders. Strategic planning allows a company to examine its purpose, vision, and values and develops a mission statement, which leads to goals, objectives, day-to-day operations, and business results that affect company stakeholders.

During the systems planning phase, an analyst reviews the business case, which is the basis, or reason, for a proposed system. A business case should describe the project clearly, provide the justification to proceed, and estimate the project's financial impact.

Systems projects are initiated to improve performance, provide more information, reduce costs, strengthen controls, or provide better service. Various internal and external factors affect systems projects, such as user requests, top management directives, existing systems, the IT department, software and hardware vendors, technology, customers, competitors, the economy, and government.

During the preliminary investigation, the analyst evaluates the systems request and determines whether the project is feasible from an operation, technical, economic, and schedule standpoint. Analysts evaluate systems requests on the basis of their expected costs and benefits, both tangible and intangible.

The steps in the preliminary investigation are to understand the problem or opportunity; define the project scope and constraints; perform fact-finding; analyze project usability, cost, benefit, and schedule data; evaluate feasibility; and present results and recommendations to management. During the preliminary investigation, analysts often use investigative tools such as fishbone or Ishikawa diagrams, Pareto charts, and XY charts. The last task in a preliminary investigation is to prepare a report to management. The report must include an estimate of time, staffing requirements, costs, benefits, and expected results for the next phase of the SDLC.

Key Terms and Phrases

biometric devices *60*
business case *52*
case for action *81*
computer resources committee *65*
constraint *74*
critical success factors *57*
customer relationship management (CRM) *63*
discretionary projects *70*
economic feasibility *68*
electronic product code (EPC) *62*
electronic proof of delivery (EPOD) *63*
encryption *60*
fishbone diagram *73*
intangible benefits *68*
Ishikawa diagram *73*
just-in-time (JIT) *63*
mission statement *55*
nondiscretionary projects *70*
operational feasibility *67*
Pareto chart *78*
preliminary investigation *71*
project creep *74*
project scope *74*
scatter diagram, *78*
schedule feasibility *69*
strategic planning *54*
SWOT analysis *54*
systems request *59*
systems review committee *65*
tangible benefits *68*
technical feasibility *67*
total cost of ownership (TCO) *68*
XY chart *78*

Learn It Online

Instructions: To complete the Learn It Online exercises, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to the resources for this chapter, and click the link for the exercise you want to complete.

1 Chapter Reinforcement

TF, MC, and SA

Click the Chapter Reinforcement link. Print the quiz by clicking Print on the File menu for each page. Answer each question.

2 Flash Cards

Click the Flash Cards link and read the instructions. Type 20 (or a number specified by your instructor) in the Number of playing cards text box, type your name in the Enter your Name text box, and then click the Flip Card button. When the flash card is displayed, read the question and then click the ANSWER box arrow to select an answer. Flip through the Flash Cards. If your score is 15 (75%) correct or greater, click Print on the File menu to print your results. If your score is less than 15 (75%) correct, then redo this exercise by clicking the Replay button.

3 Practice Test

Click the Practice Test link. Answer each question, enter your first and last name at the bottom of the page, and then click the Grade Test button. When the graded practice test is displayed on your screen, click Print on the File menu to print a hard copy. Continue to take practice tests until you score 80% or better.

4 Who Wants To Be a Computer Genius?

Click the Computer Genius link. Read the instructions, enter your first and last name at the bottom of the page, and then click the Play button. When your score is displayed, click the PRINT RESULTS link to print a hard copy.

5 Wheel of Terms

Click the Wheel of Terms link. Read the instructions, and then enter your first and last name and your school name. Click the PLAY button. When your score is displayed on the screen, right-click the score and then click Print on the shortcut menu to print a hard copy.

6 Crossword Puzzle Challenge

Click the Crossword Puzzle Challenge link. Read the instructions, and then enter your first and last name. Click the SUBMIT button. Work the crossword puzzle. When you are finished, click the Submit button. When the crossword puzzle is displayed, click the Print Puzzle button to print a hard copy.

SCR Associates Case Simulation Session 2: Analyzing the Business Case

Overview

The SCR Associates case study is a Web-based simulation that allows you to practice your skills in a real-world environment. The case study transports you to SCR's intranet, where you complete 12 work sessions, each aligning with a chapter. As you work on the case, you will receive e-mail and voice mail messages, obtain information from SCR's online libraries, and perform various tasks.

How do I use the case?

- Review the SCR background material in Chapter 1.
- Read the Preview for this session and study the Task List
- Visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to the **SCR Case Simulation**, and locate the intranet link.
- Enter your name and the password **sad9e**. An opening screen will display the 12 sessions.
- Select this session. Check your e-mail and voice mail carefully, and then work on the tasks.

Preview: Session 2

During your orientation, you found your way around the office and had a chance to explore the SCR Internet site. Now, after a week on the job, your supervisor, Jesse Baker, has explained the new TIMS system and asked you to lead the systems development effort. She suggested that you review SCR's mission statement, think about a systems review committee, draft a project scope statement, and prepare to interview people to learn more about the new system.

Task List

1. *We need a corporate goal for SCR that refers to our new training activity. Prepare a draft to show Jesse.*
2. *Jesse wants my opinion on whether or not SCR needs a system review committee. Need to prepare a recommendation and reasons.*
3. *Draft a project scope statement for the TIMS system and describe the constraints. She said be specific.*
4. *Need to identify the people I want to interview to learn more about the new training activity, and prepare a list of the questions I will ask.*

FIGURE 2-26 Task list: Session 2.

Chapter Exercises

Review Questions

1. What is a business case? How does a business case affect an IT project?
2. What is a SWOT analysis and why is it important?
3. What are five common reasons for systems projects?
4. What are some internal and external factors that affect systems projects?
5. What are some advantages and disadvantages of a systems review committee?
6. What is feasibility? List and briefly discuss four feasibility tests.
7. How do tangible benefits differ from intangible benefits?
8. What are the steps in a preliminary investigation?
9. What is project scope? What is a constraint? In what three ways are constraints classified?
10. Explain how you might use fishbone diagrams, Pareto charts, and XY charts in problem-solving. Be specific, and describe the advantages of each tool.

Discussion Topics

1. Directives from top management often trigger IT projects. Suppose that the vice president of marketing tells you to write a program to create mailing labels for a one-time advertising promotion. As the IT manager, you know that the labels can be prepared more efficiently by simply exporting the data to a word processing program with a mail merge feature. How would you handle this situation?
2. The vice president of accounting says to you, the IT director, "This systems development life cycle stuff takes too long." She tells you that her people know what they are doing and that all systems requests coming from her department are necessary and important to the organization. She suggests that the IT department bypass the initial steps for any accounting department request and immediately get to work at the solution. What would you say to her?
3. One of your coworkers says, "Mission statements are nice, but they really don't change things down here where the work gets done." How would you reply?
4. Would you continue to work for a company if you disagreed with the firm's mission statement? Why or why not?

Projects

1. Use the Internet to find an example of a corporate mission statement.
2. Many articles have been written on how to develop, understand, and evaluate a business case. Visit the Web sites for TechRepublic, CIO, or another IT magazine, and find one or more articles that might be of interest to your class. For more information, you can visit the Resources Library at the online SCR Associates case, which lists more than a dozen IT news sources. To view these sources, go to the SCR Case Simulation at the MIS CourseMate Web site at **www.cengagebrain.com**, log on to the SCR intranet, and navigate to the library. When your research is done, write a brief summary of what you learned.
3. Suppose you own a travel agency in a large city. You have many corporate clients, but growth has slowed somewhat. Some long-term employees are getting discouraged, but you feel that there might be a way to make technology work in your favor. Use your imagination and suggest at least one strength, weakness, opportunity, and threat that your business faces.
4. Write a mission statement and at least three goals for the travel agency described in Project 3.

Apply Your Knowledge

The section contains four mini-cases. Each case describes a situation, explains your role in the case, and asks you to respond to questions. You can answer the questions by applying knowledge you learned in the chapter.

1 Last Chance Securities

Situation:

The IT director opened the department staff meeting today by saying "I've got some good news and some bad news. The good news is that management approved the payroll system project this morning. The new system will reduce clerical time and errors, improve morale in the payroll department, and avoid possible fines and penalties for noncompliance. The bad news is that the system must be installed by the end of December in order to meet new federal reporting rules, costs must be within the budgeted amount, the new system must interact with existing systems, and the vice president of finance insists on approving the final design."

1. Name the constraints and indicate whether each is present, future, internal, external, mandatory, or desirable.
2. Explain why it is important to define the payroll project's scope. Explain how to define project scope.
3. Identify tangible and intangible benefits of the new payroll system.
4. What topics should be included in a report to management at the end of the preliminary investigation?

2 Way Out Bikes

Situation:

The owner of Way Out Bikes asked you for advice about acquiring an information system for her business. The company specializes in helping customers select exactly the right bicycle for their needs and lifestyles. Way Out cannot compete on price with mass merchandisers, but it seeks to offer value and expertise for which customers are willing to pay. You ask the owner whether she has long-range plans for the company, and she replies that she has not really thought beyond a one-year time frame.

1. Explain the concept of strategic planning to Way Out's owner.
2. Decide what else you might want to know about Way Out. Consider the internal and external factors described on pages 59 to 61, and make a list of questions to ask the owner.
3. Draft a mission statement for Way Out.
4. Make a list of Way Out's stakeholders.

3 The Monday IT Department Staff Meeting

Situation:

Your boss, the IT manager, was ready to explode. "Why can't we get our priorities straight?" he fumed. "Here we go again, working on a low-value project, just because it's a favorite of the marketing group. I wish we could get away from departmental politics! I want you to draft a memo that proposes a systems review committee for this company. Explain the advantages, but don't step on anyone's toes!"

1. Write a draft of the proposal, as your boss requested.
2. Write a memo to your boss explaining potential disadvantages of the committee approach.
3. Draft a set of ground rules for committee meetings. Try to suggest rules that will minimize political differences and focus on the overall benefit to the company.
4. Most people serve on a committee at some point in their lives. Write a brief memo describing your committee experiences, good or bad.

4 The Friday IT Department Staff Meeting

Situation:

By the end of the week, things quieted down. The IT staff discussed how to prioritize IT project requests, taking into account technical, operational, economic, and schedule feasibility. The IT manager asked for suggestions from the group.

1. Provide three examples of why a project might lack technical feasibility.
2. Provide three examples of why a project might lack operational feasibility.
3. Provide three examples of why a project might lack economic feasibility.
4. Provide three examples of why a project might lack schedule feasibility.

Case Studies

Case studies allow you to practice specific skills learned in the chapter. Each chapter contains several case studies that continue throughout the textbook, and a chapter capstone case.

NEW CENTURY HEALTH CLINIC

New Century Health Clinic offers preventive medicine and traditional medical care. In your role as an IT consultant, you will help New Century develop a new information system.

Background

New Century Health Clinic's office manager, Anita Davenport, recently asked permission to hire an additional office clerk because she feels the current staff can no longer handle the growing workload. The associates discussed Anita's request during a recent meeting. They were not surprised that the office staff was feeling overwhelmed by the constantly growing workload.

Because the clinic was busier and more profitable than ever, they all agreed that New Century could afford to hire another office worker. Dr. Jones then came up with another idea. He suggested that they investigate the possibility of computerizing New Century's office systems. Dr. Jones said that a computerized system could keep track of patients, appointments, charges, and insurance claim processing and reduce paperwork. All the associates were enthusiastic about the possibilities and voted to follow up on the suggestion. Dr. Jones agreed to direct the project.

Because no member of the staff had computer experience, Dr. Jones decided to hire a consultant to study the current office systems and recommend a course of action. Several friends recommended you as a person who has considerable experience with computerized business applications.

Assignments

1. Dr. Jones has arranged an introductory meeting between the associates of New Century Health Clinic and you to determine if mutual interest exists in pursuing the project. What should the associates try to learn about you? What should you try to learn in this meeting?
2. Does the proposed system present a strong business case? Why or why not?
3. For each type of feasibility, prepare at least two questions that will help you reach a feasibility determination.
4. You begin the preliminary investigation. What information is needed? From whom will you obtain it? What techniques will you use in your fact-finding?

PERSONAL TRAINER, INC.

Personal Trainer, Inc., owns and operates fitness centers in a dozen midwestern cities. The centers have done well, and the company is planning an international expansion by opening a new "supercenter" in the Toronto area. Personal Trainer's president, Cassia Umi, hired an IT consultant, Susan Park, to help develop an information system for the new facility. During the project, Susan will work closely with Gray Lewis, who will manage the new operation.

Background

At their initial meeting, Susan and Gray discussed some initial steps in planning a new information system for the new facility. The next morning, they worked together on a business profile, drew an organization chart, discussed feasibility issues, and talked about various types of information systems that would provide the best support for the supercenter's operations. Their main objective was to carry out a preliminary investigation of the new system and report their recommendations to Personal Trainer's top managers.

After the working session with Gray, Susan returned to her office and reviewed her notes. She knew that Personal Trainer's president, Cassia Umi, wanted the supercenter to become a model for the company's future growth, but she did not remember any mention of an overall strategic plan for the company. Susan also wondered whether the firm had done a SWOT analysis or analyzed the internal and external factors that might affect an information system for the supercenter.

Because the new operation would be so important to the company, Susan believed that Personal Trainer should consider an enterprise resource planning strategy that could provide a company-wide framework for information management. After she finished compiling her notes, Susan listed several topics that might need more study and called Gray to arrange another meeting the following day.

Assignments

1. Based on the background facts described in Chapter 1, draft a mission statement for Personal Trainer. Consider the firm's overall direction, and the services, products, and experiences the company might want to offer its customers in the future. In your statement, consider all the stakeholders affected by Personal Trainer's operations.
2. Susan and Gray probably will need more information about the proposed system. Make a list of people whom they might want to interview. Also, suggest other fact-finding techniques they should consider.
3. Consider the internal and external factors that affect information systems. Which factors, in your opinion, will have the greatest impact on the system proposed for the new supercenter? Explain your answer.
4. At the conclusion of the preliminary investigation, Susan and Gray will deliver a written summary of the results and deliver a brief presentation to Personal Trainer's management team. Prepare a list of recommendations that will help make their written and oral communications more effective. Put your list in priority order, starting with what you consider to be the most important suggestions. Before you complete this task, you should review Part A of the Systems Analyst's Toolkit, which provides suggestions for oral and written presentations.

ORIGINAL KAYAK ADVENTURES

Original Kayak Adventures (OKA) offers guided eco-tours and kayak rentals along the Hudson River.

Background

In Chapter 1, you learned that John and Edie Caputo founded OKA two years ago. Now John and Edie are thinking about replacing their current system, which is a mix of manual and computer-based techniques, with a new information system that would meet their current and future needs. Before you answer the following questions, you should review the fact statement in Chapter 1.

Assignments

1. Does a strong business case exist for developing an information system to support the Caputos' business? Explain your answer.
2. In a small- to medium-sized business, such as OKA, is it really important to use a structured approach for information systems development? Why or why not?
3. Based on the facts provided, draft a mission statement for OKA. In your statement, consider all the stakeholders who might be affected by OKA operations.
4. What internal and external factors might affect OKA's business success?

TOWN OF EDEN BAY

The town of Eden Bay owns and maintains a fleet of vehicles. You are a systems analyst reporting to Dawn, the town's IT manager.

Background

Eden Bay is a medium-sized municipality. The town has grown rapidly, and so has the demand for town services. Eden Bay currently owns 90 vehicles, which the town's equipment department maintains. The fleet includes police cars, sanitation trucks, fire trucks, and other vehicles assigned to town employees. The maintenance budget has risen sharply in recent years, and people are asking whether the town should continue to perform its own maintenance or outsource it to private firms.

This morning, Dawn called you into her office to discuss the situation. A summary of her comments follows.

Dawn (IT manager): When I came here two years ago, I was told that Eden Bay had a computerized information system for vehicle maintenance. What I found was a spreadsheet application designed by a part-time employee as a quick answer to a much more complex problem. It's probably better than no system at all, but I can't justify spending any time on it. The system should never have been designed as a spreadsheet in the first place.

I've discussed the situation with the equipment department people. Rather than tinker with the current system, I think we should press for a new information system project, and I've developed an initial proposal. I've code-named the new system RAVE, which stands for Repair Analysis for Vehicular Equipment. I know that commercial fleet maintenance packages exist, but they are very expensive.

I did some fact-finding, and I want you to start by reading the interview summaries I prepared.

Before You Begin …

Review the following interview summaries from Marie (town manager), Martin (equipment department manager), Phil (maintenance supervisor), Alice (maintenance clerk), and Joe (mechanic).

Marie (town manager): Maintenance costs have risen 14 to 16% annually. I'm not sure that we have any real control over these costs. Some members of the town council think we should get out of the maintenance business and contract it out to a private firm. That might mean laying off current employees, and I'm not sure whether outsourcing is the right way to go.

Both the equipment department manager and the IT manager tell me that our current record-keeping system is outdated, and I wonder if a new information system would give us a better handle on the problem. My own view is that if there's a way we can become more efficient, we should continue to perform our own maintenance.

Dawn, our IT manager, tells me that she has developed a proposal for a maintenance information system. I plan to bring it up at the next council meeting.

Martin (equipment department manager): I hear a lot of criticism about the maintenance budget, but I'm doing the best I can. We operate from one budget year to the next, without a long-term plan. I belong to a professional association of fleet maintenance managers, and I know that we should be developing a strategic plan instead of juggling annual budget figures.

I'd like to build this department into a first-class organization. Our people are great, but they could use more technical training. Our shop and equipment are generally adequate for what we do, but we haven't kept up with some of the newer diagnostic equipment. We have a real problem in record keeping. Instead of a short-term solution, Eden Bay should have developed a maintenance information system years ago. Prior to taking this position, I was assistant maintenance manager in a medium-sized city, and they had developed a system that handled scheduling and cost analysis, in addition to day-to-day maintenance operations.

Phil (maintenance supervisor): I'm in the middle — I get pressure from above to cut costs, and I get complaints from below that management doesn't know what it's doing. One thing for sure — short-term solutions are not the answer. I hope they don't ask me to cut back on preventive maintenance. The last time we did that, we extended routine oil changes and servicing, and we ended up with even more repairs than we had previously.

My mechanics are capable people, and they're doing the best they can. One problem I see is that it's hard to pull up a history for a particular vehicle. We keep the data on a computer, but different people used different codes and procedures over the years, and the system probably needs a good overhaul.

Alice (maintenance clerk): I'm in charge of maintenance record keeping. We use a spreadsheet system that was designed by a part-time employee who is no longer around. Because we work on a monthly budget, the spreadsheet has a separate page for each month. When the year is over, we start a new set of monthly pages. The spreadsheet is supposed to record labor and parts used, and assign the cost to a specific vehicle, but it doesn't always work out that way.

I also use a notebook to keep track of vehicle mileage and scheduled service intervals, so I can let the department heads know when a vehicle needs to come in for service. I write up work orders for scheduled service or necessary repairs, but often a mechanic finds other problems and has to write up an additional charges form.

Each time a vehicle comes into the shop, I start a new row on the spreadsheet. I enter the vehicle number, mileage, and date. Then I enter the rest of the data into the columns for parts, labor hours, job code, shop supplies, and miscellaneous charges. At the end of the month, I calculate total costs from the spreadsheet, and we compare these with actual payroll and parts vouchers for the month. If the totals are close, everyone is happy. If not, we try to figure out what work didn't get reported and entered into the spreadsheet.

The labor codes also are a problem. Specific codes are assigned for certain types of shop labor, but these were changed three years ago when the new Director arrived. Also, about half the labor can be coded, but the rest has to be entered manually — and there are no standards. Two mechanics might do the same job, and one records four specific tasks, while the other calls it a tune-up.

I know the mechanics don't like paperwork, but what can I do? I asked the IT manager if she could do anything to help, but she says that it isn't worthwhile to update the current system. She says she has heard some talk about developing a new information system specifically designed for vehicle fleet maintenance. It can't be soon enough for me.

Joe (mechanic): I love my job, but I hate the paperwork. We get a work order from the clerk for all scheduled maintenance, but if we find other problems, we have to handwrite an additional work ticket. Personally, I think some of these vehicles should be retired before they get too expensive to maintain.

I would hate to see the town contract out the maintenance. I've put in 17 years here, and I don't want to lose my job, but I know that some specialized repairs would be less expensive on the outside. Most of the mechanics realize this, but let management figure it out — they're the ones with the fancy computer system.

Assignments

1. Upon investigation, you learn that the town does not have a strategic plan or a mission statement. In your view, does this affect the current situation? Why or why not?
2. Based on the fact statements provided, summarize the maintenance department's most important strengths, weaknesses, opportunities, and threats.
3. Describe the specific steps you will follow during a preliminary investigation, including any fact-finding techniques you will use. Be sure to include the tools mentioned in this chapter.
4. Of the four tests of feasibility — operational, technical, economic, and schedule — which would you perform first to measure the system project's feasibility? Why?

CHAPTER CAPSTONE CASE: SoftWear, Limited

SoftWear, Limited (SWL), is a continuing case study that illustrates the knowledge and skills described in each chapter. In this case study, the student acts as a member of the SWL systems development team and performs various tasks.

Background

SWL outsources the company's payroll processing to an outside firm called Business Information Systems (BIS). SWL's payroll department submits data to BIS, which uses its own hardware and software to produce employee paychecks and generate payroll reports. BIS performs payroll processing for dozens of companies. Contractual agreements between BIS and its customers identify specific information processing services and prices.

SWL's information technology department is located at the company headquarters in Raleigh and reports to the vice president of finance. The IT staff is responsible for SWL's mainframe computer and supports the company's Web site and the inventory, marketing, customer order entry, and accounting systems.

Robert Lansing, SWL's president, believes that IT support is vital to the company's strategic long-range plans and has approved increased IT budgets and expansion of the IT staff. In addition to the mainframe, the company networked personal computers in all offices and many shop floor locations and implemented a company intranet linking all SWL locations.

Even though it could handle its own payroll processing, SWL continues to use BIS for payroll services because BIS does a good job at a reasonable cost, and it relieves SWL of this responsibility. Recently, problems with the payroll system developed, and SWL's payroll department employees had to work overtime to correct errors involving employee deductions.

SWL employees can make two types of voluntary payroll deductions. Starting in 2007, employees could contribute to the newly formed SWL credit union. To enroll or make changes, an employee must complete a deduction form. In 2009, the company gave employees an opportunity to purchase SWL company stock through payroll deductions. Employees enroll in the stock purchase plan or change their deductions by visiting the human resources department, which then sends a weekly list of transactions to SWL's payroll department.

In addition to the credit union and stock purchase deductions, SWL employees soon may have other savings and investment choices. SWL's top management, with strong support from the vice president of human resources, may consider a new Employee Savings and Investment Plan (ESIP) that allows employees to purchase mutual funds, stocks, and other investments through regular payroll deductions. Under this new 401(k) plan, an outside investment firm, Court Street Securities, manages tax-sheltered deductions and services the individual accounts. Each employee maintains direct control over his or her investments using a 24-hour toll-free number or accessing the Court Street Securities Web site. Management expects to make a final decision about the new ESIP in several months.

Request for Information Technology Services

Rob King, vice president of human resources, learned that a number of SWL employees had complained about improper paycheck deductions, and he became concerned about employee morale. He decided to discuss the subject with Michael Jeremy, vice president of finance. After the meeting, Mr. Jeremy met with Amy Calico, director of payroll, to ask her about the problem — and a recent increase in overtime pay in her group. Amy stated that the overtime became necessary because payroll operations recently required more time and effort. She also noted that, because this workload increase came about recently, she lacked the money in her budget to hire any additional people. She did not provide any specific explanation for the payroll deduction errors.

Chapter Capstone Case: SoftWear, Limited (continued)

SWL

Mr. Jeremy then decided to ask the IT department to investigate the payroll system. He prepared a systems request, as shown in Figure 2-27, and sent it to the IT department for action. In the request, he mentioned problems with the payroll system and requested help but did not identify the causes of the problems or propose a solution.

Jane Rossman, manager of applications, normally receives systems requests and does an initial review. After a quick look at Mr. Jeremy's request, Jane decided to contact her boss, Ann Hon, director of information technology. After discussing the proposal, Jane and Ann decided that a preliminary investigation should start right away. Given that the system was eight years old and had never received a major update, it seemed likely that they would find some problems. Jane assigned Rick Williams, a systems analyst, to conduct the preliminary investigation.

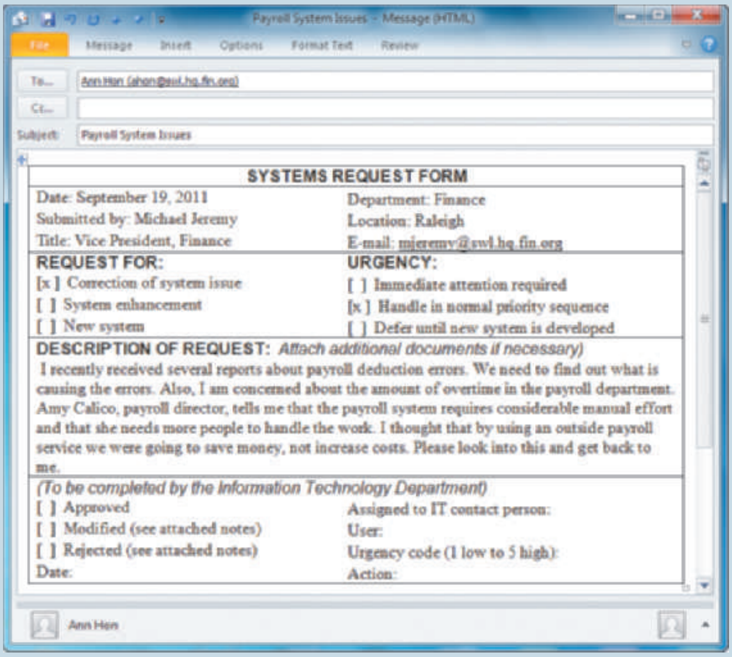


FIGURE 2-27 Michael Jeremy's systems request.

Payroll Department Organization

Rick's first meeting was with Rob King, vice president of human resources. He gave Rick copies of job descriptions for all payroll department positions but did not have a current organization chart for that group.

After reviewing the descriptions, Rick visited Amy Calico, director of payroll. She explained how the payroll department was organized. She explained that two people report directly to her: Nelson White, payroll manager, and Nancy Farmer, administrative assistant. Two payroll technicians, Britton Ellis and Debra Williams, report to Nelson White.

Interviews

Rick next decided to interview Michael Jeremy, Amy Calico, and Mike Feiner, director of human resources.

Mr. Jeremy provided an overview of the recent problems within the payroll system, including the costs of the current system. He had no specific data, but he thought that the majority of the errors involved stock purchases rather than credit union deductions.

Later that day, in his meeting with Mike Feiner, Rick found out more about the reported deduction errors. He learned that stock purchase enrollments and changes are handled differently from credit union deductions. For legal reasons, Mike explained, employees must complete a special form for stock purchase plan transactions. When enrolling or making changes, an employee visits the human resources department for a brochure and an information package called a prospectus, which also includes the form required to enroll. At the end of each week, the human resources department prepares a summary of deduction requests and sends it to the payroll department. Payroll clerks then file the changes with the employee's master record.

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

The next morning, Rick again met with Amy Calico. In the interview, Amy told Rick that some problems with deductions existed, but she did not feel that the payroll clerks were at fault. She suggested that he look elsewhere for the source of the problem. Amy stated that the payroll process generally works well, although it requires a substantial amount of manual effort. She said that if she could hire two additional clerks, it would resolve any remaining problems. During the course of the meeting, Rick began to feel that Amy's opinion might be somewhat biased. As payroll director, she might not want to call attention to problems in her department, and Rick guessed, that some other issues might be involved. He decided to keep this possibility in mind as the investigation continued.

Current Documentation

After completing the three interviews, Rick reviewed his notes and decided to find out more about the actual sequence of operations in the current system. He studied the documentation and found that it provided step-by-step procedures for preparing the payroll. When he asked the payroll clerks about those procedures, he learned that some sections were outdated. The actual sequence of events is shown in Figure 2-28.

Step 1: A new SWL employee completes an employee master sheet and a W-4 form. The human resources department then enters the employee's status and pay rate. Copies of these forms are sent to the payroll department. The payroll department updates the employee master sheet whenever changes are received from the employee or the human resources department. Updates are made with various forms, including forms for credit union and employee stock purchase plan enrollment and changes.

Step 2: On the last day of a weekly pay period, the payroll department prepares and distributes time sheets to all SWL departments. The time sheets list each employee, with codes for various status items such as regular pay, overtime, sick leave, vacation, jury duty, and personal leave.

Step 3: Department heads complete the time sheets on the first business day after the end of a pay period. The sheets then go to the payroll department, where they are reviewed. A payroll clerk enters pay rates and deduction information and forwards the time sheets to the BIS service bureau.

Step 4: BIS enters and processes the time sheet data, prints SWL paychecks, and prepares a payroll register.

Step 5: The checks, time sheets, and payroll register are returned to SWL. The payroll department distributes checks to each department, creates reports for credit union and stock purchase plan deductions, and then transfers necessary funds.

FIGURE 2-28 Sequence of events in payroll processing at SoftWear, Limited.

Rick also discovered that the payroll department never sees a copy of the form that an employee fills out in the human resources department when joining the stock purchase plan or changing deductions. Rick obtained a copy of the SWL stock purchase form from the human resources department and copies of several forms from the payroll department — including employee master sheets, employee time sheets, and credit union deduction forms. Rick put them in a file for later review.

During the preliminary investigation, Rick did not show concern with the detailed information on each form. He would review that information only after management authorized the IT department to continue with the systems analysis phase.

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

Presentation to Management

After Rick finished his investigation, he analyzed his findings, prepared a preliminary investigation report, and met with Jane and Ann to plan the presentation to management. Ann sent an advance copy of the report to Mr. Jeremy with an e-mail that announced the time and location of the presentation.

Figure 2-29 shows the preliminary investigation report. Following the presentation to SWL's top managers and department heads, a question-and-answer session took place. The management group discussed the findings and recommendations and decided that the payroll system needed further analysis. The group also wanted to know if the BIS service bureau could handle the ESIP using their current arrangement. Ann replied that no clear answer could be given, and everyone agreed that the project scope should be broadened to include that question.

Preliminary Investigation Report: SWL Payroll System **October 10, 2011**

Introduction
The IT department completed a preliminary investigation of the payroll system on October 8. This investigation was the result of a systems request submitted by Michael Jeremy, vice president, finance, on September 17.

Systems Request Summary
Two problems were mentioned in the request: incorrect deductions from employee paychecks, and excessive payroll department overtime to perform manual processing tasks and make corrections.

Preliminary Investigation Findings

1. The human resources department sends a summary of employee stock purchase deductions to the payroll department. It is likely that data errors occur during this process. Although the errors are corrected, we believe that incorrect payroll information adversely affects employee morale.

2. The payroll processing arrangement with Business Information Systems (BIS) requires considerable manual effort. BIS does not provide summary reports that SWL needs to verify and apply credit union and stock purchase deductions. Currently, the payroll department handles these tasks manually at the end of each pay period.

3. Payroll department overtime averages about eight hours per week, plus an additional eight hours at the end of the month, when stock purchase deductions are applied. Total annual overtime is about 512 hours. The average hourly base rate for payroll staff is $16.00, with an overtime rate of $24.00 per hour. The additional expense is about $12,288 per year.

4. SWL developed its current payroll procedures 10 years ago, when the company had only 75 employees. At that time, the only payroll deductions were legally required tax items. Today, the payroll system handles over 450 people and many deduction options that must be verified and applied manually.

Recommendations
The current problems will intensify as SWL continues to grow. At this point, it is unclear whether the current system can be modified to handle tasks that are being done manually. Accordingly, the IT department recommends a full analysis of the current system and possible solutions. The project should focus on two main areas: manual processing at SWL and computer-based payroll processing at BIS.

Time and Cost Estimates
We can perform a study during a two-week period. In addition to the time spent by IT staff, we will conduct about 20 hours of interviews with people outside the IT department. The following is a rough estimate of costs through the systems analysis phase:

Systems analyst	2.0 weeks @ $1,400 per week	$2,800
Other SWL staff	0.5 weeks @ $1,000 per week (average)	500
	Total:	$3,300

If the project continues beyond the systems analysis phase, total cost will depend on what development strategy is followed. If the current system can be modified, we estimate a total project effort of $20,000 to $30,000 over a four-month period. If modification is not feasible, a revised cost estimate will be submitted.

Expected Benefits
A sharp reduction in overtime costs and processing errors will avoid unnecessary expense and improve employee morale. During the systems analysis phase, the IT department will investigate various strategies and solutions to address current problems and strengthen SWL's ability to handle payroll-related IT issues in the future.

FIGURE 2-29 A typical preliminary investigation report includes findings, recommendations, and estimated costs and benefits.

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

SWL Team Tasks

1. You have been assigned to write a formal mission statement for SWL. Start by reviewing SWL's background in Chapter 1, then do Internet research to find mission statements that seem clear, focused, and easy to understand. Pay special attention to Web-based and catalog retail firms to see how they approach the issue.
2. Review the preliminary investigation report to see whether all four feasibility tests were discussed in the report. Write a brief summary of your findings.
3. Review the payroll department organization information on page nn. Using this information, prepare an organization chart for this group. In Word 2010 and Word 2007, click the Insert tab on the Ribbon, then SmartArt, then Hierarchy.
4. Rick asked you to investigate other firms that offer payroll processing services. Perform an Internet search using the term "payroll processing services." Try your search both with and without placing quotes around the phrase and notice what happens. Based on your search results, select an example of a payroll processing firm and write a brief report to Rick. Include the firm's name, Web address, and services offered.

Manage the SWL Project

You have been asked to manage SWL's new information system project. One of your most important activities will be to identify project tasks and determine when they will be performed. Before you begin, you should review the SWL case in this chapter. Then list and analyze the tasks, as follows:

LIST THE TASKS Start by listing and numbering at least 10 tasks that the SWL team needs to perform to fulfill the objectives of this chapter. Your list can include SWL Team Tasks and any other tasks that are described in this chapter. For example, Task 3 might be to Prepare a payroll department organization chart, and Task 6 might be to Review payroll department job descriptions.

ANALYZE THE TASKS Now study the tasks to determine the order in which they should be performed. First identify all concurrent tasks, which are not dependent on other tasks. In the example shown in Figure 2-30, Tasks 1, 2, 3, 4, and 5 are concurrent tasks, and could begin at the same time if resources were available.

Other tasks are called dependent tasks, because they cannot be performed until one or more earlier tasks have been completed. For each dependent task, you must identify specific tasks that need to be completed before this task can begin. For example, you would want an organization chart to help you identify the payroll department positions, so Task 6 cannot begin until Task 3 is completed, as Figure 2-30 shows.

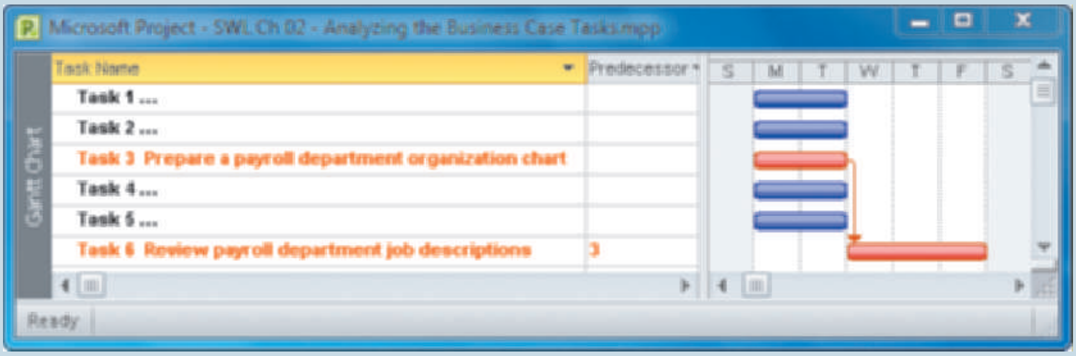


FIGURE 2-30 Tasks 1, 2, 3, 4, and 5 are concurrent tasks that could be performed at the same time. Task 6 is a dependent task that cannot be performed until Task 3 has been completed.

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

Chapter 3 describes project management tools, techniques, and software. To learn more, you can use the Features section on your Student Study Tool CD-ROM, or visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and locate the project management resources library for this book. On the Web, Microsoft offers demo versions, training, and tips for using Project 2010 and 2007. You also can visit the OpenWorkbench.org site to learn more about this free, open-source software.

Ready for a Challenge?

In additional to technical skills, IT professionals need critical thinking skills such as perception, organization, analysis, problem-solving, and decision-making. The Ready for a Challenge feature can help you learn, practice, and apply critical thinking skills that you can take to the workplace.

As an IT intern at Game Technology, you often assist analysts with feasibility studies. The work can include intense fact-finding and tight deadlines. You decide to add a new section to your journal to describe the four different types of feasibility. In your journal, you want to include a definition of each feasibility type, and a sample statement that would indicate a *lack* of feasibility.

For example:

- The hardware has limited capacity for future needs.
- Our users will resist the new system because it is the third change in 18 months.
- The project will take too long to pay for itself.
- Development cannot begin until next year, which is too late.

You also want to learn more about project constraints. You know that constraints can be grouped into various categories: present vs. future, internal vs. external, and mandatory vs. desirable. You plan to use a grid chart like the one in Figure 2-20 on page 75 to show the constraints. To get started, you come up with three sample constraints:

- The new IRS tax rates must go into effect as soon as possible.
- From now on, we should try to hire technicians with A+ certifications.
- Starting next year, government regulations will require a detailed security analysis.

Practice Tasks

A. Define each feasibility type and include an example that shows a lack of feasibility.
B. Create a grid chart that shows the sample constraints. Use Figure 2-20 as a model.

After you complete the Practice Tasks, to check your work and view sample answers, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to the resources for this chapter, and locate Ready for a Challenge?.

The Challenge

It's fortunate that you studied various feasibility types. Felisia Stukes, the IT director, wants you to review the following statements and decide which type of feasibility applies:

- Based on the future cost of support and maintenance, TCO will be very high.
- The network will not be ready until next year, which might be too late.
- Expensive training will be required.
- The current system is well liked and effective, and users see no need for change.
- The hardware is unreliable and will not integrate with other company systems.
- The new system will cause a workforce reduction, and employees are very concerned.
- The platform does not have capacity for future needs, and cannot be expanded.
- The project does not meet the company policy for acceptable return on investment.
- The projected benefits do not outweigh the estimated costs.
- The software will not be available until May, and that will cause an unacceptable delay.

Felesia wants to see a grid chart that will properly show the following constraints:

- Management told all departments to include "green" goals in next year's plan.
- The inventory system would be more effective if we add RFID capability next year.
- Management just announced a change in travel policy: No more first class air travel!
- Effective immediately, our products must meet all government standards.

Challenge Tasks

A. Reply to Felesia and indicate the type of feasibility for each statement.
B. Draw a grid chart that shows the constraints. Use Figure 2–20 as a model.

CHAPTER 3 Managing Systems Projects

Chapter 3 is the final chapter in the systems planning phase of the SDLC. In this chapter, you will learn about project management and how to plan, schedule, monitor, and report on IT projects.

INTRODUCTION

OBJECTIVES

When you finish this chapter, you will be able to:

- Explain project planning, scheduling, monitoring, and reporting
- Describe work breakdown structures, task patterns, and critical path analysis
- Explain techniques for estimating task completion times and costs
- Describe various scheduling tools, including Gantt charts and PERT/CPM charts
- Analyze task dependencies, durations, start dates, and end dates
- Describe project management software and how it can assist you in project planning, estimating, scheduling, monitoring, and reporting
- Discuss the importance of project risk management
- Understand why projects sometimes fail

Chapter 3 explains project management for IT projects. You will learn about project planning, scheduling, monitoring, reporting, and the use of project management software. You will learn how to create a work breakdown structure, identify task patterns, and calculate a critical path. You will also learn how to use Gantt charts and PERT/CPM techniques to schedule and monitor projects. Finally, you will learn how to control and manage project changes as they occur.

In addition to the project management material in this chapter, you can visit the Features section on your Student Study Tool CD-ROM, where you can learn more about Microsoft Project and Open Workbench, an open-source project management program that you can download and install. You can also visit the MIS CourseMate Web site for this book at **www.cengagebrain.com** and explore links in the SWL project management resources library.

Chapter 3 includes three Video Learning Sessions that show you how to create a work breakdown structure (WBS), how to identify task patterns, and how to calculate a project's critical path.

Chapter Introduction Case: Mountain View College Bookstore

Background: Wendy Lee, manager of college services at Mountain View College, wants a new information system that will improve efficiency and customer service at the three college bookstores.

In this part of the case, Tina Allen, systems analyst, and David Conroe, student intern, are talking about project management tools and techniques.

Participants: Tina and David
Location: Mountain View College Cafeteria, Wednesday afternoon, September 21, 2011
Discussion topics: Project planning, Gantt charts, PERT/CPM charts, Microsoft Project and Open Workbench software, project monitoring, and risk management techniques.

Tina: Hi, David. Glad I ran into you. I'd like to talk with you about project management, which we'll be using as we plan and execute the bookstore information system project.

David: *Sure. I've read a little about project management, but I don't know the specifics.*

Tina: Well, we manage business and personal projects every day, but we don't always give it much thought. To manage large-scale IT projects, you need specific tools and techniques. You also need a project manager, who is responsible for planning, leading, organizing, and controlling all the tasks.

David: *I guess that's you?*

Tina: Sure is. No matter which tools you use, the idea is to break the project down into individual tasks, determine the order in which the tasks need to be performed, and figure out how long each task will take. With this information, you can use Gantt charts or PERT/CPM charts to schedule and manage the work.

David: *I've seen Gantt charts — they're the ones that look like horizontal bar charts?*

Tina: Right. In addition to Gantt charts, we'll use PERT/CPM charts, which look like network diagrams that show all the tasks, patterns, and calculations that we'll need. We'll learn how to create PERT/CPM charts manually, and we'll also experiment with Microsoft Project and Open Workbench, which are powerful project management tools.

David: *Anything else we need to know?*

Tina: Yes. After we have a specific plan, we need to monitor it carefully, report the progress, and employ a process called risk management. If you are ready, here's a task list to get us started:

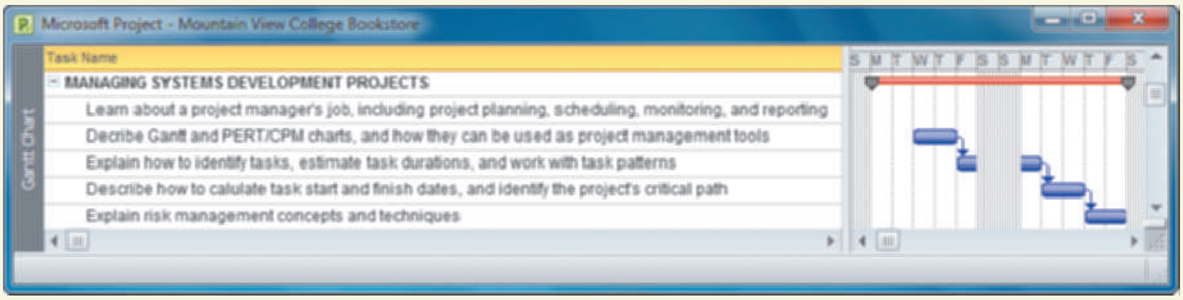


FIGURE 3-1 Typical project management tasks.

FIGURE 3-2 Building construction and systems development projects both need careful management and monitoring.

Overview of Project Management

Whether you are developing an information system or working on a construction project like the one in Figure 3-2, the process is the same. The only difference is the nature of the project. **Project management** for IT professionals includes planning, scheduling, monitoring and controlling, and reporting on information system development.

What Shapes a Project?

A successful project must be completed on time, within budget, and deliver a quality product that satisfies users and meets requirements. Project management techniques can be used throughout the SDLC. System developers can initiate a formal project as early as the preliminary investigation stage, or later on, as analysis, design, and implementation activities occur.

As the sign in Figure 3-3 suggests, sometimes you have to decide what is most important. The same concept applies to systems development, where the factors include budget limits, time constraints, and quality standards. As long as everything is in balance, like the see-saw in Figure 3-4, the project will be successful. However, if one factor changes, adjustments must be made. Because the factors interact constantly, a project manager must respond quickly. For example, if an extremely time-critical project starts to slip, the project manager might have to trim some features, seek approval for a budget increase, simplify the testing plan, or a combination of all three actions.

Unfortunately, many systems projects do fail. A report by The Standish Group noted that only a third of all software development projects were successful, in the sense that they met budget, schedule, and quality targets. Standish chairman Jim Johnson said that improvement will require better project management tools, more iterative methods, and better communication between project developers and users.

FIGURE 3-3 The sign at this imaginary garage tells an old story, which also applies to project management. Sometimes, if you can't have all three, you must choose the two you really need.

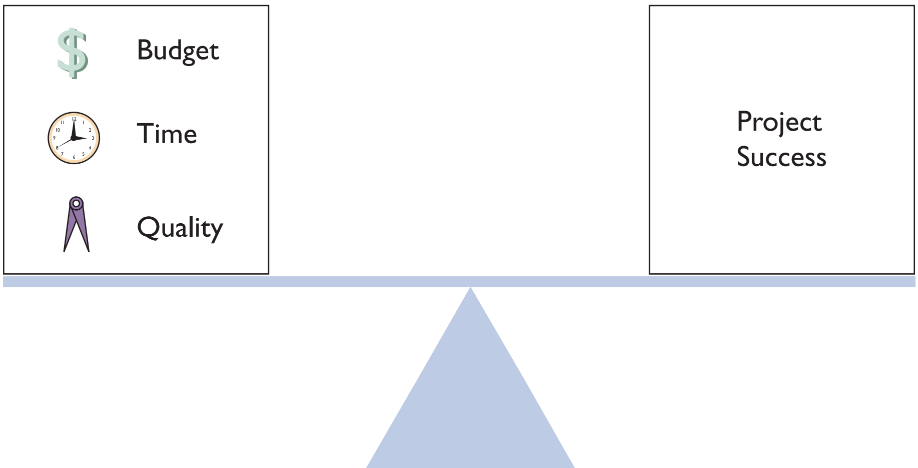


FIGURE 3-4 If one factor changes, adjustments must be made to keep things in balance.

What Does a Project Manager Do?

Whether a project involves a new office building or an information system, good leadership is essential. In a systems project, the **project manager**, or **project leader**, usually is a senior systems analyst or an IT department manager if the project is large. An analyst or a programmer/analyst might manage smaller projects. In addition to the project manager, most large projects have a project coordinator. A **project coordinator** handles administrative responsibilities for the team and negotiates with users who might have conflicting requirements or want changes that would require additional time or expense. Project managers typically perform four activities, or functions: planning, scheduling, monitoring, and reporting.

- **Project planning** includes identifying all project tasks and estimating the completion time and cost of each.
- **Project scheduling** involves the creation of a specific timetable, usually in the form of charts that show tasks, task dependencies, and critical tasks that might delay the project. Scheduling also involves selecting and staffing the project team and assigning specific tasks to team members. Project scheduling uses Gantt charts and PERT/CPM charts, which are explained in the following sections.
- **Project monitoring** requires guiding, supervising, and coordinating the project team's workload. The project manager must monitor the progress, evaluate the results, and take corrective action when necessary to control the project and stay on target.
- **Project reporting** includes regular progress reports to management, users, and the project team itself. Effective reporting requires strong communication skills and a sense of what others want and need to know about the project.

CASE IN POINT 3.1: SPRING FORWARD PRODUCTS

After three years with the company, you recently were asked to manage several IT projects. You are confident that you have the technical skills you need, but you are concerned about morale at the company. There has been some downsizing, and many employees are worried about the future.

As a longtime fan of the Dilbert cartoon strip, you know that maintaining morale can be a real challenge. Your current project involves a team of a dozen people, several of whom remind you of Dilbert and his coworkers. What are some techniques that you might use to motivate the team and inspire its members? What are some things you might *not* want to do?

Project Activities and Planning Steps

On any given day, a project manager might perform one or more of the activities listed above. However, as Figure 3-5 suggests, each activity is part of a larger framework, which includes three key steps in project planning:

- Create a work breakdown structure.
- Identify task patterns.
- Calculate the critical path.

The matrix in Figure 3-5 on the next page shows typical activities that the project leader performs as the project develops. When the project becomes operational, he or she also manages the people, the schedule, the budget, and the progress.

	Planning	Scheduling	Monitoring	Reporting
STEP1: Create a work breakdown structure	✓			
STEP 2: Identify task patterns	✓	✓		
STEP 3: Calculate the critical path	✓	✓		
Manage the operational project			✓	✓

FIGURE 3-5 The matrix is a typical sample of management activities performed while the project is being developed, and when it is launched.

The following sections describe the three project development steps. You can view a Video Learning Session before, during, or after you study each step.

STEP 1: CREATE A WORK BREAKDOWN STRUCTURE

A **work breakdown structure** (**WBS**) involves breaking a project down into a series of smaller tasks. Before creating work breakdown structures, you should understand the two primary chart types: Gantt charts and PERT/CPM charts.

What Is a Gantt Chart?

Gantt charts were developed almost 100 years ago by Henry L. Gantt, a mechanical engineer and management consultant. His goal was to design a chart that could show planned and actual progress on a project. A **Gantt chart** is a horizontal bar chart that represents a set of tasks. For example, the Gantt chart in Figure 3-6 displays five tasks in a vertical array, with time shown on the horizontal axis. The position of the bar shows the planned starting and ending time of each task, and the length of the bar indicates its duration. On the horizontal axis, time can be shown as elapsed time from a fixed starting point, or as actual calendar dates. A Gantt chart also can simplify a complex project by combining several activities into a **task group**. For example, in Figure 3-6, Task 4 might consist of five separate tasks, which are hidden in this view.

VIDEO LEARNING SESSION: WORK BREAKDOWN STRUCTURES

Video Learning Sessions can help you understand key concepts, practice your skills, and check your work. To access the sessions, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and navigate to the **Video Learning Sessions** for this book. This session is about work breakdown structures. You'll learn what a work breakdown is and why it is important, how to create a work breakdown structure, and how to use Microsoft Project to display a work breakdown structure.

ON THE WEB

To learn more about Gantt charts, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Gantt Charts link.

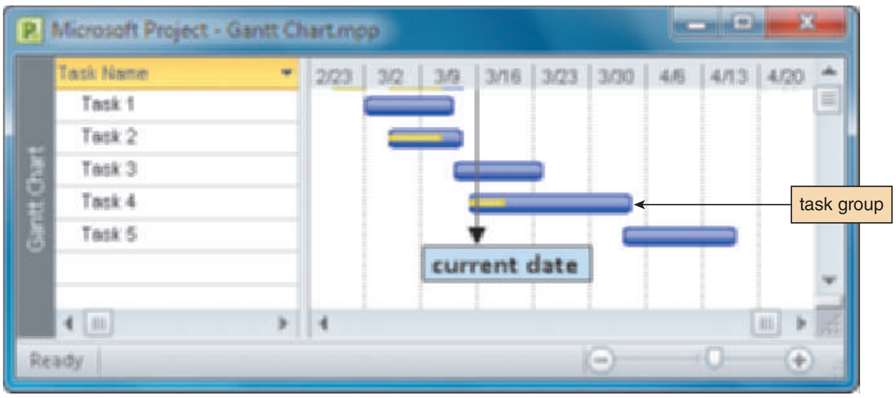


FIGURE 3-6 In this Gantt chart, notice the task group represented by Task 4. Also notice the yellow bars that show the percentage of task completion.

A Gantt chart can show task status by adding a contrasting color to the horizontal bars. For example, a vertical arrow marks the current date in Figure 3-6. With a fixed reference point, it is easy to see that Task 1 is way behind schedule, Task 2 is only about 80 percent done and is running behind schedule, Task 3 should have started, but no work has been done, Task 4 actually is running ahead of schedule, and Task 5 will begin in several weeks.

Gantt charts can present an overview of the project's status, but they do not provide enough detailed information, which is necessary when managing a complex project. Most project managers find that PERT/CPM charts, which are discussed in the following section, are better tools for managing large projects.

ON THE WEB

To learn more about PERT/CPM, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the PERT/CPM link.

What Is a PERT/CPM Chart?

The **Program Evaluation Review Technique** (**PERT**) was developed by the U.S. Navy to manage very complex projects, such as the construction of nuclear submarines. At approximately the same time, the **Critical Path Method** (**CPM**) was developed by private industry to meet similar project management needs. The distinction between the two methods has disappeared over time, and today the technique is called either PERT, CPM, or **PERT/CPM**. The textbook will use the term PERT chart.

PERT is a **bottom-up technique**, because it analyzes a large, complex project as a series of individual tasks. To create a PERT chart, you first identify all the project tasks and estimate how much time each task will take to perform. Next, you must determine the logical order in which the tasks must be performed. For example, some tasks cannot start until other tasks have been completed. In other situations, several tasks can be performed at the same time.

Once you know the tasks, their durations, and the order in which they must be performed, you can calculate the time that it will take to complete the project. You also can identify the specific tasks that will be critical to the project's on-time completion. An example of a PERT chart, which Microsoft calls a **network diagram**, is shown in the lower screen in Figure 3-7.

Which Type of Chart Is Better?

Although a Gantt chart offers a valuable snapshot view of the project, PERT charts are more useful for scheduling, monitoring, and controlling the actual work. With a PERT chart, a project manager can convert task start and finish times to actual dates by laying out the entire project on a calendar. Then, on any given day, the manager can compare what *should* be happening with what *is* taking place, and react accordingly. Also, a PERT chart displays complex task patterns and relationships. This information is valuable to a manager who is trying to address high priority issues. PERT and Gantt charts are not mutually exclusive techniques, and project managers often use both methods.

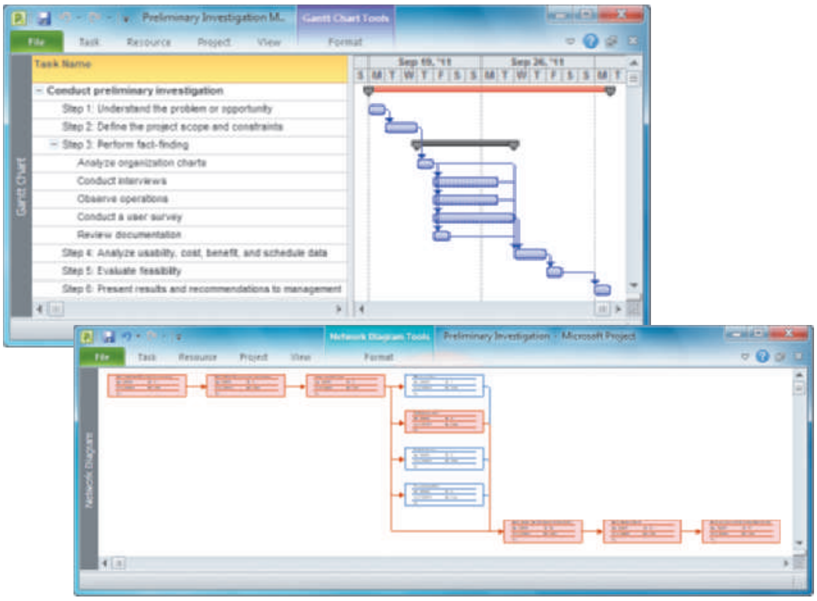


FIGURE 3-7 The top screen is a Gantt chart with 11 tasks, and a PERT chart in the bottom screen shows the same project. Although not visible in this zoomed-out view, the PERT chart boxes provide detailed information about task duration, start dates, and finish dates.

Figure 3-7 shows both chart types. The top screen is a Gantt chart with 11 tasks. The PERT chart in the bottom screen shows the same project, using a separate box for each task instead of a horizontal bar. Although they both show the task patterns and flow, the PERT chart boxes can provide more detailed information, such as task duration, start date, and finish date. You will learn how to create PERT charts in following sections.

Identifying Tasks in a Work Breakdown Structure

A work breakdown structure must clearly identify each task and include an estimated duration. A **task**, or **activity**, is any work that has a beginning and an end and requires the use of company resources such as people, time, or money. Examples of tasks include conducting interviews, designing a report, selecting software, waiting for the delivery of equipment, or training users. Tasks are basic units of work that the project manager plans, schedules, and monitors — so they should be relatively small and manageable.

In addition to tasks, every project has events, or milestones. An **event**, or **milestone**, is a recognizable reference point that you can use to monitor progress. For example, an event might be the start of user training, the conversion of system data, or the completion of interviews. A milestone such as *Complete 50 percent of program testing* would not be useful information unless you could determine exactly when that event will occur.

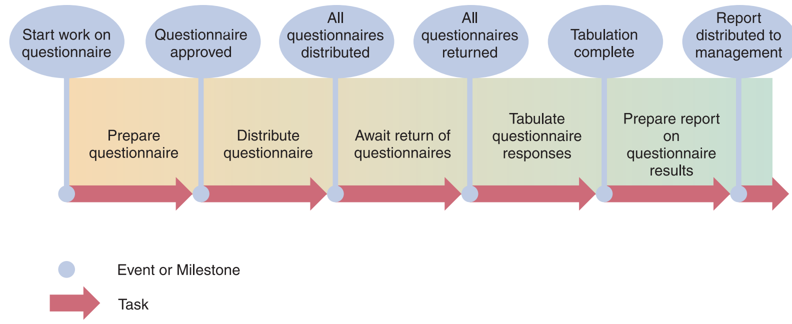


FIGURE 3-8 Using a questionnaire requires a series of tasks and events to track the progress. The illustration shows the relationship between the tasks and the events, or milestones, that mark the beginning and end of each task.

Figure 3-8 shows tasks and events that might be involved in the creation, distribution, and tabulation of a questionnaire. Notice that the beginning and end of each task is marked by a recognizable event. If you tried to manage a project as one large task, it would be impossible. Instead, you break the project down into smaller tasks, creating a work breakdown structure (WBS). The first step in creating a WBS is to list all the tasks.

LISTING THE TASKS While this step sounds simple, it can be challenging, because the tasks might be embedded in a document, such as the one shown in the first version of Figure 3-9. One trick is to start by highlighting the individual tasks, as shown in the second version. Adding bullets makes the tasks stand out more clearly, as shown in the third version. The next step is to number the tasks and create a table, similar to the one shown in Figure 3-10, with columns for task number, description, duration, and predecessor tasks.

First version

First, reserve the meeting room. Then order the marketing materials and brief the managers. After the briefings, send out customer e-mails and burn sample DVDs. When the e-mails are sent and the DVDs are ready, load the new software. When the marketing materials have arrived and the software is ready, do a dress rehearsal.

Second version

First, *reserve the meeting room*. Then *order the marketing materials* and *brief the managers*. After the briefings, *send out customer e-mails* and *burn sample DVDs*. When the e-mails are sent and the DVDs are ready, *load the new software*. When the marketing materials have arrived and the software is ready, *do a dress rehearsal*.

Third version

- First, *reserve the meeting room*.
- Then *order the marketing materials* and *brief the managers*.
- After the briefings, *send out customer e-mails* and *burn sample DVDs*.
- When the e-mails are sent and the DVDs are ready, *load the new software*.
- When the marketing materials have arrived and the software is ready, *do a dress rehearsal*.

FIGURE 3-9 The three versions show how to transform a task statement into a list of specific tasks for a work breakdown structure.

Task No.	Description	Duration (Days)	Predecessor Tasks
1	Reserve the meeting room		
2	Order the marketing materials		
3	Brief the managers		
4	Send out customer e-mails		
5	Burn sample DVDs		
6	Load the new software		
7	Do a dress rehearsal		

FIGURE 3-10 In this table, columns have been added for task number, description, duration, and predecessor tasks.

CASE IN POINT 3.2: PARALLEL SERVICES

The project management team at Parallel Services is having a debate about how to define tasks in the work breakdown structure (WBS). Ann, the project manager, wants to break tasks down into the smallest possible units. For example, she objected to a broad task statement called *Develop a training schedule*. Instead, she suggested three subtasks: (1) *Determine availability of training room*, (2) *Determine availability of attendees*, and (3) *Select specific dates and training times*.

Karen, another project team member, disagrees. She feels that the broader task statement is better, because it allows more flexibility and will produce the same result. Karen says that if you break tasks into pieces that are too small, you risk overmanaging the work and spending more time on monitoring than actually performing the tasks. As a member of the team, would you tend to agree more with Ann or Karen? What are the pros and cons of each approach?

ESTIMATING TASK DURATION Task duration can be hours, days, or weeks — depending on the project. Because the following example uses days, the units of measurement are called person-days. A **person-day** represents the work that one person can complete in one day. This approach, however, can present some problems. For example, if it will take one person 20 days to perform a particular task, it might not be true that two people could complete the same task in 10 days or that 10 people could perform the task in two days. Some tasks can be divided evenly so it is possible to use different combinations of time and people, up to a point. For instance, if it takes two person-days to install the cables for a new local area network, one person might do the task in two days, two people in one day, or four people in half a day. In most systems analysis tasks, however, time and people are not interchangeable. If one analyst needs two hours to interview a user, two analysts also will need two hours to do the same interview.

Project managers often use a weighted formula for estimating the duration of each task. The project manager first makes three time estimates for each task: an optimistic, or **best-case estimate** (B), a **probable-case estimate** (P), and a pessimistic, or **worst-case estimate** (W). The manager then assigns a **weight**, which is an importance value, to each estimate. The weight can vary, but a common approach is to use a ratio of B = 1, P = 4, and W = 1. The expected task duration is calculated as follows:

$$\frac{(B+4P+W)}{6}$$

For example, a project manager might estimate that a file-conversion task could be completed in as few as 20 days or could take as many as 34 days, but most likely will require 24 days. Using the formula, the expected task duration is 25 days, calculated as follows:

$$\frac{(20+(4*24)+34)}{6} = 25$$

Factors Affecting Duration

When developing duration estimates, project managers consider four factors:

- Project size
- Human resources
- Experience with similar projects
- Constraints

PROJECT SIZE You learned in Chapter 1 that information systems have various characteristics that affect their complexity and cost. In addition to considering those factors, a project manager must estimate the time required to complete each project phase. To develop accurate estimates, a project manager must identify all project tasks, from initial fact-finding to system implementation. Regardless of the systems development methodology used, the project manager must determine how much time will be needed to perform each task. In developing an estimate, the project manager must allow time for meetings, project reviews, training, and any other factors that could affect the productivity of the development team.

HUMAN RESOURCES Companies must invest heavily in cutting-edge technology and Web-based systems to remain competitive in a connected world. In many areas, skilled IT professionals are in great demand, and firms must work hard to attract and retain the talent they need. A project manager must assemble and guide a development team that has the skill and experience to handle the project. If necessary, additional systems analysts or programmers must be hired or trained, and this must be accomplished within a specific time frame. After a project gets under way, the project manager must deal with turnover, job vacancies, and escalating salaries in the technology sector — all of which can affect whether the project can be completed on time and within budget.

EXPERIENCE WITH SIMILAR PROJECTS A project manager can develop time and cost estimates based on the resources used for similar, previously developed information systems. The experience method works best for small- or medium-sized projects where the two systems are similar in size, basic content, and operating environment. In large systems with more variables, the estimates are less reliable.

In addition, you might not be able to use experience from projects that were developed in a different environment. For example, when you use a new Web-based database application, you might not have previous experience to measure in this environment. In this situation, you could design a prototype or pilot system to gain technical and cost estimating experience.

CONSTRAINTS You learned in Chapter 2 that constraints are defined during the preliminary investigation. A constraint is a condition, restriction, or requirement that the system must satisfy. For example, a constraint might involve maximums for one or more resources, such as time, dollars, or people. A project manager must define system requirements that can be achieved realistically within the required constraints. In the

CASE IN POINT 3.3: SUNRISE SOFTWARE

A lively discussion is under way at Sunrise Software, where you are a project manager. The main question is whether the person-days concept has limitations. In other words, if a task will require 100 person-days, does it matter whether the work is performed by two people in 50 days, five people in 20 days, 10 people in 10 days, or some other combination that adds up to 100?

Programmers Paula and Ethan seem to think it doesn't matter. On the other hand, Hector, a systems analyst, says it is ridiculous to think that any combination would work. To support his point, he offers this extreme example: Could a task estimated at 100 person-days be accomplished by 100 people in one day?

Is Hector correct? If so, what are the limits in the people versus days equation? Taking the concept a step farther, is there an *optimum* number of people to be assigned to a task? If so, how would that number be determined? You need to offer some guidance at the next project team meeting. What will you say?

absence of constraints, the project manager simply calculates the resources needed. However, if constraints are present, the project manager must adjust other resources or change the scope of the project. This approach is similar to the what-if analysis that is described in Chapter 12.

Displaying the Work Breakdown Structure

After you enter the task durations, the work breakdown structure will look like Figure 3-11. If you are managing a complex project with many tasks, you can use task groups, just as you would in a Gantt chart, to simplify the list. If you are using Microsoft Project, the WBS might resemble Figure 3-12.

Task No.	Description	Duration (Days)	Predecessor Tasks
1	Reserve the meeting room	1	
2	Order the marketing materials	9	
3	Brief the managers	2	
4	Send out customer e-mails	3	
5	Burn sample DVDs	3	
6	Load the new software	2	
7	Do a dress rehearsal	1	

FIGURE 3-11 Task durations have been added, and the WBS is complete except for predecessor task information. The predecessor tasks will determine task patterns and sequence of performance.

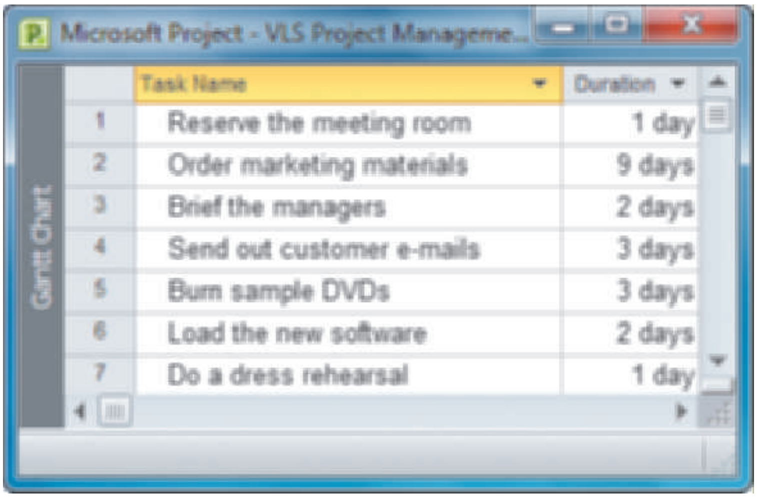


FIGURE 3-12 A Microsoft project screen displays the same WBS, including task number, description, during, and predecessors.

VIDEO LEARNING SESSION: Task Patterns

Video Learning Sessions can help you understand key concepts, practice your skills, and check your work. To access the sessions, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and navigate to the **Video Learning Sessions** for this book. This session is about task patterns. You'll learn what task patterns are and why are they important, how to identify and use task patterns, and how to use Microsoft Project to show task patterns.

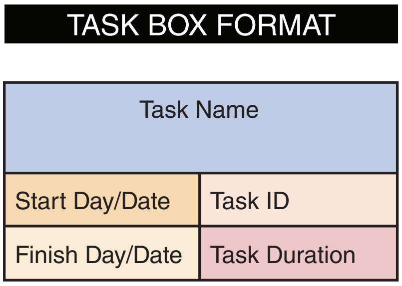


FIGURE 3-13 Each section of the task box contains important information about the task, including the Task Name, Task ID, Task Duration, Start Day/Date, and Finish Day/Date.

STEP 2: IDENTIFY TASK PATTERNS

Tasks in a work breakdown structure must be arranged in a logical sequence called a **task pattern**. This section will show you how to understand and create graphical models of these patterns.

What Are Task Patterns?

In any project, large or small, tasks depend on each other and must be performed in a sequence, not unlike the commands in a software program. Task patterns can involve dependent tasks, multiple successor tasks, and multiple predecessor tasks. In larger projects, these patterns can be very complex, and an analyst must study the logical flow carefully.

How Do I Use Task Boxes to Create a Model?

In a PERT/CPM chart, project tasks are shown as rectangular boxes, arranged in the sequence in which they must be performed. Each rectangular box, called a **task box**, has five sections, as shown in Figure 3-13. Each section of the task box contains important information about the task, including the Task Name, Task ID, Task Duration, Start Day/Date, and Finish Day/Date.

TASK NAME The **task name** should be brief and descriptive, but it does not have to be unique in the project. For example, a task named *Conduct Interviews* might occur in several phases of the project.

TASK ID The **task ID** can be a number or code that provides unique identification.

TASK DURATION The **duration** is the amount of time it will take to complete a task. All tasks must use the same time units, which can be hours, days, weeks, or months, depending on the project. An actual project starts on a specific date, but can also be measured from a point in time, such as *Day 1*.

START DAY/DATE The **start day/date** is the time that a task is scheduled to begin. For example, suppose that a simple project has two tasks: Task 1 and Task 2. Also suppose that Task 2 cannot begin until Task 1 is finished. An analogy might be that you cannot run a program until you turn on your computer. If Task 1 begins on Day 1 and has a duration of three days, it will finish on Day 3. Because Task 2 cannot begin until Task 1 is completed, the start time for Task 2 is Day 4, which is the day *after* Task 1 is finished.

FINISH DAY/DATE The **finish day/date** is the time that a task is scheduled to be completed. To calculate the finish day or date, you add the duration to the start day or date. When you do this, you must be very careful not to add too many days. For example, if a task starts on Day 10 and has a duration of 5 days, then the finish would be on Day 14 — *not* Day 15.

What Are the Main Types of Task Patterns?

A project is based on a pattern of tasks. In a large project the overall pattern would be quite complex, but it can be broken down into three basic patterns: dependent tasks, multiple successor tasks, and multiple predecessor tasks.

DEPENDENT TASKS When tasks must be completed one after another, like the relay race shown in Figure 3-14, they are called **dependent tasks**, because one depends on the other. For example, Figure 3-15 shows that Task 2 depends on Task 1, because Task 2 cannot start until Task 1 is completed. In this example, the finish time of Task 1, Day 5, controls the start date of Task 2, which is Day 6.

MULTIPLE SUCCESSOR TASKS When several tasks can start at the same time, each is called a **concurrent task**. Often, two or more concurrent tasks depend on a single prior task, which is called a **predecessor task**. In this situation, each concurrent task is called a **successor task**. In the example shown in Figure 3-16, successor Tasks 2 and 3 both can begin as soon as Task 1 is finished. Notice that the finish time for Task 1 determines the start time for both Tasks 2 and 3. In other words, the earliest that Task 1 can finish is day 30, so day 31 is the earliest that Tasks 2 and 3 can start.

MULTIPLE PREDECESSOR TASKS Suppose that a task requires two or more prior tasks to be completed before it can start. Figure 3-17 on the next page shows that example, because Task 3 cannot begin until Tasks 1 and 2 are both completed. Since the two tasks might not finish at the same time, the longest (latest) predecessor task becomes the controlling factor. Notice that the start for Task 3 is Day 16, *not* Day 6. Why is this so? Because Task 3 depends on two predecessor tasks, Tasks 1 and 2, Task 3 cannot begin until the *later* of those tasks is complete. Therefore, the start time for a successor task must be the latest (largest) finish time for any of its preceding tasks. In the example shown, Task 1 ends on Day 15, while Task 2 ends on Day 5, so Task 1 controls the start time for Task 3.

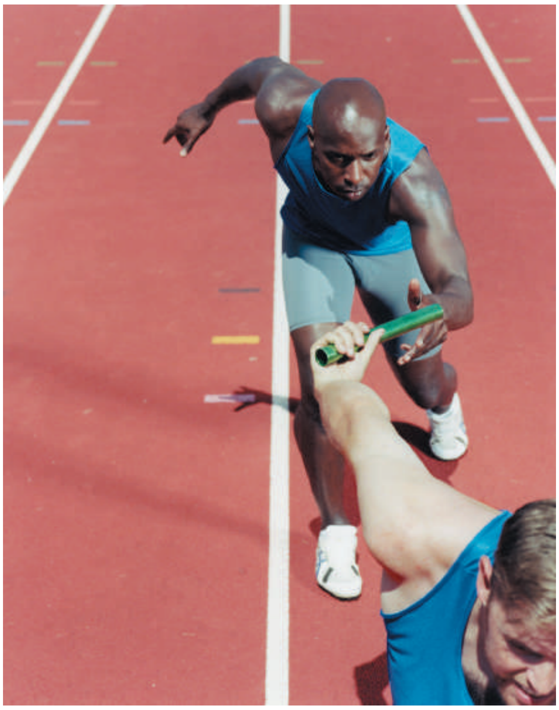


FIGURE 3-14 In a relay race, each runner is dependent on the preceding runner and cannot start until the earlier runner finishes.

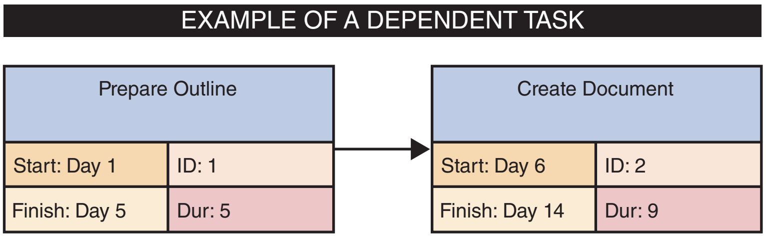


FIGURE 3-15 This example of a dependent task shows that the finish time of Task 1, Day 5, controls the start date of Task 2, which is Day 6.

How Do I Identify Task Patterns?

You can identify task patterns by looking carefully at the wording of the task statement. Words like ***then***, ***when***, or ***and*** are action words that signal a sequence of events. Here are three simple examples:

- *Do Task 1, **then** do Task 2* describes dependent tasks that must be completed one after the other.
- ***When** Task 2 is finished, start two tasks: Task 3 **and** Task 4* describes multiple successor tasks that can both start as soon as Task 2 is finished.
- ***When** Tasks 5 **and** 6 are done, start Task 7* indicates that Task 7 is a multiple predecessor task because it can't start until two or more previous tasks all are completed.

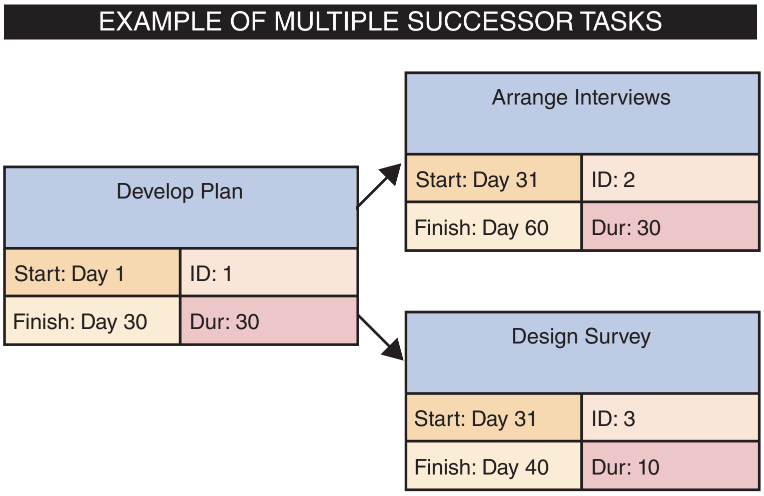


FIGURE 3-16 This example of multiple successor tasks shows that the finish time for Task 1 determines the start time for both Tasks 2 and 3.

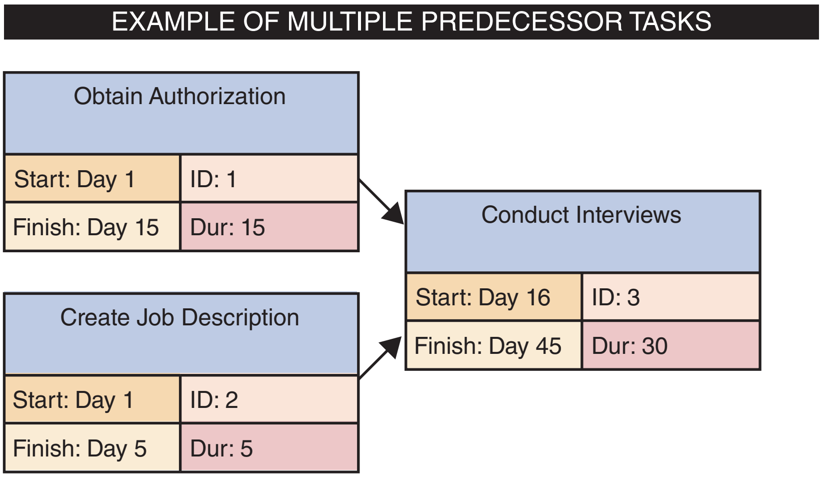


FIGURE 3-17 This example of multiple predecessor tasks shows that the start time for a successor task must be the latest (largest) finish time for any of its preceding tasks. In the example shown,Task 1 ends on Day 15,while Task 2 ends on Day 5, so Task 1 controls the start time for Task 3.

How Do I Work With Complex Task Patterns?

When several task patterns combine, you must study the facts very carefully to understand the logic and sequence. A project schedule will not be accurate if the underlying task pattern is incorrect. For example, consider the following three fact statements and the task patterns they represent. Examples of the task patterns are shown Figures 3-18, 3-19, and 3-20.

DEPENDENT TASKS Perform Task 1. When Task 1 is complete, perform Task 2.

DEPENDENT TASKS AND MULTIPLE SUCCESSOR TASKS Perform Task 1. When Task 1 is complete, perform Task 2. When Task 2 is finished, start two tasks: Task 3 and Task 4. When Task 3 is complete, start two more tasks: Task 5 and Task 6.

DEPENDENT TASKS, MULTIPLE SUCCESSOR TASKS, AND MULTIPLE PREDECESSOR TASKS Perform Task 1. When Task 1 is complete, perform Task 2. When Task 2 is finished, start two Tasks: Task 3 and Task 4. When Task 3 is complete, start two more tasks: Task 5 and Task 6. When Tasks 5 and 6 are done, start Task 7. Then, when Tasks 4 and 7 are finished, perform Task 8.

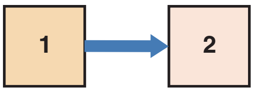


FIGURE 3-18 Dependent tasks.

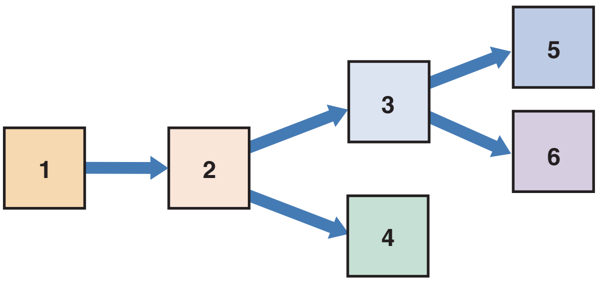


FIGURE 3-19 Dependent tasks and multiple successor tasks.

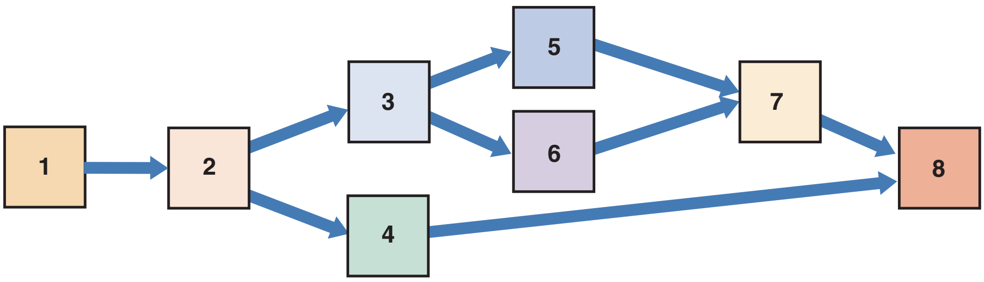


FIGURE 3-20 Dependent tasks, multiple successor tasks, and multiple predecessor tasks.

VIDEO LEARNING SESSION: CRITICAL PATH ANALYSIS

Video Learning Sessions can help you understand key concepts, practice your skills, and check your work. To access the sessions, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and navigate to the **Video Learning Sessions** for this book. This session is about critical path analysis. You'll learn what a critical path is and why it is important, how to calculate task dates and identify the critical path, and how to use Microsoft Project to analyze and display the critical path.

STEP 3: CALCULATE THE CRITICAL PATH

Task patterns determine the order in which the tasks are performed. Once the task sequence has been defined, a project manager can schedule the tasks and calculate the critical path.

What Is a Critical Path?

A **critical path** is a series of tasks which, if delayed, would affect the completion date of the overall project. If any task on the critical path falls behind schedule, the entire project will be delayed. For example, suppose that you invite Joan and Jim to your home for dinner. Joan arrives on time, but Jim arrives 30 minutes late. Jim's arrival is part of the critical path, because you do not want to start without him, so the meal will be served 30 minutes later than originally planned.

Project managers always must be aware of the critical path, so they can respond quickly to keep the project on track. Microsoft Project and other **project management software** can highlight the series of tasks that form the critical path.

How Do I Calculate the Critical Path?

Figure 3-21 shows a training project with five tasks. Notice that the analyst has arranged the tasks and entered task names, IDs, and durations. First, you should review the task patterns. In this example, Task 1 is followed by Task 2, which is a dependent task. Task 2 has two successor tasks: Task 3 and Task 4. Tasks 3 and 4 are predecessor tasks for Task 5.

The next step is to determine start and finish dates, which will determine the critical path for the project. The following explanation will guide you through a step-by-step process. The result is shown in Figure 3-22 on the next page.

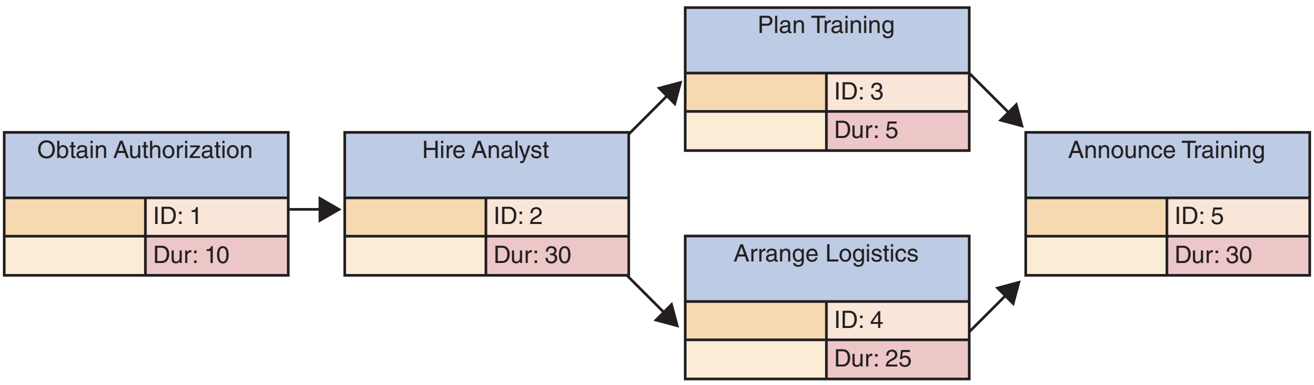


FIGURE 3-21 Example of a PERT/CPM chart with five tasks. Task 2 is a dependent task that has multiple successor tasks. Task 5 has multiple predecessor tasks. In this figure, the analyst has arranged the tasks and entered task names, IDs, and durations.

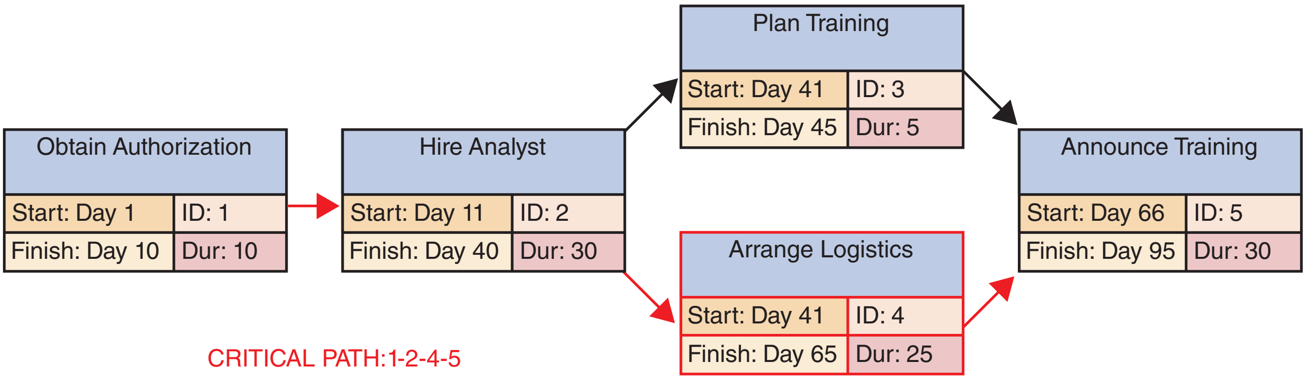


FIGURE 3-22 Now the analyst has entered the start and finish times, using the rules explained in this section. Notice that the overall project has a duration of 95 days.

- Task 1 starts on Day 1 and has a duration of 10 days, so the finish date is Day 10.
- Task 2, which is dependent on Task 1, can start on Day 11 — the day *after* Task 1 ends. With a duration of 30 days, Task 2 will end on Day 40.
- Tasks 3 and 4 are multiple successor tasks that can start after Task 2 is done. Task 2 ends on Day 40, so Tasks 3 and 4 *both* can start on Day 41. Task 3 has a duration of 5 days, and will end on Day 45. Task 4 has a duration of 25 days, and will not end until Day 65.
- Task 5 depends on Tasks 3 and 4, which are multiple predecessors. Because Task 5 depends on *both* tasks, it cannot start until the later of the two tasks is complete. In this example, Task 3 ends earlier, but Task 4 will not be completed until Day 65, so Task 5 cannot start until Day 66.

Recall that the critical path is a series of tasks which, if delayed, would affect the final completion date of the overall project. In this example, Tasks 1 and 2 are the first tasks on the critical path. Now look at Task 5, which cannot start until both Tasks 3 and 4 are done. In this case, Task 4 is the controlling factor, because Task 4 finishes on Day 65, which is 20 days later than Task 3, which is completed on Day 45. Therefore, the start date for Task 5 is determined by the finish date for Task 4. In contrast, Task 3 has slack time, and could be delayed up to 20 days without affecting Task 5. **Slack time** is the amount of time that the task could be late without pushing back the completion date of the entire project. Tasks 1, 2, 4, and 5 represent the critical path, which is highlighted with red arrows in Figure 3-22.

Project Monitoring and Control

Regardless of whether the project was planned and scheduled with project management software or in some other manner, the project manager must keep track of the tasks and progress of team members, compare actual progress with the project plan, verify the completion of project milestones, and set standards and ensure that they are followed.

Monitoring and Control Techniques

To help ensure that quality standards are met, many project managers institute structured walk-throughs. A **structured walk-through** is a review of a project team member's work by other members of the team. Generally, systems analysts review the work of other systems analysts, and programmers review the work of other programmers, as a form of peer review. Structured walk-throughs take place throughout the SDLC and are called **design reviews**, **code reviews**, or **testing reviews**, depending on the phase in which they occur.

Maintaining a Schedule

Maintaining a project schedule can be challenging, and most projects run into at least some problems or delays. By monitoring and controlling the work, the project manager tries to anticipate problems, avoid them or minimize their impact, identify potential solutions, and select the best way to solve the problem.

The better the original plan, the easier it will be to control the project. If clear, verifiable milestones exist, it will be simple to determine if and when those targets are achieved. If enough milestones and frequent checkpoints exist, problems will be detected rapidly. A project that is planned and scheduled with PERT/CPM can be tracked and controlled using these same techniques. As work continues, the project manager revises the plan to record actual times for completed tasks and revises times for tasks that are not yet finished.

Project managers spend most of their time tracking the tasks along the critical path, because delays in those tasks have the greatest potential to delay or jeopardize the project. Other tasks cannot be ignored, however. For example, suppose that a task not on the critical path takes too long and depletes the allotted slack time. At that point, the task actually becomes part of the critical path, and any further delay will push back the overall project.

REPORTING

Members of the project team regularly report their progress to the project manager, who in turn reports to management and users. As shown in Figure 3-23, the project manager collects, verifies, organizes, and evaluates the information he or she receives from the team. Then the manager decides which information needs to be passed along, prepares a summary that can be understood easily, adds comments and explanations if needed, and submits it to management and users.

TOOLKIT TIME

The Communication Tools in Part A of the Systems Analyst's Toolkit can help you develop better reports and presentations. To learn more about these tools, turn to Part A of the four-part Toolkit that follows Chapter 12.

Project Status Meetings

Project managers, like the one shown in Figure 3-24, schedule regular meetings to update the team and discuss project status, issues, problems, and opportunities. Although meetings can be time consuming, most project managers believe they are worth the effort. The sessions give team members an opportunity to share information, discuss common problems, and explain new techniques. The meetings also give the project manager an opportunity to seek input and conduct brainstorming sessions.

Project Status Reports

Before going further, you should read the *Question of Ethics* feature on page 125, which describes an interesting conflict at Final Four Industries.

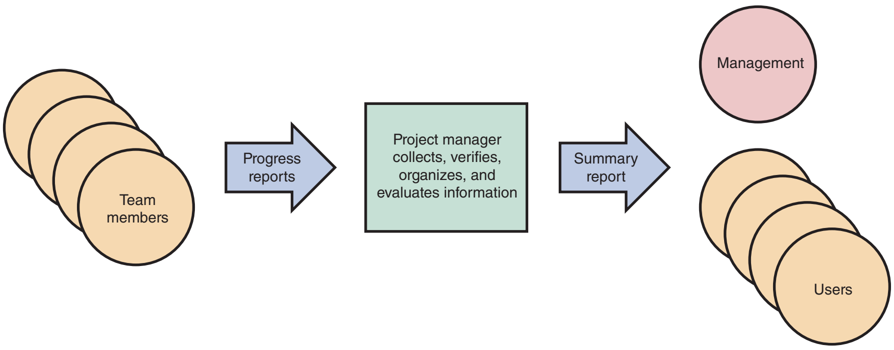


FIGURE 3-23 Members of the project team regularly report their progress to the project manager, who in turn reports to management and users.

FIGURE 3-24 Project managers schedule regular meetings to update the project team and discuss project status, issues, problems, and opportunities.

A project is in trouble, but the project manager is reluctant to report the problems. The case highlights important ethical issues that often arise in this situation.

A project manager must report regularly to his or her immediate supervisor, upper management, and users. Although a progress report might be given verbally to an immediate supervisor, reports to management and users usually are written. Gantt charts often are included in progress reports to show project status graphically. Deciding how to handle potential problems can be difficult. At what point should you inform management about the possibility of cost overruns, schedule delays, or technical problems? At one extreme is the overly cautious project manager who alerts management to every potential snag and slight delay. The danger here is that the manager loses credibility over a period of time, and management might ignore potentially serious situations. At the other extreme is the project manager who tries to handle all situations single-handedly and does not alert management until a problem is serious. By the time management learns of the problem, little time might remain in which to react or devise a solution.

A project manager's best course of action lies somewhere between the two extremes, but is probably closer to the first. If you are unsure of the consequences, you should be cautious and warn management about the possibility of a problem. When you report the situation, you also should explain what you are doing to handle and monitor the problem. If you believe the situation is beyond your control, you might want to suggest possible actions that management can take to resolve the situation. Most managers recognize that problems do occur on most projects; it is better to alert management sooner rather than later.

Project Management Examples

You can use these examples to practice the skills you learned in this chapter. You will also see how you can use project management software to help you manage and display the tasks.

PERT/CPM Example

Figure 3-25 shows a list of 11 tasks. The example is more complex, but the same guidelines apply. Notice that each task has an ID, a description, a duration, and a reference to predecessor tasks, if any, which must be completed before the task can begin. Also notice that dependent tasks can have one predecessor task, or several. You construct a PERT/CPM chart from this task list in a two-step process:

Task No.	Description	Duration (Days)	Predecessor Tasks
1	Develop Plan	1	-
2	Assign Tasks	4	1
3	Obtain Hardware	17	1
4	Programming	70	2
5	Install Hardware	10	3
6	Program Test	30	4
7	Write User Manual	25	5
8	Convert Files	20	5
9	System Test	25	6
10	User Training	20	7, 8
11	User Test	25	9,10

FIGURE 3-25 Example of a table listing 11 tasks, together with their descriptions, durations, and predecessor tasks.

STEP 1: CREATE THE WORK BREAKDOWN STRUCTURE In the first step, as shown in Figure 3-26 on the next page, you identify the tasks, determine task dependencies, and enter the task name, ID, and duration. Notice that this example includes dependent tasks, multiple successor tasks, and multiple predecessor tasks.

STEP 2: ENTER START AND FINISH TIMES In the second step, as shown in Figure 3-27, you enter the start and finish times by applying the guidelines in this section. For example, Task 1 has a one-day duration, so you enter the start and finish times for Task 1 as Day 1. Then you enter Day 2 as the start time for successor Tasks 2 and 3. Continuing from left to right, you add the task duration for each task

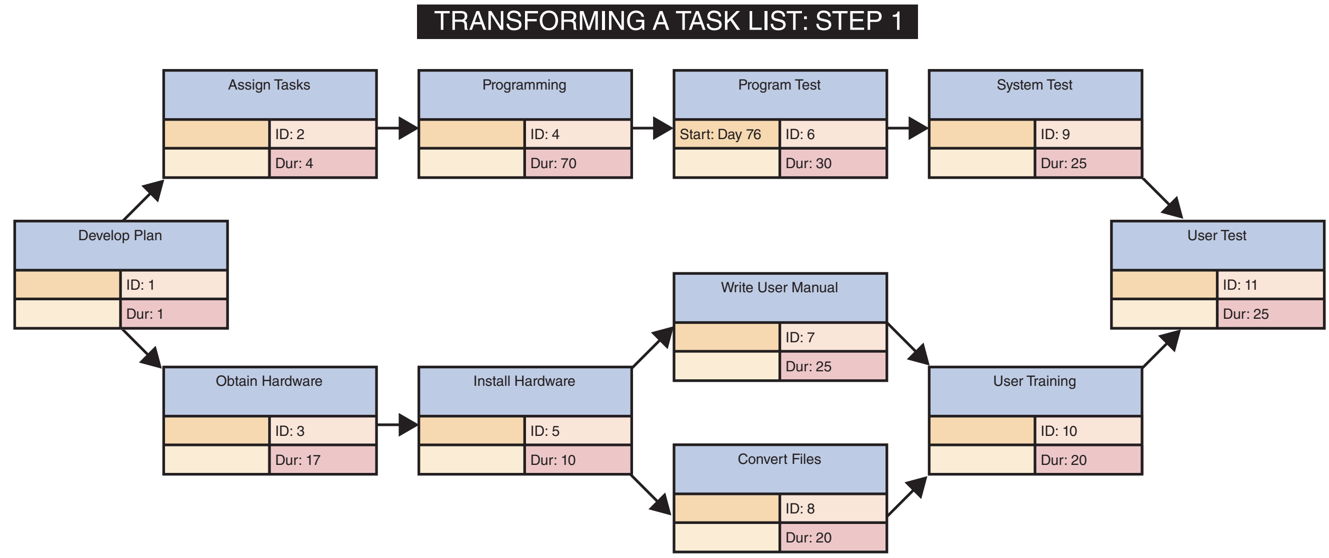


FIGURE 3-26 To transform a task list into a PERT/CPM chart, you first enter the task name, ID, duration, and predecessors for each task. Notice that this example includes dependent tasks, tasks with multiple successors, and tasks with multiple predecessors.

to its start time to determine its finish time. As you proceed, there are three important rules you must keep in mind:

- If a successor task has more than one predecessor task, use the *latest* finish time of the predecessor tasks to determine the start time for the successor task.
- If a predecessor task has more than one successor task, use the predecessor task's finish time to determine the start time for *all* successor tasks.
- Continuing from left to right, add the task duration for each task to its start time to determine and enter its finish time. Again, be very careful not to add too many days. For example, if a task starts on Day 10 and has a duration of 5 days, then the finish would be Day 14 — *not* Day 15.

When you enter all the start and finish times, you determine that the project will be completed on Day 155. Also, you note that Tasks 1, 2, 4, 6, 9, and 11 represent the critical path shown by the red arrows.

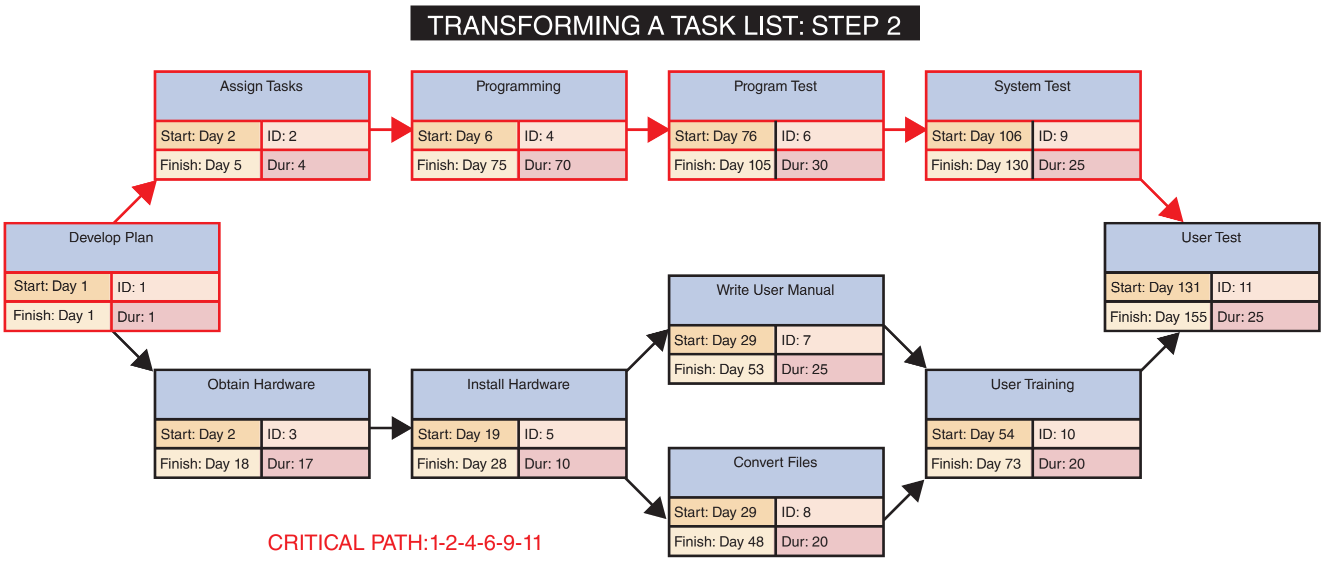


FIGURE 3-27 To complete the PERT/CPM chart, you apply the guidelines explained in this section. For example, Task 1 has a one-day duration, so you enter the start and finish for Task 1 as Day 1. Then you enter Day 2 as the start for successor Tasks 2 and 3.

ON THE WEB

To learn more about project management software, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Project Management Software link.

Software-Driven Example

Project managers use powerful software to help plan, schedule, monitor, and report on a project. Most project management programs offer features such as PERT/CPM, Gantt charts, resource scheduling, project calendars, and cost tracking.

Microsoft Project is a full-featured project management program that holds the dominant share of the market. On the Web, Microsoft offers demo versions, training, and tips. Although Microsoft is the industry leader, many other vendors offer project management software, and you can explore these options by searching on the Web. One product, **Open Workbench**, is available as free software, complete with manuals and sample projects, as shown in Figure 3-28. You can download the latest version from the Open Workbench site at *openworkbench.org*, or you can use the download link in the Features section of the Student Study Tool CD-ROM, which also contains a user manual for Open Workbench.

As the Web site explains, Open Workbench is **open-source software** that is supported by a large group of users and developers. Support options include community forums that are open to all users, various training packages, and third-party support. For many small to medium-sized projects, Open Workbench would be a cost-effective alternative that would compare favorably to Microsoft Project. Open Workbench also can exchange files with Microsoft Project by importing and exporting the data in XML file format.

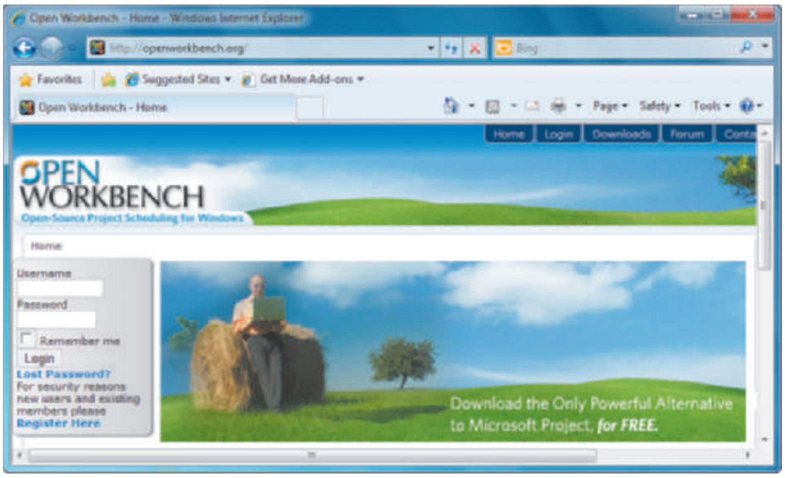


FIGURE 3-28 Open Workbench is a free, open-source project management program with powerful features and capabilities.

When you use project management software, you follow the same step-by-step process to develop a WBS and create various types of charts. The following sections focus on some basic concepts, but most programs include many powerful features, such as automated reporting tools, and convenient data import-export tools. You can sample these features by reviewing the Open Workbench User Guide, which is included in your Student Study Tool CD-ROM.

Please study the following task summary:

- First, we will review the systems request. That will take three days.
- Then, two tasks can begin at once: We can review the documentation, which will take three days, and review the Internet access delays, which will take two days.
- When the documentation and the Internet access delays have been analyzed, we can contact managers about the interviews, which will take two days.
- After we contact the managers, we can plan the interview schedule, which will take two days.
- Next, we can prepare the preliminary investigation report, which will take two days.
- When the report is ready, we can deliver our presentation to the committee, which will take two days.
- After the presentation, three tasks can begin at once: We plan the interview questions, which will take one day; contact the interviewees, which will take one day; and send out the questionnaire, which will be returned in five days.
- When the interview questions are ready and the interviewees have been contacted, we can conduct the interviews, which will take three days.
- Finally, when the interviews have been conducted and the questionnaire results are back, we can tabulate all results, which will take one day.

FIGURE 3-29 A sample task summary.

WORK BREAKDOWN STRUCTURE You already know how to create a work breakdown structure. If you are using Microsoft Project or Open Workbench, the process is exactly the same. You must identify the tasks, durations, and task patterns. You might have to develop this information on your own, or you might work with a task summary like the one in Figure 3-29. Your manager would like you to create a Gantt chart and a PERT chart that show all tasks, dependencies, dates, and total project duration. Your first step is to create a Gantt chart showing the necessary information. You decide to use Microsoft Project to construct the chart. As you enter each task, you also enter the duration and the predecessor tasks, if any.

GANTT CHART As you enter the tasks, durations, and predecessor tasks, the program automatically performs the calculations, detects the task patterns, and creates a Gantt chart similar to the one shown in Figure 3-30. The chart consists of 12 horizontal bars, connected with arrows that indicate the task dependencies. Notice that Saturdays and Sundays are shown as shaded columns, because no work will be performed on those days. The program makes these adjustments automatically. For example, Task 2, which has a duration of three days, starts on Thursday and ends on Monday.

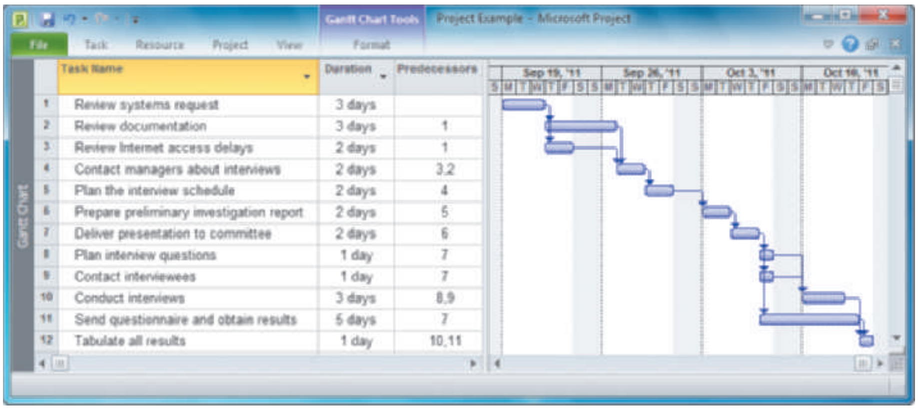


FIGURE 3-30 Open Workbench can show the sample project as a Gantt chart, or as a PERT chart that includes tasks, durations, dependencies, and a highlighted critical path.

NETWORK DIAGRAM After you complete the Gantt chart, you decide to view the data in the form of a Microsoft Project network diagram, which is similar to a PERT chart. When you select the Network Diagram option on the View menu, you can see the project tasks and dependencies, as shown in Figure 3-31. You study the diagram and see that the program has calculated a start and finish date for each task. Notice that the diagram displays the same information as the Gantt chart, including task dependencies, and also includes a red line that indicates the project's critical path. According to the diagram, if the

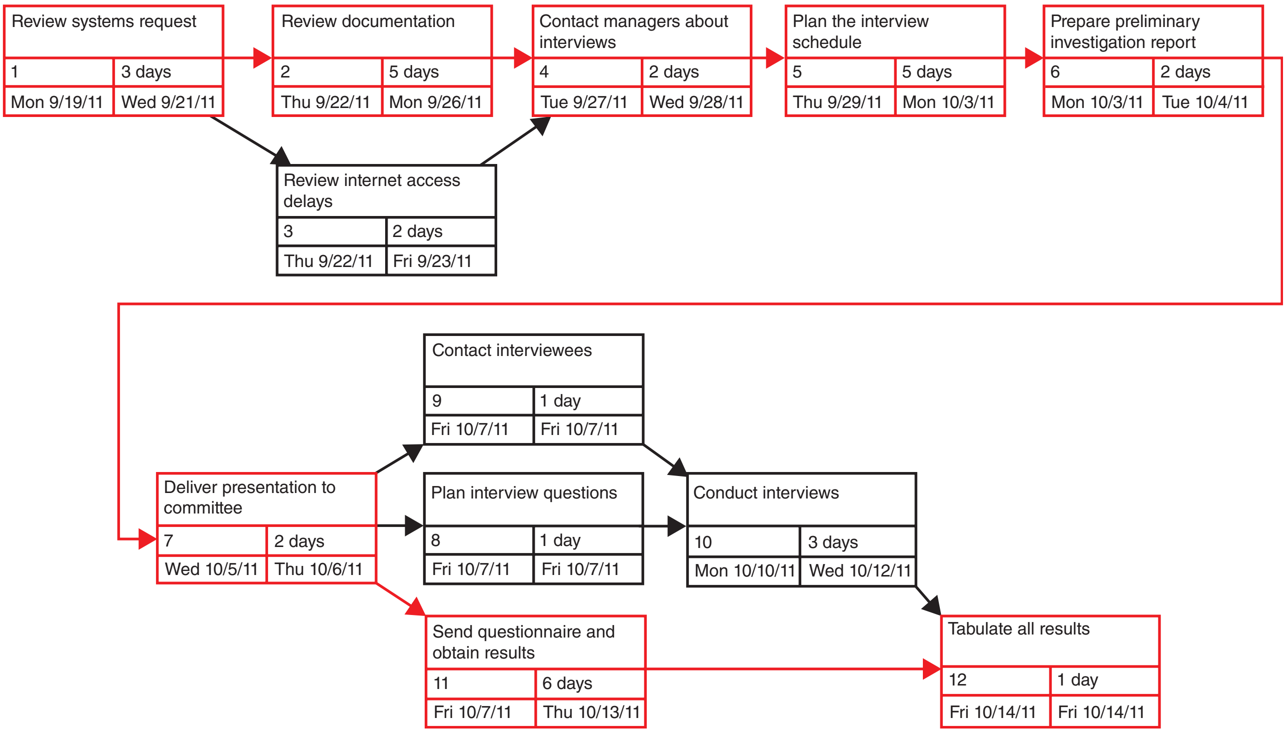


FIGURE 3-31 Using Microsoft Project, you can display a network diagram, which is similar to a PERT chart. Notice that the critical path appears as a red line.

project remains on schedule, the last task will be completed on Friday, October 14, 2011. Notice that the task boxes in Microsoft Project are similar to PERT/CPM task boxes. Using Microsoft Project, you can assign each task to one or more people, assign budget targets, produce progress reports, and readjust schedules and deadlines as necessary.

The latest version of Project is Microsoft Project 2010. This release is offered in a Standard version, a Professional version, and a Server version that includes support for large, enterprise-wide projects. In addition to providing a full description, demos, and training on its Web site, Microsoft also offers a free 60-day trial version that allows you to install, use, and evaluate the program.

An alternative to Microsoft Project is the Open Workbench program, which is free. Figure 3-32 shows the Open Workbench version of the same project shown in Figure 3-30 on the previous page. Using Open Workbench, you create tasks and durations, indicate dependencies, and assign resources, just as you would in Microsoft Project. Notice that the critical path is highlighted, both in the Gantt chart and the network diagram.

Regardless of which software you use, you can see from these examples that project schedules, task estimates, and personnel assignments all are interrelated. Therefore, project planning is a dynamic task and involves constant change. One significant advantage of integrated interactive project management software is that it allows the project manager to adjust schedules, estimates, and resource assignments rapidly to develop a workable plan.

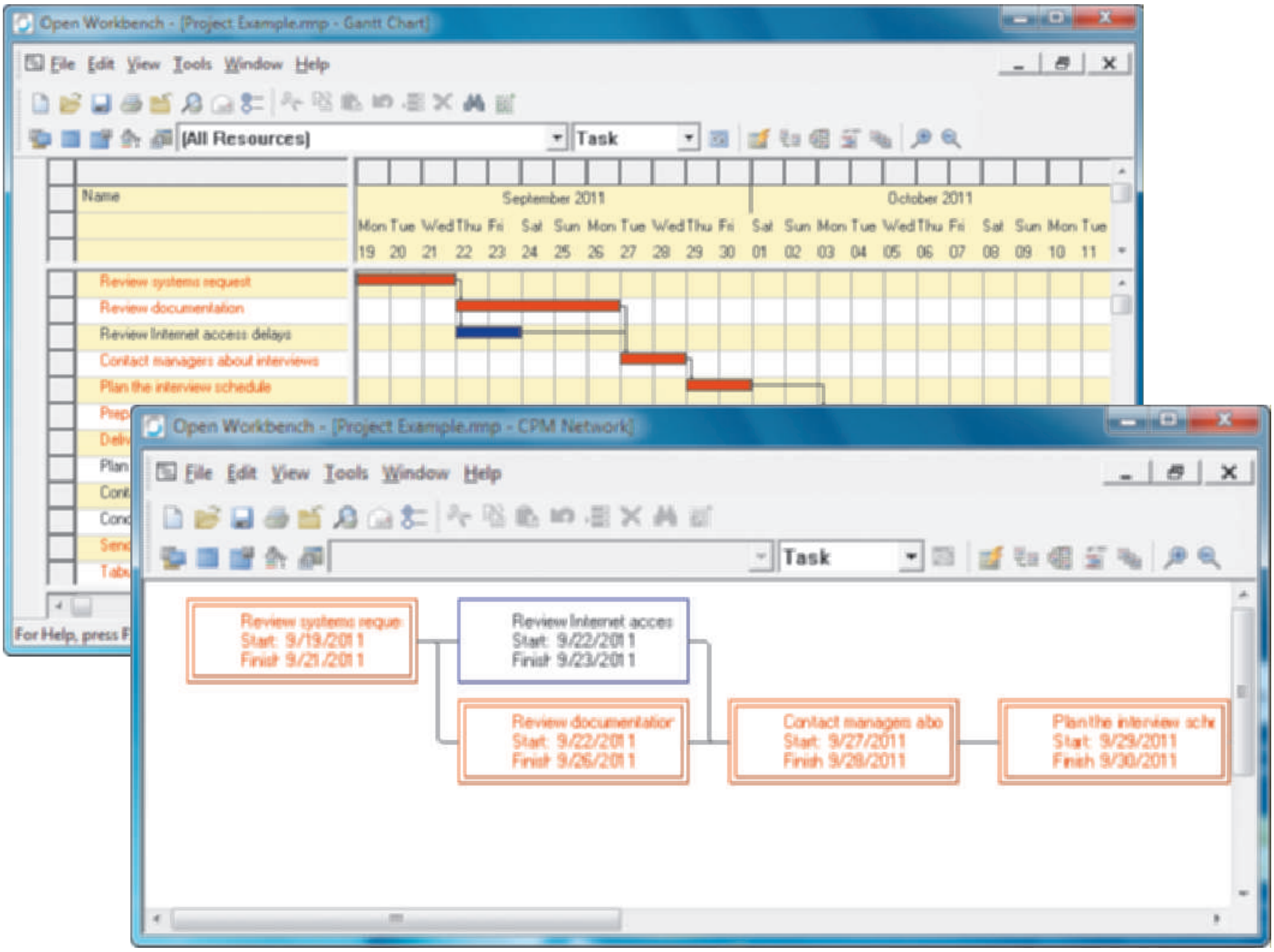


FIGURE 3-32 Open Workbench can show the sample project as a Gantt chart, or as a PERT chart that includes tasks, durations, dependencies, and a highlighted critical path.

CASE IN POINT 3.4: CENSUS 2010

In April 2008, the U.S. Commerce Department canceled a plan to acquire 500,000 handheld computers to tabulate data during the 2010 census. According to Commerce Secretary Carlos Gutierrez, costs had skyrocketed. He blamed the problem on "a lack of effective communications with one of our major suppliers."

Apparently, there was plenty of blame to go around. Secretary Gutierrez noted that the Census Bureau had submitted numerous technical changes to the vendor, Harris Corporation. This greatly increased the cost and the complexity of the devices. Gutierrez stated, "The Census Bureau was unaccustomed to working with an outside vendor on such a large contract." He also pointed out that the vendor had submitted an initial estimate of $36 million to operate a help desk to assist census-takers, but that figure had jumped to $217 million. "It was a bad estimate. I can't think of a better way to say it. Harris gave us the number. We accepted it. It was totally underestimated."

What can be learned from the failure of this project, and could it have been prevented? Suppose you were asked to head up a similar project. What would you do to prevent a similar outcome?

RISK MANAGEMENT

Every IT project involves risks that systems analysts and project managers must address. A **risk** is an event that could affect the project negatively. **Risk management** is the process of identifying, analyzing, anticipating, and monitoring risks to minimize their impact on the project.

Steps in Risk Management

The first step in risk management is to develop a specific plan. Although project management experts differ with regard to the number of steps or phases, a basic list would include the following tasks:

- *Develop a risk management plan*. A **risk management plan** includes a review of the project's scope, stakeholders, budget, schedule, and any other internal or external factors that might affect the project. The plan should define project roles and responsibilities, risk management methods and procedures, categories of risks, and contingency plans.
- *Identify the risks*. **Risk identification** lists each risk and assesses the likelihood that it could affect the project. The details would depend on the specific project, but most lists would include a means of identification, and a brief description of the risk, what might cause it to occur, who would be responsible for responding, and the potential impact of the risk.
- *Analyze the risks*. This typically is a two-step process: Qualitative risk analysis and quantitative risk analysis. **Qualitative risk analysis** evaluates each risk by estimating the probability that it will occur and the degree of impact. Project managers can use a formula to weigh risk and impact values, or they can display the results in a two-axis grid. For example, a Microsoft Excel XY chart can be used to display the matrix, as shown in Figure 3-33. In the chart, notice the various combinations of risk and impact ratings for the five sample values. This tool can help a project manager focus on the most critical areas, where risk probability and potential impact are high.

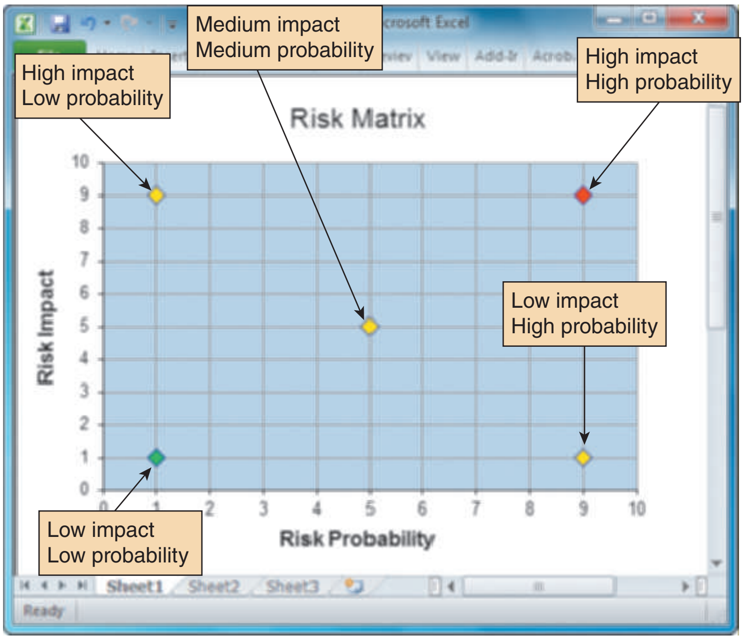


FIGURE 3-33 You can use a Microsoft Excel XY chart type to display a risk matrix that shows risk probability and potential impact.

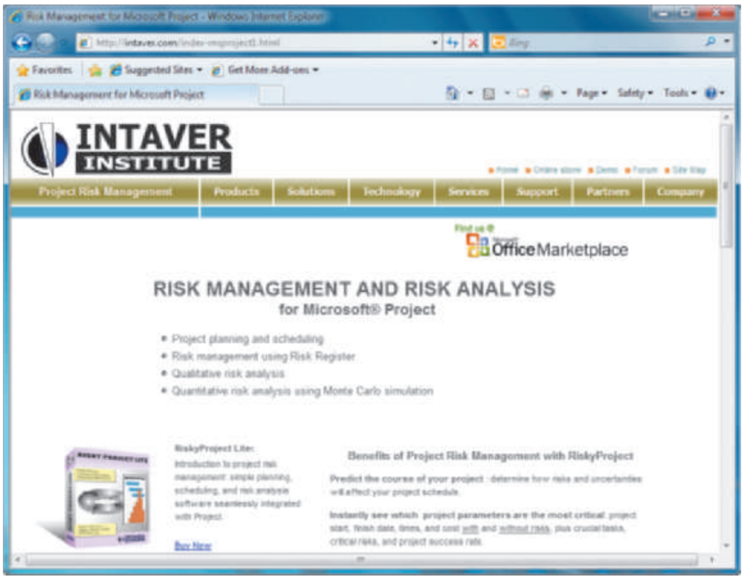


FIGURE 3-34 Intaver Institute offers an add-on risk management package for Microsoft Project.

The purpose of **quantitative risk analysis** is to understand the actual impact in terms of dollars, time, project scope, or quality. Quantitative risk analysis can involve a modeling process called what-if analysis, which allows a project manager to vary one or more element(s) in a model to measure the effect on other elements. This topic is discussed in more detail in Chapter 12, Managing Systems Support and Security.

- Create a risk response plan. A **risk response plan** is a proactive effort to anticipate a risk and describe an action plan to deal with it. An effective risk response plan can reduce the overall impact by triggering timely and appropriate action.
- Monitor risks. This activity is ongoing throughout the risk management process. It is important to conduct a continuous tracking process that can identify new risks, notice changes in existing risks, and update any other areas of the risk management plan.

Risk Management Software

Most project management software includes powerful features that allow a project manager to assign specific dates as constraints, align task dependencies, note external factors that might affect a task, track progress, and display tasks that are behind schedule. In addition, some vendors offer risk management add-ons, such as the one shown in Figure 3-34.

The enterprise edition of Microsoft Project, Microsoft Project Server 2010, has a built-in risk management capability that can be used for large, corporate-wide projects. Microsoft claims that the software can link risks with specific tasks and projects, specify probability and impact, assign ownership, and track progress to manage projects more efficiently. Microsoft's risk management model includes the following factors:

- Probability, which represents the likelihood that the risk will happen, expressed as a percentage
- Impact, which indicates the degree of adverse effect should the risk occur, on a scale of 1 to 10
- Cost, which indicates the potential financial impact of the risk
- Category, which specifies the risk type
- Description, which specifies the nature of the risk
- Mitigation plan, which identifies plans to control or limit the risk

- Contingency plan, which specifies actions to be taken if the risk occurs
- Trigger, which identifies a condition that would initiate the contingency plan

Armed with this information, the IT team can make a recommendation regarding the risks associated with the project. Depending on the nature and magnitude of the risks, the final decision might be made by management.

MANAGING FOR SUCCESS

To be successful, an information system must satisfy business requirements, stay within budget, be completed on time, and — most important of all — be managed effectively. When a project develops problems, the reasons typically involve business, budget, or schedule issues, as explained in the following sections. In addition to planning and managing the project, a project manager must be able to recognize problems and deal with them effectively.

Business Issues

The major objective of every system is to provide a solution to a business problem or opportunity. If the system does not do this, then it is a failure — regardless of positive reaction from users, acceptable budget performance, or timely delivery. When the information system does not meet business requirements, causes might include unidentified or unclear requirements, inadequately defined scope, imprecise targets, shortcuts or sloppy work during systems analysis, poor design choices, insufficient testing or inadequate testing procedures, and lack of change control procedures. Systems also fail because of changes in the organization's culture, funding, or objectives. A system that falls short of business needs also produces problems for users and reduces employee morale and productivity.

As you learned in Chapter 2, projects without clear scope definitions are risky, because they tend to expand gradually, without specific authorization, in a process called **project creep**. However, even when a project is clearly described, it must be managed constantly.

Budget Issues

Cost overruns typically result from one or more of the following:

- Unrealistic estimates that are too optimistic or based on incomplete information
- Failure to develop an accurate forecast that considers all costs over the life of the project
- Poor monitoring of progress and slow response to early warning signs of problems
- Schedule delays due to factors that were not foreseen
- Human resource issues, including turnover, inadequate training, and motivation

Schedule Issues

Problems with timetables and project milestones can indicate a failure to recognize task dependencies, confusion between effort and progress, poor monitoring and control methods, personality conflicts among team members, or turnover of project personnel. The failure of an IT project also can be caused by poor project management techniques.

If the project manager fails to plan, staff, organize, supervise, communicate, motivate, evaluate, direct, and control properly, then the project is certain to fail. Even when factors outside his or her control contribute to the failure, the project manager is responsible for recognizing the early warning signs and handling them effectively.

The Bottom Line

Project management is a challenging task. Project managers must be alert, technically competent, and highly resourceful. They also must be good communicators with strong human resource skills. A project manager can be proud when he or she handles a successful project that helps the company achieve its business objectives, such as the Apple product launch shown in Figure 3-35.

Unfortunately, projects can and do get derailed for a wide variety of reasons. When problems occur, the project manager's ability to handle the situation becomes the critical factor. When a project manager first recognizes that a project is in trouble, what options are available? Alternatives can include trimming the project requirements, adding to the project resources, delaying the project deadline, and improving management controls and procedures. Sometimes, when a project experiences delays or cost overruns, the system still can be delivered on time and within budget if several less critical requirements are trimmed. The system can be delivered to satisfy the most necessary requirements, and additional features can be added later as a part of a maintenance or enhancement project.

If a project is in trouble because of a lack of resources or organizational support, management might be willing to give the project more commitment and higher priority. For example, management might agree to add more people to a project that is behind schedule. Adding staff, however, will reduce the project's completion time only if the additional people can be integrated effectively into the development team. If team members lack experience with certain aspects of the required technology, temporary help might be obtained from IT consultants or part-time staff. Adding staff can mean training and orienting the new people, however. In some situations, adding more people to a project actually might increase the time necessary to complete the project because of a principle called **Brooks' Law**. This interesting concept was stated by Frederick Brooks, Jr., an IBM engineer, who observed that adding manpower to a late software project only makes it later. Brooks reached this conclusion when he saw that new workers on a project first had to be educated and instructed by existing employees whose own productivity was reduced accordingly.

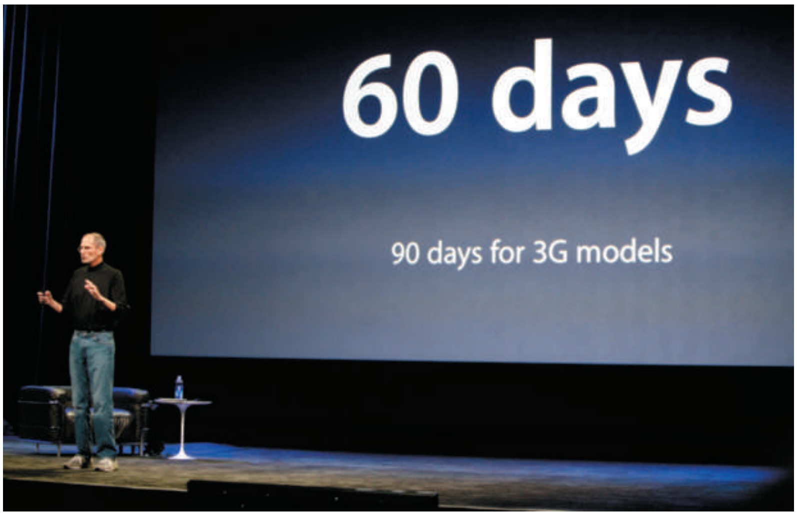


FIGURE 3-35 When a complex project is successful, the project manager has a right to be proud.

A QUESTION OF ETHICS

"Better blow the whistle," says Roy, your friend and project teammate at Final Four Industries. "The project is out of control, and you know it!" "Maybe so," you respond, "But that's not my call — I'm not the project manager." What you don't say is that Stephanie, the project manager, feels like her career is on the line and she is reluctant to bring bad news to management at this time. She honestly believes that the project can catch up, and says that a bad report on a major project could result in bad publicity for the firm and frighten potential customers.

To be fair, the next management progress report is scheduled in three weeks. It is possible that the team could catch up, but you doubt it. You wonder if there is an ethical question here: Even though the report isn't due yet, should a significant problem be reported to management as soon as possible? You are concerned about the issue, and you decide to discuss it with Stephanie. What will you say to her?

CHAPTER SUMMARY

Project management is the process of planning, scheduling, monitoring, and reporting on the development of an information system. A successful project must be completed on time, within budget, and deliver a quality product that satisfies users and meets requirements. Project management techniques can be used throughout the SDLC.

Project managers are responsible for project planning, scheduling, monitoring, and reporting. Planning includes identifying all project tasks and estimating the completion time and cost of each. Project scheduling involves the creation of a specific timetable, usually in the form of charts that show tasks, task dependencies, and critical tasks that might delay the project. Project monitoring requires guiding, supervising, and coordinating the project team's workload. The project manager must monitor the progress, evaluate the results, and take corrective action when necessary to control the project and stay on target. Project reporting includes regular progress reports to management, users, and the project team itself. Effective reporting requires strong communication skills and a sense of what others want and need to know about the project.

Planning, scheduling, monitoring and reporting all take place within a larger project development framework, which includes three key steps: creating a work breakdown structure, identifying task patterns, and calculating the critical path.

A work breakdown structure must clearly identify each task and include an estimated duration. A task, or activity, is any work that has a beginning and an end and requires the use of company resources such as people, time, or money. Time and cost estimates for tasks usually are made in person-days. A person-day represents the work that one person can accomplish in one day. Estimating the time for project activities is more difficult with larger systems. Project managers must consider the project size and scope, IT resources, prior experience with similar projects or systems, and applicable constraints. In addition to tasks, every project has events, or milestones. An event, or milestone, is a recognizable reference point that you can use to monitor progress.

Task patterns establish the sequence of work in a project. Task patterns involve dependent tasks, multiple successor tasks, and multiple predecessor tasks. In larger projects, these patterns can be very complex.

A critical path is a series of tasks which, if delayed, would affect the completion date of the overall project. If any task on the critical path falls behind schedule, the entire project will be delayed. Tasks on the critical path cannot have slack time. To identify the critical path, you calculate the start and finish date for each task, which will determine the critical path for the project.

In project scheduling, the project manager develops a specific time for each task, based on available resources and whether or not the task is dependent on other predecessor tasks. The manager can use graphical tools such as Gantt charts and PERT charts to assist in the scheduling process.

A Gantt chart is a horizontal bar chart that represents the project schedule with time on the horizontal axis and tasks arranged vertically. It shows individual tasks and task groups, which include several tasks. In a Gantt chart, the length of the bar indicates the duration of the tasks. A Gantt chart can display progress, but does not show task dependency details or resource assignment unless the chart was created with a project management program that supports dependency linking and the entry of other information.

A PERT/CPM chart shows the project as a network diagram with tasks connected by arrows. Using a prescribed calculation method, the project manager uses a PERT chart to determine the overall duration of the project and provide specific information for each task, including the task IDs, their durations, start and finish times, and the order in which they must be performed. With this information, the manager can determine the critical path, which is the sequence of tasks that have no slack time and must be performed on schedule in order to meet the overall project deadline.

Most project managers use powerful software such as Microsoft Project and Open Workbench to plan, schedule, and monitor projects. Project managers are responsible for risk management, which is the process of identifying, analyzing, anticipating, and monitoring risks to minimize their impact on the project.

In the end, every successful information system must support business requirements, stay within budget, and be completed on time. Sound project management involves the same skills as any other management. The project manager must be perceptive, analytical, well-organized, and a good communicator. If the project manager senses that the project is off-track, he or she must take immediate steps to diagnose and solve the problem. If the project manager fails to plan, staff, organize, supervise, communicate, motivate, evaluate, direct, and control properly, then the project is certain to fail. Even when factors outside his or her control contribute to the failure, the project manager is responsible for recognizing the early warning signs and handling them effectively.

Key Terms and Phrases

activity *106*
best-case estimate *107*
bottom-up technique *105*
Brooks' Law *124*
code review *114*
concurrent tasks *111*
critical path *113*
Critical Path Method (CPM) *105*
dependent task *111*
design review *114*
duration *110*
event *106*
finish day/date *110*
Gantt chart *104*
Microsoft Project *118*
milestone *106*
network diagram *105*
Open Workbench *118*
open-source software *118*
person-day *107*
PERT/CPM *105*
predecessor task *111*
probable-case estimate *107*
Program Evaluation Review Technique (PERT) *105*
project coordinator *103*
project creep *123*
project leader *103*
project management *102*
project management software *113*
project manager *103*
project monitoring *103*
project planning *103*
project reporting *103*
project scheduling *103*
qualitative risk analysis *121*
quantitative risk analysis *122*
risk *121*
risk identification *121*
risk management *121*
risk management plan *121*
risk response plan *122*
slack time *114*
start day/date *110*
structured walk-through *114*
successor task *111*
task *105*
task box *110*
task group *104*
task ID *110*
task name *110*
task pattern *110*
testing review *114*
weight *107*
work breakdown structure (WBS) *104*
worst-case estimate *107*

Learn It Online

Instructions: To complete the Learn It Online exercises, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to the resources for this chapter, and click the link for the exercise you want to complete.

1 Chapter Reinforcement

TF, MC, and SA

Click the Chapter Reinforcement link. Print the quiz by clicking Print on the File menu for each page. Answer each question.

2 Flash Cards

Click the Flash Cards link and read the instructions. Type 20 (or a number specified by your instructor) in the Number of playing cards text box, type your name in the Enter your name text box, and then click the Flip Card button. When the flash card is displayed, read the question and then click the ANSWER box arrow to select an answer. Flip through the Flash Cards. If your score is 15 (75%) correct or greater, click Print on the File menu to print your results. If your score is less than 15 (75%) correct, then redo this exercise by clicking the Replay button.

3 Practice Test

Click the Practice Test link. Answer each question, enter your first and last name at the bottom of the page, and then click the Grade Test button. When the graded practice test is displayed on your screen, click Print on the File menu to print a hard copy. Continue to take practice tests until you score 80% or better.

4 Who Wants To Be a Computer Genius?

Click the Computer Genius link. Read the instructions, enter your first and last name at the bottom of the page, and then click the Play button. When your score is displayed, click the PRINT RESULTS link to print a hard copy.

5 Wheel of Terms

Click the Wheel of Terms link. Read the instructions, and then enter your first and last name and your school name. Click the PLAY button. When your score is displayed on the screen, right-click the score and then click Print on the shortcut menu to print a hard copy.

6 Crossword Puzzle Challenge

Click the Crossword Puzzle Challenge link. Read the instructions, and then enter your first and last name. Click the SUBMIT button. Work the crossword puzzle. When you are finished, click the SUBMIT button. When the crossword puzzle is redisplayed, click the Print Puzzle button to print a hard copy.

SCR Associates Case Simulation Session 3: Managing Systems Projects

Overview

The SCR Associates case study is a Web-based simulation that allows you to practice your skills in a real-world environment. The case study transports you to SCR's intranet, where you complete 12 work sessions, each aligning with a chapter. As you work on the case, you will receive e-mail and voice mail messages, obtain information from SCR's online libraries, and perform various tasks.

How do I use the case?

- Review the SCR background material in Chapter 1.
- Read the Preview for this session and study the Task List
- Visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to the **SCR Case Simulation**, and locate the intranet link.
- Enter your name and the password **sad9e**. An opening screen will display the 12 sessions.
- Select this session. Check your e-mail and voice mail carefully, and then work on the tasks.

Preview: Session 3

The TIMS system was approved by top management. Jesse Baker, systems group manager, has asked you to help her manage the project. She said it will be a great opportunity to learn more about project management and practice your skills, so you'll be able to handle a future project on your own. She specifically suggested that you become familiar with work breakdown structures, task patterns, and critical path calculations. To make sure that you are prepared, you decide to study the project management chapter in your textbook before starting the tasks.

Task List

1. Jesse wants me to investigate Open Workbench software to determine whether it would be suitable for SCR. She asked me to prepare a summary of pros and cons, and a sample of screen shots and information.

2. Jesse likes the idea of using task completion estimates with best-case, probable-case, and worst-case estimates. She said that I should use typical formulas and weight values to create a Microsoft Excel spreadsheet that would make it easier to calculate expected task durations.

3. To practice my skills, Jesse asked me to create an imaginary project with 10 tasks, which include dependent, multiple predecessor, and multiple successor tasks. She wants me to create a list showing the tasks and dependencies, and then lay it out on paper to show the logical flow, and the duration, start, and finish for each task.

4. I'm excited to be part of the project team, and Jesse wants me to prepare a brief handout for the other team members with some do's and don'ts regarding project management. She said to make it look like a checklist of keys to project success.

FIGURE 3-36 Task list: Session 3.

Chapter Exercises

Review Questions

1. What is project management, and what are its main objectives?
2. What is the relationship between tasks and events, or milestones?
3. What is a work breakdown structure? How do you create one?
4. What are task patterns, and how can you recognize them?
5. Compare the advantages and disadvantages of Gantt and PERT/CPM charts.
6. Define the following terms: best-case estimate, probable-case estimate, and worst-case estimate, and describe how project managers use these concepts.
7. How does a project manager calculate start and finish times?
8. What is a critical path, and why is it important to project managers? How do you identify the critical path?
9. What are some project reporting and communication techniques?
10. What is risk management, and why is it important?

Discussion Topics

1. In *Poor Richard's Almanac*, Benjamin Franklin penned the familiar lines: "For the want of a nail the shoe was lost, for the want of a shoe the horse was lost, for the want of a horse the rider was lost, for the want of a rider the battle was lost, for the want of a battle the kingdom was lost — and all for the want of a horseshoe nail." Looking at the outcome in hindsight, could project management concepts have avoided the loss of the kingdom? Explain your answers.
2. Microsoft Project is an example of software that is very powerful, but quite expensive. As a project manager, how would you justify the purchase of this software? Also, would you consider using Open Workbench? Why or why not?
3. Suppose you want to manage a relatively small project, but you have no access to project management software of any kind. How could you use a spreadsheet program or a database program to manage the project? Share your ideas with the class.
4. Many managers claim to have "seat of the pants" intuition when it comes to project management. In your view, does this kind of intuition actually exist? Can you think of examples to support your views?

Projects

1. Think of all the tasks that you perform when you purchase a car. Include any research, decisions, or financial issues that relate to the purchase. Draw a Gantt chart that shows all the tasks and the estimated duration of each.
2. Perform Internet research to learn more about project risk management, and write a summary of the results. Be sure to search for a book titled *Waltzing with Bears: Managing Risk on Software Projects*, by Tom Demarco and Timothy Lister.
3. Go to Microsoft's Web site and navigate to the Download and Trials area. Select Microsoft Project Professional 2010, download the program, and install it. Then create a project based on the five tasks shown in Figure 3-21 on page 113. When the project is complete, click View, then click Network Diagram. Do the tasks resemble Figure 3-22 on page 114? Is the critical path the same?
4. Describe three personal experiences where a project management approach would have been helpful.

Apply Your Knowledge

The Apply Your Knowledge section contains four mini-cases. Each case describes a situation, explains your role in the case, and asks you to respond to questions. You can answer the questions by applying knowledge you learned in the chapter.

1 Countywide Construction

At Countywide Construction, you are trying to convince your boss that he should consider modern project management techniques to manage a complex project. Your boss says that he doesn't need anything fancy, and that he can guess the total time by the seat of his pants.

To prove your point, you decide to use a very simple example of a commercial construction project, with eight tasks. You create a hypothetical work breakdown structure, as follows:

- Prepare the foundation (10 days). Then assemble the building (4 days).
- When the building is assembled, start two tasks at once: Finish the interior work (4 days) and set up an appointment for the final building inspection (30 days).
- When the interior work is done, start two more tasks at once: landscaping (5 days) and driveway paving (2 days).
- When the landscaping and driveway are done, do the painting (5 days).
- Finally, when the painting is done and the final inspection has occurred, arrange the sale (3 days).

Now you ask your boss to estimate the total time and write his answer on a piece of paper. You look at the paper and see that his guess is wrong.

1. What is the correct answer?
2. What is the critical path?
3. Create a Gantt chart that shows the WBS.
4. Create a PERT/CPM chart.

2 Pleasantville High School Class

The computer science instructor at Pleasantville High School has asked you to visit her class and give a presentation about project management. You have just a few days to prepare, and you need to develop a presentation that briefly describes project management tools and techniques. You can be creative, and you might want to include examples of actual projects that you know about. In any case, try to describe how projects are planned, scheduled, monitored, and reported upon. Your presentation can be in the form of a Microsoft Word outline with notes, or as a set of PowerPoint slides.

1. Prepare opening comments that give the class an overview of project management.
2. Provide the class with a glossary of the most important project management terms and definitions.
3. Think of a common event like buying a new home, and show the class how a project manager might handle the matter.
4. Create a short scenario with 4 – 6 tasks, some of which depend on each other. You can use the two preceding cases as a model. Develop a sample answer that you will show the students after you give them a chance to analyze the tasks.

3 Lightfoot Industries

You have been asked to lead a training session for new employees at Lightfoot Industries. You must develop a specific schedule for the tasks listed below (the estimated task duration for each is shown in parentheses):

- First, you need to contact the participants and explain their roles (1 day). Then you must obtain approval from their department managers (5 days).
- After you obtain the approval, two tasks can begin at the same time: You can arrange the meeting room (4 days) and prepare an agenda for the initial session (11 days).
- When the agenda is ready, you can start two more concurrent tasks: Prepare the information packets (4 days) and create visual aids (8 days).
- When the meeting room is arranged and the information packets are ready, you can send out an e-mail to participants (1 day).
- Finally, after the e-mail is sent to participants and the visual aids are ready, you can conduct the JAD sessions (5 days).

1. Prepare a list showing all tasks and their durations.
2. Analyze the fact situation carefully to determine which tasks are concurrent and which ones are dependent on other tasks.
3. Using PERT/CPM techniques, develop a chart that shows the project. Use a format similar to Figure 3-27 on page 117. If project management software is available, use it to develop the chart.
4. What is the critical path for this project? How do you know?

4 Riverside Financial

At Riverside Financial, where you work as a project manager, you have been asked to conduct user training sessions during the implementation phase for a new information system. You must develop a specific schedule for the tasks (the estimated task duration for each is shown in parentheses):

- First, you need to send an e-mail message to all department managers announcing the training sessions (1 day).
- After the e-mail message goes out, two tasks can begin at the same time: You can develop the training material (4 days) and confirm arrangements for the training facility you plan to use (11 days).
- As soon as the training material is complete, you can work on two tasks at once: Arrange to have copies of handout material printed (3 days) and develop a set of PowerPoint slides (4 days).
- When the PowerPoint slides are ready, you conduct a practice training session with the instructor who will assist you (1 day).
- Finally, when the practice session is over, the handout material is ready, and the training facility is confirmed, you conduct the user training sessions (3 days).

1. Prepare a list showing all tasks and their durations.
2. Analyze the fact situation carefully to determine which tasks are concurrent and which ones are dependant on other tasks.
3. Using PERT/CPM techniques, develop a chart that shows the project. Use a format similar to Figure 3-27 on page 117. If project management software is available, use it to develop the chart.
4. What is the critical path for this project? How do you know?

Case Studies

Case studies allow you to practice specific skills learned in the chapter. Each chapter contains several case studies that continue throughout the textbook, and a chapter capstone case.

NEW CENTURY HEALTH CLINIC

New Century Health Clinic offers preventive medicine and traditional medical care. In your role as an IT consultant, you will help New Century develop a new information system.

Background

To ensure the quality, cost, and timeliness of the new information system, New Century is considering a project management approach. To obtain a better understanding of project management, Dr. Jones contacted Precision Planning, a consulting firm that specializes in managing projects of this type. He invited the company to deliver a brief presentation on project management concepts and advantages, and to submit a proposal for project management consulting services.

You joined Precision Planning two years ago as a project assistant, after working two summers as a student intern. Your supervisor, Charlie West, asked you to develop the presentation for New Century and you are excited about the opportunity. Charlie said that the main objective is to provide a clear, informative presentation.

Charlie wants you to include the following topics in your presentation: an overview of project management and its history, a description of the process, and an explanation of the most important terms and concepts. Charlie also wants you to describe task identification, various types of relationships among tasks, and schedule development. He says you should show how Gantt and PERT/CPM charts are developed, and how they can be used to plan, track, and control projects. Charlie also said that your presentation should include a specific example to illustrate all the main points.

Assignments

1. Create a Microsoft PowerPoint presentation that will meet the requirements that Charlie outlined to you.
2. Create a Microsoft Word handout that will meet the requirements that Charlie outlined to you.
3. Create a project management example with at least six tasks. Assign durations and task dependencies. At least three of the tasks should be dependent on other tasks. Use this example to display a Gantt chart.
4. Use the same data as Assignment 3 to display a PERT/CPM chart.

PERSONAL TRAINER, INC.

Personal Trainer, Inc. owns and operates fitness centers in a dozen Midwestern cities. The centers have done well, and the company is planning an international expansion by opening a new "supercenter" in the Toronto area. Personal Trainer's president, Cassia Umi, hired an IT consultant, Susan Park, to help develop an information system for the new facility. During the project, Susan will work closely with Gray Lewis, who will manage the new operation.

Background

You are enjoying your job as a student intern at Personal Trainer. Last week, Susan asked you to help her plan the new information system project. Susan knows that you have completed several information systems courses at the local college, and that you have studied project management tools and techniques.

Specifically, she wants you to get ready for the next set of systems development tasks, which will be requirements modeling for the new system. Yesterday, Susan called you into her office to discuss the specific tasks she wants you to perform. After meeting with Susan, you sit down and review your notes. She wants you to treat the set of tasks as a project, and to use project management skills to plan the tasks.

Here is what she suggested to you as a work breakdown structure, including the duration she estimated for each task:

- First, you need to meet with fitness center managers at other Personal Trainer locations (10 days).
- After these meetings, you can conduct a series of interviews (8 days).
- When the interviews are complete, two tasks can begin at the same time: You can review company records (2 days) and observe business operations (7 days).
- When you have reviewed the records and observed business operations, you can analyze the BumbleBee accounting software (3 days) and study a sample of sales and billing transactions (1 day).

You are excited about the opportunity to practice your skills, and you start to work on the following list.

Assignments

1. Create a table listing all tasks separately, with their duration.
2. Identify all dependent tasks, and indicate what predecessor tasks are required.
3. Construct a PERT/CPM chart similar to the one in Figure 3-27 on page 117. If you have access to Microsoft Project or other project management software, you can use it to help you create the chart.
4. Determine the overall duration of the project, and identify the critical path.

CHAPTER CAPSTONE CASE: SoftWear, Limited

SoftWear, Limited (SWL) is a continuing case study that illustrates the knowledge and skills described in each chapter. In this case study, the student acts as a member of the SWL systems development team and performs various tasks.

Background

At a recent management meeting, SWL's president, Robert Lansing, announced a major effort to control costs and improve quality. To help achieve this goal, Mr. Lansing stated that SWL would use project management tools and techniques to plan and manage all major corporate projects. He named several people who would work as an interdepartmental team to coordinate SWL's project management efforts. Team members included April Lane, director of planning; Mike Feiner, director of human resources; and Ann Hon, director of information technology.

The Interdepartmental Team

At their first meeting, the team came up with three main goals: Establish a company-wide understanding of project management concepts, identify suitable project management software, and develop comprehensive training for all SWL managers. Since Ann Hon had the most experience with project management, she agreed to serve as team leader. She also agreed to develop a list of concepts that the team could use as a starting point.

Project Management Concepts

The team met again a week later, and Ann distributed a list of 10 key questions:

1. What is a project?
2. What are project characteristics, constraints, and risks?
3. What is a project stakeholder?
4. What is the role of a project manager?
5. What is project planning?
6. What is project scheduling?
7. What is project monitoring and controlling?
8. What is project reporting?
9. What is project risk management?
10. What are the indications of project success or failure?

As the team members reviewed the list, Ann said that a set of working definitions would be a good first step in developing a company-wide approach for managing projects. She suggested that the answers were available from various sources, including a considerable body of literature and numerous online links. She also pointed out that the answers would be a key part of the proposed training program for SWL managers. The team decided to split up the research tasks and share the results at the next meeting.

Project Management Software

Ann made arrangements for the other team members to obtain a copy of Microsoft Project, which is the leading project management program. She also suggested that each of them try the brief Project training courses that are available on the Microsoft Web site. She then walked them though a two-hour session that demonstrated the software. She showed examples of Gantt charts, PERT charts, milestones, task dependencies, and resource assignments.

(continues)

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

Ann also pointed out that other software alternatives exist, including free, open-source programs, such as Open Workbench, which is supported by a large user group. For now, the team agreed to obtain pricing and licensing information for Microsoft Project, and to look into other alternatives to determine whether the other programs could exchange data with Microsoft Project.

Project Management Training

Ann suggested that the team compare the pros and cons of in-house training versus vendor-supplied training options. Again, Ann suggested the Microsoft Web site as a good starting point to evaluate third-party solutions. Using information on the site, the team was able to identify three training providers. After contacting these firms, the team had some realistic time and cost estimates for outside training solutions.

Ann suggested that the team should also consider a train-the-trainer approach where she would instruct an initial group from all SWL departments, and the training team would then provide training sessions within their respective departments. Meanwhile, Mike Feiner wondered whether any current SWL employees had listed project management experience and skills in their applications or résumés.

SWL Team Tasks

1. Using the material in this chapter and other reference material if necessary, develop a set of answers to the 10 questions that Ann presented to the team.
2. Suppose that Ann asked you to create an outline for her two-hour demo session. You can use Microsoft Project if it is available to you, or you can download a free demo version from the Microsoft Web site. In your outline, try to mention the basic information that a user would need to get started with a simple project.
3. Visit the Web site for Open Workbench and write a description of the product. Try to include as many features as possible, and list the pros and cons of the program. Determine whether the program can exchange information with Microsoft Project, and whether any special techniques are necessary to accomplish the transfer.
4. Microsoft has launched MPUG, which stands for Microsoft Project User Group. MPUG's stated mission is to deliver Microsoft Office Project content, resources, opportunities, and community networking worldwide. Explore the site at mpug.com and note the various levels of membership. Should SWL encourage IT staff members to join this group? Write up a recommendation with your reasons.

Manage the SWL Project

You have been asked to manage SWL's new information system project. One of your most important activities will be to identify project tasks and determine when they will be performed. Before you begin, you should review the SWL case in this chapter. Then list and analyze the tasks, as follows:

LIST THE TASKS Start by listing and numbering at least 10 tasks that the SWL team needs to perform to fulfill the objectives of this chapter. Your list can include SWL Team Tasks and any other tasks that are described in this chapter. For example, Task 3 might be to Identify the project tasks and Task 6 might be to Analyze task relationships.

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

ANALYZE THE TASKS Now study the tasks to determine the order in which they should be performed. First identify all concurrent tasks, which are not dependent on other tasks. In the example shown in Figure 3-37, Tasks 1, 2, 3, 4, and 5 are concurrent tasks, and could begin at the same time if resources are available.

Other tasks are dependent tasks, because they cannot be performed until one or more earlier tasks have been completed. For each dependent task, you must identify specific tasks that need to be completed before these tasks can begin. For example, you need to identify the project tasks before you can analyze the task relationships, so Task 6 cannot begin until Task 3 is completed, as Figure 3-37 shows.

This chapter describes project management tools, techniques, and software. To learn more, you can use the Features section on your Student Study Tool CD-ROM, or visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and locate the project management resources library for this book. On the Web, Microsoft offers demo versions, training, and tips for using Project 2010. You also can visit the OpenWorkbench.org site to learn more about this free, open-source software.

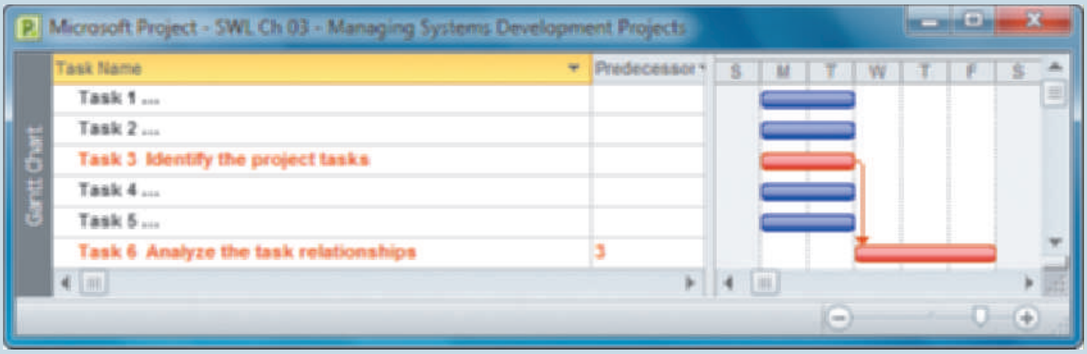


FIGURE 3-37 Tasks 1, 2, 3, 4, and 5 are concurrent tasks that could be performed at the same time. Task 6 is a dependent task that cannot be performed until Task 3 has been completed.

Ready for a Challenge?

In additional to technical skills, IT professionals need critical thinking skills such as perception, organization, analysis, problem-solving, and decision-making. The Ready for a Challenge feature can help you learn, practice, and apply critical thinking skills that you can take to the workplace.

The IT group at Game Technology has received approval to add a new entry-level systems analyst position. You will help set up the hiring timetable and tasks. After speaking to several people in the department, you identified seven tasks and the order in which they should be performed.

Using your notes, you write up the following statement: First, review the applications (2 days). Then start three tasks: notify the IT team (2 days), reserve a conference room (1 day), and set up an interview schedule (5 days). When the team has been notified and the conference room reserved, prepare a set of interview questions (2 days). When the questions are ready and the interview schedule has been developed, conduct the interviews (5 days). After the interviews are done, meet with the hiring committee to select a candidate (1 day).

Practice Tasks

A. Prepare a work breakdown structure showing tasks, durations, and predecessor tasks.
B. Calculate the start and finish date for each task, and determine the critical path.

After you complete the Practice Tasks, to check your work and view sample answers, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to the resources for this chapter, and locate Ready for a Challenge?.

The Challenge

The hiring committee made a job offer to Sharon Adams, and she accepted. She will start her new job in three weeks. As IT department intern, you were asked to set up an orientation for her, including meetings with human resources, users, and the IT team. She also would go on a tour of the company, and confirm a security clearance. After thinking about this, you come up with a suggested orientation schedule, as follows:

First, Sharon would meet with human resources to learn about company benefits and policies (3 hours). When that meeting is over, she could handle three tasks at once: she could tour the company offices (4 hours), meet with users (3 hours), and work on the necessary paperwork (2 hours). After the tour and user meetings, she could meet with the IT managers (1 hour). Then, after meeting the IT managers and with all the paperwork completed, she would meet with the entire IT team for a brief introduction (1 hour).

Challenge Tasks

A. Prepare a work breakdown structure showing tasks, durations, and predecessor tasks.
B. Calculate the start and finish date for each task, and determine the critical path.

PHASE 2 SYSTEMS ANALYSIS

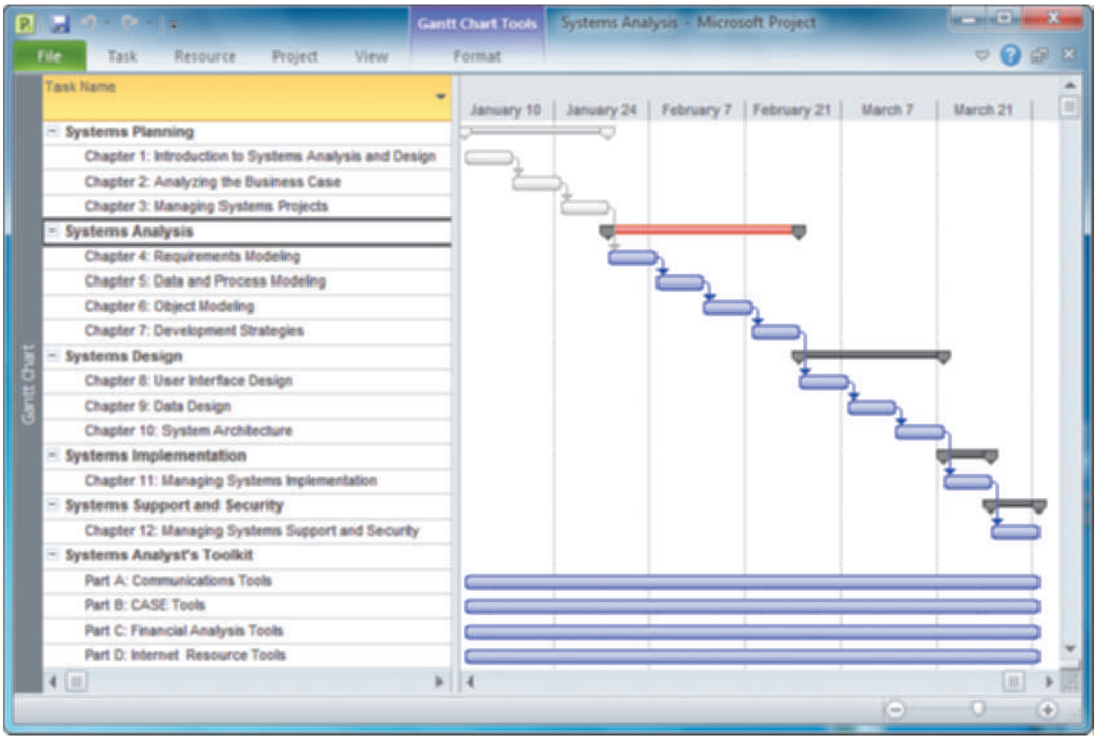
DELIVERABLE
System requirements document

TOOLKIT SUPPORT
Communication, CASE, and financial analysis tools

VIDEO LEARNING SESSIONS
Functional Decomposition Diagrams, DFDs, Decision Tables, and Object Models

As the Dilbert cartoon suggests, a successful project manager must determine the requirements before starting the design process, not the other way around. You will learn more about fact-finding and modeling system requirements in the systems analysis phase.

Systems analysis is the second of five phases in the systems development life cycle. In the previous phase, systems planning, you conducted a preliminary investigation to determine the project's feasibility. Now you will use requirements modeling, data and process modeling, and object modeling techniques to represent the new system. You also will consider various development strategies for the new system, and plan for the transition to systems design tasks. The deliverable for this phase is the system requirements document.



CHAPTER 4 Requirements Modeling

Chapter 4 is the first of four chapters in the systems analysis phase. This chapter describes the process of gathering facts about a systems project, preparing documentation, and creating models that will be used to design and develop the system.

INTRODUCTION

OBJECTIVES

When you finish this chapter, you will be able to:

- Describe systems analysis phase activities
- Explain joint application development (JAD), rapid application development (RAD), and agile methods
- Use a functional decomposition diagram (FDD) to model business functions and processes
- Describe the Unified Modeling Language (UML) and examples of UML diagrams
- List and describe system requirements, including outputs, inputs, processes, performance, and controls
- Explain the concept of scalability
- Use fact-finding techniques, including interviews, documentation review, observation, questionnaires, sampling, and research
- Define total cost of ownership (TCO)
- Conduct a successful interview
- Develop effective documentation methods to use during systems development

After an overview of the systems analysis phase, this chapter describes requirements modeling techniques and team-based methods that systems analysts use to visualize and document new systems. The chapter then discusses system requirements and fact-finding techniques, which include interviewing, documentation review, observation, surveys and questionnaires, sampling, and research.

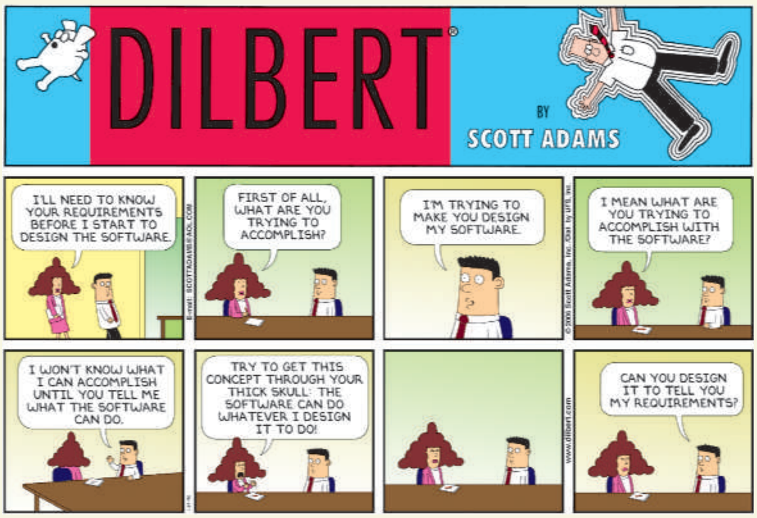
Chapter 4 includes a Video Learning Session that shows you how to use a functional decomposition diagram (FDD) to model business functions and processes.

CHAPTER INTRODUCTION CASE: Mountain View College Bookstore

Background: Wendy Lee, manager of college services at Mountain View College, wants a new information system that will improve efficiency and customer service at the three college bookstores.

In this part of the case, Tina Allen (systems analyst) and David Conroe (student intern) are talking about requirements modeling tasks and concepts.

Participants: Tina and David
Location: Tina's office, Monday morning, October 3, 2011
Project status: The project has advanced to the systems analysis phase. Now, Tina and David will work on modeling, fact-finding, and the documentation they need to build a requirements model for the proposed bookstore information system.
Discussion topics: Modeling, team-based development strategies, fact-finding techniques, and documentation



DILBERT: © Scott Adams/Dist. by United Feature Syndicate, Inc.

Tina: Before I tell you about the project, look at this Dilbert cartoon. You'll like it!

David: *It's funny, but scary too. Hope it doesn't apply to us!*

Tina: Me too. That's why we have to do a good job of requirements modeling.

David: *So, what do we do next?*

Tina: We need to create a model of the new system. We call this a requirements model, because it will include all the outputs, inputs, processes, and controls for the new system. The model will consist of various diagrams, charts, and documentation.

David: *How will we use the model when we're done?*

Tina: We'll study it carefully and review it frequently with system users.

David: *Who are the users?*

Tina: Users might include bookstore staff, students, faculty members, and the college business office. External users might include textbook publishers and suppliers of bookstore merchandise. The main thing is to work with users every step of the way. We'll perform fact-finding, and we'll document everything carefully. Here's a task list to get us started:

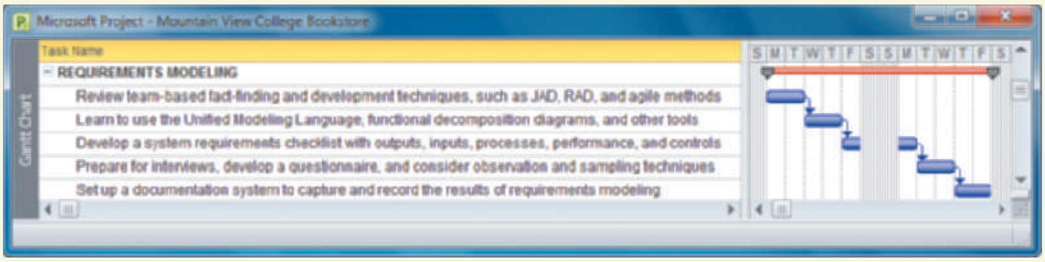


FIGURE 4-1 Typical requirements modeling task list.

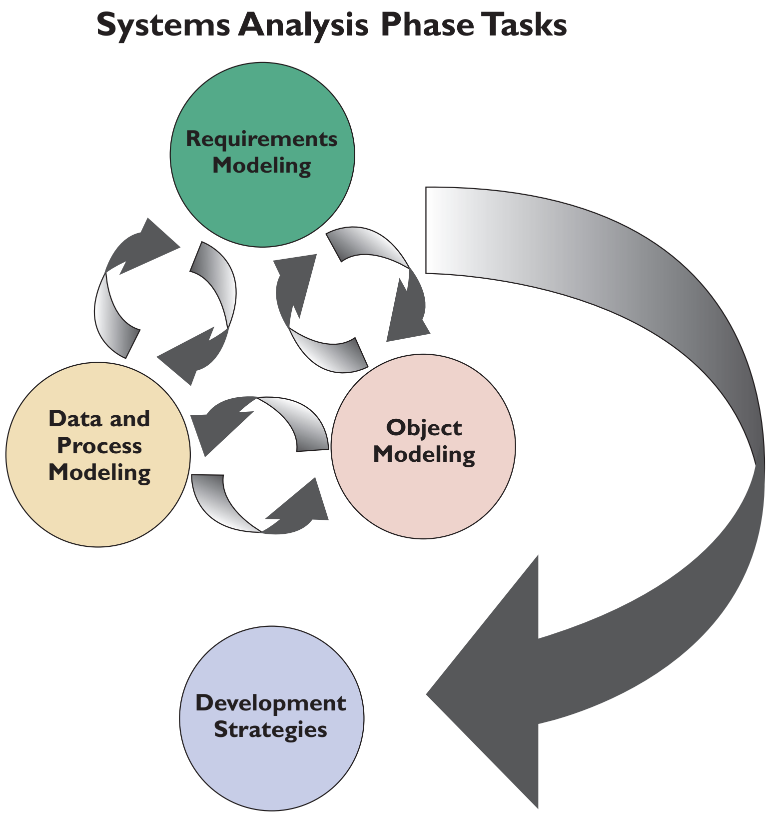


FIGURE 4-2 The systems analysis phase consists of requirements modeling, data and process modeling, object modeling, and consideration of development strategies. Notice that the systems analysis tasks are interactive, even though the waterfall model generally depicts sequential development.

SYSTEMS ANALYSIS PHASE OVERVIEW

The overall objective of the systems analysis phase is to understand the proposed project, ensure that it will support business requirements, and build a solid foundation for system development. In this phase, you use models and other documentation tools to visualize and describe the proposed system.

Systems Analysis Activities

The systems analysis phase includes the four main activities shown in Figure 4-2: requirements modeling, data and process modeling, object modeling, and consideration of development strategies.

Although the waterfall model shows sequential SDLC phases, it is not uncommon for several phases (or certain tasks within a phase) to interact during the development process, just as they would in an adaptive model. For example, this occurs whenever new facts are learned or system requirements change during the modeling process. Figure 4-2 shows typical interaction among the three modeling tasks: requirements modeling, data and process modeling, and object modeling.

VIDEO LEARNING SESSIONS

To learn more about data flow diagrams, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and navigate to the **Video Learning Sessions** for this book. These sessions can help you understand key concepts, practice your skills, and check your work.

REQUIREMENTS MODELING This chapter describes **requirements modeling**, which involves fact-finding to describe the current system and identification of the requirements for the new system, such as outputs, inputs, processes, performance, and security. **Outputs** refer to electronic or printed information produced by the system. **Inputs** refer to necessary data that enters the system, either manually or in an automated manner. **Processes** refer to the logical rules that are applied to transform the data into meaningful information. **Performance** refers to system characteristics such as speed, volume, capacity, availability, and reliability. **Security** refers to hardware, software, and procedural controls that safeguard and protect the system and its data from internal or external threats.

DATA AND PROCESS MODELING In Chapter 5, Data and Process Modeling, you will continue the modeling process by learning how to represent graphically system data and processes using traditional structured analysis techniques. As you learned in Chapter 1, structured analysis identifies the data flowing into a process, the business rules that transform the data, and the resulting output data flow.

OBJECT MODELING Chapter 6 discusses object modeling, which is another popular modeling technique. While structured analysis treats processes and data as separate components, object-oriented analysis (O-O) combines data and the processes that act on the data into things called objects. These objects represent actual people, things, transactions, and events that affect the system. During the system development process, analysts often use both modeling methods to gain as much information as possible.

DEVELOPMENT STRATEGIES In Chapter 7, Development Strategies, you will consider various development options and prepare for the transition to the systems design phase of the SDLC. You will learn about software trends, acquisition and development alternatives, outsourcing, and formally documenting requirements for the new system.

The deliverable, or end product, of the systems analysis phase is a **system requirements document**, which is an overall design for the new system. In addition, each activity within the systems analysis phase has an end product and one or more milestones. As you learned in Chapter 3, project managers use various tools and techniques to coordinate people, tasks, timetables, and budgets.

VIDEO LEARNING SESSIONS

To learn more about obect modeling, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and navigate to the **Video Learning Sessions** for this book. These sessions can help you understand key concepts, practice your skills, and check your work.

Systems Analysis Skills

You will need strong analytical and interpersonal skills to build an accurate model of the new system. **Analytical skills** enable you to identify a problem, evaluate the key elements, and develop a useful solution. **Interpersonal skills** are especially valuable to a systems analyst who must work with people at all organizational levels, balance conflicting needs of users, and communicate effectively.

Because information systems affect people throughout the company, you should consider team-oriented strategies as you begin the systems analysis phase.

ON THE WEB

To learn more about interpersonal skills, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Interpersonal Skills link.

Team-Based Techniques: JAD, RAD, and Agile Methods

The IT department's goal is to deliver the best possible information system, at the lowest possible cost, in the shortest possible time. To achieve the best results, system developers view users as partners in the development process. Greater user involvement usually results in better communication, faster development times, and more satisfied users.

The traditional model for systems development was an IT department that used structured analysis and consulted users only when their input or approval was needed. Although the IT staff still has a central role, and structured analysis remains a popular method of systems development, most IT managers invite system users to participate actively in various development tasks.

As you learned in Chapter 1, team-based approaches have been around for some time. A popular example is **joint application development** (**JAD**), which is a user-oriented technique for fact-finding and requirements modeling. Because it is not linked to a specific development methodology, systems developers use JAD whenever group input and interaction are desired.

Another popular user-oriented method is **rapid application development** (**RAD**). RAD resembles a condensed version of the entire SDLC, with users involved every step of the way. While JAD typically focuses only on fact-finding and requirements determination, RAD provides a fast-track approach to a full spectrum of system development tasks, including planning, design, construction, and implementation.

Finally, as you learned in Chapter 1, **agile methods** represent a recent trend that stresses intense interaction between system developers and users. JAD, RAD, and agile methods are discussed in the following sections.

JOINT APPLICATION DEVELOPMENT

Joint application development (JAD) is a popular fact-finding technique that brings users into the development process as active participants.

User Involvement

ON THE WEB

To learn more about JAD, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the JAD link.

Users have a vital stake in an information system, and they should participate fully in the development process. Until recent years, the IT department usually had sole responsibility for systems development, and users had a relatively passive role. During the development process, the IT staff would collect information from users, define system requirements, and construct the new system. At various stages of the process, the IT staff might ask users to review the design, offer comments, and submit changes.

Today, users typically have a much more active role in systems development. IT professionals now recognize that successful systems must be user-oriented, and users need to be involved, formally or informally, at every stage of system development.

One popular strategy for user involvement is a JAD team approach, which involves a task force of users, managers, and IT professionals that works together to gather information, discuss business needs, and define the new system requirements.

JAD Participants and Roles

A JAD team usually meets over a period of days or weeks in a special conference room or at an off-site location. Either way, JAD participants should be insulated from the distraction of day-to-day operations. The objective is to analyze the existing system, obtain user input and expectations, and document user requirements for the new system.

The JAD group usually has a project leader, who needs strong interpersonal and organizational skills, and one or more members who document and record the results and decisions. Figure 4-3 describes typical JAD participants and their roles. IT staff members often serve as JAD project leaders, but that is not always the case. Systems analysts on the JAD team participate in discussions, ask questions, take notes, and provide support to the team. If CASE tools are available, analysts can develop models and enter documentation from the JAD session directly into the CASE tool.

A typical JAD session agenda is shown in Figure 4-4. The JAD process involves intensive effort by all team members. Because of the wide range of input and constant interaction among the participants, many companies believe that a JAD group produces the best possible definition of the new system.

JAD PARTICIPANT	ROLE
JAD project leader	Develops an agenda, acts as a facilitator, and leads the JAD session
Top management	Provides enterprise-level authorization and support for the project
Managers	Provide department-level support for the project and understanding of how the project must support business functions and requirements
Users	Provide operational-level input on current operations, desired changes, input and output requirements, user interface issues, and how the project will support day-to-day tasks
Systems analysts and other IT staff members	Provide technical assistance and resources for JAD team members on issues such as security, backup, hardware, software, and network capability
Recorder	Documents results of JAD sessions and works with systems analysts to build system models and develop CASE tool documentation

FIGURE 4-3 Typical JAD participants and roles.

Project leader	• Introduce all JAD team members • Discuss ground rules, goals, and objectives for the JAD sessions • Explain methods of documentation and use of CASE tools, if any
Top management (sometimes called the project owner or sponsor)	• Explain the reason for the project and express top management authorization and support
Project leader	• Provide overview of the current system and proposed project scope and constraints • Present outline of specific topics and issues to be investigated
Open discussion session, moderated by project leader	• Review the main business processes, tasks, user roles, input, and output • Identify specific areas of agreement or disagreement • Break team into smaller groups to study specific issues and assign group leaders
JAD team members working in smaller group sessions, supported by IT staff	• Discuss and document all system requirements • Develop models and prototypes
Group leaders	• Report on results and assigned tasks and topics • Present issues that should be addressed by the overall JAD team
Open discussion session, moderated by project leader	• Review reports from small group sessions • Reach consensus on main issues • Document all topics
Project leader	• Present overall recap of JAD session • Prepare report that will be sent to JAD team members

FIGURE 4-4 Typical agenda for a JAD session.

JAD Advantages and Disadvantages

Compared with traditional methods, JAD is more expensive and can be cumbersome if the group is too large relative to the size of the project. Many companies find, however, that JAD allows key users to participate effectively in the requirements modeling process. When users participate in the systems development process, they are more likely to feel a sense of ownership in the results, and support for the new system. When properly used, JAD can result in a more accurate statement of system requirements, a better understanding of common goals, and a stronger commitment to the success of the new system.

Rapid Application Development

Rapid application development (RAD) is a team-based technique that speeds up information systems development and produces a functioning information system. Like JAD, RAD uses a group approach, but goes much further. While the end product of JAD is a requirements model, the end product of RAD is the new information system. RAD is a complete methodology, with a four-phase life cycle that parallels the traditional SDLC phases. Companies use RAD to reduce cost and development time, and increase the probability of success.

RAD relies heavily on prototyping and user involvement. The RAD process allows users to examine a working model as early as possible, determine if it meets their needs, and suggest necessary changes. Based on user input, the prototype is modified and the interactive process continues until the system is completely developed and users are satisfied. The project team uses CASE tools to build the prototypes and create a continuous stream of documentation.

RAD Phases and Activities

The RAD model consists of four phases: requirements planning, user design, construction, and cutover, as shown in Figure 4-5. Notice the continuous interaction between the user design and construction phases.

ON THE WEB

To learn more about RAD, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the RAD link.

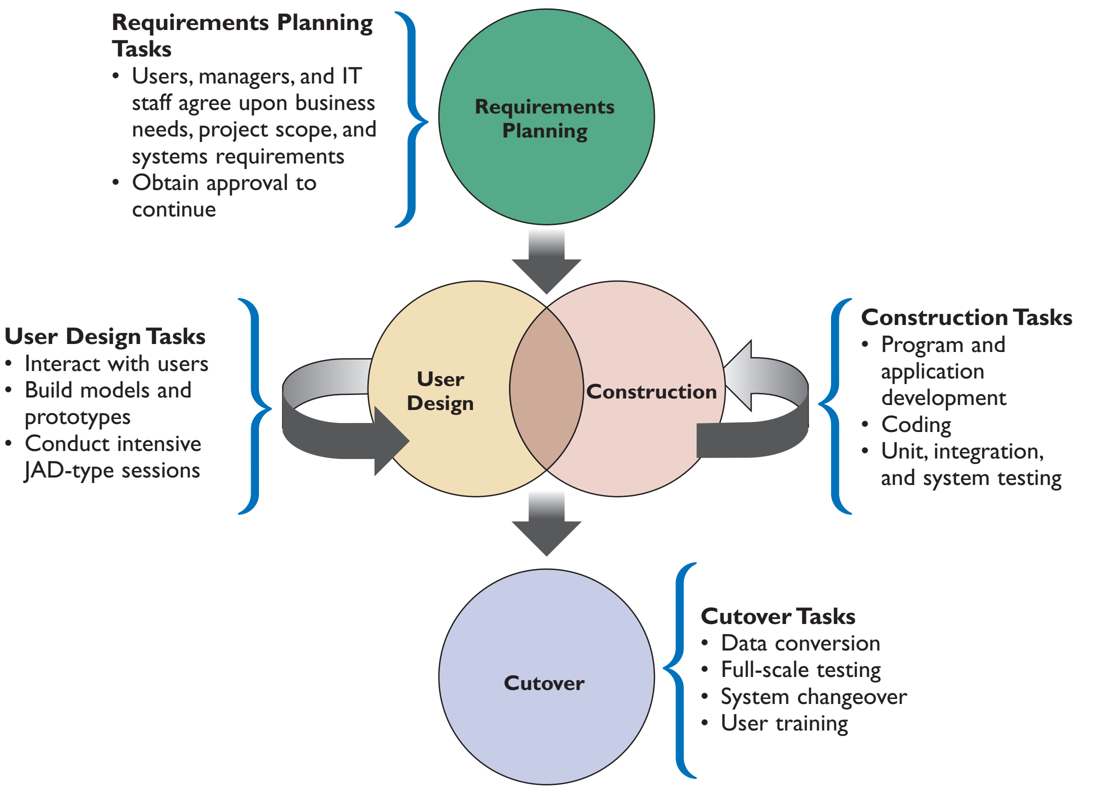


FIGURE 4-5 The four phases of the RAD model are requirements planning, user design, construction, and cutover. Notice the continuous interaction between the user design and construction phases.

REQUIREMENTS PLANNING The **requirements planning phase** combines elements of the systems planning and systems analysis phases of the SDLC. Users, managers, and IT staff members discuss and agree on business needs, project scope, constraints, and system requirements. The requirements planning phase ends when the team agrees on the key issues and obtains management authorization to continue.

USER DESIGN During the **user design phase**, users interact with systems analysts and develop models and prototypes that represent all system processes, outputs, and inputs. The RAD group or subgroups typically use a combination of JAD techniques and CASE tools to translate user needs into working models. User design is a continuous, interactive process that allows users to understand, modify, and eventually approve a working model of the system that meets their needs.

CONSTRUCTION The **construction phase** focuses on program and application development tasks similar to the SDLC. In RAD, however, users continue to participate and still can suggest changes or improvements as actual screens or reports are developed.

CUTOVER The **cutover phase** resembles the final tasks in the SDLC implementation phase, including data conversion, testing, changeover to the new system, and user training. Compared with traditional methods, the entire process is compressed. As a result, the new system is built, delivered, and placed in operation much sooner.

RAD Objectives

The main objective of all RAD approaches is to cut development time and expense by involving users in every phase of systems development. Because it is a continuous process, RAD allows the development team to make necessary modifications quickly, as the design evolves. In times of tight corporate budgets, it is especially important to limit the cost of changes that typically occur in a long, drawn-out development schedule.

In addition to user involvement, a successful RAD team must have IT resources, skills, and management support. Because it is a dynamic, user-driven process, RAD is especially valuable when a company needs an information system to support a new business function. By obtaining user input from the beginning, RAD also helps a development team design a system that requires a highly interactive or complex user interface.

RAD Advantages and Disadvantages

RAD has advantages and disadvantages compared with traditional structured analysis methods. The primary advantage is that systems can be developed more quickly with significant cost savings. A disadvantage is that RAD stresses the mechanics of the system itself and does not emphasize the company's strategic business needs. The risk is that a system might work well in the short term, but the corporate and long-term objectives for the system might not be met. Another potential disadvantage is that the accelerated time cycle might allow less time to develop quality, consistency, and design standards. RAD can be an attractive alternative, however, if an organization understands the possible risks.

AGILE METHODS

In Chapter 1, you learned that agile methods attempt to develop a system incrementally, by building a series of prototypes and constantly adjusting them to user requirements. As the agile process continues, developers revise, extend, and merge earlier versions into the final product. An agile approach emphasizes continuous feedback, and each incremental step is affected by what was learned in the prior steps.

As agile methods become more popular, a large community of agile-related software and services has evolved. For example, Visual Paradigm offers Agilian, which includes a set of agile modeling tools, as shown in Figure 4-6 on the next page. The Agilian modeling toolset includes support for many modeling tools, such as the Unified Modeling Language, entity-relationship diagrams, data flow diagrams, and business process modeling, among others.

FIGURE 4-6 Visual Paradigm's Agilian includes many types of agile modeling tools.

> **ON THE WEB**
>
> To learn more about agile methods, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Agile Methods link.

Some agile developers prefer not to use CASE tools at all, and rely instead on whiteboard displays and arrangements of movable sticky notes. This approach, they believe, reinforces the agile strategy: simple, rapid, flexible, and user-oriented.

Scrum is another agile approach. The name is derived from the rugby term *scrum* (Figure 4-7), where team members prepare to lunge at each other to achieve their objectives. The systems development version of Scrum involves the same intense interaction, though more mental than physical. In a Scrum session, agile team members play specific roles, including colorful designations as *pigs* or *chickens*. These roles are based on the old joke about the pig and chicken who discuss a restaurant where ham and eggs would be served. However, the pig declines, because that role would require a total commitment, while for the chicken, it would only be a contribution.

In the agile world, the *pigs* include the product owner, the facilitator, and the development team; while the *chickens* include users, other stakeholders, and managers. Scrum sessions have specific guidelines that emphasize time blocks, interaction, and team-based activities that result in deliverable software.

FIGURE 4-7 In a rugby scrum, team members prepare to lunge at each other to achieve their objectives.

Agile Method Advantages and Disadvantages

Agile, or adaptive, methods are very flexible and efficient in dealing with change. They are popular because they stress team interaction and reflect a set of community-based values. Also, frequent deliverables constantly validate the project and reduce risk.

However, some potential problems exist. For example, team members need a high level of technical and interpersonal skills. Also, a lack of structure and documentation can introduce risk factors. Finally, the overall project may be subject to significant change in scope as user requirements continue to evolve during the project.

CASE IN POINT 4.1: North Hills College

North Hills College has decided to implement a new registration system that will allow students to register online, as well as in person. As IT manager, you decide to set up a JAD session to help define the requirements for the new system. The North Hills organization is fairly typical, with administrative staff that includes a registrar, a student support and services team, a business office, an IT group, and a number of academic departments. Using this information, you start work on a plan to carry out the JAD session. Who would you invite to the session, and why? What would be your agenda for the session, and what would take place at each stage of the session?

Modeling Tools and Techniques

Models help users, managers, and IT professionals understand the design of a system. Modeling involves graphical methods and nontechnical language that represent the system at various stages of development. During requirements modeling, you can use various tools to describe business processes, requirements, and user interaction with the system.

In Chapter 1, you learned about CASE tools that offer powerful modeling features. CASE tool modeling is discussed in detail in Part B of the Systems Analyst's Toolkit.

Systems analysts use modeling and fact-finding interactively — first they build fact-finding results into models, then they study the models to determine whether additional fact-finding is needed. To help them understand system requirements, analysts use functional decomposition diagrams, business process models, data flow diagrams, and Unified Modeling Language diagrams. Any of these diagrams can be created with CASE tools or standalone drawing tools if desired.

VIDEO LEARNING SESSION: FUNCTIONAL DECOMPOSITION DIAGRAMS (FDDs)

Video Learning Sessions can help you understand key concepts, practice your skills, and check your work. To access the sessions, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and navigate to the **Video Learning Sessions** for this book. This session is about functional decomposition diagrams (FDDs). You'll learn about FDDs and why they are important, how to use FDDs to model business functions and processes, and how to use CASE tools to create FDDs.

Functional Decomposition Diagrams

A **functional decomposition diagram (FDD)** is a top-down representation of a function or process. Using an FDD, an analyst can show business functions and break them down into lower-level functions and processes. Creating an FDD is similar to drawing an organization chart — you start at the top and work your way down. Figure 4-8 shows an FDD of a library system drawn with the Visible Analyst CASE tool. FDDs can be used at several stages of systems development. During requirements modeling, analysts use FDDs to model business functions and show how they are organized into lower-level processes. Those processes translate into program modules during application development.

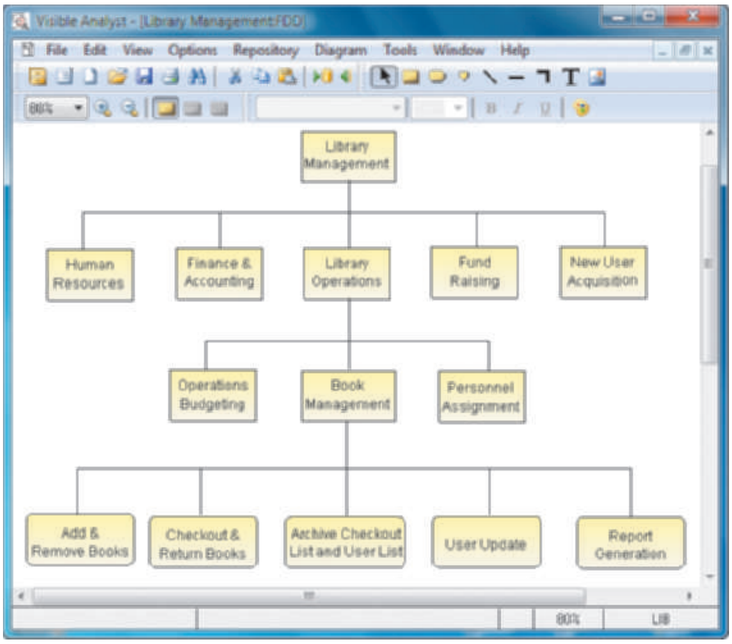


FIGURE 4-8 FDD showing five top-level functions. The Library Operations function includes two additional levels that show processes and subprocesses.

TOOLKIT TIME

The CASE tools in Part B of the Systems Analyst's Toolkit can help you document business functions and processes, develop graphical models, and provide an overall framework for information system development. To learn more about these tools, turn to Part B of the four-part Toolkit that follows Chapter 12.

Business Process Modeling

As you learned in Chapter 1, a **business process model (BPM)** describes one or more business processes, such as handling an airline reservation, filling a product order, or updating a customer account. During requirements modeling, analysts often create models that use a standard language called **business process modeling notation (BPMN)**. BPMN includes various shapes and symbols to represent events, processes, and workflows.

When you create a business process model using a CASE tool such as Visible Analyst, your diagram automatically becomes part of the overall model. In the example shown in Figure 4-9, using BPMN terminology, the overall diagram is called a **pool**, and the designated customer areas are called **swim lanes**. Integrating BPM into the CASE development process leads to faster results, fewer errors, and reduced cost. Part B of the Systems Analyst's Toolkit describes business process modeling in more detail.

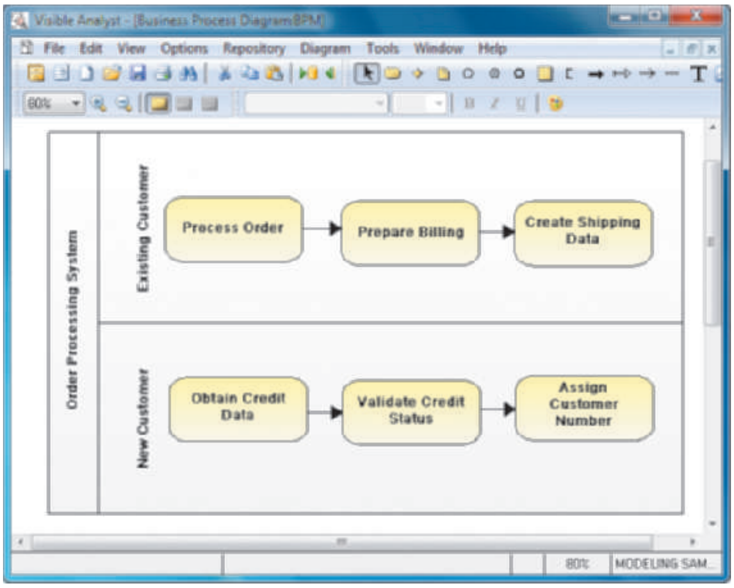


FIGURE 4-9 Using the Visible Analyst CASE tool, an analyst can create a business process diagram. The overall diagram is called a pool, and the two separate customer areas are called swim lanes.

Data Flow Diagrams

Working from a functional decomposition diagram, analysts can create **data flow diagrams (DFDs)** to show how the system stores, processes, and transforms data. The DFD in Figure 4-10 describes adding and removing books, which is a function shown in the Library Management diagram in Figure 4-8. Notice that the two shapes in the DFD represent processes, each with various inputs and outputs. Additional levels of information and detail are depicted in other, related DFDs. Data and process modeling is described in detail in Chapter 5.

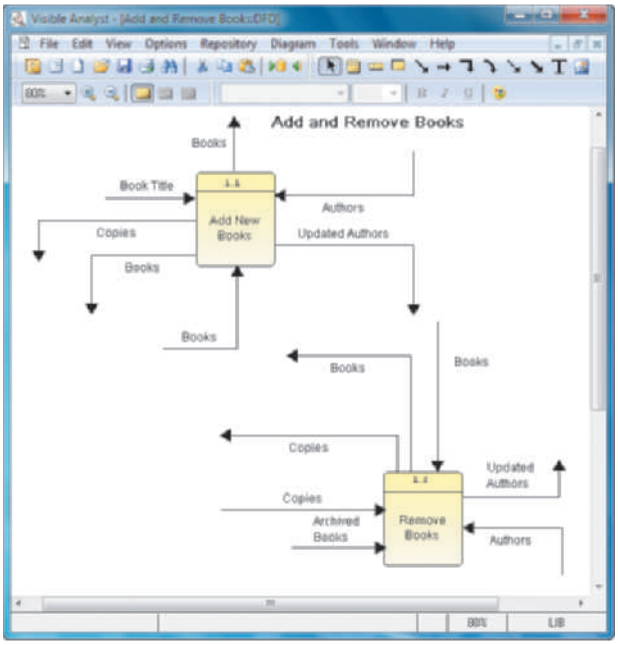


FIGURE 4-10 A library system DFD shows how books are added and removed.

Unified Modeling Language

The **Unified Modeling Language (UML)** is a widely used method of visualizing and documenting software systems design. UML uses object-oriented design concepts, but it is independent of any specific programming language and can be used to describe business processes and requirements generally.

UML provides various graphical tools, such as use case diagrams and sequence diagrams. During requirements modeling, a systems analyst can utilize the UML to represent the information system from a user's viewpoint. Use case diagrams, sequence diagrams, and other UML concepts are discussed in more detail in Chapter 6, along with other object-oriented analysis concepts. A brief description of each technique follows.

USE CASE DIAGRAMS During requirements modeling, systems analysts and users work together to document requirements and model system functions. A **use case diagram** visually represents the interaction between users and the information system. In a use case diagram, the user becomes an **actor**, with a specific role that describes how he or she interacts with the system. Systems analysts can draw use case diagrams freehand or use CASE tools that integrate the use cases into the overall system design.

Figure 4-11 shows a simple use case diagram for a sales system where the actor is a customer and the use case involves a credit card validation that is performed by the system. Because use cases depict the system through the eyes of a user, common business

ON THE WEB

To learn more about the Unified Modeling Language, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate The Unified Modeling Language link.

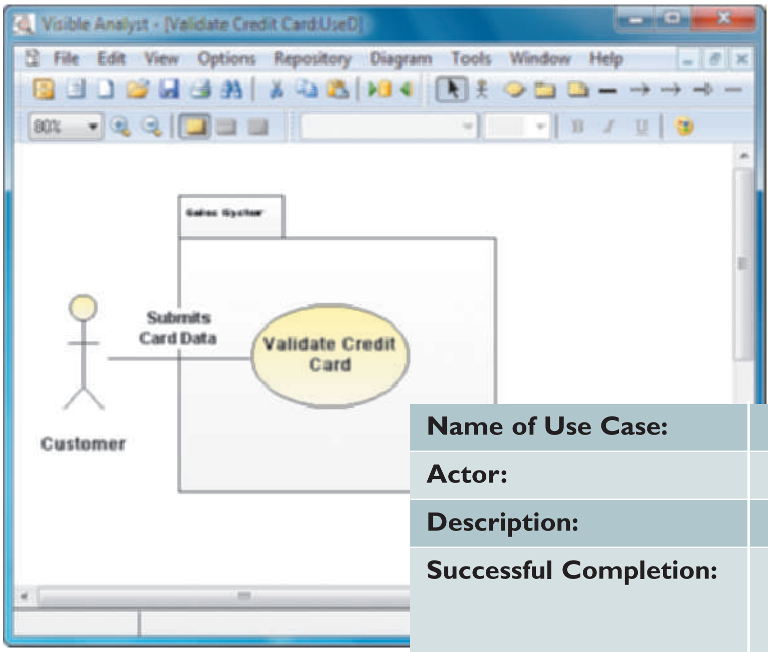


FIGURE 4-11 Use case diagram of a sales system, where the actor is a customer and the use case involves a credit card validation.

Name of Use Case:	Credit card validation process
Actor:	Customer
Description:	Describes the credit card validation process
Successful Completion:	1. Customer clicks the input selector and enters credit card number and expiration date 2. System verifies card 3. System sends authorization message
Alternative:	1. Customer clicks the input selector and enters credit card number and expiration date 2. System rejects card 3. System sends rejection message
Precondition:	Customer has selected at least one item and has proceeded to checkout area
Postcondition:	Credit card information has been validated Customer can continue with order
Assumptions:	None

FIGURE 4-12 A table documents the credit card validation use case shown in Figure 4-11.

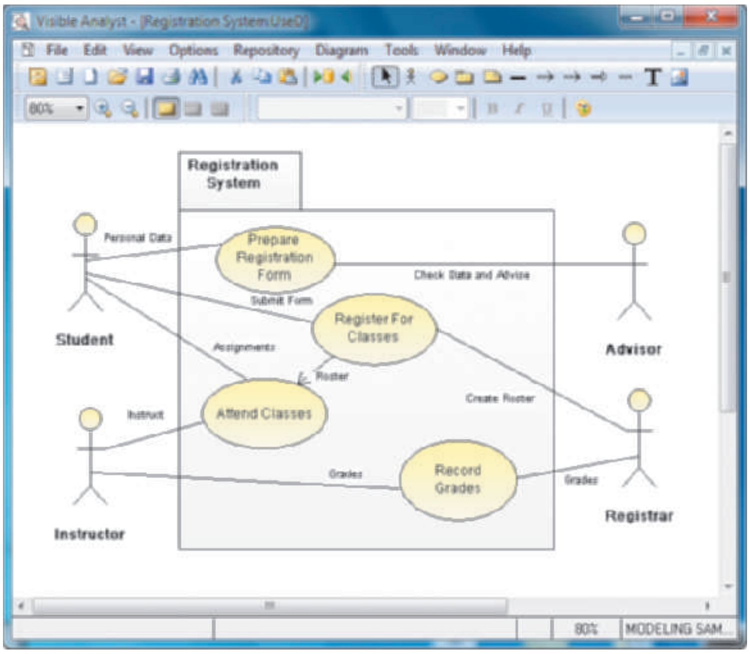


FIGURE 4-13 Use case diagram of a student records system.

language can be used to describe the transactions. For example, Figure 4-12 shows a table that documents the credit card validation use case, and Figure 4-13 shows a student records system, with several use cases and actors.

SEQUENCE DIAGRAMS A **sequence diagram** shows the timing of interactions between objects as they occur. A systems analyst might use a sequence diagram to show all possible outcomes, or focus on a single scenario. Figure 4-14 shows a simple sequence diagram of a successful credit card validation. The interaction proceeds from top to bottom along a vertical timeline, while the horizontal arrows represent messages from one object to another.

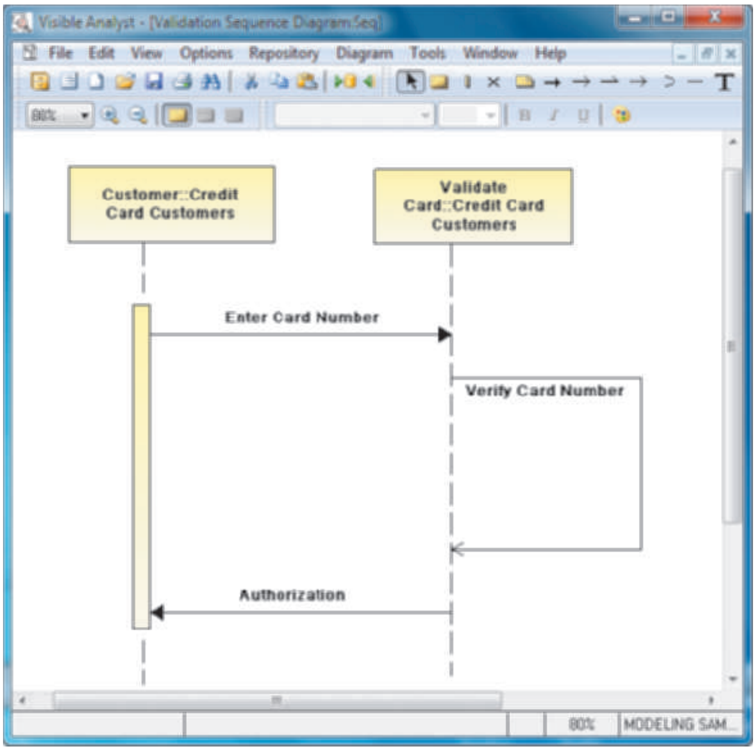


FIGURE 4-14 Sequence diagram showing a credit card validation process.

SYSTEM REQUIREMENTS CHECKLIST

During requirements modeling, systems developers must identify and describe all system requirements. A **system requirement** is a characteristic or feature that must be included in an information system to satisfy business requirements and be acceptable to users. System requirements serve as benchmarks to measure the overall acceptability of the finished system.

System requirements fall into five general categories: outputs, inputs, processes, performance, and controls. Typical examples of system requirements for each category are listed below.

Output Examples

- ✓ The Web site must report online volume statistics every four hours, and hourly during peak periods.
- ✓ The inventory system must produce a daily report showing the part number, description, quantity on hand, quantity allocated, quantity available, and unit cost of all sorted by part number.
- ✓ The contact management system must generate a daily reminder list for all sales reps.
- ✓ The purchasing system must provide suppliers with up-to-date specifications.
- ✓ The sales tracking system must produce a daily fast-moving-item report, listing all products that exceed the forecasted sales volume grouped by style, color, size, and reorder status.
- ✓ The customer analysis system must produce a quarterly report that identifies changes in ordering patterns or trends with statistical comparisons to the previous four quarters.

Input Examples

- ✓ Manufacturing employees must swipe their ID cards into online data collection terminals that record labor costs and calculate production efficiency.
- ✓ The department head must enter overtime hours on a separate screen.
- ✓ Student grades must be entered on machine-scannable forms prepared by the instructor.
- ✓ Each input form must include date, time, product code, customer number, and quantity.
- ✓ Data entry screens must be uniform, except for background color, which can be changed by the user.
- ✓ A data entry person at the medical group must input patient services into the billing system.

Process Examples

- ✓ The student records system must calculate the GPA at the end of each semester.
- ✓ As the final step in year-end processing, the payroll system must update employee salaries, bonuses, and benefits and produce tax data required by the IRS.
- ✓ The warehouse distribution system must analyze daily orders and create a routing pattern for delivery trucks that maximizes efficiency and reduces unnecessary mileage.
- ✓ The human resources system must interface properly with the existing payroll system.
- ✓ The video rental system must not execute new rental transactions for customers who have overdue videos.
- ✓ The prescription system must automatically generate an insurance claim form.

Performance Examples

- ✓ The system must support 25 users online simultaneously.
- ✓ Response time must not exceed four seconds.
- ✓ The system must be operational seven days a week, 365 days a year.
- ✓ The accounts receivable system must prepare customer statements by the third business day of the following month.
- ✓ The student records system must produce class lists within five hours after the end of registration.
- ✓ The online inventory control system must flag all low-stock items within one hour after the quantity falls below a predetermined minimum.

Control Examples

- ✓ The system must provide logon security at the operating system level and at the application level.
- ✓ An employee record must be added, changed, or deleted only by a member of the human resources department.
- ✓ The system must maintain separate levels of security for users and the system administrator.

✓ All transactions must have audit trails.
✓ The manager of the sales department must approve orders that exceed a customer's credit limit.
✓ The system must create an error log file that includes the error type, description, and time.

Future Growth, Costs, and Benefits

In addition to the system requirements, systems analysts must consider scalability, which determines how a system will handle future growth and demands, and the total cost of ownership, which includes all future operational and support costs.

Scalability

Scalability refers to a system's ability to handle increased business volume and transactions in the future. Because it will have a longer useful life, a scalable system offers a better return on the initial investment.

To evaluate scalability, you need information about projected future volume for all outputs, inputs, and processes. For example, for a Web-based order processing system, you would need to know the maximum projected number of concurrent users, the periods of peak online activity, the number and types of data items required for each transaction, and the method of accessing and updating customer files.

Even to print customer statements, you need to know the number of active accounts and have a forecast for one, two, or five years, because that information affects future hardware decisions. In addition, with realistic volume projections, you can provide reliable cost estimates for related expenses, such as postage and online charges.

Similarly, to ensure that a Web-based hotel reservation system is sufficiently scalable, you would need to project activity levels for several years of operation. For example, you might forecast the frequency of online queries about room availability and estimate the time required for each query and the average response time. With that information, you could estimate server transaction volume and network requirements.

Transaction volume has a significant impact on operating costs. When volume exceeds a system's limitations, maintenance costs increase sharply. Volume can change dramatically if a company expands or enters a new line of business. For example, a new Internet-based marketing effort might require an additional server and 24-hour technical support.

Data storage also is an important scalability issue. You need to determine how much data storage is required currently and predict future needs based on system activity and growth. Those requirements affect hardware, software, and network bandwidth needed to maintain system performance. You also must consider data retention requirements and determine whether data can be deleted or archived on a specific timetable.

TOOLKIT TIME

The financial analysis tools in Part C of the Systems Analyst's Toolkit can help you analyze project costs, benefits, and economic feasibility. To learn more about these tools, turn to Part C of the four-part Toolkit that follows Chapter 12.

Total Cost of Ownership

In addition to direct costs, systems developers must identify and document indirect expenses that contribute to the **total cost of ownership** (**TCO**). TCO is especially important if the development team is assessing several alternatives. After considering the indirect costs, which are not always apparent, a system that seems inexpensive initially might actually turn out to be the most costly choice. One problem is that cost estimates tend to understate indirect costs such as user support and downtime productivity losses. Even if accurate figures are unavailable, systems analysts should try to identify indirect costs and include them in TCO estimates.

ON THE WEB

To learn more about REJ, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the REJ link.

Microsoft has developed a method for measuring total costs and benefits, called **Rapid Economic Justification (REJ)**, which is described in Figure 4-15. According to Microsoft, REJ is a framework to help IT professionals analyze and optimize IT investments. Notice that the primary emphasis is on business improvement, rather than operational efficiency. As the Web site points out, the strategic role of IT investments should be included, even when the specific benefits are difficult to quantify.

FIGURE 4-15 Microsoft developed Rapid Economic Justification (REJ) as a framework to help IT professionals analyze and optimize IT investments.

FACT-FINDING

Now that you understand the categories of system requirements, scalability, and TCO, the next step is to begin collecting information. Whether you are working on your own or as a member of a JAD team, during requirements modeling you will use various fact-finding techniques, including interviews, document review, observation, surveys and questionnaires, sampling, and research.

Fact-Finding Overview

Although software can help you to gather and analyze facts, no program actually performs fact-finding for you. First, you must identify the information you need. Typically, you begin by asking a series of questions, such as these:

- What business functions are supported by the current system?
- What strategic objectives and business requirements must be supported by the new system?
- What are the benefits and TCO of the proposed system?
- What transactions will the system process?
- What information do users and managers need from the system?

- Must the new system interface with legacy systems?
- What procedures could be eliminated by business process reengineering?
- What security issues exist?
- What risks are acceptable?
- What budget and timetable constraints will affect system development?

To obtain answers to these questions, you develop a fact-finding plan, which can involve another series of questions (*who*, *what*, *where, when*, and *how*), or use a more structured approach such as the Zachman Framework, which is explained in a following section. Either way, you will develop a strategy, carry out fact-finding techniques, document the results, and prepare a system requirements document, which is presented to management.

Who, What, Where, When, How, and Why?

Fact-finding involves answers to five familiar questions: *who*, *what*, *where, when*, and *how*. For each of those questions, you also must ask another very important question: *why*. Some examples of these questions are:

1. *Who*? Who performs each of the procedures within the system? Why? Are the correct people performing the activity? Could other people perform the tasks more effectively?
2. *What*? What is being done? What procedures are being followed? Why is that process necessary? Often, procedures are followed for many years and no one knows why. You should question why a procedure is being followed at all.
3. *Where*? Where are operations being performed? Why? Where could they be performed? Could they be performed more efficiently elsewhere?
4. *When*? When is a procedure performed? Why is it being performed at this time? Is this the best time?
5. *How*? How is a procedure performed? Why is it performed in that manner? Could it be performed better, more efficiently, or less expensively in some other manner?

There is a difference between asking what *is* being done and what *could* or *should* be done. The systems analyst first must understand the current situation. Only then can he or she tackle the question of what *should* be done. Figure 4-16 lists the basic questions and when they should be asked. Notice that the first two columns relate to the current system, but the third column focuses on the proposed system.

CURRENT SYSTEM		PROPOSED SYSTEM
Who does it?	Why does this person do it?	Who should do it?
What is done?	Why is it done?	What should be done?
Where is it done?	Why is it done there?	Where should it be done?
When is it done?	Why is it done then?	When should it be done?
How is it done?	Why is it done this way?	How should it be done?

FIGURE 4-16 Sample questions during requirements modeling as the focus shifts from the current system to the proposed system.

> **ON THE WEB**
>
> To learn more about the Zachman Framework, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate The Zachman Framework link.

The Zachman Framework

In the 1980s, John Zachman observed how industries such as architecture and construction handled complex projects, and he suggested that the same ideas could be applied to information systems development. His concept, the **Zachman Framework for Enterprise Architecture**, is a model that asks the traditional fact-finding questions in a systems development context, as shown in Figure 4-17. The Zachman Framework is a popular approach, and the Visible Analyst CASE tool now includes a Zachman Framework interface that allows users to view a systems project from different perspectives and levels of detail. The Zachman Framework helps managers and users understand the model and ensures that overall business goals translate into successful IT projects.

FIGURE 4-17 Visible Analyst uses the Zachman Framework for Enterprise Architecture. The Zachman concept presents traditional fact-finding questions in a systems development context.

Interviews

Interviewing is an important fact-finding tool during the systems analysis phase. An **interview** is a planned meeting during which you obtain information from another person. You must have the skills needed to plan, conduct, document, and evaluate interviews successfully.

After you identify the information you need, as described earlier in the chapter, you can begin the interviewing process, which consists of seven steps for each interview:

1. Determine the people to interview.
2. Establish objectives for the interview.
3. Develop interview questions.
4. Prepare for the interview.
5. Conduct the interview.
6. Document the interview.
7. Evaluate the interview.

Step 1: Determine the People to Interview

To get an accurate picture, you must select the right people to interview and ask them the right questions. During the preliminary investigation, you talked mainly to middle managers or department heads. Now, during the systems analysis phase, you might need to interview people from all levels of the organization.

Although you can select your interview candidates from the formal organization charts that you reviewed earlier, you also must consider any informal structures that exist in the organization. Informal structures usually are based on interpersonal relationships and can develop from previous work assignments, physical proximity, unofficial procedures, or personal relationships such as the informal gathering shown in Figure 4-18. In an **informal structure**, some people have more influence or knowledge than appears on an organization chart. Your knowledge of the company's formal and informal structures helps you determine the people to interview during the systems analysis phase.

Should you interview several people at the same time? Group interviews can save time and provide an opportunity to observe interaction among the participants. Group interviews also can present problems. One person might dominate the conversation, even when questions are addressed specifically to others. Organization level also can present a problem, as the presence of senior managers in an interview might prevent lower-level employees from expressing themselves candidly.


FIGURE 4-18 When setting up interviews, an analyst should look outside a formal organization chart to identify people who might provide valuable information.

Step 2: Establish Objectives for the Interview

After deciding on the people to interview, you must establish objectives for the session. First, you should determine the general areas to be discussed, and then list the facts you want to gather. You also should try to solicit ideas, suggestions, and opinions during the interview.

The objectives of an interview depend on the role of the person being interviewed. Upper-level managers can provide the big picture and help you to understand the system as a whole. Specific details about operations and business processes are best learned from people who actually work with the system on a daily basis.

In the early stages of systems analysis, interviews usually are general. As the fact-finding process continues, however, the interviews focus more on specific topics. Interview objectives also vary at different stages of the investigation. By setting specific objectives, you create a framework that helps you decide what questions to ask and how to phrase the questions.

Step 3: Develop Interview Questions

Creating a standard list of interview questions helps to keep you on track and avoid unnecessary tangents. Also, if you interview several people who perform the same job, a standard question list allows you to compare their answers. Although you have a list of specific questions, you might decide to depart from it because an answer to one question leads to another topic that you want to pursue. That question or topic then should be included in a revised set of questions used to conduct future interviews. If the question proves to be extremely important, you may need to return to a previous interviewee to query him or her on the topic.

The interview should consist of several different kinds of questions: open-ended, closed-ended, or questions with a range of responses. When you phrase your questions, you should avoid **leading questions** that suggest or favor a particular reply. For example, rather than asking, "What advantages do you see in the proposed system?" you might ask, "Do you see any advantages in the proposed system?"

OPEN-ENDED QUESTIONS **Open-ended questions** encourage spontaneous and unstructured responses. Such questions are useful when you want to understand a larger process or draw out the interviewee's opinions, attitudes, or suggestions. Here are some examples of open-ended questions: What are users saying about the new system? How is this task performed? Why do you perform the task that way? How are the checks reconciled? What added features would you like to have in the new billing system? Also, you can use an open-ended question to probe further by asking: Is there anything else you can tell me about this topic?

CLOSED-ENDED QUESTIONS **Closed-ended questions** limit or restrict the response. You use closed-ended questions when you want information that is more specific or when you need to verify facts. Examples of closed-ended questions include the following: How many personal computers do you have in this department? Do you review the reports before they are sent out? How many hours of training does a clerk receive? Is the calculation procedure described in the manual? How many customers ordered products from the Web site last month?

RANGE-OF-RESPONSE QUESTIONS **Range-of-response questions** are closed-ended questions that ask the person to evaluate something by providing limited answers to specific responses or on a numeric scale. This method makes it easier to tabulate the answers and interpret the results. Range-of-response questions might include these: On a scale of 1 to 10, with 1 the lowest and 10 the highest, how effective was your training? How would you rate the severity of the problem: low, medium, or high? Is the system shutdown something that occurs never, sometimes, often, usually, or always?

Step 4: Prepare for the Interview

After setting the objectives and developing the questions, you must prepare for the interview. Careful preparation is essential because an interview is an important meeting and not just a casual chat. When you schedule the interview, suggest a specific day and time and let the interviewee know how long you expect the meeting to last. It is also a good idea to send an e-mail or place a reminder call the day before the interview.

Remember that the interview is an interruption of the other person's routine, so you should limit the interview to no more than one hour. If business pressures force a postponement of the meeting, you should schedule another appointment as soon as it is convenient. Remember to keep department managers informed of your meetings with their staff members. Sending a message to each department manager listing your planned appointments is a good way to keep them informed. Figure 4-19 is an example of such a message.

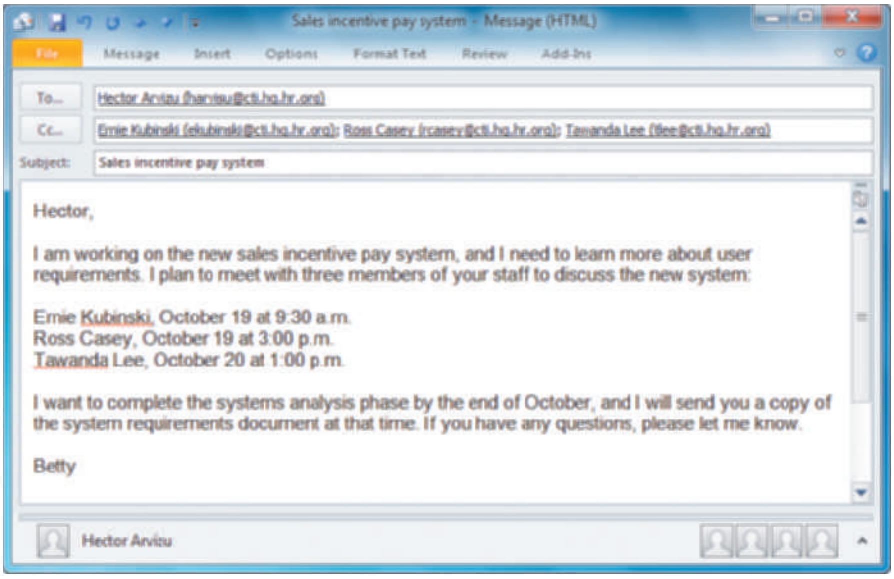


FIGURE 4-19 Sample message to a department head about interviews.

You should send a list of topics to an interviewee several days before the meeting, especially when detailed information is needed, so the person can prepare for the interview and minimize the need for a follow-up meeting. Figure 4-20 shows a sample message that lists specific questions and confirms the date, time, location, purpose, and anticipated duration of the interview.

If you have questions about documents, ask the interviewee to have samples available at the meeting. Your advance memo should include a list of the documents you want to discuss, if you know what they are. Also, you can make a general request for documents, as the analyst did in her e-mail shown in Figure 4-20.

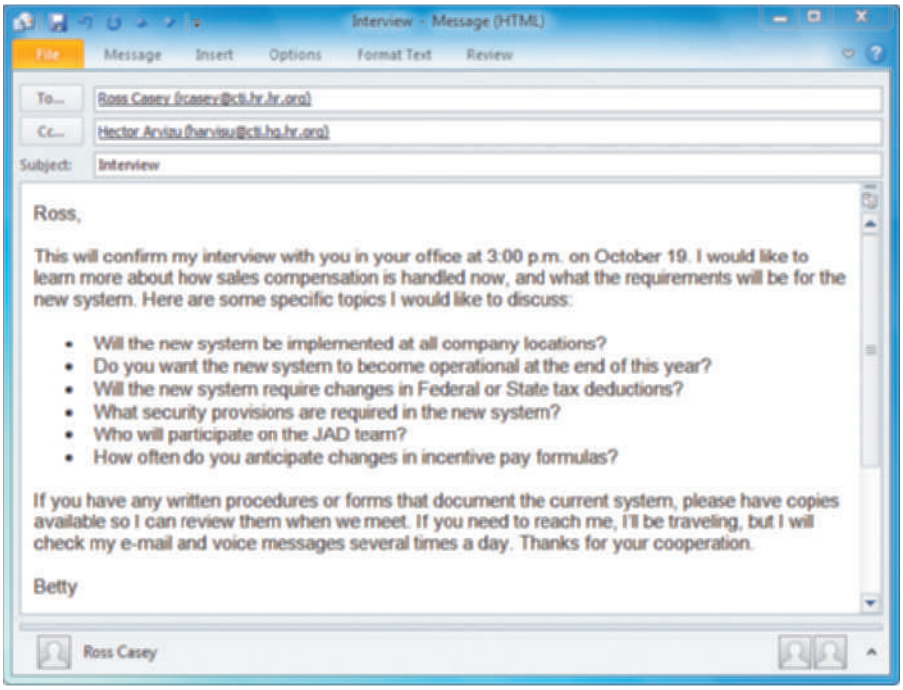


FIGURE 4-20 Sample message to confirm an interview.

Two schools of thought exist about the best location for an interview. Some analysts believe that interviews should take place in the interviewee's office, whereas other analysts feel that a neutral location such as a conference room is better.

Supporters of interviews in the interviewee's office believe that is the best location because it makes the interviewee feel comfortable during the meeting. A second argument in favor of the interviewee's office is that the office is where he or she has the easiest access to supporting material that might be needed during the discussion. If you provide a complete list of topics in advance, however, the interviewee can bring the necessary items to a conference room or other location.

Supporters of neutral locations stress the importance of keeping interruptions to a minimum so both people can concentrate fully. In addition, an interview that is free of interruptions takes less time. If the meeting does take place in the interviewee's office, you should suggest tactfully that all calls be held until the conclusion of the interview.

Step 5: Conduct the Interview

After determining the people to interview, setting your objectives, and preparing the questions, you should develop a specific plan for the meeting. When conducting an interview, you should begin by introducing yourself, describing the project, and explaining your interview objectives.

During the interview, ask questions in the order in which you prepared them, and give the interviewee sufficient time to provide thoughtful answers. Establishing a good rapport with the interviewee is important, especially if this is your first meeting. If the other person feels comfortable and at ease, you probably will receive more complete and candid answers. Your primary responsibility during an interview is to listen carefully to the answers. Analysts sometimes hear only what they expect to hear. You must concentrate on what is said and notice any nonverbal communication that takes place. This process is called **engaged listening**.

After asking a question, allow the person enough time to think about the question and arrive at an answer. Studies have shown that the maximum pause during a conversation is usually three to five seconds. After that interval, one person will begin talking. You will need to be patient and practice your skills in many actual interview situations to be successful.

When you finish asking your questions, summarize the main points covered in the interview and explain the next course of action. For example, mention that you will send a follow-up memo or that the interviewee should get back to you with certain information. When you conclude the interview, thank the person and encourage him or her to contact you with any questions or additional comments. Also, when the interview ends, it is a good idea to ask the interviewee whether he or she can suggest any additional topics that should be discussed.

After an interview, you should summarize the session and seek a confirmation from the other person. By stating your understanding of the discussion, the interviewee can respond and correct you, if necessary. One approach is to rephrase the interviewee's answers. For example, you can say, "If I understand you correctly, you are saying that ..." and then reiterate the information given by the interviewee.

Step 6: Document the Interview

Although taking notes during an interview has both advantages and disadvantages, the accepted view is that note taking should be kept to a minimum. Although you should write down a few notes to jog your memory after the interview, you should avoid writing everything that is said. Too much writing distracts the other person and makes it harder to establish a good rapport.

After conducting the interview, you must record the information quickly. You should set aside time right after the meeting to record the facts and evaluate the information. For that reason, try not to schedule back-to-back interviews. Studies have shown that 50 percent of a conversation is forgotten within 30 minutes. You, therefore, should use your notes to record the facts immediately so you will not forget them. You can summarize the facts by preparing a narrative describing what took place or by recording the answers you received next to each question on your prepared question list.

Tape recorders are effective tools for an interview; however, many people feel uncomfortable when recorders are present. Before using a recorder, you should discuss its use with the interviewee. Assure the interviewee that you will erase the tape after you transcribe your notes and that you will stop and rewind the tape anytime during the interview at his or her request. If you ask sensitive questions or the interviewee wants to answer a question without being recorded, explain that you will turn off the tape for a period of time during the interview.

Even with a tape recorder in use, you should listen carefully to the interviewee's responses so you can ask good follow-up questions. Otherwise, you might have to return for a second visit to ask the questions you missed the first time. Also, remember that each recorded interview takes twice the amount of time, because you must listen to or view the recorded meeting again after conducting the interview itself.

After the interview, send a memo to the interviewee expressing your appreciation for his or her time and cooperation. In the memo, you should note the date, time, location, purpose of the interview, and the main points you discussed so the interviewee has a written summary and can offer additions or corrections.

Step 7: Evaluate the Interview

In addition to recording the facts obtained in an interview, try to identify any possible biases. For example, an interviewee who tries to protect his or her own area or function might give incomplete answers or refrain from volunteering information. Or, an interviewee with strong opinions about the current or future system might distort the facts. Some interviewees might answer your questions in an attempt to be helpful even though they do not have the necessary experience to provide accurate information.

CASE IN POINT 4.2: DEEP RIVER COLLEGE

Deep River College is a two-year school in Southern California. Twice a year, the fund-raising office at Deep River mails requests for donations to the alumni. The staff uses a word processing program and a personal information database to create personalized letters. Data on past contributions and other alumni information, however, is stored manually. The dean, Alexandra Ali, recently submitted a systems request asking the college's IT department to develop a computerized alumni information system. The school does not have a formal systems review committee, and each department has an individual budget for information services.

Eddie Bateman, a systems analyst, performed a preliminary investigation and he concluded that the system met all the feasibility tests. After reading his report, Alexandra asked him to proceed with the systems analysis phase. Eddie has scheduled an interview with her, and he has asked you to help him prepare for the meeting. Specifically, he wants you to list all the topics he should cover during the interview. Eddie also wants you to prepare a list of specific questions that he should ask. Be sure to include open-ended, closed-ended, and range-of-response questions.

Unsuccessful Interviews

No matter how well you prepare for interviews, some are not successful. One of the main reasons could be that you and the interviewee did not get along well. Such a situation can be caused by several factors. For example, a misunderstanding or personality conflict could affect the interview negatively, or the interviewee might be afraid that the new system will eliminate or change his or her job.

In other cases, the interviewee might give only short or incomplete responses to your open-ended questions. If so, you should switch to closed-ended questions or questions with a range of responses, or try rephrasing your open-ended questions into those types of questions. If that still does not help, you should find a tactful way to conclude the meeting.

Continuing an unproductive interview is difficult. The interviewee could be more cooperative later, or you might find the information you seek elsewhere. If failure to obtain specific information will jeopardize the success of the project, inform your supervisor, who can help you decide what action to take. Your supervisor might contact the interviewee's supervisor, ask another systems analyst to interview the person, or find some other way to get the needed information.

CASE IN POINT 4.3: FastPak Overnight Package System

FastPak, the nation's fourth-largest overnight package system, is headquartered in Los Angeles, California. Jesse Evans is a systems analyst on an IT team that is studying ways to update FastPak's package tracking system. Jesse prepared well for her interview with Jason Tanya, FastPak's executive vice president. Mr. Tanya did not ask his assistant to hold his calls during the meeting, however. After several interruptions, Jesse tactfully suggested that she could come back another time, or perhaps that Mr. Tanya might ask his assistant to hold his calls. "No way," he replied. "I'm a very busy man and we'll just have to fit this in as we can, even if it takes all day." Jesse was unprepared for his response. What are her options? Is an analyst always in control of this kind of situation? Why or why not?

Other Fact-Finding Techniques

In addition to interviewing, systems analysts use other fact-finding techniques, including document review, observation, questionnaires and surveys, sampling, and research. Such techniques are used before interviewing begins to obtain a good overview and to help develop better interview questions.

Document Review

Document review can help you understand how the current system is supposed to work. Remember that system documentation sometimes is out of date. Forms can change or be discontinued, and documented procedures often are modified or eliminated. You should obtain copies of actual forms and operating documents currently in use. You also should review blank copies of forms, as well as samples of actual completed forms. You usually can obtain document samples during interviews with the people who perform that procedure. If the system uses a software package, you should review the documentation for that software.

Observation

The **observation** of current operating procedures is another fact-finding technique. Seeing the system in action gives you additional perspective and a better understanding of system procedures. Personal observation also allows you to verify statements made in interviews and determine whether procedures really operate as they are described.

Through observation, you might discover that neither the system documentation nor the interview statements are accurate.

Personal observation also can provide important advantages as the development process continues. For example, recommendations often are better accepted when they are based on personal observation of actual operations. Observation also can provide the knowledge needed to test or install future changes and can help build relationships with the users who will work with the new system.

Plan your observations in advance by preparing a checklist of specific tasks you want to observe and questions you want to ask. Consider the following issues when you prepare your list:

1. Ask sufficient questions to ensure that you have a complete understanding of the present system operation. A primary goal is to identify the methods of handling situations that are not covered by standard operating procedures. For example, what happens in a payroll system if an employee loses a time card? What is the procedure if an employee starts a shift 10 minutes late but then works 20 minutes overtime? Often, the rules for exceptions such as these are not written or formalized; therefore, you must try to document any procedures for handling exceptions.
2. Observe all the steps in a transaction and note the documents, inputs, outputs, and processes involved.
3. Examine each form, record, and report. Determine the purpose each item of information serves.
4. Consider each user who works with the system and the following questions: What information does that person receive from other people? What information does this person generate? How is the information communicated? How often do interruptions occur? How much downtime occurs? How much support does the user require, and who provides it?
5. Talk to the people who receive current reports to see whether the reports are complete, timely, accurate, and in a useful form. Ask whether information can be eliminated or improved and whether people would like to receive additional information.

As you observe people at work, as shown in Figure 4-21, consider a factor called the **Hawthorne Effect**. The name comes from a well-known study performed in the Hawthorne plant of the Western Electric Company in the 1920s. The purpose of the study was to determine how various changes in the work environment would affect employee productivity. The surprising result was that productivity improved during observation whether the conditions were made better or worse. Researchers concluded that productivity seemed to improve whenever the workers knew they were being observed.

Although some recent studies have raised questions about the original findings, you should be aware that observation can and does have an effect on normal

FIGURE 4-21 The Hawthorne study suggested that worker productivity improves during observation. Always consider the Hawthorne Effect when observing the operation of an existing system.

operations. With this in mind, always give advance notice to the supervisor in that area. In some situations, it might be helpful to explain the purpose of your visit to the people being observed.

Questionnaires and Surveys

In projects where it is desirable to obtain input from a large number of people, a questionnaire can be a valuable tool. A **questionnaire**, also called a **survey**, is a document containing a number of standard questions that can be sent to many individuals.

Questionnaires can be used to obtain information about a wide range of topics, including workloads, reports received, volumes of transactions handled, job duties, difficulties, and opinions of how the job could be performed better or more efficiently. Figure 4-22 shows a sample questionnaire that includes several different question and response formats.

PURCHASE REQUISITION QUESTIONNAIRE

Pat Kline, Vice President, Finance, has asked us to investigate the purchase requisition process to see if it can be improved. Your input concerning this requisition process will be very valuable. We would greatly appreciate it if you could complete the following questionnaire and return it by March 10 to Dana Juarez in information technology. If you have any questions, please call Dana at x2561.

A. **YOUR OBSERVATIONS**
Please answer each question by checking one box.

1. How many purchase requisitions did you process in the past five working days? ____________

2. What percentage of your time is spent processing requisitions?
[ ] under 20%   [ ] 60–79%
[ ] 21–39%   [ ] 80% or more
[ ] 40–59%

3. Do you believe too many errors exist on requisitions?
[ ] yes
[ ] no

4. Out of every 100 requisitions you process, how many contain errors?
[ ] fewer than 5   [ ] 20 to 29
[ ] 5 to 9   [ ] 30 to 39
[ ] 10 to 14   [ ] 40 to 49
[ ] 15 to 19   [ ] 50 or more

5. What errors do you see most often on requisitions? (Place a 1 next to the most common error, place a 2 next to the second, etc.)
[ ] incorrect charge number   [ ] missing authorization
[ ] missing charge information   [ ] other (please explain) ____________
[ ] arithmetic errors
[ ] incorrect discount percent used

B. **YOUR SUGGESTIONS**
Please be specific, and give examples if possible.

1. If the currently used purchase requisition form were to be redesigned, what changes to the form would you recommend?
____
____
____
____
____
(If necessary, please attach another sheet)

2. Would you be interested in meeting with an information technology representative to discuss your ideas further? If so, please complete the following information:

Name ____________ Department ____________

Telephone ____________ E-mail address ____________

FIGURE 4-22 Sample questionnaire. Does it follow the suggested guidelines?

A typical questionnaire starts with a heading, which includes a title, a brief statement of purpose, the name and telephone number of the contact person, the deadline date for completion, and how and where to return the form. The heading usually is followed by general instructions that provide clear guidance on how to answer the questions. Headings also are used to introduce each main section or portion of the survey and include instructions when the type of question or response changes. A long questionnaire might end with a conclusion that thanks the participants and reminds them how to return the form.

What about the issue of anonymity? Should people be asked to sign the questionnaire, or is it better to allow anonymous responses? The answer depends on two questions. First, does an analyst really need to know who the respondents are in order to match or correlate information? For example, it might be important to know what percentage of users need a certain software feature, but specific usernames might not be relevant. Second, does the questionnaire include any sensitive or controversial topics? Many people do not want to be identified when answering a question such as "How well has your supervisor explained the system to you?" In such cases, anonymous responses might provide better information.

When designing a questionnaire, the most important rule of all is to make sure that your questions collect the right data in a form that you can use to further your fact-finding. Here are some additional ideas to keep in mind when designing your questionnaire:

- Keep the questionnaire brief and user-friendly.
- Provide clear instructions that will answer all anticipated questions.
- Arrange the questions in a logical order, going from simple to more complex topics.
- Phrase questions to avoid misunderstandings; use simple terms and wording.
- Try not to lead the response or use questions that give clues to expected answers.
- Limit the use of open-ended questions that are difficult to tabulate.
- Limit the use of questions that can raise concerns about job security or other negative issues.
- Include a section at the end of the questionnaire for general comments.
- Test the questionnaire whenever possible on a small test group before finalizing it and distributing to a large group.

A questionnaire can be a traditional paper form, or you can create a **fill-in form** and collect data on the Internet or a company intranet. For example, you can use Microsoft Word, as shown in Figure 4-23, to create form fields, including text boxes, date pickers, and drop-down lists where users can click selections. Before you publish the form, you should protect it so users can fill it in but cannot change the layout or design. Forms also can be automated, so if a user answers *no* to question three, he or she goes directly to question eight, where the form-filling resumes.

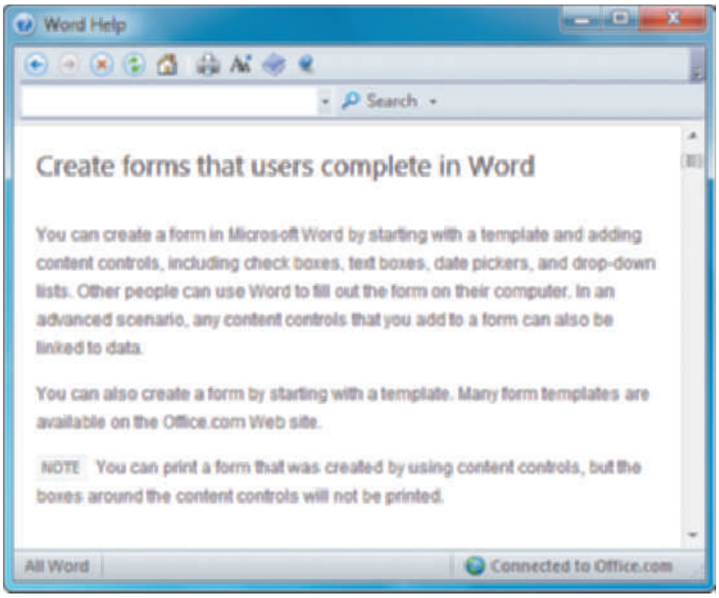


FIGURE 4-23 Using Microsoft Word, you can create a fill-in form with text boxes, date pickers, and drop-down lists.

Sampling

When studying an information system, you should collect examples of actual documents using a process called **sampling**. The samples might include records, reports, operational logs, data entry documents, complaint summaries, work requests, and various types of forms. Sampling techniques include systematic sampling, stratified sampling, and random sampling.

Suppose you have a list of 200 customers who complained about errors in their statements, and you want to review a representative sample of 20 customers. A **systematic sample** would select every tenth customer for review. If you want to ensure that the sample is balanced geographically, however, you could use a **stratified sample** to select five customers from each of four zip codes. Another example of stratified sampling is to select a certain percentage of transactions from each zip code, rather than a fixed number. Finally, a **random sample** selects any 20 customers.

ON THE WEB

To learn more about sampling, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Sampling link.

The main objective of a sample is to ensure that it represents the overall population accurately. If you are analyzing inventory transactions, for example, you should select a sample of transactions that are typical of actual inventory operations and do not include unusual or unrelated examples. For instance, if a company performs special processing on the last business day of the month, that day is not a good time to sample *typical* daily operations. To be useful, a sample must be large enough to provide a fair representation of the overall data.

You also should consider sampling when using interviews or questionnaires. Rather than interviewing everyone or sending a questionnaire to the entire group, you can use a sample of participants. You must use sound sampling techniques to reflect the overall population and obtain an accurate picture.

Research

Research is another important fact-finding technique. Your research can include the Internet, IT magazines, and books to obtain background information, technical material, and news about industry trends and developments. In addition, you can attend professional meetings, seminars, and discussions with other IT professionals, which can be very helpful in problem solving.

FIGURE 4-24 InfoWorld's Web site offers many resources for IT professionals.

The Internet is an extremely valuable resource. Part D of the Systems Analyst's Toolkit describes a variety of Internet resource tools. Using the Internet, you also can access information from federal and state governments, as well as from publishers, universities, and libraries around the world. Online forums and newsgroups are good resources for exchanging information with other professionals, seeking answers to questions, and monitoring discussions that are of interest to you.

All major hardware and software vendors maintain sites on the Web where you can obtain information about products and services offered by the company and send e-mail with specific questions to company representatives. In addition to contacting specific firms, you can access Web sites maintained by publishers and independent firms that provide links to hundreds of hardware and software vendors, as shown in Figure 4-24. Such sites are one-stop information centers where IT professionals can find information, share ideas, and keep posted on developments in technology.

Research also can involve a visit to a physical location, called a **site visit**, where the objective is to observe a system in use at another location. If you are studying your firm's human resources information system, for example, you might want to see how another company's system works. Site visits also are important when considering the purchase of a software package. If the software vendor suggests possible sites to visit, be aware that such sites might constitute a biased sample. A single site visit seldom gives you true pictures, so you should try to visit more than one installation.

Before a site visit such as the one shown in Figure 4-25, prepare just as you would for an interview. Contact the appropriate manager and explain the purpose of your visit. Decide what questions you will ask and what processes you will observe. During your visit, observe how the system works and note any problems or limitations. You also will want to learn about the support provided by the vendor, the quality of the system documentation, and so on.

FIGURE 4-25 A site visit provides an opportunity to observe a system in use.

Interviews versus Questionnaires

When you seek input from a large group, a questionnaire is a very useful tool. On the other hand, if you require detailed information from only a few people, then you probably should interview each person individually. Is it better to interview or use a questionnaire? Each situation is different, and you must consider the type of information, time constraints, and expense factors.

The interview is more familiar and personal than a questionnaire. People who are unwilling to put critical or controversial comments in writing might talk more freely in person. Moreover, during a face-to-face interview, you can react immediately to anything the interviewee says. If surprising or confusing statements are made, you can pursue the topic with additional questions. In addition, during a personal interview, you can watch for clues to help you determine if responses are knowledgeable and unbiased. Participation in interviews also can affect user attitudes, because people who are asked for their opinions often view the project more favorably.

Interviewing, however, is a costly and time-consuming process. In addition to the meeting itself, both people must prepare, and the interviewer has to do follow-up work. When a number of interviews are planned, the total cost can be quite substantial. The personal interview usually is the most expensive fact-finding technique.

In contrast, a questionnaire gives many people the opportunity to provide input and suggestions. Questionnaire recipients can answer the questions at their convenience and do not have to set aside a block of time for an interview. If the questionnaire allows anonymous responses, people might offer more candid responses than they would in an interview.

Preparing a good questionnaire, however, like a good interview, requires skill and time. If a question is misinterpreted, you cannot clarify the meaning as you can in a face-to-face interview. Furthermore, unless questionnaires are designed well, recipients might view them as intrusive, time-consuming, and impersonal. As an analyst, you should select the technique that will work best in a particular situation.

Another popular method of obtaining input is called **brainstorming**, which refers to a small group discussion of a specific problem, opportunity, or issue. This technique encourages new ideas, allows team participation, and enables participants to build on each other's inputs and thoughts. Brainstorming can be structured or unstructured. In **structured brainstorming**, each participant speaks when it is his or her turn, or passes. In **unstructured brainstorming**, anyone can speak at any time. At some point, the results are recorded and made part of the fact-finding documentation process.

CASE IN POINT 4.4: CyberStuff

Ann Ellis is a systems analyst at CyberStuff, a large company that sells computer hardware and software via telephone, mail order, and the Internet. CyberStuff processes several thousand transactions per week on a three-shift operation and employs 50 full-time and 125 part-time employees. Lately, the billing department has experienced an increase in the number of customer complaints about incorrect bills. During the preliminary investigation, Ann learned that some CyberStuff representatives did not follow established order entry procedures. She feels that with more information, she might find a pattern and identify a solution for the problem.

Ann is not sure how to proceed. She came to you, her supervisor, with two separate questions. First, is a questionnaire the best approach, or would interviews be better? Second, whether she uses interviews, a questionnaire, or both techniques, should she select the participants at random, include an equal number of people from each shift, or use some other approach? As Ann's supervisor, what would you suggest, and why?

Documentation

Keeping accurate records of interviews, facts, ideas, and observations is essential to successful systems development. The ability to manage information is the mark of a successful systems analyst and an important skill for all IT professionals.

The Need for Recording the Facts

As you gather information, the importance of a single item can be overlooked or complex system details can be forgotten. The basic rule is to write it down. You should document your work according to the following principles:

- Record information as soon as you obtain it.
- Use the simplest recording method possible.
- Record your findings in such a way that they can be understood by someone else.
- Organize your documentation so related material is located easily.

Often, systems analysts use special forms for describing a system, recording interviews, and summarizing documents. One type of documentation is a narrative list with simple statements about what is occurring, apparent problems, and suggestions for improvement. Other forms of documentation that are described in Chapter 4 include data flow diagrams, flowcharts, sample forms, and screen captures.

Software Tools

Many software programs are available to help you record and document information. Some examples are described here.

CASE TOOLS You can use CASE tools at every stage of systems development. This chapter contains several examples of CASE tools. Part B of the Systems Analyst's Toolkit describes other features and capabilities of CASE tools.

PRODUCTIVITY SOFTWARE **Productivity software** includes word processing, spreadsheet, database management, presentation graphics, and collaboration software programs. Although Microsoft Office is the best-known set of productivity software programs, other vendors offer products in each of these categories.

Using word processing software such as Microsoft Word, Corel WordPerfect, or OpenOffice.org Writer, you can create reports, summaries, tables, and forms. In addition to standard document preparation, the program can help you organize a presentation with templates, bookmarks, annotations, revision control, and an index. You can consult the program's Help system for more information about those and other features. You also can create fill-in forms to conduct surveys and questionnaires, as described earlier in this chapter.

Spreadsheet software, such as Microsoft Excel, Corel Quattro Pro, or OpenOffice.org Calc, can help you track and manage numeric data or financial information. You also can generate graphs and charts that display the data and show possible patterns, and you can use the statistical functions in a spreadsheet to tabulate and analyze questionnaire data. A graphical format often is used in quality control analysis because it highlights problems and their possible causes, and it is effective when presenting results to management. A common tool for showing the distribution of questionnaire or sampling results is a vertical bar chart called a **histogram**. Most spreadsheet programs can create histograms and other charts that can display data you have collected. Figure 4-26 displays a typical histogram that might have resulted from the questionnaire shown in Figure 4-22 on page 166.

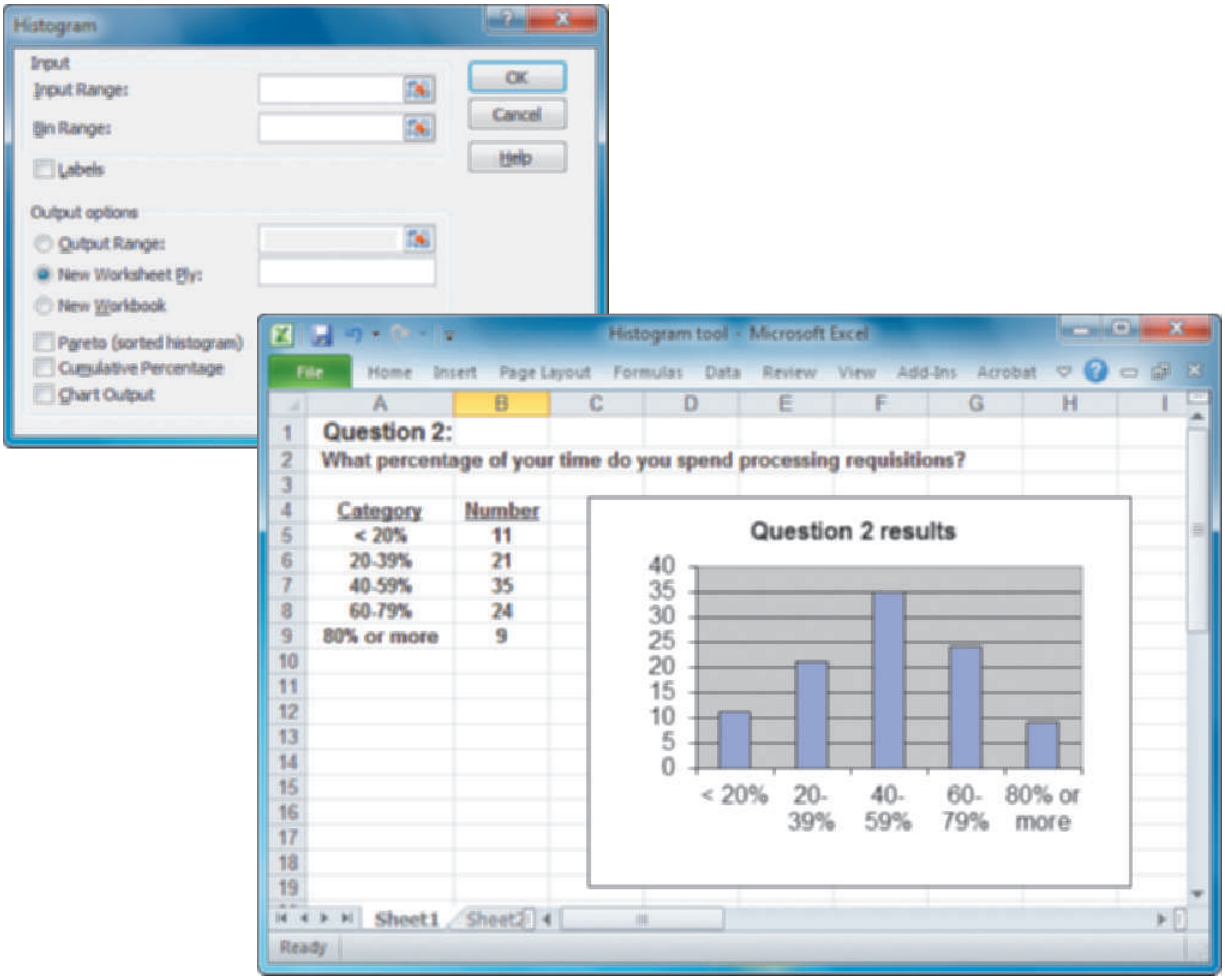


FIGURE 4-26 This histogram displays results from Question 2 in the questionnaire shown in Figure 4-22 on page 166.

Database management software allows you to document and organize fact-finding results such as events, observations, and data samples. You can use a database program such as Microsoft Access to manage the details of a complex project, create queries to retrieve specific information, and generate custom reports.

Presentation graphics software, such as Microsoft PowerPoint, Apple Keynote, or OpenOffice.org Impress, is a powerful tool for organizing and developing your formal presentation. Presentation graphics programs enable you to create organization charts that can be used in a preliminary investigation and later during requirements modeling. These high-quality charts also can be included in written reports and management presentations.

Collaboration software is the latest weapon in the struggle to boost productivity. More than ever, people work in teams and use Web-based software such as Google Docs and Microsoft Web Apps to access data and share files. Google and others are betting that cloud computing will create a virtual workplace, where people will be able to interact in real time, with all the benefits of a traditional face-to-face workplace, but none of the limitations.

GRAPHIC MODELING SOFTWARE Microsoft Visio is a popular graphic modeling tool that can produce a wide range of charts and diagrams. Visio includes a library of templates, stencils, and shapes. An analyst can use Visio to create many types of visual models, including business processes, flowcharts, network diagrams, organization charts, and Web site maps, such as the one shown in Figure 4-27.

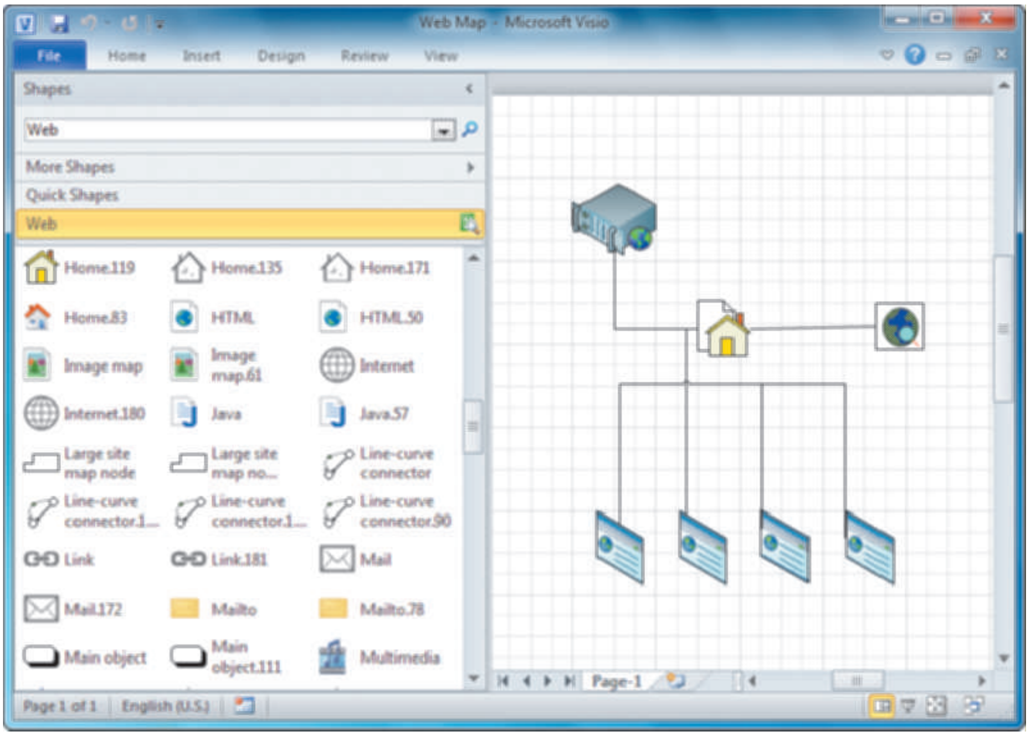


FIGURE 4-27 This Microsoft Visio screen shows shapes that can be used to create a Web site map.

PERSONAL INFORMATION MANAGERS A busy analyst needs to keep track of meetings, interviews, appointments, and deadlines. A **personal information manager (PIM)**, such as Microsoft Outlook or IBM's Lotus Organizer, can help manage those tasks using a personal calendar and a to-do list, with priorities and the capability to check off completed items.

In addition to desktop-based organizers, handheld computers are popular. Some handheld computers, also called **personal digital assistants (PDAs)**, accept handwritten input, while others have small keyboards. These devices can handle calendars, schedules, appointments, telephone lists, and calculations. A PDA can be standalone, Bluetooth-capable to synchronize with a desktop, or fully wireless-enabled, such as the HP iPAQ shown in Figure 4-28.

FIGURE 4-28 HP's iPAQ is a powerful wireless device that HP describes as a mobile handheld computer.

WIRELESS COMMUNICATION DEVICES Even in the dynamic world of IT, the recent explosion in wireless technology is almost unprecedented. The latest wireless standard, called **4G (fourth generation)**, is opening new frontiers in broadband Web access, e-mail, social networking, file exchange, and streaming multimedia. Users enjoy new hardware and software, easy synchronization with office networks, and innovative services designed for a *wired* generation.

The rapid growth of wireless communication has resulted in a merger of various technologies. Many people swear by **all-in-one devices** such as Research in Motion's BlackBerry or **smart phones**, such as the Apple iPhone. Others are devoted to products that use Google's Android operating system, which is a **mobile device platform** adopted by many hardware vendors, including Motorola, Kyocera, and LG. Figure 4-29 on the next page shows some examples of these products.

Beyond hardware choices, users can select from literally thousands of portable applications for business and personal use. No one can predict the future with certainty, but it is apparent that portable wireless technology is having an enormous impact on business practices, everyday communications, and social interaction.

FIGURE 4-29 Three popular examples of current wireless technology.

Preview of Logical Modeling

At the conclusion of requirements modeling, systems developers should have a clear understanding of business processes and system requirements. The next step is to construct a logical model of the system.

Data and process modeling, which is described in Chapter 5, uses a structured analysis approach. Structured analysis is a popular, traditional technique that describes the system in terms of data and the processes that act on that data.

An alternative to structured analysis modeling is object modeling, which is described in Chapter 6. Object modeling is a methodology that combines data and processes into things called objects that represent actual people, things, transactions, and events. Systems analysts use object models to visualize and document real-world business processes and operations.

IT professionals have differing views about systems development methodologies, and no universally accepted approach exists. By studying both structured analysis and object-oriented methods, you gain valuable knowledge, skills, and perspective. You then can use that information to determine what method, or combination of methods, is best for the different situations you will face in your career.

A QUESTION OF ETHICS

Your supervisor manages the corporate office where you work as a systems analyst. Several weeks ago, after hearing rumors of employee dissatisfaction, he asked you to create a survey for all IT employees. After the responses were returned and tabulated, he was disappointed to learn that many employees assigned low ratings to morale and management policies.

This morning he called you into his office and asked whether you could identify the departments that submitted the lowest ratings. No names were used on the individual survey forms. However, with a little analysis, you probably could identify the departments, because several questions were department-related.

Now you are not sure how to respond. The expectation was that the survey would be anonymous. Even though no individuals would be identified, would it be ethical to reveal which departments sent in the low ratings? Would your supervisor's motives for wanting this information matter?

Chapter Summary

The systems analysis phase includes three activities: requirements modeling, data and process modeling, and consideration of development strategies. The main objective is to understand the proposed project, ensure that it will support business requirements, and build a solid foundation for the systems design phase.

During requirements modeling, you identify the business-related requirements for the new information system, including outputs, inputs, processes, performance, and controls. You consider scalability to ensure that the system can support future growth and expansion. You also estimate total cost of ownership (TCO) to identify all costs, including indirect costs.

Popular team-based approaches include JAD, RAD, and agile methods. Joint application development (JAD) is a popular, team-based approach to fact-finding and requirements modeling. JAD involves an interactive group of users, managers, and IT professionals who participate in requirements modeling and develop a greater commitment to the project and to their common goals.

Rapid application development (RAD) is a team-based technique that speeds up information systems development and produces a functioning information system. RAD is a complete methodology, with a four-phase life cycle that parallels the traditional SDLC phases.

Agile methods attempt to develop a system incrementally, by building a series of prototypes and constantly adjusting them to user requirements.

Systems analysts use various tools and techniques to model system requirements. Unified Modeling Language (UML) is a widely used method of visualizing and documenting software design through the eyes of the business user. UML tools include use case diagrams and sequence diagrams to represent actors, their roles, and the sequence of transactions that occurs.

A functional decomposition diagram (FDD) is a model of business functions and processes. A CASE tool can generate a set of data flow diagrams directly from a FDD.

The fact-finding process includes interviewing, document review, observation, questionnaires, sampling, and research. Successful interviewing requires good planning and strong interpersonal and communication skills. The systems analyst must decide on the people to interview, set interview objectives, and prepare for, conduct, and analyze interviews. The analyst also might find it helpful to use one or more software tools during fact-finding.

Systems analysts should carefully record and document factual information as it is collected, and various software tools can help an analyst visualize and describe an information system. The chapter concluded with a preview of logical modeling. Data and process modeling is a structured analysis approach that views the system in terms of data and the processes that act on that data. Object modeling is an approach that views the system in terms of data and the processes that act on that data.

Key Terms and Phrases

4G (fourth generation) *173*
actor *151*
agile methods *143*
analytical skills *143*
brainstorming *169*
business process model (BPM) *150*
business process modeling notation (BPMN) *150*
closed-ended questions *160*
construction phase *147*
cutover phase *147*
data flow diagram (DFD) *151*
document review *164*
engaged listening *162*
fill-in form *167*
functional decomposition diagram (FDD) *150*
Hawthorne Effect *165*
histogram *171*
informal structure *159*
inputs *142*
interpersonal skills *143*
interview *159*
joint application development (JAD) *143*
leading questions *160*
observation *164*
open-ended questions *160*
outputs *142*
performance *142*
personal digital assistant (PDA) *173*
personal information manager (PIM) *173*
pool *151*
processes *142*
productivity software *170*
questionnaire *166*
random sample *168*
range-of-response questions *160*
rapid application development (RAD) *143*
Rapid Economic Justification (REJ) *156*
requirements modeling *142*
requirements planning phase *146*
research *168*
sampling *167*
scalability *155*
Scrum *148*
security *142*
sequence diagram *152*
site visit *169*
smart phone *173*
stratified sample *168*
structured brainstorming *169*
survey *166*
swim lanes *151*
system requirement *153*
system requirements document *143*
systematic sample *168*
total cost of ownership (TCO) *155*
Unified Modeling Language (UML) *151*
unstructured brainstorming *169*
use case diagram *151*
user design phase *146*
Zachman Framework for Enterprise Architecture *158*

Learn It Online

Instructions: To complete the Learn It Online exercises, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to the resources for this chapter, and click the link for the exercise you want to complete.

1 Chapter Reinforcement

TF, MC, and SA

Click one of the Chapter Reinforcement links for Multiple Choice, True/False, or Short Answer. Answer each question and submit to your instructor.

2 Flash Cards

Click the Flash Cards link and read the instructions. Type 20 (or a number specified by your instructor) in the Number of playing cards text box, type your name in the Enter your Name text box, and then click the Flip Card button. When the flash card is displayed, read the question and then click the ANSWER box arrow to select an answer. Flip through the Flash Cards. If your score is 15 (75%) correct or greater, click Print on the File menu to print your results. If your score is less than 15 (75%) correct, then redo this exercise by clicking the Replay button.

3 Practice Test

Click the Practice Test link. Answer each question, enter your first and last name at the bottom of the page, and then click the Grade Test button. When the graded practice test is displayed on your screen, click Print on the File menu to print a hard copy. Continue to take practice tests until you score 80% or better.

4 Who Wants To Be a Computer Genius?

Click the Computer Genius link. Read the instructions, enter your first and last name at the bottom of the page, and then click the Play button. When your score is displayed, click the PRINT RESULTS link to print a hard copy.

5 Wheel of Terms

Click the Wheel of Terms link. Read the instructions, and then enter your first and last name and your school name. Click the PLAY button. When your score is displayed on the screen, right-click the score and then click Print on the shortcut menu to print a hard copy.

6 Crossword Puzzle Challenge

Click the Crossword Puzzle Challenge link. Read the instructions, and then click the Continue button. Work the crossword puzzle. When you are finished, click the Submit button. When the crossword puzzle is redisplayed, submit it to your instructor.

SCR Associates Case Simulation Session 4: Requirements Modeling

Overview

The SCR Associates case study is a Web-based simulation that allows you to practice your skills in a real-world environment. The case study transports you to SCR's intranet, where you complete 12 work sessions, each aligning with a chapter. As you work on the case, you will receive e-mail and voice mail messages, obtain information from SCR's online libraries, and perform various tasks.

How do I use the case?

- Review the SCR background material in Chapter 1.
- Read the Preview for this session and study the Task List.
- Visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to the **SCR Case Simulation**, and locate the intranet link.
- Enter your name and the password **sad9e**. An opening screen will display the 12 sessions.
- Select this session. Check your e-mail and voice mail carefully, and then work on the tasks.

Preview: Session 4

As you begin the requirements modeling process, you receive specific directions from your supervisor, Jesse Baker. She wants you to conduct a survey of former and prospective students, lead a JAD group session, and draft a list of system requirements based on the results of the JAD session. She also wants to see a functional decomposition diagram showing the main TIMS functions.

Task List

1. *Group managers said it was OK for their people to attend a three-day JAD session next week. Send a message to the JAD team members, with a brief explanation of JAD methods and a proposed agenda.*
2. *Design a questionnaire for former and potential students in SCR's training classes. Also, reply to Jesse's message about sampling. Give her a recommendation and reasons.*
3. *Read the JAD session summary in the Data Library and put together a list of system requirements, including outputs, inputs, processes, performance, and controls.*
4. *Draw an FDD of the main functions for TIMS and send it to Jesse. Be sure to show at least one or two levels of detail.*

FIGURE 4-30 Task list: Session 4.

Chapter Exercises

Review Questions

1. What are the five questions typically used in fact-finding? What additional question can be asked during this process?
2. What is a systems requirement, and how are systems requirements classified?
3. What are JAD and RAD, and how do they differ from traditional fact-finding methods? What are their pros and cons?
4. What is total cost of ownership (TCO), and why is it important?
5. Provide examples of closed-ended, open-ended, and range-of-response questions.
6. What are three types of sampling, and why would you use them?
7. What is the Hawthorne Effect? Why is it significant?
8. What is a functional decomposition diagram (FDD) and why would you use one? Explain how to create an FDD.
9. What are agile methods, and what are some pros and cons of this approach?
10. To what three different audiences might you have to give a presentation? How would the presentation differ for each?

TOOLKIT TIME

Answer question 10 after you complete the presentations section in Part A of the four-part Systems Analyst's Toolkit that follows Chapter 12.

Discussion Topics

1. A group meeting sometimes is suggested as a useful compromise between interviews and questionnaires. In such a group meeting, one systems analyst meets with and asks questions of a number of users at one time. Discuss the advantages and disadvantages of such a group meeting.
2. JAD requires strong interpersonal and communication skills on the part of the systems analyst. Are those skills different from the ones that an analyst needs when conducting one-to-one interviews? Explain your answer.
3. Research the Internet, magazines, or textbooks to find examples of each of the following types of visual aids: bar chart, pie chart, line chart, table, diagram, and bulleted list of key points. How effective do you think each aid is? Find at least one example that you feel could be improved. Discuss its shortcomings and prepare an improved version.
4. Review the presentations section in Part A of the Systems Analyst's Toolkit, then attend a speech or presentation and analyze its effectiveness. Consider the speaker's delivery and how he or she organized the material, used visual aids, and handled audience questions. Describe specifically how the speech or presentation was most effective, as well as how it could have been improved.

Projects

1. Design a questionnaire to learn more about the registration process at your school or how customers place orders at a local business. Apply the guidelines you learned in this chapter.
2. Use Microsoft Word or another word processing program to design a simple form, using the program's form-filling feature.
3. Create a functional decomposition diagram similar to the one in Figure 4-8 on page 150, but showing a typical U.S. post office.
4. Use the Internet to find a Web site that contains current IT industry news, information, and links. Bookmark the site and print a copy of the initial screen.

Apply Your Knowledge

The Apply Your Knowledge section contains four mini-cases. Each case describes a situation, explains your role in the case, and asks you to respond to questions. You can answer the questions by applying knowledge you learned in the chapter.

1 Elmwood College

Situation:

The school is considering a new system that will speed up the registration process. As a systems analyst, you are asked to develop a plan for fact-finding.

1. List all the possible techniques that you might use.
2. Describe an advantage for each technique.
3. Suppose the development budget is tight. How might that affect the fact-finding process?
4. What are five important questions to use during fact-finding?

2 JAD Session 1

Situation:

You are an IT advisor to a JAD team that is studying a new inventory system. The proposed system will provide more information and faster updates, and automatically monitor fast- or slow-moving items. Some controversy exists about whether to use an on-site or off-site location for the JAD sessions.

1. How would you advise the project leader?
2. Who should be on the JAD team, and what would be their roles as team members?
3. The JAD project leader asked for advice about how to get the first session started. How would you reply?
4. You invited the senior vice president to the opening JAD session, but she says she is quite busy and might not be able to attend unless it is really important. What would you say to her?

3 JAD Session 2

Situation:

The JAD team wants you to draw up a checklist of requirements for the new system.

1. List the five main categories of system requirements.
2. Use your imagination and provide at least one example per category of a system requirement that might be appropriate for an inventory system.
3. The project leader wants you to explain the concept of scalability to the team. How will you do that?
4. Several managers on the team have heard of TCO but are not quite sure what it is. How will you explain it to them?

4 Better Hardware Marketing System

Situation:

Your boss, the IT director, wants you to explain the UML to a group of company managers and users who will serve on a systems development team for the new marketing system.

1. Describe the Unified Modeling Language (UML) and how it can be used during systems development.
2. Explain use case diagrams to the group, and provide a simple example.
3. Explain sequence diagrams to the group, and provide a simple example.
4. During the meeting, a manager asks you to explain why it is desirable to describe the system through the eyes of a user. How would you answer?

Case Studies

Case studies allow you to practice specific skills learned in the chapter. Each chapter contains several case studies that continue throughout the textbook, and a chapter capstone case.

NEW CENTURY HEALTH CLINIC

New Century Health Clinic offers preventive medicine and traditional medical care. In your role as an IT consultant, you will help New Century develop a new information system.

Background

New Century Health Clinic has decided to computerize its office systems. The associates hired you, a local computer consultant, to perform a preliminary investigation. You had several meetings with Dr. Tim Jones to discuss the various office records and accounting systems. Anita Davenport, New Century's office manager, participated in those meetings.

In a report to the associates at the end of your investigation, you recommended conducting a detailed analysis of the patient record system, the patient and insurance billing systems, and the patient scheduling system. You believe that New Century would benefit most from implementing those three systems. Although the systems could be developed independently, you recommended analyzing all three systems together because of the significant interaction among them.

You presented your findings and recommendations at a late afternoon meeting of the associates. After answering several questions, you left the meeting so they could discuss the matter privately. Dr. Jones began the discussion by stating that he was impressed with your knowledge and professionalism, as well as your report and presentation.

Dr. Jones recommended accepting your proposal and hiring you immediately to conduct the systems analysis phase. Dr. Garcia, however, was not as enthusiastic and pointed out that such a study would certainly disrupt office procedures. The staff already had more work than they could handle, she argued, and taking time to answer your questions would only make the situation worse. Dr. Jones countered that the office workload was going to increase in any event, and that it was important to find a long-term solution to the problem. After some additional discussion, Dr. Garcia finally agreed with Dr. Jones's assessment. The next morning, Dr. Jones called you and asked you to go ahead with the systems analysis phase of the project.

Assignments

1. Review the office organization chart you prepared in Chapter 1 for New Century.
2. List the individuals you would like to interview during the systems analysis phase.
3. Prepare a list of objectives for each of the interviews you will conduct.
4. Prepare a list of specific questions for each individual you will interview.
5. Conduct the interviews. (Consult your instructor regarding how to accomplish this. One possibility is through role-playing.)
6. Prepare a written summary of the information gained from each of the interviews. (Your instructor may want you to use a standard set of interview results.)
7. Design a questionnaire that will go to a sample of New Century patients to find out if they were satisfied with current insurance and scheduling procedures. Your questionnaire should follow the suggestions in this chapter. Also, decide what sampling method you will use and explain the reason for your choice.

PERSONAL TRAINER, INC.

Personal Trainer, Inc., owns and operates fitness centers in a dozen Midwestern cities. The centers have done well, and the company is planning an international expansion by opening a new "supercenter" in the Toronto area. Personal Trainer's president, Cassia Umi, hired an IT consultant, Susan Park, to help develop an information system for the new facility. During the project, Susan will work closely with Gray Lewis, who will manage the new operation.

Background

During requirements modeling for the new system, Susan Park met with fitness center managers at several Personal Trainer locations. She conducted a series of interviews, reviewed company records, observed business operations, analyzed the BumbleBee accounting software, and studied a sample of sales and billing transactions. Susan's objective was to develop a list of system requirements for the proposed system.

Fact-Finding Summary

- A typical center has 300–500 members, with two membership levels: full and limited. Full members have access to all activities. Limited members are restricted to activities they have selected, but they can participate in other activities by paying a usage fee. All members have charge privileges. Charges for merchandise and services are recorded on a charge slip, which is signed by the member. At the end of each day, cash sales and charges are entered into the BumbleBee accounting software, which runs on a computer workstation at each location. Daily cash receipts are deposited in a local bank and credited to the corporate Personal Trainer account. The BumbleBee program produces a daily activity report with a listing of all sales transactions. At the end of the month, the local manager uses BumbleBee to transmit an accounts receivable summary to the Personal Trainer headquarters in Chicago, where member statements are prepared and mailed. Members mail their payments to the Personal Trainer headquarters, where the payment is applied to the member account.
- The BumbleBee program stores basic member information, but does not include information about member preferences, activities, and history.
- Currently, the BumbleBee program produces one local report (the daily activity report) and three reports that are prepared at the headquarters location: a monthly member sales report, an exception report for inactive members and late payers, and a quarterly profit-and-loss report that shows a breakdown of revenue and costs for each separate activity.

During the interviews, Susan received a number of "wish list" comments from local managers and staff members. For example, many managers wanted more analytical features so they could spot trends and experiment with what-if scenarios for special promotions and discounts. The most frequent complaint was that managers wanted more frequent information about the profitability of the business activities at their centers.

To enhance their business, managers wanted to offer a computerized activity and wellness log, a personal coach service, and e-mail communication with members. Managers also wanted better ways to manage information about part-time instructors and staff. Several staff members suggested a redesign for the charge slips or scannable ID cards.

Assignments

1. List the system requirements, with examples for each category. Review the information that Susan gathered, and assume that she will add her own ideas to achieve more effective outputs, inputs, processes, performance, and controls.
2. Are there scalability issues that Susan should consider? What are they?
3. If Susan wants to conduct a survey of current or prospective members to obtain their input, what type of sampling should she use? Why?
4. Draw an FDD that shows the main operations described in the fact statement.

BAXTER COMMUNITY COLLEGE

Baxter Community College is a two-year school in New Jersey. Twice a year, the records office at Baxter mails requests for donations to the alumni. The staff uses a word processing merge file to create personalized letters, but the data on past contributions and other alumni information is stored manually. The registrar, Mary Louise, recently submitted a systems request asking the college's IT department to develop a computerized alumni information system. The school does not have a formal systems review committee, and each department head has an individual budget for routine information services.

Todd Wagner, a systems analyst, was assigned to perform a preliminary investigation. After reading his report, Mary asked him to proceed with the systems analysis phase, saying that a formal presentation was unnecessary. Todd has scheduled an interview tomorrow with her, and he asked you to help him prepare for the meeting.

Assignments

1. Make a list of the topics that you think Todd should cover during the interview.
2. Prepare a list of specific questions that Todd should ask. Include open-ended, closed-ended, and range-of-response questions.
3. Conduct student-to-student interviews, with half the students assuming Todd's role and the other half playing the registrar.
4. Document the information covered during the interviews.

TOWN OF EDEN BAY

The town of Eden Bay owns and maintains a fleet of vehicles. You are a systems analyst reporting to Dawn, the town's IT manager.

Background

In Chapter 2, you learned that the town's maintenance budget has risen sharply in recent years. Based on a preliminary investigation, the town has decided to develop a new information system to manage maintenance information and costs more effectively. The new system will be named RAVE, which stands for Repair Analysis for Vehicular Equipment.

Dawn has asked you to perform additional fact-finding to document the requirements for the new system.

Assignments

1. Review the interview summaries in Chapter 2. For each person (Marie, Martin, Phil, Alice, and Joe), develop three additional questions: an open-ended question, a closed-ended question, and a range-of-response question.
2. Based on what you know so far, list the system requirements for the new system. You can use your imagination if the facts are insufficient. Consider outputs, inputs, processes, performance, and controls. Include at least two examples for each category.
3. You decide to analyze a sample of vehicle records. What sampling methods are available to you? Which one should you use, and why?
4. Dawn thinks it would be a good idea to conduct a JAD session to perform additional fact-finding. Draft a message to the participants, with a brief explanation of JAD methods and a proposed agenda.

CHAPTER CAPSTONE CASE: SoftWear, Limited

SWL

SoftWear, Limited (SWL), is a continuing case study that illustrates the knowledge and skills described in each chapter. In this case study, the student acts as a member of the SWL systems development team and performs various tasks.

Background

In Chapter 2, you learned that SWL's vice president of finance, Michael Jeremy, requested an investigation into problems with the company's payroll system. Jane Rossman, applications manager, assigned systems analyst Rick Williams to conduct a preliminary investigation.

Rick found several problems, including input errors and a need for manual preparation of reports. The payroll department often required overtime to correct those errors and produce the reports.

The IT department recommended an in-depth analysis of the problem areas and Mr. Jeremy approved the study. Now, as the systems analysis phase begins, the next step is requirements modeling.

Human Resources Department Interview

During the preliminary investigation phase, Rick prepared the organization chart of the human resources department shown in Figure 4-31.

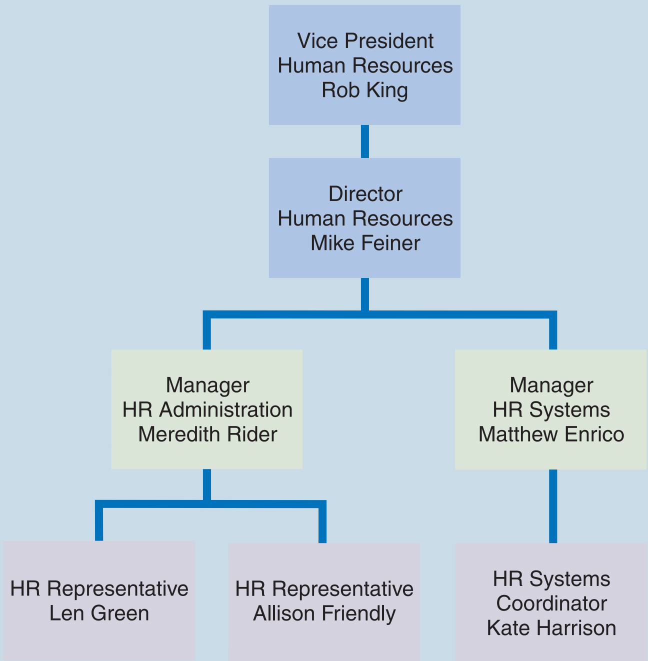


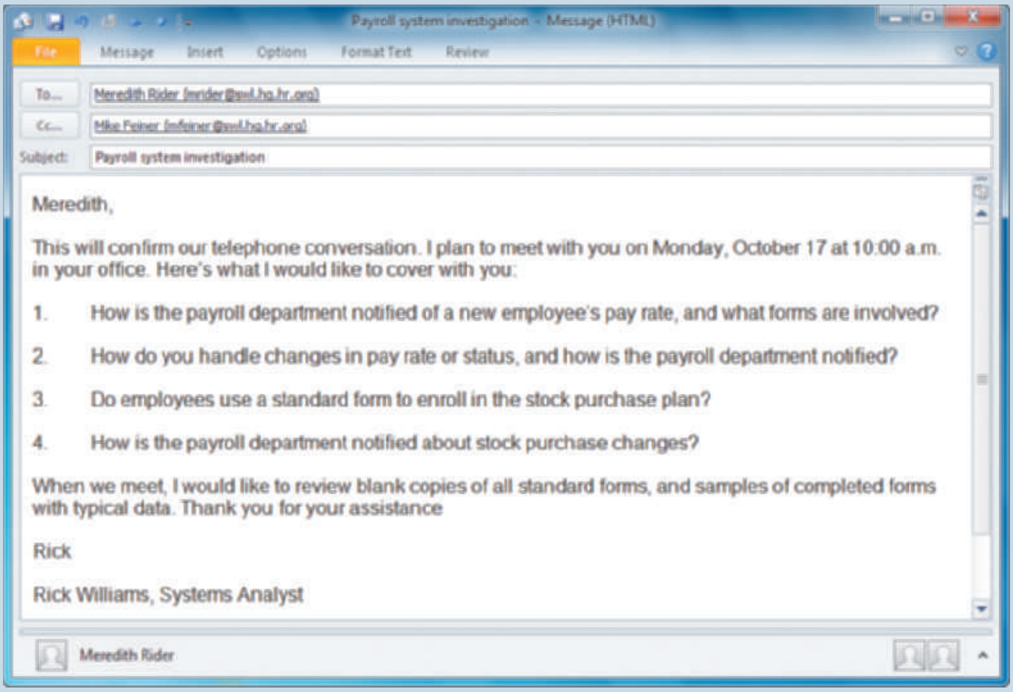
FIGURE 4-31 Human resources department organization chart.

Rick learned that some errors involved employee stock purchase deductions, so he decided to study that process. He knew that the human resources department initiates stock purchase deductions, so he decided to interview Meredith Rider, manager of human resources administration. Meredith is responsible for completing the paperwork for newly hired employees and sending the forms to the payroll department.

Rick called Meredith to make an appointment and sent her the confirmation message shown in Figure 4-32 that described the topics and requested copies of related forms.

Chapter Capstone Case: SoftWear, Limited (continued)

SWL


Payroll system investigation - Message (HTML)

File Message Insert Options Format Text Review

To... Meredith Rider (mrider@swl.hq.hr.org)

Cc... Mike Feiner (mfeiner@swl.hq.hr.org)

Subject: Payroll system investigation

Meredith,

This will confirm our telephone conversation. I plan to meet with you on Monday, October 17 at 10:00 a.m. in your office. Here's what I would like to cover with you:

1. How is the payroll department notified of a new employee's pay rate, and what forms are involved?
2. How do you handle changes in pay rate or status, and how is the payroll department notified?
3. Do employees use a standard form to enroll in the stock purchase plan?
4. How is the payroll department notified about stock purchase changes?

When we meet, I would like to review blank copies of all standard forms, and samples of completed forms with typical data. Thank you for your assistance.

Rick

Rick Williams, Systems Analyst

Meredith Rider

FIGURE 4-32 Rick Williams's message to Meredith Rider regarding preparation for the interview.

In the interview, Meredith explained that new employees fill in the top portion of a Payroll Master Record Form (Form PR-1). The human resources department then adds the pay rate and other data and sends a copy of the PR-1 form to the payroll department. Meredith showed Rick a blank copy of an online PR-1 form shown in Figure 4-33. She explained that because payroll and personnel information is confidential, she could not give Rick a completed form.


PAYROLL MASTER RECORD FORM

(Form PR-1)

SSN

Name — last, first, middle

Address — street, city, state, zip

Department

Job title

Pay rate — per

Status ☐ Exempt ☐ Non-exempt ☐ Orientation ☐ Insurance

☐ W-4 OK ☐ Citizenship OK ☐ Credit Union ☐ Handbook

Checked by — Date

FIGURE 4-33 Payroll Master Record Form (Form PR-1).

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

SWL

Also, when an employee's pay rate or status changes, the human resources department completes the online Payroll Status Change Form (Form PR-2) shown in Figure 4-34 and sends a copy to the payroll department.

PAYROLL STATUS CHANGE FORM

(Form PR-2)

SSN

Name
last first middle

Department **Location**

Status change ☐ Old: New:

Pay change ☐ Old: per New: per

Position change ☐ Old: New:

Human Resources Department Review by:

Date

FIGURE 4-34 Payroll Status Change Form (Form PR-2).

Meredith also explained that after a 90-day probationary period, employees can participate in the SWL Credit Union. An employee submits the Payroll Deduction Change Form (Form PR-3) shown in Figure 4-35 to the human resources department, which forwards it to the payroll department.

PAYROLL DEDUCTION CHANGE FORM

(Form PR-3)

SSN **Date**

Name
last first middle

Department **Location**

W-4 exemption change ☐ Old: New:

Credit Union change ☐ Old: New:

Position change ☐ Old: New:

(For Human Resources Department Use Only)

Effective date of change

By: Human Resources Department

FIGURE 4-35 Payroll Deduction Change Form (Form PR-3).

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued) SWL

SWL also has an Employee Stock Purchase Plan, which an employee can join after 180 days. To enroll, the employee completes an Employee Stock Purchase Plan Enrollment and Change Form (Form PR-4). The human resources department prepares a weekly report of all stock plan enrollments and changes on the Employee Stock Purchase Plan Weekly Deduction Summary Report (Form PR-5) shown in Figure 4-36 and sends a copy to the payroll department.

EMPLOYEE STOCK PURCHASE PLAN
Weekly Deduction Summary Report

(Form PR-5) Week ending:

Code: N= New C = Change	**SSN**	**Employee Name**	**Deduction Amount**

FIGURE 4-36 Employee Stock Purchase Plan Weekly Deduction Summary Report (Form PR-5).

After the interview with Meredith, Rick sent the follow-up message shown in Figure 4-37 and attached a copy of the interview documentation shown in Figure 4-37.

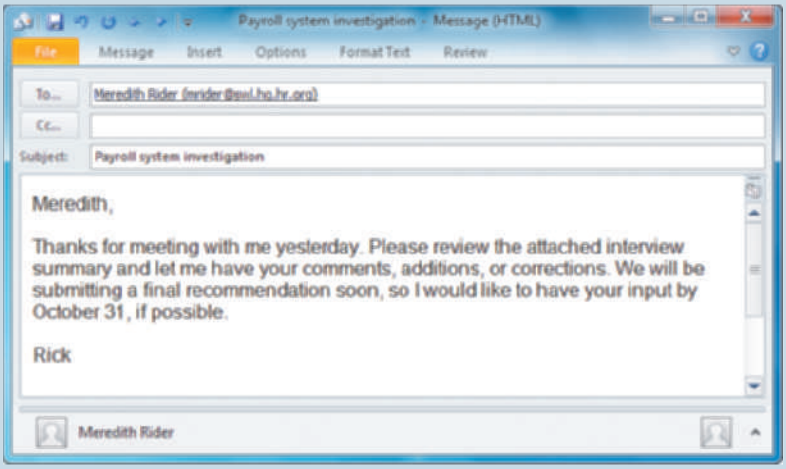


FIGURE 4-37 Follow-up message from Rick Williams to Meredith Rider, with a request for her comments on the interview summary.

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

Name of System: Payroll
Date: October 21, 2011
Prepared by: Rick Williams
Title: Systems Analyst
Purpose: Interview Summary: Meredith Rider, Manager of Human Resources Administration
Location: Raleigh

Five basic forms are used by the human resources department that relate to the payroll system:
I. Payroll Master Record Form (Form PR-I)
2. Payroll Status Change Form (Form PR-2)
3. Payroll Deduction Change Form (Form PR-3)
4. Employee Stock Purchase Plan Enrollment and Change Form (Form PR-4)
S. Employee Stock Purchase Plan Weekly Deduction Summary Report (Form PR-S)

When an employee is hired, the following takes place:
I. The human resources department prepares a Payroll Master Record Form (Form PR-I) with employee data, including Social Security number, name, address, telephone, emergency contact, and information about the position, title, and initial pay rate.
2. A copy of this form is sent to the payroll department, where it is filed and maintained.
3. Subsequent pay rate or status changes are submitted by the human resources department to the payroll department on a Payroll Status Change Form (Form PR-2). Payroll then files these change forms with the employee's PR-l form.

After 180 days of employment, the employee is eligible to enroll in the SWL Stock Purchase Plan.
1. To enroll, an employee completes an Employee Stock Purchase Plan Enrollment and Change Form (Form PR-4).
2. The human resources department prepares an Employee Stock Purchase Plan Weekly Deduction Summary Report Form (Form PR-S) and sends it to the payroll department with copies of the PR-4 forms, which then are filed with the employee's PR-I form.

I have identified several problems with the current procedures:
1. Data errors can occur when the human resources staff prepares the weekly summary of employee stock purchase deductions, and no system verification takes place until incorrect deductions are reported.
2. The system performs no verification of employment dates, and it is possible that the 90- and 180-day eligibility periods are applied incorrectly.
3. The filing of the PR-2, PR-3, and PR-4 forms with the Payroll Master Record Forms in the payroll department could lead to problems. If any of the forms are lost or misfiled, incorrect data is entered into the s stem.

FIGURE 4-38 Documentation of the interview with Meredith Rider.

Payroll Department Interview

Rick's next interview was with the lead payroll clerk, Nelson White. Nelson confirmed that when an employee is hired, a PR-1 form is completed in the human resources department, and a copy is sent to payroll. He also explained that each week, the payroll department sends a time sheet to every SWL department manager. The time sheet lists the employees, with space to record regular hours, vacation, sick leave, jury duty, and other codes.

After each pay period, SWL managers complete the time sheets and return them to the payroll department. Payroll then enters the pay rates and deduction information, and delivers the sheets to Business Information Systems (BIS), the service bureau that prepares SWL's payroll.

After BIS runs the payroll, it returns the time sheets, paychecks, and the payroll register to SWL. The director of payroll, Amy Calico, sends the paychecks to SWL department heads for distribution to employees.

Nelson uses the weekly payroll register to prepare a report of credit union deductions and a check to the credit union for the total amount deducted. Stock purchases, on the other hand, are processed monthly, based on the stock's closing price on the last business day of

Chapter Capstone Case: SoftWear, Limited (continued)

the month. Using the weekly payroll registers, Nelson manually prepares a monthly report of employee stock purchases and forwards a copy of the report and a funds transfer authorization to Carolina National Bank, which is SWL's stock transfer agent.

Rick asked Nelson why BIS did not produce a report on employee stock purchase deductions. Nelson replied that although the payroll is run weekly, the stock deductions are invested only once a month. Because the two cycles do not match, the BIS system could not handle the task.

Nelson then referred Rick to the SWL Systems and Procedures Manual page that describes how monthly Employee Stock Purchase Plan investment amounts are calculated, as shown in Figure 4-39. After blanking out the employee's name and Social Security number, Nelson also gave Rick a sample of two monthly deduction registers, as shown in Figure 4-40.

Procedures Manual

Employee Stock Purchase Plan

To enroll in the SWL stock purchase plan, an employee submits a PR-4 form. Human resources sends a copy of the form to the payroll department. At the end of the month, accumulated deductions for that month are invested in shares of SWL stock at the current market price.

However, because weeks and months do not match up exactly, the following calculation must be used:

a. Divide the weekly deduction by seven to get a daily deduction rate.
b. Then multiply the number of days in the month times the daily deduction rate. The result is the total stock investment amount for that month.

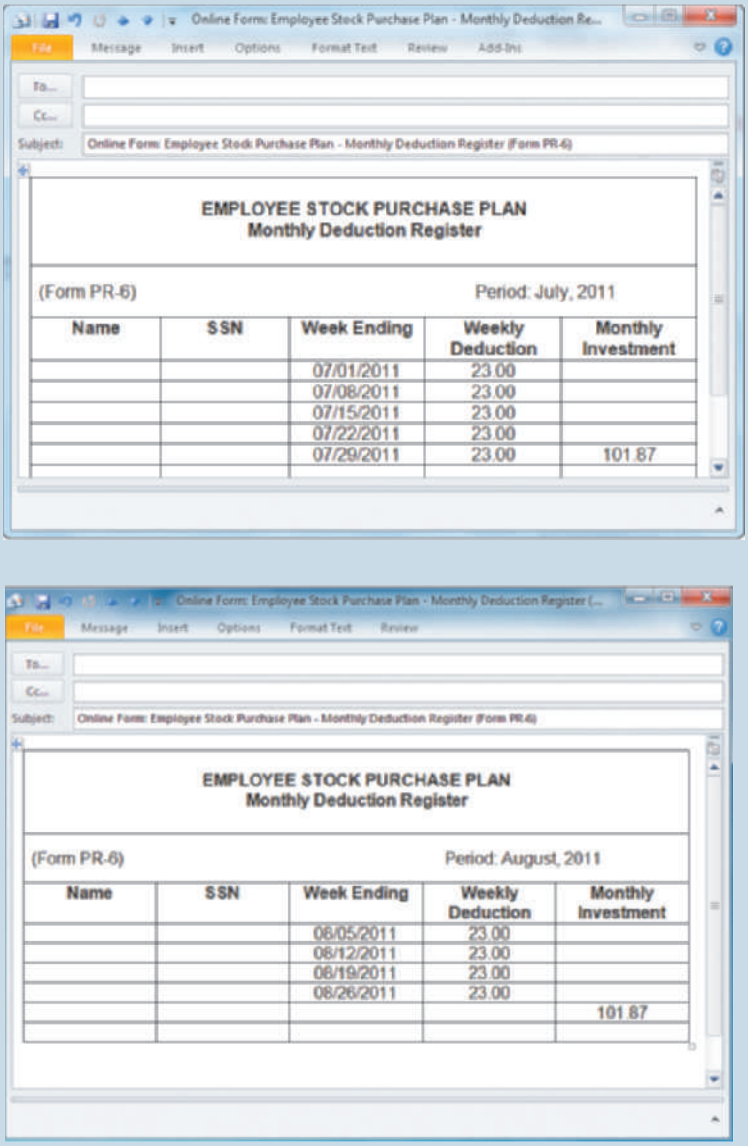
Here is an example for the month of January:

- Employee A authorizes a weekly stock plan deduction of $20.00.
- $20.00 divided by 7 = a $2.857 daily deduction rate.
- January has 31 calendar days, so 31 × $2.857 = $88.57, which will be the stock investment amount for January.

At the end of each month, the payroll department prepares a deduction register (PR-6) that shows weekly deductions and monthly totals.

FIGURE 4-39 Sample page from SWL Systems and Procedures Manual.

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)



EMPLOYEE STOCK PURCHASE PLAN
Monthly Deduction Register

(Form PR-6) Period: July, 2011

Name	SSN	Week Ending	Weekly Deduction	Monthly Investment
		07/01/2011	23.00	
		07/08/2011	23.00	
		07/15/2011	23.00	
		07/22/2011	23.00	
		07/29/2011	23.00	101.87

EMPLOYEE STOCK PURCHASE PLAN
Monthly Deduction Register

(Form PR-6) Period: August, 2011

Name	SSN	Week Ending	Weekly Deduction	Monthly Investment
		08/05/2011	23.00	
		08/12/2011	23.00	
		08/19/2011	23.00	
		08/26/2011	23.00	
				101.87

FIGURE 4-40 Sample of the ESIP Monthly Deduction Register for July and August, 2009.

Rick began to see why it was taking so much effort to prepare the reports. The interview with Nelson provided much more detail than the general description that Rick had received during the preliminary investigation.

BIS Interview

Rick decided that he should talk with someone at the BIS service bureau to find out more about its operations. He learned from Nelson that Linda DeMarco was BIS's customer relations manager, so he scheduled an appointment with her.

When Rick arrived at BIS, Linda greeted him warmly. She explained that she had planned to meet with members of SWL's payroll department within the next month or two to discuss the latest developments. Because Rick now was working on SWL's payroll system, however, this meeting would save her a trip. Rick temporarily abandoned his interview plan and asked Linda what she had in mind.

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

"The payroll system that your company is using, which we call GAPP, for Generalized Automated Payroll Program, originally was developed here at BIS about eight years ago," Linda began. "In fact, SoftWear, Limited was one of our very first customers. We've worked together for a long time, and we are very committed to your firm. As you know, GAPP was modified and updated many times. But let's face it, even with the patches, GAPP is an antique! Anyway, I have some exciting news. We decided to develop a new, state-of-the-art payroll system. We are going to call it CHIPS, for Comprehensive High-powered Interactive Payroll System. I'm looking forward to working with SWL when you switch over to CHIPS," Linda said.

Rick took a few moments to consider this surprising development. He then asked what would happen with GAPP. Linda stated that GAPP would be available to customers for another year or two, but that BIS would make no further enhancements to the system. Using BIS resources to maintain an obsolete system would not make sense, she explained.

Before this meeting, Rick had hoped that BIS could make some minor changes to solve SWL's payroll problems. He now realized that was impossible, so he decided to learn more about CHIPS.

Rick described the problem with the mismatched deduction cycles and asked if CHIPS would handle that. Linda said that she already had looked into the matter. She pointed out that SWL was their only customer with more than one deduction application cycle. From BIS's point of view, programming CHIPS to handle multiple cycle reports did not make sense. Linda suggested that perhaps a special add-on module could be written, once CHIPS was up and running. BIS could do that kind of job on a contract basis, she added.

Rick then asked when the new system would be available and what the cost would be. Linda stated that current plans were to begin offering CHIPS sometime in the following year. She explained that the system was still in development, and she could not be more specific about timetables and costs. She was sure, however, that the monthly fee for CHIPS would not increase more than 30 percent above the current GAPP charges.

As Rick was preparing to leave, Linda urged him to keep in touch. In the next few months, she explained, plans for CHIPS would become more specific, and she would be able to answer all his questions.

New Developments

When Rick returned from his meeting with Linda, he immediately went to his manager, Jane Rossman. After he described his visit to BIS, Jane telephoned Ann Hon, director of information technology. Within the hour, Jane and Rick held a meeting with Ann in her office. Rick repeated the details of his visit, and Ann asked for his opinion on how the developments at BIS would affect SWL's current systems analysis.

Rick explained that one of the problems — possible input errors when transferring data from the human resources summary list — might be solved easily by developing a new form or procedure. Nevertheless, he saw no obvious solutions for the stock purchase deduction problems, except to change the scope of the payroll project.

Jane, Rick, and Ann then analyzed the situation. They all agreed that because of the upcoming changes at BIS, the current payroll system project would produce very limited results and should be expanded in scope. They totaled the costs of the SWL project to that point and prepared estimates for a detailed investigation of the entire payroll system in order to meet SWL's current and future needs.

Later that week, Ann met with Michael Jeremy, vice president of finance, to discuss the situation and present her proposal to expand the project. Before she even started, however, Mr. Jeremy filled her in on the latest announcement from SWL's top management: The company had decided to move forward with the new Employee Savings and Investment Plan (ESIP)

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

under consideration. He said that in December, Robert Lansing, SWL's president, would announce a target date of April 1, 2012, for the new ESIP plan. Mr. Jeremy explained that the new plan would be a 401(k) plan with tax advantages for employees.

Facing the new constraints on top of the existing payroll system problems, it looked like SWL would need a new payroll system after all.

The Revised Project

Jane Rossman assigned Carla Moore, a programmer-analyst, to work with Rick Williams on the revised system project. Because they now had to determine the requirements for the complete payroll system, Rick and Carla conducted follow-up interviews with Nelson White and Meredith Rider, as well as Allison Friendly, a human resources representative, and both payroll clerks, Britton Ellis and Debra Williams. During the payroll department interviews, the payroll staff prepared samples of all the existing payroll reports. At the end of the fact-finding process, Rick and Carla decided to prepare the functional decomposition diagram shown in Figure 4-41. The diagram shows the main functions identified during the interviews.

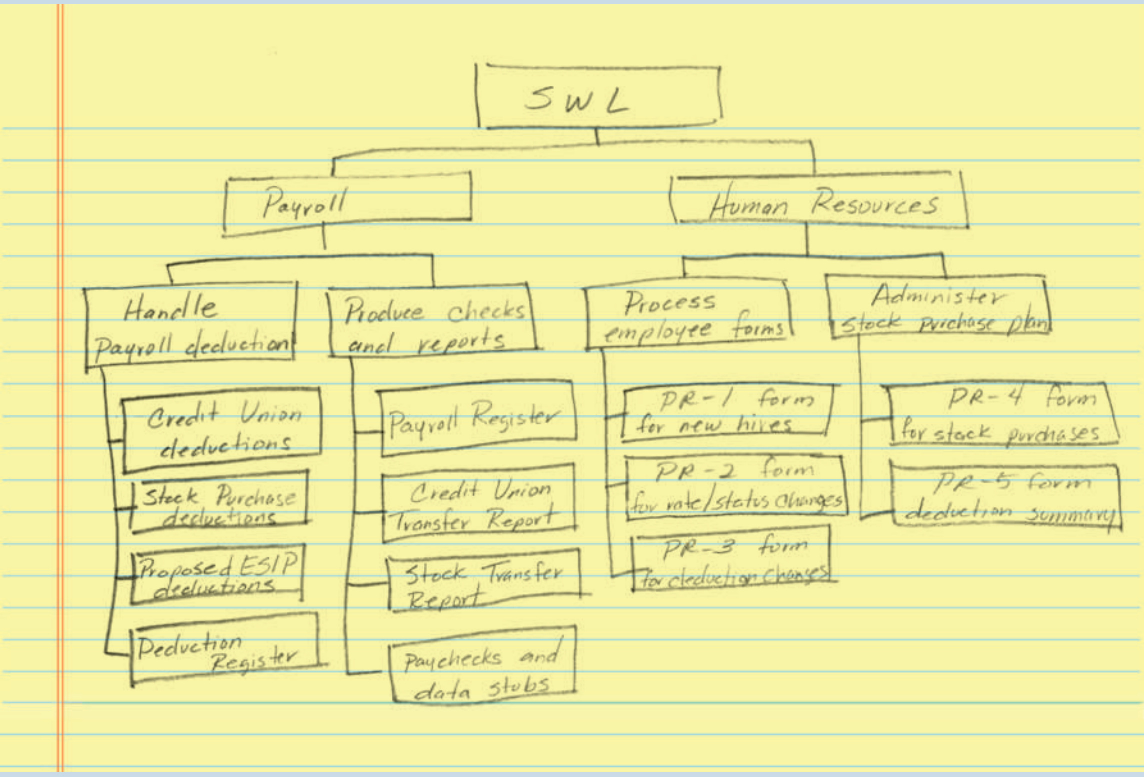


FIGURE 4-41 A functional decomposition diagram (FDD) shows the main functions that were identified during the interviews.

The Payroll Register is shown in Figure 4-42. Each employee is listed on a separate line, along with his or her earnings, deductions, and net pay. BIS creates three copies of this report each week. One copy is sent to Michael Jeremy, and one copy goes to Amy Calico. The third copy is used by the payroll department for determining SWL's obligation for tax withholding and FICA payments and for applying credit union and stock purchase plan deductions.

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

SWL

PAYROLL REGISTER

Week Ending ______________ ***Page 1***

Employee Data		**Earnings**			**Deductions**					**Net Pay**	
Name	SSN	Regular Pay	Overtime Pay	Total Pay	Federal Tax	State Tax	FICA	Credit Union	Stock Plan	Net Amount	Check Number

FIGURE 4-42 Sample page of SWL Payroll Register report.

Mr. Jeremy receives a weekly overtime report from BIS that lists every employee who worked overtime that week. When Carla asked him about that report, he stated that he consulted it occasionally but admitted that he did not need the report every week. He also receives an accounting report, but he routinely forwards it to the accounting department. He mentioned that an overall financial summary was more valuable to him.

SWL Team Tasks

1. When Rick Williams met with Meredith Rider in the human resources department, he asked for copies of actual reports and forms that contained confidential information, but Meredith declined to provide them. Rick has asked you to suggest a reasonable compromise between confidentiality requirements and the need for analysts to review actual records, instead of fictitious data. Think about this, and write a message to Rick with your views.
2. Assume that you were with Rick at the meeting with Linda DeMarco. Review the fact statement, then write an interview summary that documents the main topics that Rick and Linda discussed.
3. Rick asked you to design a questionnaire that would measure employee satisfaction with the current payroll deduction system. Review the sample questionnaire in the chapter, and prepare a draft for Rick. Rick also wants you to suggest various sampling methods so he can make a choice. Include a brief description of various methods, and be sure to include your recommendation and reasons.
4. Rick wants you to interview several employees to learn more about their levels of satisfaction with the current system. Prepare a set of interview questions, and be sure to include at least examples of open-ended, closed-ended, and range-of-response questions. If possible, conduct role-play interviews with other students.

Manage the SWL Project

You have been asked to manage SWL's new information system project. One of your most important activities will be to identify project tasks and determine when they will be performed. Before you begin, you should review the SWL case in this chapter. Then list and analyze the tasks, as follows:

LIST THE TASKS Start by listing and numbering at least 10 tasks that the SWL team needs to perform to fulfill the objectives of this chapter. Your list can include SWL Team Tasks and any other tasks that are described in this chapter. For example, Task 3 might be to Identify people to interview, and Task 6 might be to Conduct interviews.

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

ANALYZE THE TASKS Now study the tasks to determine the order in which they should be performed. First identify all concurrent tasks, which are not dependent on other tasks. In the example shown in Figure 4-43, Tasks 1, 2, 3, 4, and 5 are concurrent tasks, and could begin at the same time if resources were available.

Other tasks are called dependent tasks, because they cannot be performed until one or more earlier tasks have been completed. For each dependent task, you must identify specific tasks that need to be completed before this task can begin. For example, you would need to identify the people to interview before you conducted the interviews, so Task 6 cannot begin until Task 3 is completed, as Figure 4-43 shows.

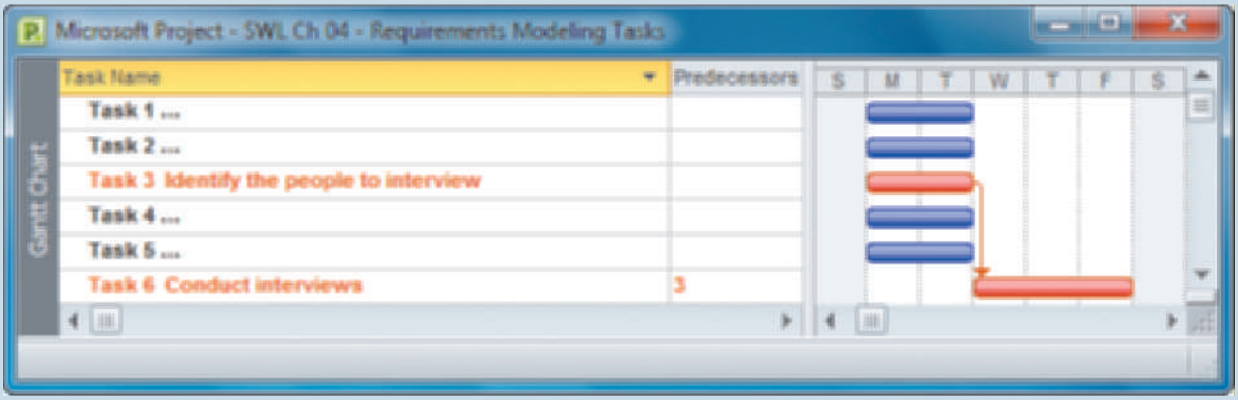


FIGURE 4-43 Tasks 1, 2, 3, 4, and 5 are concurrent tasks that could be performed at the same time. Task 6 is a dependent task that cannot be performed until Task 3 has been completed.

Chapter 3 describes project management tools, techniques, and software. To learn more, you can use the Features section on your Student Study Tool CD-ROM, or visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and locate the project management resources library for this book. On the Web, Microsoft offers demo versions, training, and tips for using Project 2010. You also can visit the OpenWorkbench.org site to learn more about this free, open-source software.

Ready for a Challenge?

In additional to technical skills, IT professionals need critical thinking skills such as perception, organization, analysis, problem-solving, and decision-making. The Ready for a Challenge feature can help you learn, practice, and apply critical thinking skills that you can take to the workplace.

As a leader in the game development field, Game Technology has many customers. The company wants to add a customer contact management feature to the sales system, and you will assist the IT team on this project. One of your first tasks is to interview users to find out what functions they want. When you have this information, you can construct a functional decomposition diagram (FDD) to list and organize the functions.

Before you talk to the users, you decide to practice your interviewing skills. Specifically, you want to use a mix of open-ended, closed-ended, and range-of-response questions.

To test yourself, you ask a team member to develop some practice questions for you to identify, as follows:

Question	Code: O = open-ended C = closed-ended R = range-of-response
Do you foresee any problems with this system? On a scale of one (low) to five (high), how important is the project? Do you maintain current e-mail addresses for customers? Who is responsible for entering these records? What would be some benefits of the new feature?	

After you finish the interviews, you study the following results before preparing the FDD.

Interview results

The starting point should be an overview screen that allows users to look up a customer, add a customer, edit a customer, delete a customer, or get help. The Help option should allow a user to search a knowledge base or contact the IT Help Desk. If users select the knowledge base option, they can search by keyword or by topic. If they select the IT Help Desk, they can either select e-mail or a telephone call-back option.

Practice Tasks

A. For each question listed, enter a code that correctly identifies the question.
B. Draw an FDD that follows the guidelines in the textbook.

After you complete the Practice Tasks, to check your work and view sample answers, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to the resources for this chapter, and locate Ready for a Challenge?.

The Challenge

The interviews were successful, but you want more details. Specifically, you want to learn more about the user interface, built-in error checking, and reports that could be generated for users. Using these topics, or others you think would be relevant, prepare two samples of each type of question.

Also, your FDD was good, but now you must add a Reports function to the overview screen. By selecting that option, users should be able to list all reports by name or by type. Users also should be able to view or print a report. If they select the Print option, they should be able to print the full report, the current page, or selected pages.

Challenge Tasks

A. Prepare the six questions described above (two of each type), indicating the type for each question.
B. Draw an FDD that shows the Reports function and subfunctions.

CHAPTER 5 Data and Process Modeling

Chapter 5 is the second of four chapters in the systems analysis phase of the SDLC. This chapter discusses data and process modeling techniques that analysts use to show how the system transforms data into useful information. The deliverable, or end product, of data and process modeling is a logical model that will support business operations and meet user needs.

INTRODUCTION

OBJECTIVES

When you finish this chapter, you will be able to:

- Describe data and process modeling concepts and tools, including data flow diagrams, a data dictionary, and process descriptions
- Describe the symbols used in data flow diagrams and explain the rules for their use
- Draw data flow diagrams in a sequence, from general to specific
- Explain how to level and balance a set of data flow diagrams
- Describe how a data dictionary is used and what it contains
- Use process description tools, including structured English, decision tables, and decision trees
- Describe the relationship between logical and physical models

During the requirements modeling process described in Chapter 4, you used fact-finding techniques to investigate the current system and identify user requirements. Now, in Chapters 5 and 6 you will use that information to develop a logical model of the proposed system and document the system requirements. A **logical model** shows *what* the system must do, regardless of how it will be implemented physically. Later, in the systems design phase, you build a **physical model** that describes *how* the system will be constructed. Data and process modeling involves three main tools: data flow diagrams, a data dictionary, and process descriptions.

Chapter 5 includes four Video Learning Sessions that show you how to work with DFD symbols and diagrams, how to create a context diagram, how to create a diagram 0 DFD, and how to use a decision table.

CHAPTER INTRODUCTION CASE: Mountain View College Bookstore

Background: Wendy Lee, manager of college services at Mountain View College, wants a new information system that will improve efficiency and customer service at the three college bookstores.

In this part of the case, Tina Allen (systems analyst) and David Conroe (student intern) are talking about data and process modeling tasks and concepts.

Participants: Tina and David
Location: Tina's office, Monday afternoon, October 17, 2011.
Project status: Tina and David have completed fact-finding for the new system and are ready to develop a requirements model using various diagrams and a data dictionary that will describe and document the proposed system.
Discussion topics: Data flow diagrams, data dictionaries, and process description tools

Tina: Hi, David. Any questions about the fact-finding we did?

David: *Well, I found out that fact-finding is hard work.*

Tina: Yes, but it was worth it. Look at what we learned — now we understand how the current system operates, and we know what users expect in the new system. This information will help us build a requirements model that we can present to Wendy and her staff.

David: *What's the next step?*

Tina: We need to draw a set of data flow diagrams, or DFDs for short.

David: *Do we use a CASE tool to draw the DFDs?*

Tina: We can draw the initial versions by hand. We'll use a CASE tool to prepare the final version of the diagrams.

David: *What goes into a DFD?*

Tina: DFDs use four basic symbols that represent processes, data flows, data stores, and entities. You'll learn about these as we go along. I'll also show you how we use techniques called leveling and balancing to develop accurate, consistent DFDs.

David: *Apart from the diagrams, do we need to develop any other documentation?*

Tina: Yes, we need to create a data dictionary and process descriptions. The data dictionary is an overall storehouse of information about the system, and serves as a central clearinghouse for all documentation. We use process descriptions to explain the logical steps that each process performs. To create these descriptions, we use three tools: structured English statements, decision tables, and decision trees.

David: *Sounds like a lot to do. Where do we begin?*

Tina: Here's a task list to get us started:

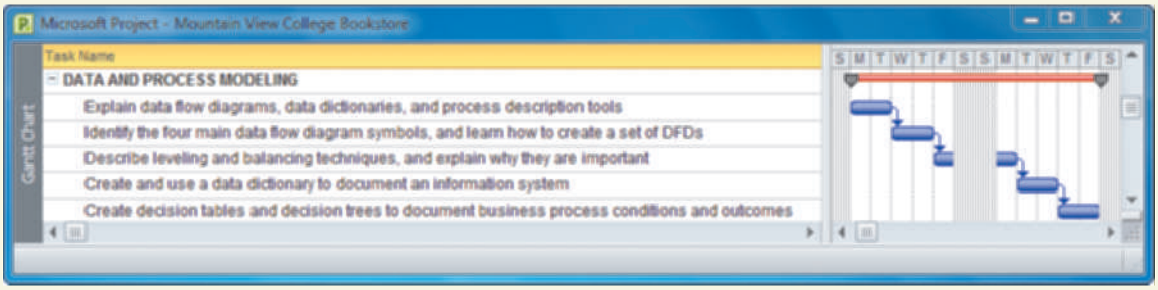


FIGURE 5-1 Typical data and process modeling task list.

Overview of Data and Process Modeling Tools

TOOLKIT TIME

The CASE Tools in Part B of the Systems Analyst's Toolkit can help you document business functions and processes, develop graphical models, and provide an overall framework for information system development. To learn more about these tools, turn to Part B of the four-part Toolkit that follows Chapter 12.

Systems analysts use many graphical techniques to describe an information system. One popular method is to draw a set of data flow diagrams. A **data flow diagram (DFD)** uses various symbols to show how the system transforms input data into useful information. Other graphical tools include object models, which are explained in Chapter 6 (Object Modeling), and entity-relationship diagrams, which are described in Chapter 9 (Data Design).

Data Flow Diagrams

In Part A of the Systems Analyst's Toolkit, you learn how to use visual aids to help explain a concept, as shown in Figure 5-2. Similarly, during the systems analysis phase, you learn how to create a visual model of the information system using a set of data flow diagrams.

A data flow diagram (DFD) shows how data moves through an information system but does not show program logic or processing steps. A set of DFDs provides a logical model that shows *what* the system does, not *how* it does it. That distinction is important because focusing on implementation issues at this point would restrict your search for the most effective system design.

FIGURE 5-2 Systems analysts often use visual aids during presentations.

VIDEO LEARNING SESSION: DFD Symbols and Diagrams

Video Learning Sessions can help you understand key concepts, practice your skills, and check your work. To access the sessions, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and navigate to the **Video Learning Sessions** for this book. This session is about DFD symbols and diagrams. You'll learn why DFDs are important modeling tools, how to use DFD symbols, and how you can use a CASE tool to create DFDs.

DFD Symbols

DFDs use four basic symbols that represent processes, data flows, data stores, and entities. Several different versions of DFD symbols exist, but they all serve the same purpose. DFD examples in this textbook use the **Gane and Sarson** symbol set. Another popular symbol set is the **Yourdon** symbol set. Figure 5-3 shows examples of both versions. Symbols are referenced by using all capital letters for the symbol name.

PROCESS SYMBOL A **process** receives input data and produces output that has a different content, form, or both. For instance, the process for calculating pay uses two inputs (pay rate and hours worked) to produce one output (total pay). Processes can be very simple or quite complex. In a typical company, processes might include calculating sales trends, filing online insurance claims, ordering inventory from a supplier's system, or verifying e-mail addresses for Web customers. Processes contain the **business logic**, also called **business rules**, that transform the data and produce the required results.

The symbol for a process is a rectangle with rounded corners. The name of the process appears inside the rectangle. The process name identifies a specific function and consists of a verb (and an adjective, if necessary) followed by a singular noun. Examples of process names are APPLY RENT PAYMENT, CALCULATE COMMISSION, ASSIGN FINAL GRADE, VERIFY ORDER, and FILL ORDER.

Processing details are not shown in a DFD. For example, you might have a process named DEPOSIT PAYMENT. The process symbol does not reveal the business logic for the DEPOSIT PAYMENT process. To document the logic, you create a process description, which is explained later in this chapter.

ON THE WEB

To learn more about Yourdon symbols, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Yourdon Symbols link.

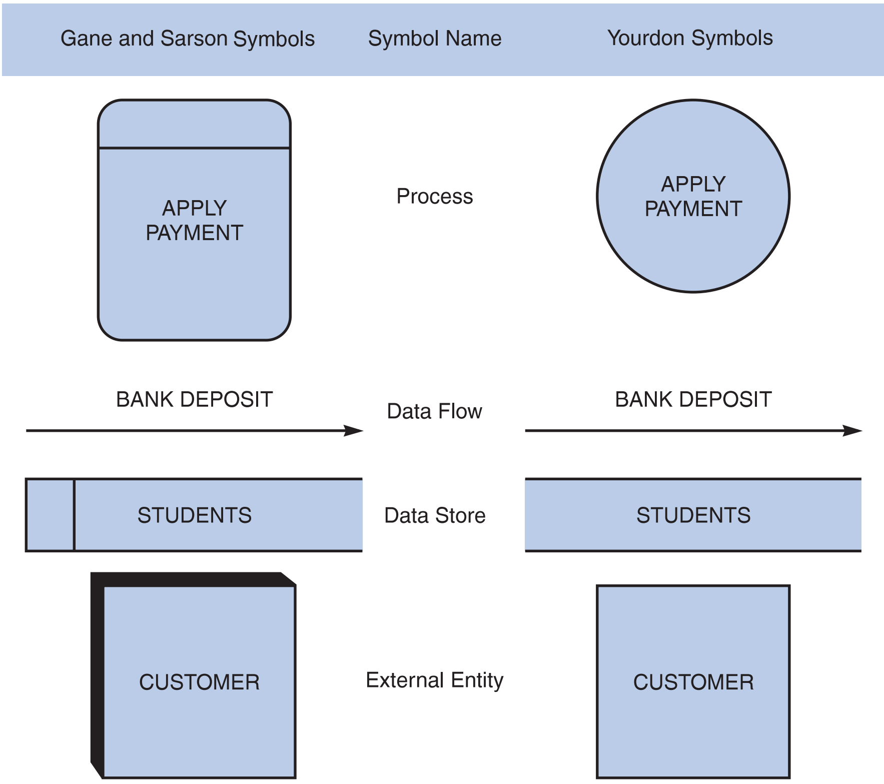


FIGURE 5-3 Data flow diagram symbols, symbol names, and examples of the Gane and Sarson and Yourdon symbol sets.

In DFDs, a process symbol can be referred to as a **black box**, because the inputs, outputs, and general functions of the process are known, but the underlying details and logic of the process are hidden. By showing processes as black boxes, an analyst can create DFDs that show how the system functions, but avoid unnecessary detail and clutter. When the analyst wishes to show additional levels of detail, he or she can zoom in on a process symbol and create a more in-depth DFD that shows the process's internal workings — which might reveal even more processes, data flows, and data stores. In this manner, the information system can be modeled as a series of increasingly detailed pictures.

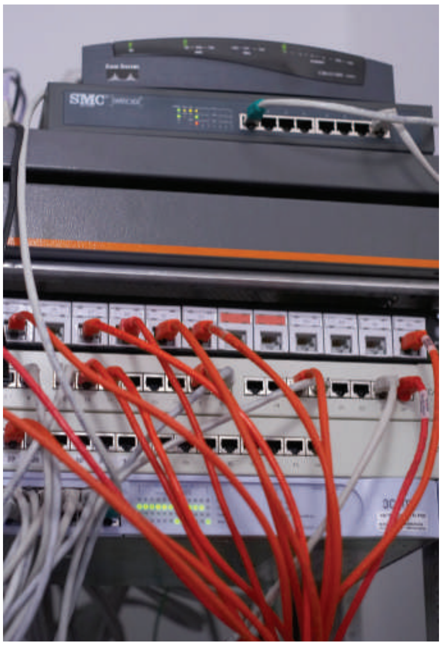


FIGURE 5-4 Networks use various devices that act like *black boxes*. Cables carry data in and out, but internal operations are hidden inside the case.

The network router shown in Figure 5-4 is an example of a black box. An observer can see cables that carry data into and out of the router, but the router's internal operations are not revealed — only the results are apparent.

DATA FLOW SYMBOL A **data flow** is a path for data to move from one part of the information system to another. A data flow in a DFD represents one or more data items. For example, a data flow could consist of a single data item (such as a student ID number) or it could include a set of data (such as a class roster with student ID numbers, names, and registration dates for a specific class). Although the DFD does not show the detailed contents of a data flow, that information is included in the data dictionary, which is described later in this chapter.

The symbol for a data flow is a line with a single or double arrowhead. The data flow name appears above, below, or alongside the line. A data flow name consists of a singular noun and an adjective, if needed. Examples of data flow names are DEPOSIT, INVOICE PAYMENT, STUDENT GRADE, ORDER, and COMMISSION. Exceptions to the singular name rule are data flow names, such as GRADING PARAMETERS, where a singular name could mislead you into thinking a single parameter or single item of data exists.

Figure 5-5 shows correct examples of data flow and process symbol connections. Because a process changes the data's content or form, at least one data flow must enter and one data flow must exit each process symbol, as they do in the CREATE INVOICE process. A process symbol

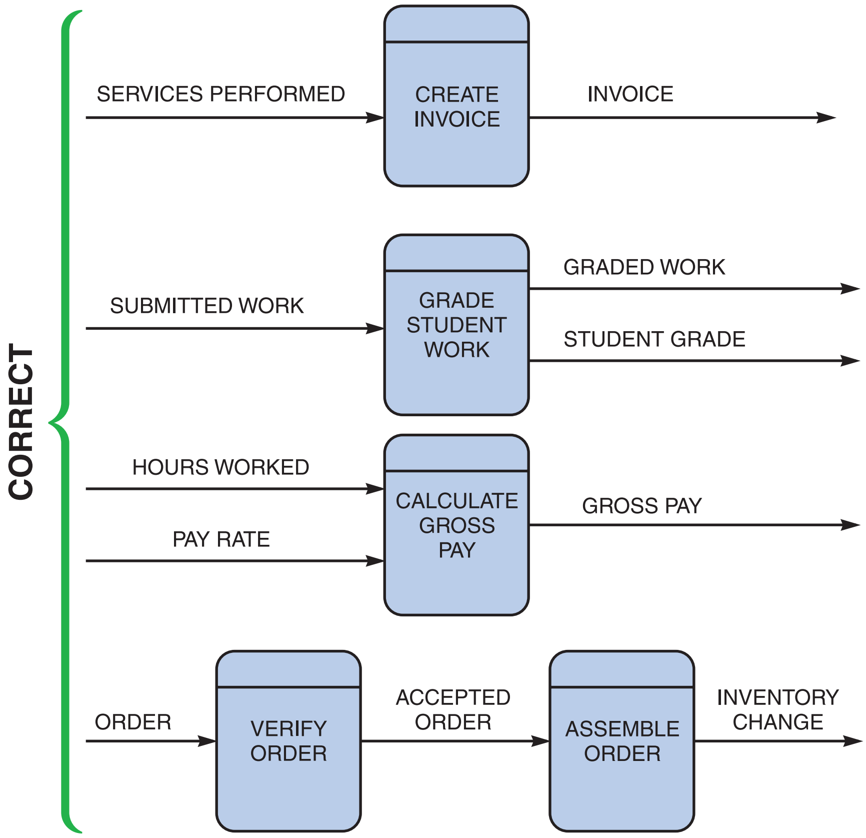


FIGURE 5-5 Examples of correct combinations of data flow and process symbols.

can have more than one outgoing data flow, as shown in the GRADE STUDENT WORK process, or more than one incoming data flow, as shown in the CALCULATE GROSS PAY process. A process also can connect to any other symbol, including another process symbol, as shown by the connection between VERIFY ORDER and ASSEMBLE ORDER in Figure 5-5. A data flow, therefore, *must* have a process symbol on at least one end.

Figure 5-6 shows three data flow and process combinations that you must avoid:

- **Spontaneous generation.** The APPLY INSURANCE PREMIUM process, for instance, produces output, but has no input data flow. Because it has no input, the process is called a spontaneous generation process.
- **Black hole.** The CALCULATE GROSS PAY is called a black hole process, which is a process that has input, but produces no output.
- **Gray hole.** A gray hole is a process that has at least one input and one output, but the input obviously is insufficient to generate the output shown. For example, a date of birth input is not sufficient to produce a final grade output in the CALCULATE GRADE process.

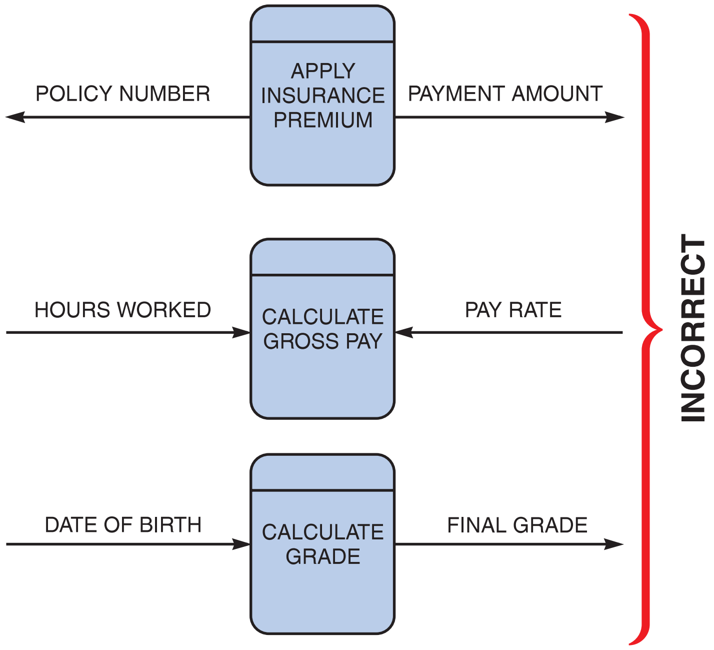


FIGURE 5-6 Examples of incorrect combinations of data flow and process symbols. APPLY INSURANCE PREMIUM has no input and is called a spontaneous generation process. CALCULATE GROSS PAY has no outputs and is called a black hole process. CALCULATE GRADE has an input that is obviously unable to produce the output. This process is called a gray hole.

Spontaneous generation, black holes, and gray holes are impossible logically in a DFD because a process must act on input, shown by an incoming data flow, and produce output, represented by an outgoing data flow.

DATA STORE SYMBOL A **data store** is used in a DFD to represent data that the system stores because one or more processes need to use the data at a later time. For instance, instructors need to store student scores on tests and assignments during the semester so they can assign final grades at the end of the term. Similarly, a company stores employee salary and deduction data during the year in order to print W-2 forms with total earnings and deductions at the end of the year. A DFD does not show the detailed contents of a data store — the specific structure and data elements are defined in the data dictionary, which is discussed later in this chapter.

The physical characteristics of a data store are unimportant because you are concerned only with a logical model. Also, the length of time that the data is stored is unimportant — it can be a matter of seconds while a transaction is processed or a period of months while data is accumulated for year-end processing. What is important is that a process needs access to the data at some later time.

In a DFD, the Gane and Sarson symbol for a data store is a flat rectangle that is open on the right side and closed on the left side. The name of the data store appears between the lines and identifies the data it contains. A data store name is a plural name consisting of a noun and adjectives, if needed. Examples of data store names are STUDENTS, ACCOUNTS RECEIVABLE, PRODUCTS, DAILY PAYMENTS, PURCHASE ORDERS, OUTSTANDING CHECKS, INSURANCE POLICIES, and EMPLOYEES. Exceptions to the plural name rule are collective nouns that represent multiple occurrences of objects. For example, GRADEBOOK represents a group of students and their scores.

A data store must be connected to a process with a data flow. Figure 5-7 illustrates typical examples of data stores. In each case, the data store has at least one incoming and one outgoing data flow and is connected to a process symbol with a data flow.

Violations of the rule that a data store must have at least one incoming and one outgoing data flow are shown in Figure 5-8. In the first example, two data stores are connected incorrectly because no process is between them. Also, COURSES has no incoming data flow and STUDENTS has no outgoing data flow. In the second and third examples, the data stores lack either an outgoing or incoming data flow.

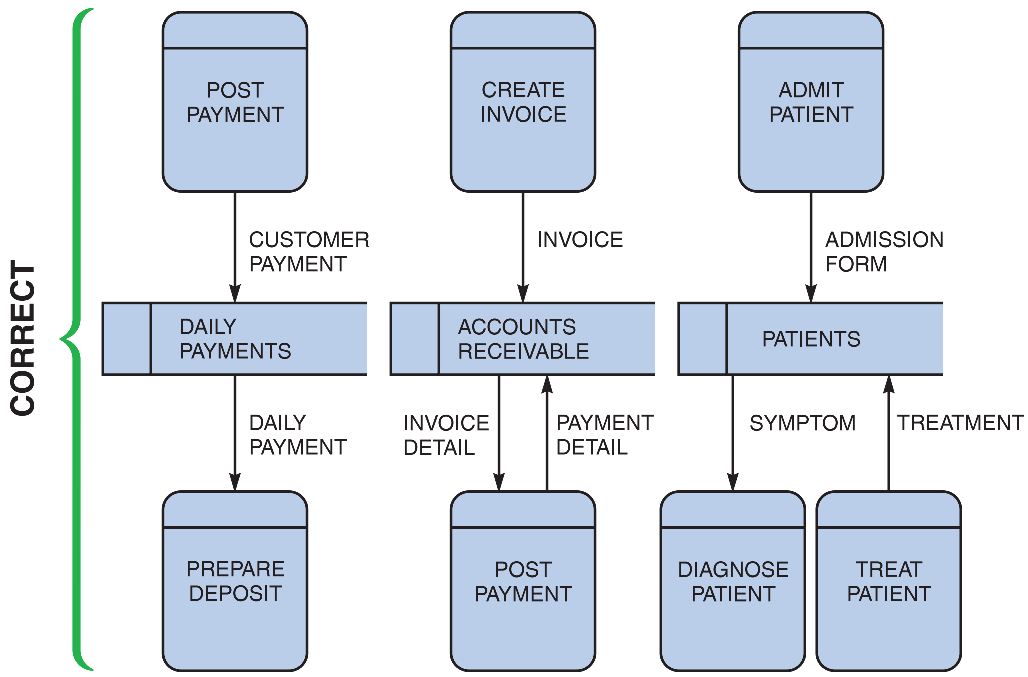


FIGURE 5-7 Examples of correct uses of data store symbols in a data flow diagram.

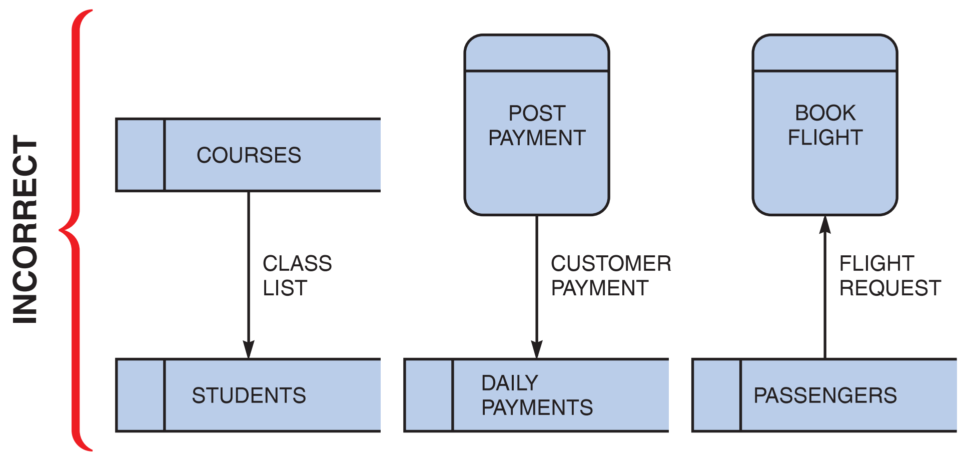


FIGURE 5-8 Examples of incorrect uses of data store symbols: Two data stores cannot be connected by a data flow without an intervening process, and each data store should have an outgoing and incoming data flow.

There is an exception to the requirement that a data store must have at least one incoming and one outgoing data flow. In some situations, a data store has no input data flow because it contains fixed reference data that is not updated by the system. For example, consider a data store called TAX TABLE, which contains withholding tax data that a company downloads from the Internal Revenue Service. When the company runs its payroll, the CALCULATE WITHHOLDING process accesses data from this data store. On a DFD, this would be represented as a one-way outgoing data flow from the TAX TABLE data store into the CALCULATE WITHHOLDING process.

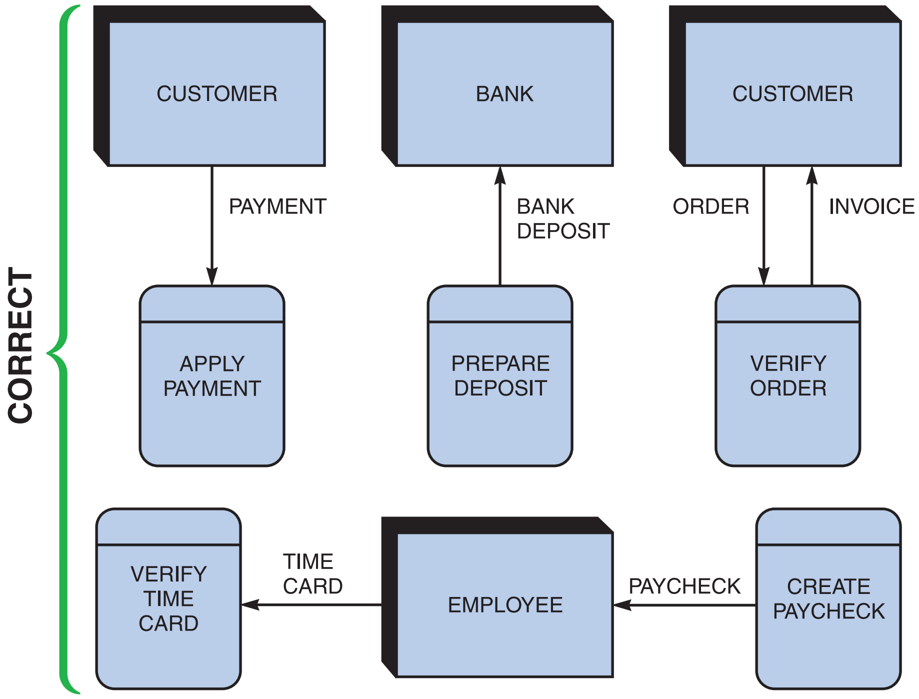


FIGURE 5-9 Examples of correct uses of external entities in a data flow diagram.

ENTITY SYMBOL The symbol for an **entity** is a rectangle, which may be shaded to make it look three-dimensional. The name of the entity appears inside the symbol.

A DFD shows only external entities that provide data to the system or receive output from the system. A DFD shows the boundaries of the system and how the system interfaces with the outside world. For example, a customer entity submits an order to an order processing system. Other examples of entities include a patient who supplies data to a medical records system, a homeowner who receives a bill from a city property tax system, or an accounts payable system that receives data from the company's purchasing system.

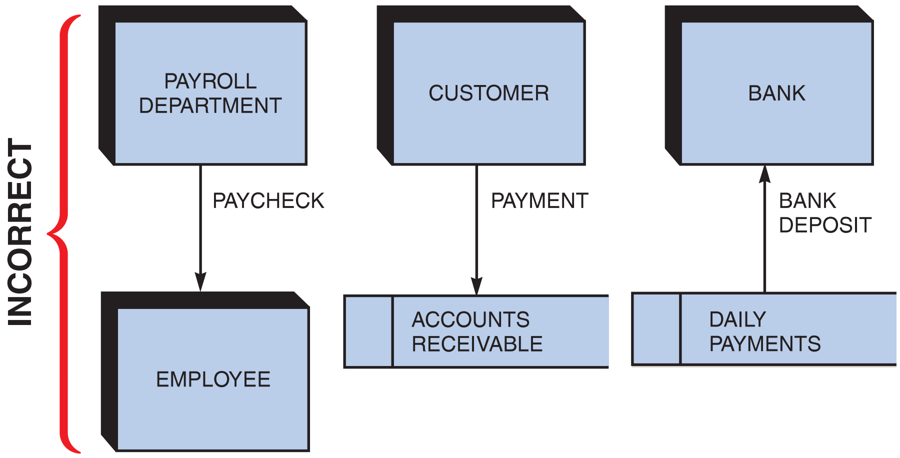


FIGURE 5-10 Examples of incorrect uses of external entities. An external entity must be connected by a data flow to a process, and not directly to a data store or to another external entity.

DFD entities also are called **terminators**, because they are data origins or final destinations. Systems analysts call an entity that supplies data to the system a **source**, and an entity that receives data from the system a **sink**. An entity name is the singular form of a department, outside organization, other information system, or person. An external entity can be a source or a sink or both, but each entity must be connected to a process by a data flow. Figures 5-9 and 5-10 show correct and incorrect examples of this rule.

With an understanding of the proper use of DFD symbols, you are ready to construct diagrams that use these symbols. Figure 5-11 on the next page shows a summary of the rules for using DFD symbols.

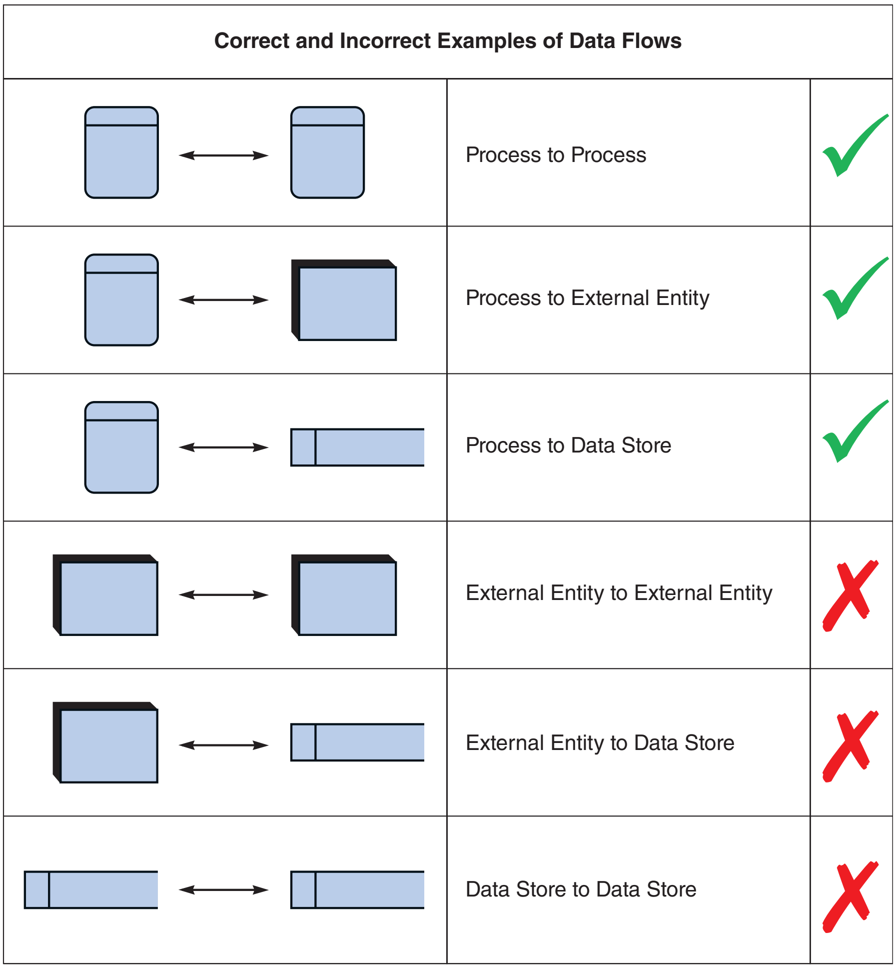


FIGURE 5-11 Examples of correct and incorrect uses of data flows.

Creating a Set of DFDs

During requirements modeling, you used interviews, questionnaires, and other techniques to gather facts about the system, and you learned how the various people, departments, data, and processes fit together to support business operations. Now you are ready to create a graphical model of the information system based on your fact-finding results.

To learn how to construct DFDs, you will use examples of two information systems. The first example is a grading system that instructors use to assign final grades based on the scores that students receive during the term. The second example is an order system that a company uses to enter orders and apply payments against a customer's balance. First, you will review a set of guidelines for drawing DFDs. Then you will learn how to apply these guidelines and create a set of DFDs using a three-step process.

VIDEO LEARNING SESSION: DFD Context Diagrams

Video Learning Sessions can help you understand key concepts, practice your skills, and check your work. To access the sessions, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and navigate to the **Video Learning Sessions** for this book. This session is about DFD context diagrams, why they are important, how to construct a context diagram, and how you can use a CASE tool to create a context diagram.

Guidelines for Drawing DFDs

When you draw a context diagram and other DFDs, you should follow several guidelines:

- Draw the context diagram so it fits on one page.
- Use the name of the information system as the process name in the context diagram. For example, the process name in Figure 5-12 is GRADING SYSTEM. Notice that the process name is the same as the system name. This is because the

context diagram shows the entire information system as if it were a single process. For processes in lower-level DFDs, you would use a verb followed by a descriptive noun, such as ESTABLISH GRADEBOOK, ASSIGN FINAL GRADE, or PRODUCE GRADE REPORT.

- Use unique names within each set of symbols. For instance, the diagram in Figure 5-12 shows only one entity named STUDENT and only one data flow named FINAL GRADE. Whenever you see the entity STUDENT on any other DFD in the grading system, you know that you are dealing with the same entity. Whenever the FINAL GRADE data flow appears, you know that you are dealing with the same data flow. The naming convention also applies to data stores.
- Do not cross lines. One way to achieve that goal is to restrict the number of symbols in any DFD. On lower-level diagrams with multiple processes, you should not have more than nine process symbols. Including more than nine symbols usually is a signal that your diagram is too complex and that you should reconsider your analysis. Another way to avoid crossing lines is to duplicate an entity or data store. When duplicating a symbol on a diagram, make sure to document the duplication to avoid possible confusion. A special notation, such as an asterisk, next to the symbol name and inside the duplicated symbols signifies that they are duplicated on the diagram.
- Provide a unique name and reference number for each process. Because it is the highest-level DFD, the context diagram contains process 0, which represents the entire information system, but does not show the internal workings. To describe the next level of detail inside process 0, you must create a DFD named diagram 0, which will reveal additional processes that must be named and numbered. As you continue to create lower-level DFDs, you assign unique names and reference numbers to all processes, until you complete the logical model.
- Obtain as much user input and feedback as possible. Your main objective is to ensure that the model is accurate, easy to understand, and meets the needs of its users.

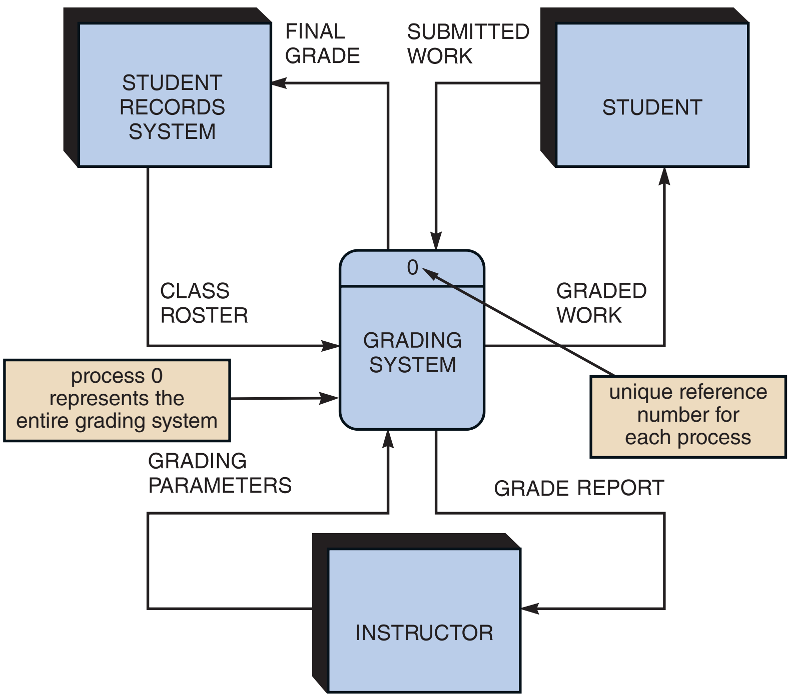


FIGURE 5-12 Context diagram DFD for a grading system.

Step 1: Draw a Context Diagram

The first step in constructing a set of DFDs is to draw a context diagram. A **context diagram** is a top-level view of an information system that shows the system's boundaries and scope. To draw a context diagram, you start by placing a single process symbol in the center of the page. The symbol represents the entire information system, and you identify it as **process 0** (the numeral zero, and not the letter O). Then you place the system entities around the perimeter of the page and use data flows to connect the entities to the central process. Data stores are not shown in the context diagram because they are contained within the system and remain hidden until more detailed diagrams are created.

How do you know which entities and data flows to place in the context diagram? You begin by reviewing the system requirements to identify all external data sources and destinations. During that process, you identify the entities, the name and content of the data flows, and the direction of the data flows. If you do that carefully, and you did a good job of fact-finding in the previous stage, you should have no difficulty drawing the context diagram. Now review the following context diagram examples.

EXAMPLE: CONTEXT DIAGRAM FOR A GRADING SYSTEM The context diagram for a grading system is shown in Figure 5-12 on the previous page. The GRADING SYSTEM process is at the center of the diagram. The three entities (STUDENT RECORDS SYSTEM, STUDENT, and INSTRUCTOR) are placed around the central process. Interaction among the central process and the entities involves six different data flows. The STUDENT RECORDS SYSTEM entity supplies data through the CLASS ROSTER data flow and receives data through the FINAL GRADE data flow. The STUDENT entity supplies data through the SUBMITTED WORK data flow and receives data through the GRADED WORK data flow. Finally, the INSTRUCTOR entity supplies data through the GRADING PARAMETERS data flow and receives data through the GRADE REPORT data flow.

EXAMPLE: CONTEXT DIAGRAM FOR AN ORDER SYSTEM The context diagram for an order system is shown in Figure 5-13. Notice that the ORDER SYSTEM process is at the center of the diagram and five entities surround the process. Three of the entities, SALES REP, BANK, and ACCOUNTING, have single incoming data flows for

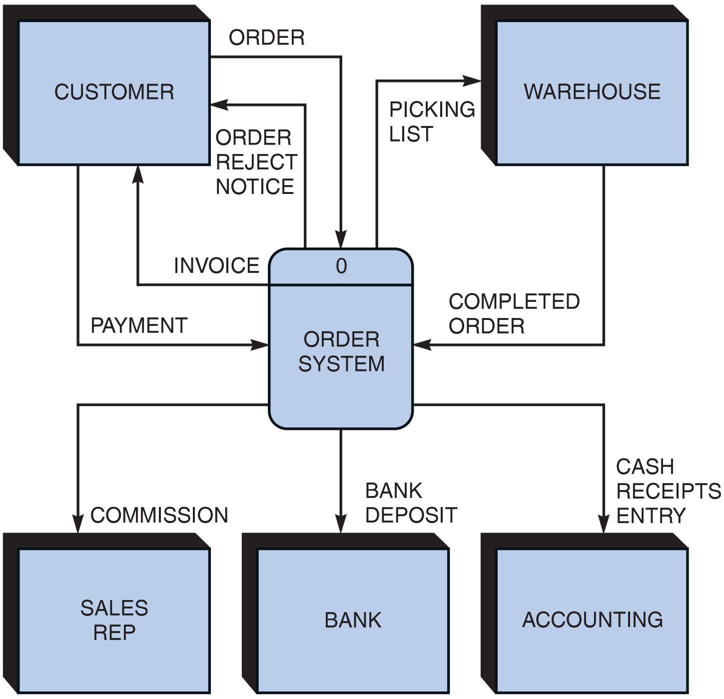


FIGURE 5-13 Context diagram DFD for an order system.

COMMISSION, BANK DEPOSIT, and CASH RECEIPTS ENTRY, respectively. The WAREHOUSE entity has one incoming data flow — PICKING LIST — that is, a report that shows the items ordered and their quantity, location, and sequence to pick from the warehouse. The WAREHOUSE entity has one outgoing data flow, COMPLETED ORDER. Finally, the CUSTOMER entity has two outgoing data flows, ORDER and PAYMENT, and two incoming data flows, ORDER REJECT NOTICE and INVOICE.

The context diagram for the order system appears more complex than the grading system because it has two more entities and three more data flows. What makes one system more complex than another is the number of components, the number of levels, and the degree of interaction among its processes, entities, data stores, and data flows.

VIDEO LEARNING SESSION: DFD DIAGRAM 0

Video Learning Sessions can help you understand key concepts, practice your skills, and check your work. To access the sessions, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and navigate to the **Video Learning Sessions** for this book. This session is about DFD diagram 0. You'll learn what a DFD diagram 0 is, how to create a diagram 0, and how you can use a CASE tool to create a DFD diagram 0.

Step 2: Draw a Diagram 0 DFD

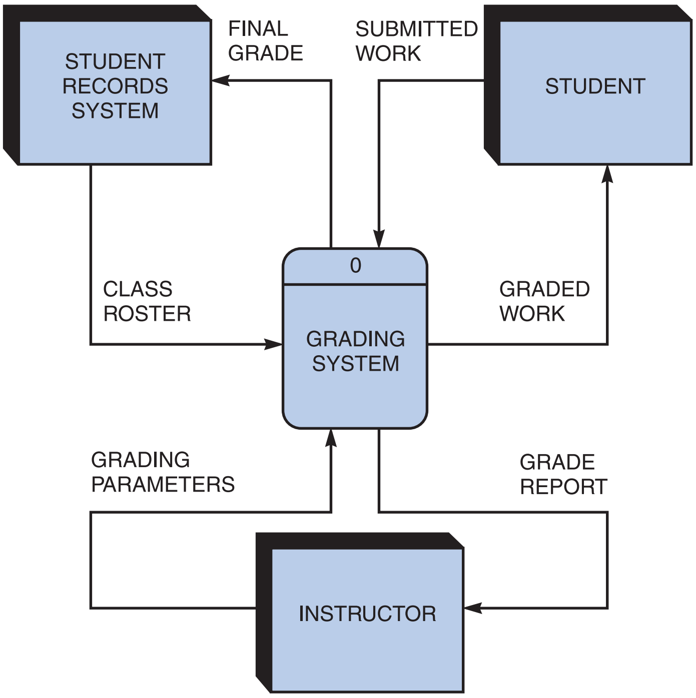
In the previous step, you learned that a context diagram provides the most general view of an information system and contains a single process symbol, which is like a black box. To show the detail inside the black box, you create DFD diagram 0. **Diagram 0** (the numeral zero, and not the letter O) zooms in on the system and shows major internal processes, data flows, and data stores. Diagram 0 also repeats the entities and data flows that appear in the context diagram. When you expand the context diagram into DFD diagram 0, you must retain all the connections that flow into and out of process 0.

FIGURE 5-14 Complex manufacturing systems require many interactive processes and data sources.

The real-life scene in Figure 5-14 represents a complex manufacturing system with many interactive processes and data. In a large system such as this, each process in diagram 0 could represent a separate system such as inventory, production control, and scheduling. Diagram 0 provides an overview of all the components that interact to form the overall system. Now review the following diagram 0 examples.

EXAMPLE: DIAGRAM 0 DFD FOR A GRADING SYSTEM

Figure 5-15 on the next page shows a context diagram at the top and diagram 0 beneath it. Notice that diagram 0 is an expansion of process 0. Also notice that the three same entities (STUDENT RECORDS SYSTEM, STUDENT, and INSTRUCTOR) and the same six data flows (FINAL GRADE, CLASS ROSTER, SUBMITTED WORK, GRADED WORK, GRADING PARAMETERS, and GRADE REPORT) appear in both diagrams. In addition, diagram 0 expands process 0 to reveal four internal processes, one data store, and five additional data flows.



CONTEXT DIAGRAM FOR GRADING SYSTEM

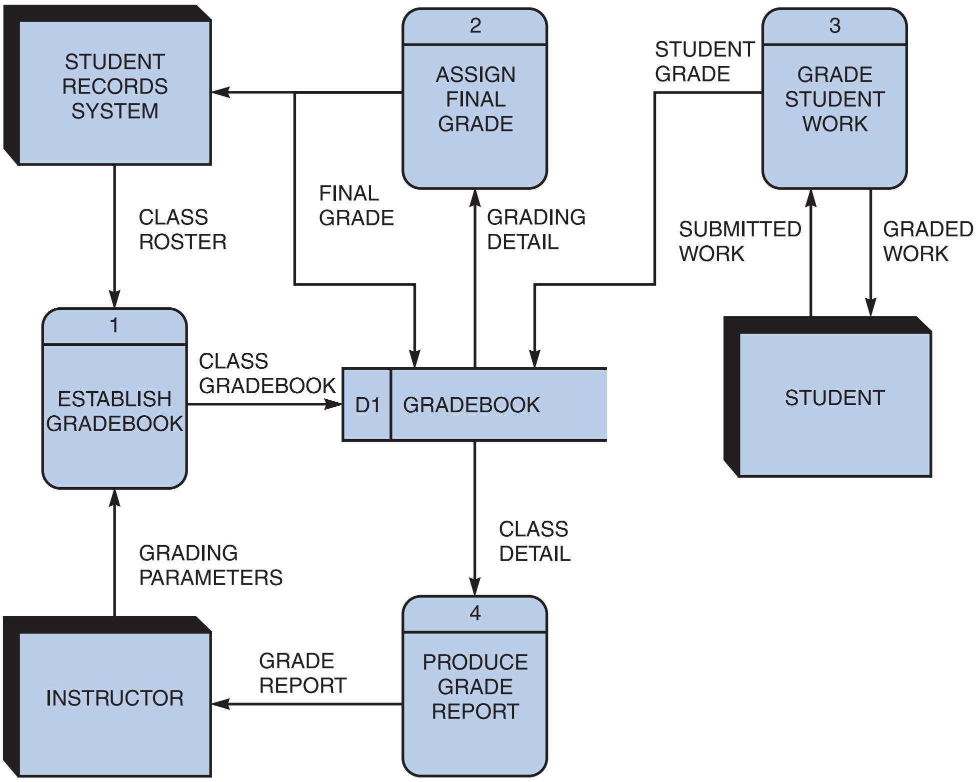


DIAGRAM 0 FOR GRADING SYSTEM

FIGURE 5-15 Context diagram and diagram 0 for the grading system.

Notice that each process in diagram 0 has a reference number: ESTABLISH GRADEBOOK is 1, ASSIGN FINAL GRADE is 2, GRADE STUDENT WORK is 3, and PRODUCE GRADE REPORT is 4. These reference numbers are important because they identify a series of DFDs. If more detail were needed for ESTABLISH GRADEBOOK, for example, you would draw a diagram 1, because ESTABLISH GRADEBOOK is process 1.

The process numbers do not suggest that the processes are accomplished in a sequential order. Each process always is considered to be available, active, and awaiting data to be processed. If processes must be performed in a specific sequence, you document the information in the process descriptions (discussed later in this chapter), not in the DFD.

The FINAL GRADE data flow output from the ASSIGN FINAL GRADE process is a diverging data flow that becomes an input to the STUDENT RECORDS SYSTEM entity and to the GRADEBOOK data store. A **diverging data flow** is a data flow in which the same data travels to two or more different locations. In that situation, a diverging data flow is the best way to show the flow rather than showing two identical data flows, which could be misleading.

If the same data flows in both directions, you can use a double-headed arrow to connect the symbols. To identify specific data flows into and out of a symbol, however, you use separate data flow symbols with single arrowheads. For example, in Figure 5-15, the separate data flows (SUBMITTED WORK and GRADED WORK) go into and out of the GRADE STUDENT WORK process.

Because diagram 0 is an exploded version of process 0, it shows considerably more detail than the context diagram. You also can refer to diagram 0 as a partitioned or decomposed view of process 0. When you explode a DFD, the higher-level diagram is called the **parent diagram**, and the lower-level diagram is referred to as the **child diagram**. The grading system is simple enough that you do not need any additional DFDs to model the system. At that point, the four processes, the one data store, and the 10 data flows can be documented in the data dictionary.

When you create a set of DFDs for a system, you break the processing logic down into smaller units, called functional primitives, that programmers will use to develop code. A **functional primitive** is a process that consists of a single function that is not exploded further. For example, each of the four processes shown in the lower portion of Figure 5-15 is a functional primitive. You document the logic for a functional primitive by writing a process description in the data dictionary. Later, when the logical design is implemented as a physical system, programmers will transform each functional primitive into program code and modules that carry out the required steps. Deciding whether to explode a process further or determine that it is a functional primitive is a matter of experience, judgment, and interaction with programmers who must translate the logical design into code.

EXAMPLE: DIAGRAM 0 DFD FOR AN ORDER SYSTEM Figure 5-16 on the next page shows the diagram 0 for an order system. Process 0 on the order system's context diagram is exploded to reveal three processes (FILL ORDER, CREATE INVOICE, and APPLY PAYMENT), one data store (ACCOUNTS RECEIVABLE), two additional data flows (INVOICE DETAIL and PAYMENT DETAIL), and one diverging data flow (INVOICE).

The following walkthrough explains the DFD shown in Figure 5-16:

1. A CUSTOMER submits an ORDER. Depending on the processing logic, the FILL ORDER process either sends an ORDER REJECT NOTICE back to the customer or sends a PICKING LIST to the WAREHOUSE.
2. A COMPLETED ORDER from the WAREHOUSE is input to the CREATE INVOICE process, which outputs an INVOICE to both the CUSTOMER process and the ACCOUNTS RECEIVABLE data store.
3. A CUSTOMER makes a PAYMENT that is processed by APPLY PAYMENT. APPLY PAYMENT requires INVOICE DETAIL input from the ACCOUNTS

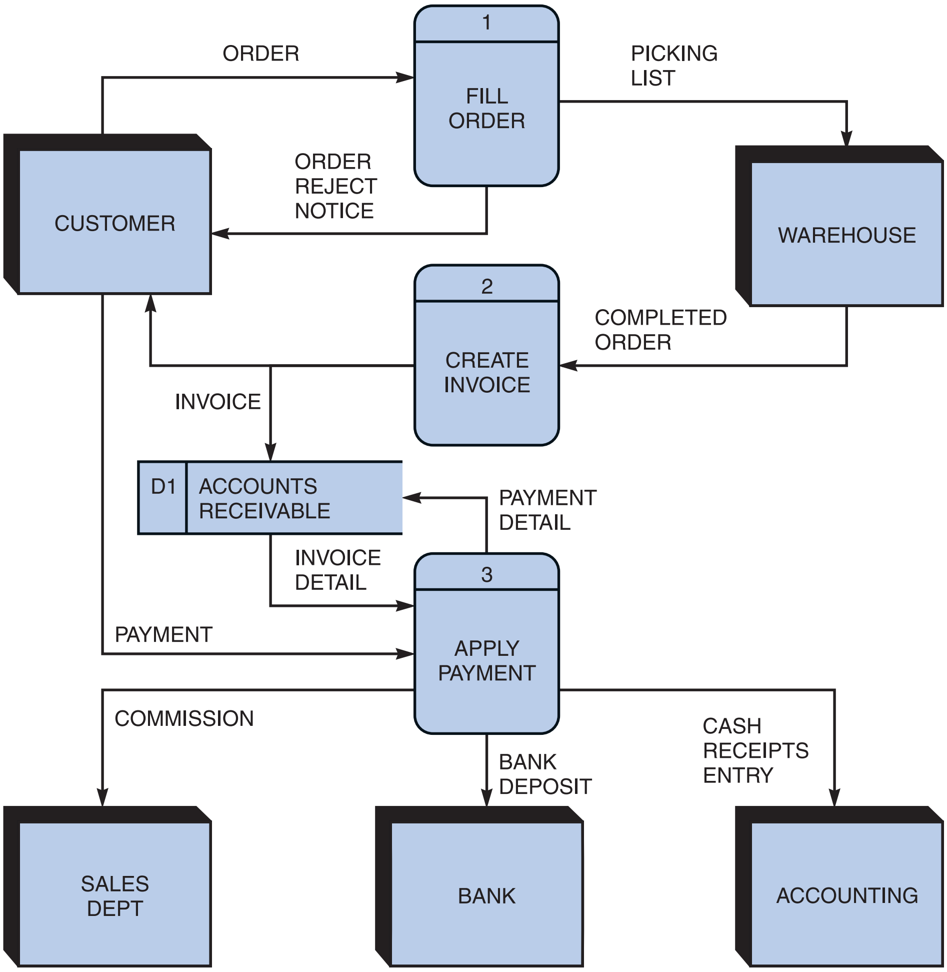


FIGURE 5-16 Diagram 0 DFD for the order system.

RECEIVABLE data store along with the PAYMENT. APPLY PAYMENT also outputs PAYMENT DETAIL back to the ACCOUNTS RECEIVABLE data store and outputs COMMISSION to the SALES DEPT, BANK DEPOSIT to the BANK, and CASH RECEIPTS ENTRY to ACCOUNTING.

The walkthrough of diagram 0 illustrates the basic requirements of the order system. To learn more, you would examine the detailed description of each separate process.

Step 3: Draw the Lower-Level Diagrams

This set of lower-level DFDs is based on the order system. To create lower-level diagrams, you must use leveling and balancing techniques. **Leveling** is the process of drawing a series of increasingly detailed diagrams, until all functional primitives are identified. **Balancing** maintains consistency among a set of DFDs by ensuring that input and output data flows align properly. Leveling and balancing are described in more detail in the following sections.

LEVELING EXAMPLES Leveling uses a series of increasingly detailed DFDs to describe an information system. For example, a system might consist of dozens, or even hundreds, of separate processes. Using leveling, an analyst starts with an overall view, which is a context diagram with a single process symbol. Next, the analyst creates diagram 0, which shows more detail. The analyst continues to create lower-level DFDs until all processes are identified as functional primitives, which represent single processing functions. More complex systems have more processes, and analysts must work through many levels to identify the functional primitives. Leveling also is called **exploding**, **partitioning**, or **decomposing**.

Figures 5-16 and 5-17 provide an example of leveling. Figure 5-16 shows diagram 0 for an order system, with the FILL ORDER process labeled as process 1. Now consider Figure 5-17, which provides an exploded view of the FILL ORDER process. Notice that FILL ORDER (process 1) actually consists of three processes: VERIFY ORDER (process 1.1), PREPARE REJECT NOTICE (process 1.2), and ASSEMBLE ORDER (process 1.3).

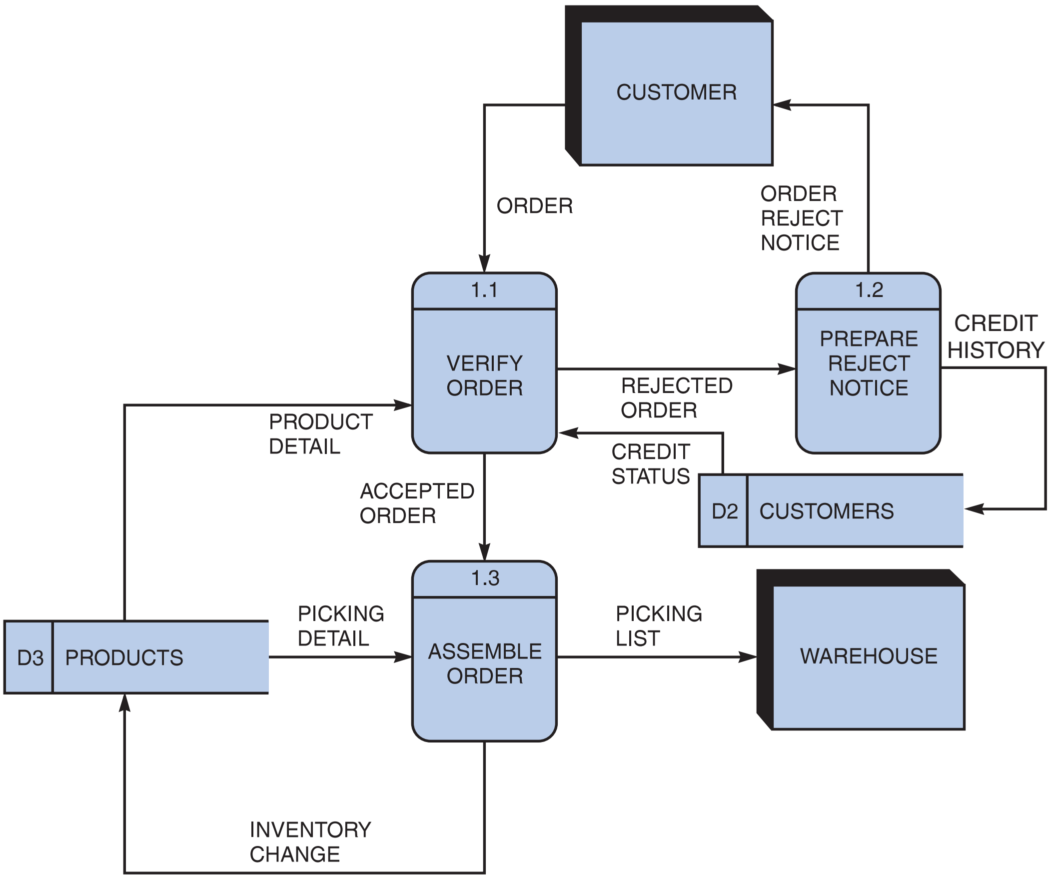


FIGURE 5-17 Diagram 1 DFD shows details of the FILL ORDER process in the order system.

As Figure 5-17 shows, all processes are numbered using a decimal notation consisting of the parent's reference number, a decimal point, and a sequence number within the new diagram. In Figure 5-17, the parent process of diagram 1 is process 1, so the processes in diagram 1 have reference numbers of 1.1, 1.2, and 1.3. If process 1.3, ASSEMBLE ORDER, is decomposed further, then it would appear in diagram 1.3 and the processes in diagram 1.3 would be numbered as 1.3.1, 1.3.2, 1.3.3, and so on. This numbering technique makes it easy to integrate and identify all DFDs.

When you compare Figures 5-16 and 5-17, you will notice that Figure 5-17 (the exploded FILL ORDER process) shows two data stores (CUSTOMERS and PRODUCTS) that do not appear on Figure 5-16, which is the parent DFD. Why not? The answer is based on a simple rule: When drawing DFDs, you show a data store only when two or more processes use that data store. The CUSTOMERS and PRODUCTS data stores were internal to the FILL ORDER process, so the analyst did not show them on diagram 0, which is the parent. When you explode the FILL ORDER process into diagram 1 DFD, however, you see that three processes (1.1, 1.2, and 1.3) interact with the two data stores, which now are shown.

Now compare Figure 5-17 (on the previous page) and Figure 5-18. Notice that Figure 5-18 shows the same data flows as Figure 5-17, but does not show the CUSTOMER and WAREHOUSE entities. Analysts often use this technique to simplify a DFD and reduce unnecessary clutter. Because the missing symbols appear on the parent DFD, you can refer to that diagram to identify the source or destination of the data flows.

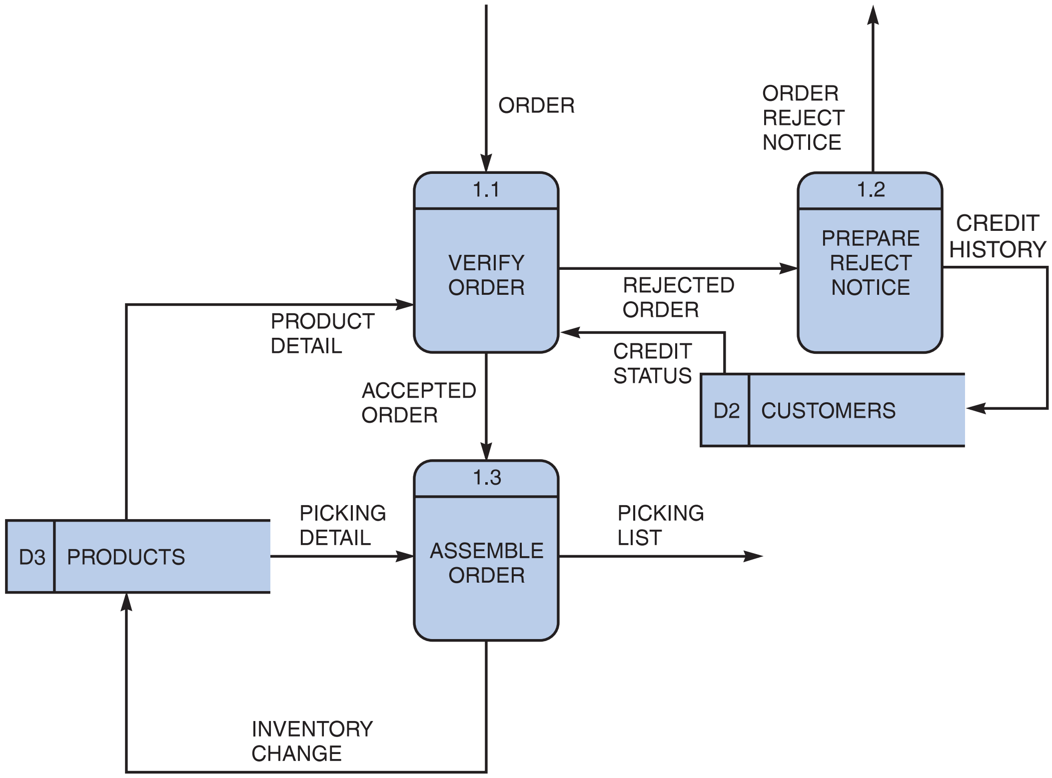


FIGURE 5-18 This diagram does not show the symbols that connect to data flows entering or leaving FILL ORDER on the context diagram.

BALANCING EXAMPLES Balancing ensures that the input and output data flows of the parent DFD are maintained on the child DFD. For example, Figure 5-19 shows two DFDs: The order system diagram 0 is shown at the top of the figure, and the exploded diagram 3 DFD is shown at the bottom.

The two DFDs are balanced, because the child diagram at the bottom has the same input and output flows as the parent process 3 shown at the top. To verify the balancing, notice that the parent process 3, APPLY PAYMENT, has one incoming data flow from an external entity, and three outgoing data flows to external entities. Now examine the child DFD, which is diagram 3. Now, ignore the internal data flows and count the data flows to and from external entities. You will see that the three processes maintain the same one incoming and three outgoing data flows as the parent process.

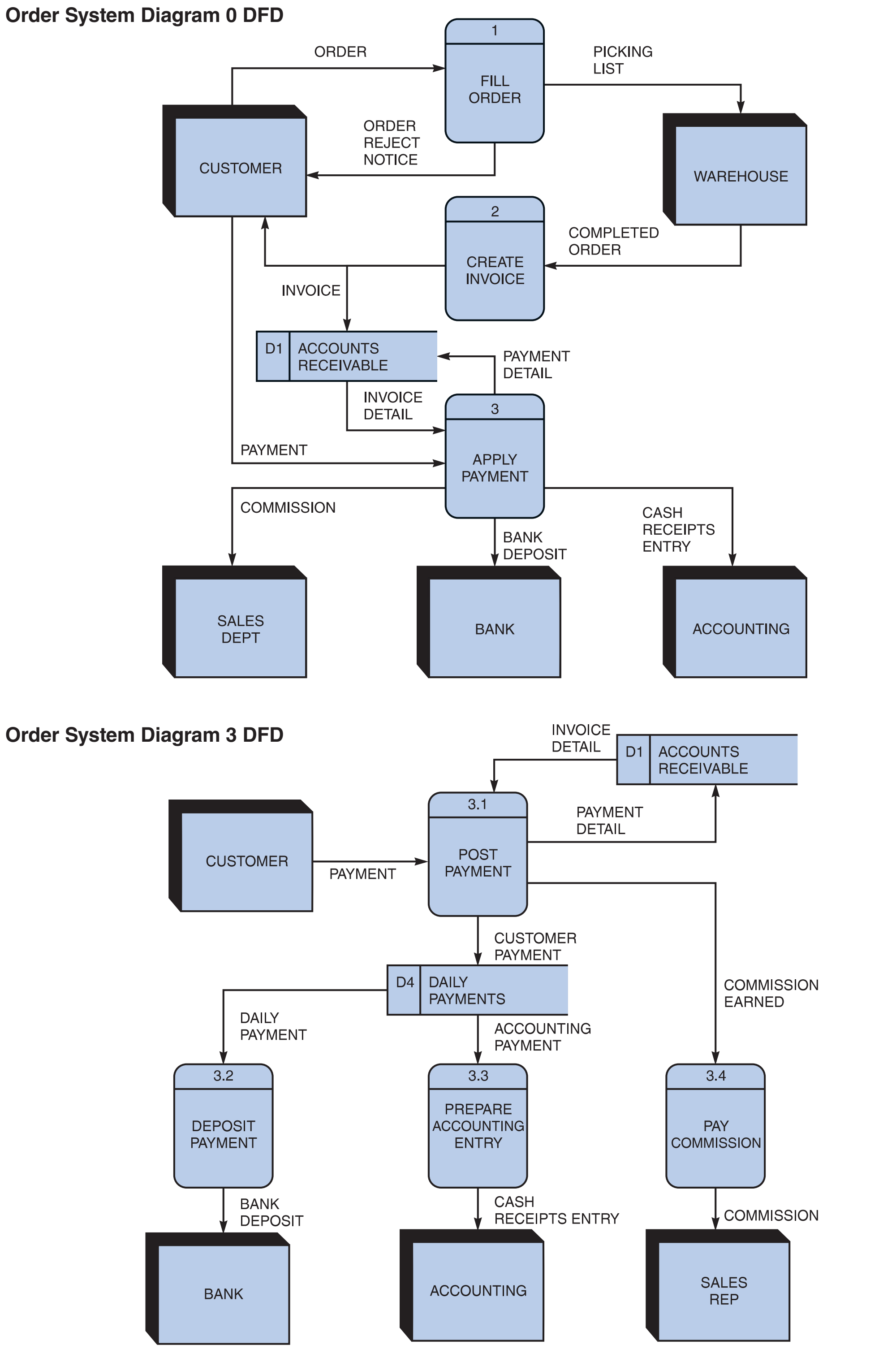


FIGURE 5-19 The order system diagram 0 is shown at the top of the figure, and exploded diagram 3 DFD (for the APPLY PAYMENT process) is shown at the bottom. The two DFDs are balanced, because the child diagram at the bottom has the same input and output flows as the parent process 3 shown at the top.

Another example of balancing is shown in Figures 5-20 and 5-21 on the next page. The DFDs in these figures were created using Visible Analyst, a popular CASE tool.

Figure 5-20 shows a sample context diagram. The process 0 symbol has two input flows and two output flows. Notice that process 0 can be considered as a black box, with no internal detail shown. In Figure 5-21, process 0 (the parent DFD) is exploded into the next level of detail. Now three processes, two data stores, and four internal data flows are visible. Notice that the details of process 0 are shown inside a dashed line, just as if you could see inside the process.

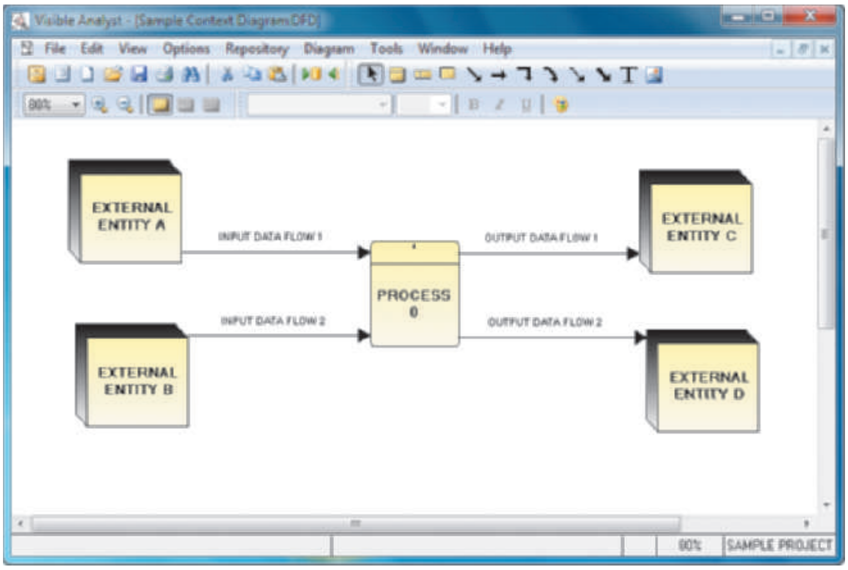


FIGURE 5-20 Example of a parent DFD diagram, showing process 0 as a black box.

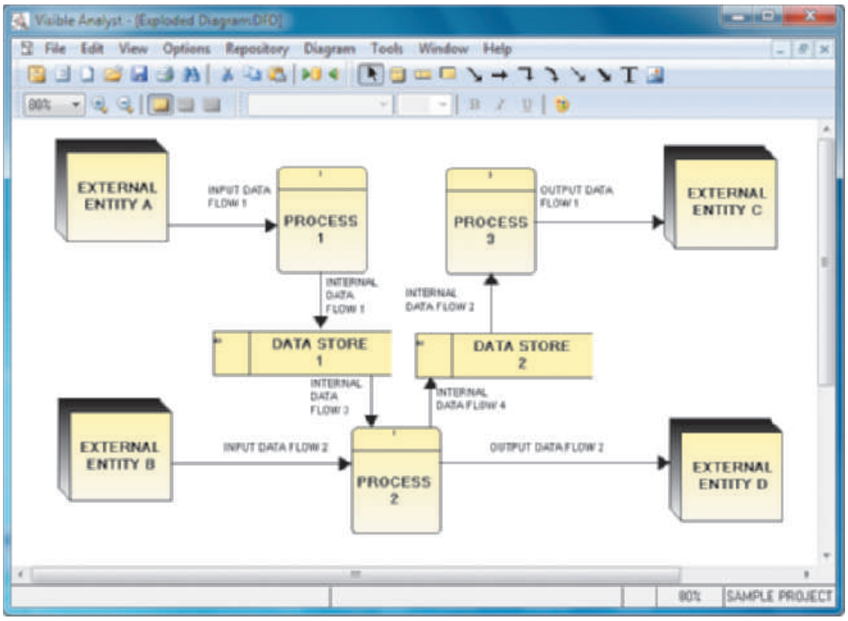


FIGURE 5-21 In the next level of detail, the process 0 black box reveals three processes, two data stores, and four internal data flows — all of which are shown inside the dashed line.

The DFDs in Figures 5-20 and 5-21 are balanced, because the four data flows into and out of process 0 are maintained on the child DFD. The DFDs also are leveled, because each internal process is numbered to show that it is a child of the parent process.

CASE IN POINT 5.1: BIG TEN UNIVERSITY

You are the IT director at Big Ten University. As part of a training program, you decide to draw a DFD that includes some obvious mistakes to see whether your newly hired junior analysts can find them. You came up with the diagram 0 DFD shown in Figure 5-22. Based on the rules explained in this chapter, how many problems should the analysts find?

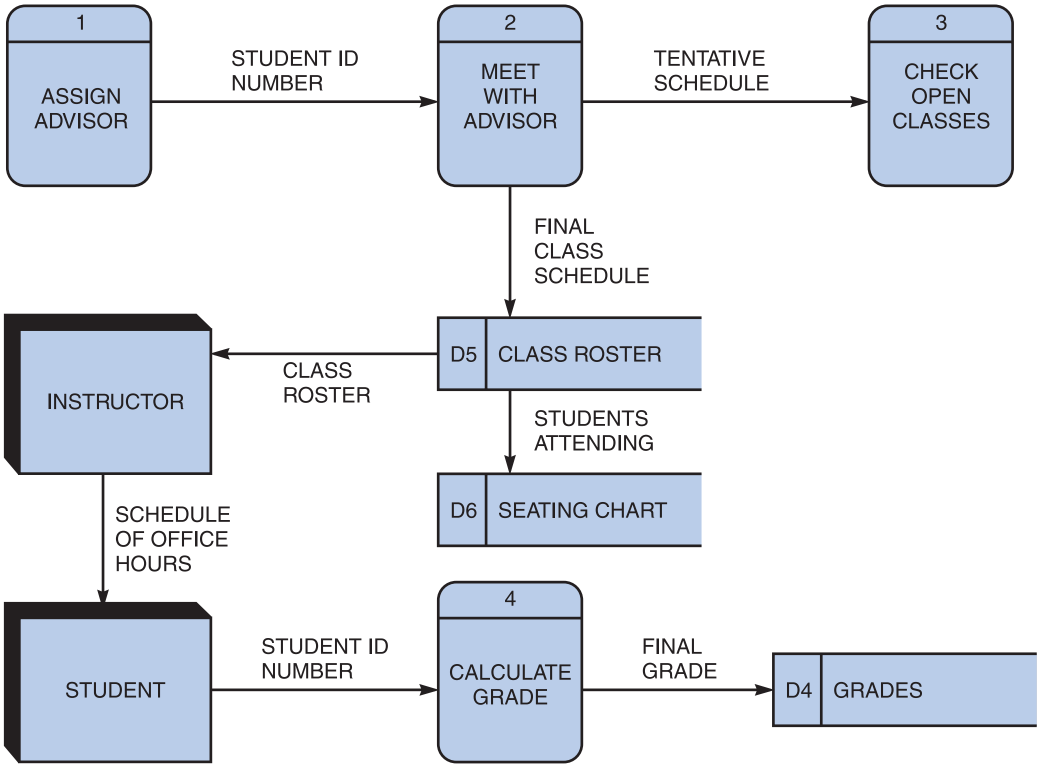


FIGURE 5-22 What are the mistakes in this diagram 0 DFD?

DATA DICTIONARY

A set of DFDs produces a logical model of the system, but the details within those DFDs are documented separately in a data dictionary, which is the second component of structured analysis.

A **data dictionary**, or **data repository**, is a central storehouse of information about the system's data. An analyst uses the data dictionary to collect, document, and organize specific facts about the system, including the contents of data flows, data stores, entities, and processes. The data dictionary also defines and describes all data elements and meaningful combinations of data elements. A **data element**, also called a **data item** or **field**, is the smallest piece of data that has meaning within an information system. Examples of data elements are student grade, salary, Social Security number, account balance, and company name. Data elements are combined into **records**, also called **data structures**. A record is a meaningful combination of related data elements that is included in a data flow or retained in a data store. For example, an auto parts store inventory record might include part number, description, supplier code, minimum and maximum stock levels, cost, and list price.

ON THE WEB

To learn more about data dictionaries, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Data Dictionaries link.

Significant relationships exist among the items in a data dictionary. For example, data stores and data flows are based on data structures, which in turn are composed of data elements. Data flows are connected to data stores, entities, and processes. Accurately documenting these relationships is essential so the data dictionary is consistent with the DFDs. You can use CASE software to help you document the design.

TOOLKIT TIME

The CASE tools in Part B of the Systems Analyst's Toolkit can help you document business functions and processes. To learn more about these tools, turn to Part B of the four-part Toolkit that follows Chapter 12.

Using CASE Tools for Documentation

The more complex the system, the more difficult it is to maintain full and accurate documentation. Fortunately, modern CASE tools simplify the task. For example, in the Visible Analyst CASE tool, documentation automatically flows from the modeling diagrams into the central repository, along with information entered by the user. This section contains several examples of Visible Analyst screens that show the data repository and its contents.

A CASE repository ensures data consistency, which is especially important where multiple systems require the same data. In a large company, for example, the sales, accounting, and shipping systems all might use a data element called CUSTOMER NUMBER. Once the CUSTOMER NUMBER element has been defined in the repository, it can be accessed by other processes, data flows, and data stores. The result is that all systems across the enterprise can share data that is up to date and consistent. You will learn more about CASE tools in Part B of the Systems Analyst's Toolkit.

Documenting the Data Elements

You must document every data element in the data dictionary. Some analysts like to record their notes on online or manual forms. Others prefer to enter the information directly into a CASE tool. Several of the DFDs and data dictionary entries that appear in this chapter were created using a popular CASE tool called Visible Analyst. Although other CASE tools might use other terms or display the information differently, the objective is the same: to provide clear, comprehensive information about the data and processes that make up the system.

Figure 5-23 shows how the analyst used an online documentation form to record information for the SOCIAL SECURITY NUMBER data element. Notice that the figure caption identifies eight specific characteristics for this data element.

1. Online or manual documentation entries often indicate which system is involved. This is not necessary with a CASE tool because all information is stored in one file that is named for the system.
2. The data element has a standard label that provides consistency throughout the data dictionary.
3. The data element can have an alternative name, or alias.
4. This entry indicates that the data element consists of nine numeric characters.
5. Depending on the data element, strict limits might be placed on acceptable values.
6. The data comes from the employee's job application.
7. This entry indicates that only the payroll department has authority to update or change this data.
8. This entry indicates the individual or department responsible for entering and changing data.

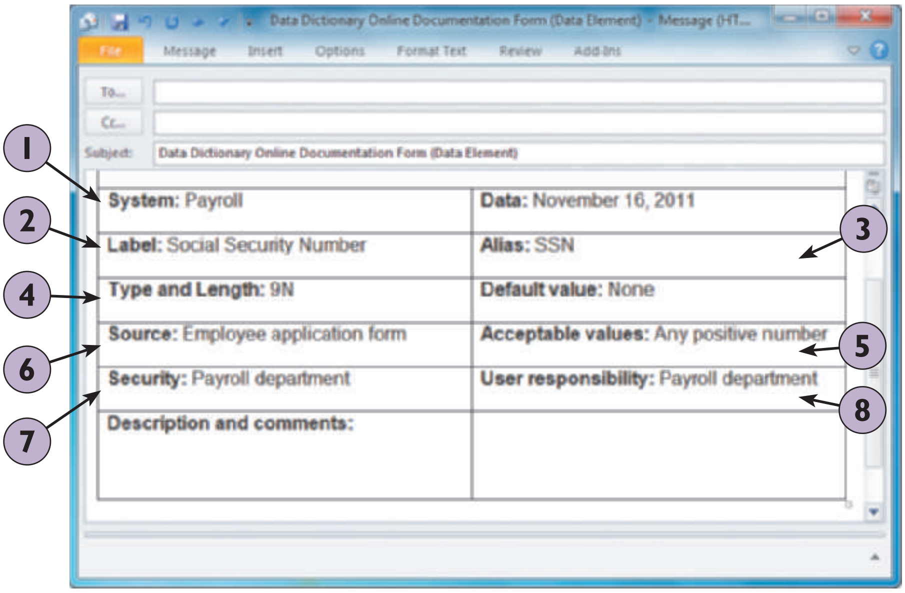


FIGURE 5-23 Using an online documentation form, the analyst has recorded information for a data element named SOCIAL SECURITY NUMBER. Later, the analyst will create a data dictionary entry using a CASE tool.

Figure 5-24 shows a sample screen that illustrates how the SOCIAL SECURITY NUMBER data element might be recorded in the Visible Analyst data dictionary.

Regardless of the terminology or method, the following attributes usually are recorded and described in the data dictionary:

Data element name or label. The data element's standard name, which should be meaningful to users.

Alias. Any name(s) other than the standard data element name; this alternate name is called an **alias**. For example, if you have a data element named CURRENT BALANCE, various users might refer to it by alternate names such as OUTSTANDING BALANCE, CUSTOMER BALANCE, RECEIVABLE BALANCE, or AMOUNT OWED.

Type and length. Type refers to whether the data element contains numeric, alphabetic, or character values. **Length** is the maximum number of characters for an alphabetic or character data element or the maximum number of digits and number of decimal positions for a numeric data element. In addition to text and numeric data, sounds and images also can be stored in digital form. In some systems, these binary data objects are managed and processed just as traditional data elements are. For example, an employee record might include a digitized photo image of the person.

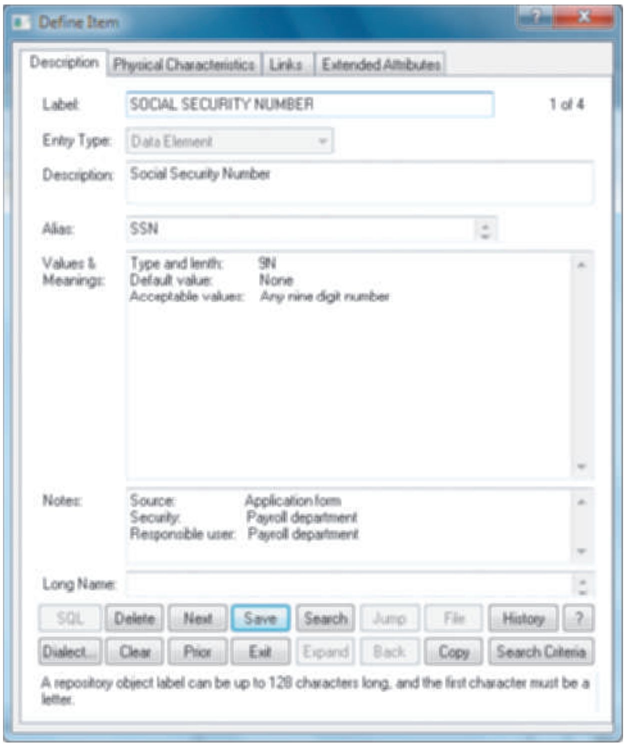


FIGURE 5-24 A Visible Analyst screen describes the data element named SOCIAL SECURITY NUMBER. Notice that many of the items were entered from the online form shown in Figure 5-23.

Default value. The value for the data element if a value otherwise is not entered for it. For example, all new customers might have a default value of $500 for the CREDIT LIMIT data element.

Acceptable values. Specification of the data element's **domain**, which is the set of values permitted for the data element; these values either can be specifically listed or referenced in a table, or can be selected from a specified range of values. You also would indicate if a value for the data element is optional. Some data elements have additional **validity rules**. For example, an employee's salary must be within the range defined for the employee's job classification.

Source. The specification for the origination point for the data element's values. The source could be a specific form, a department or outside organization, another information system, or the result of a calculation.

Security. Identification for the individual or department that has access or update privileges for each data element. For example, only a credit manager has the authority to change a credit limit, while sales reps are authorized to access data in a read-only mode.

Responsible user(s). Identification of the user(s) responsible for entering and changing values for the data element.

Description and comments. This part of the documentation allows you to enter additional notes.

Documenting the Data Flows

In addition to documenting each data element, you must document all data flows in the data dictionary. Figure 5-25 shows a definition for a data flow named COMMISSION. The information on the manual form at the top was entered into the CASE tool data dictionary at the bottom of Figure 5-25.

Although terms can vary, the typical attributes are as follows:

Data flow name or label. The data flow name as it appears on the DFDs.

Description. Describes the data flow and its purpose.

Alternate name(s). Aliases for the DFD data flow name(s).

Origin. The DFD beginning, or source, for the data flow; the origin can be a process, a data store, or an entity.

Destination. The DFD ending point(s) for the data flow; the destination can be a process, a data store, or an entity.

Record. Each data flow represents a group of related data elements called a record or data structure. In most data dictionaries, records are defined separately from the data flows and data stores. When records are defined, more than one data flow or data store can use the same record, if necessary.

Volume and frequency. Describes the expected number of occurrences for the data flow per unit of time. For example, if a company has 300 employees, a TIME CARD data flow would involve 300 transactions and records each week, as employees submit their work hour data.

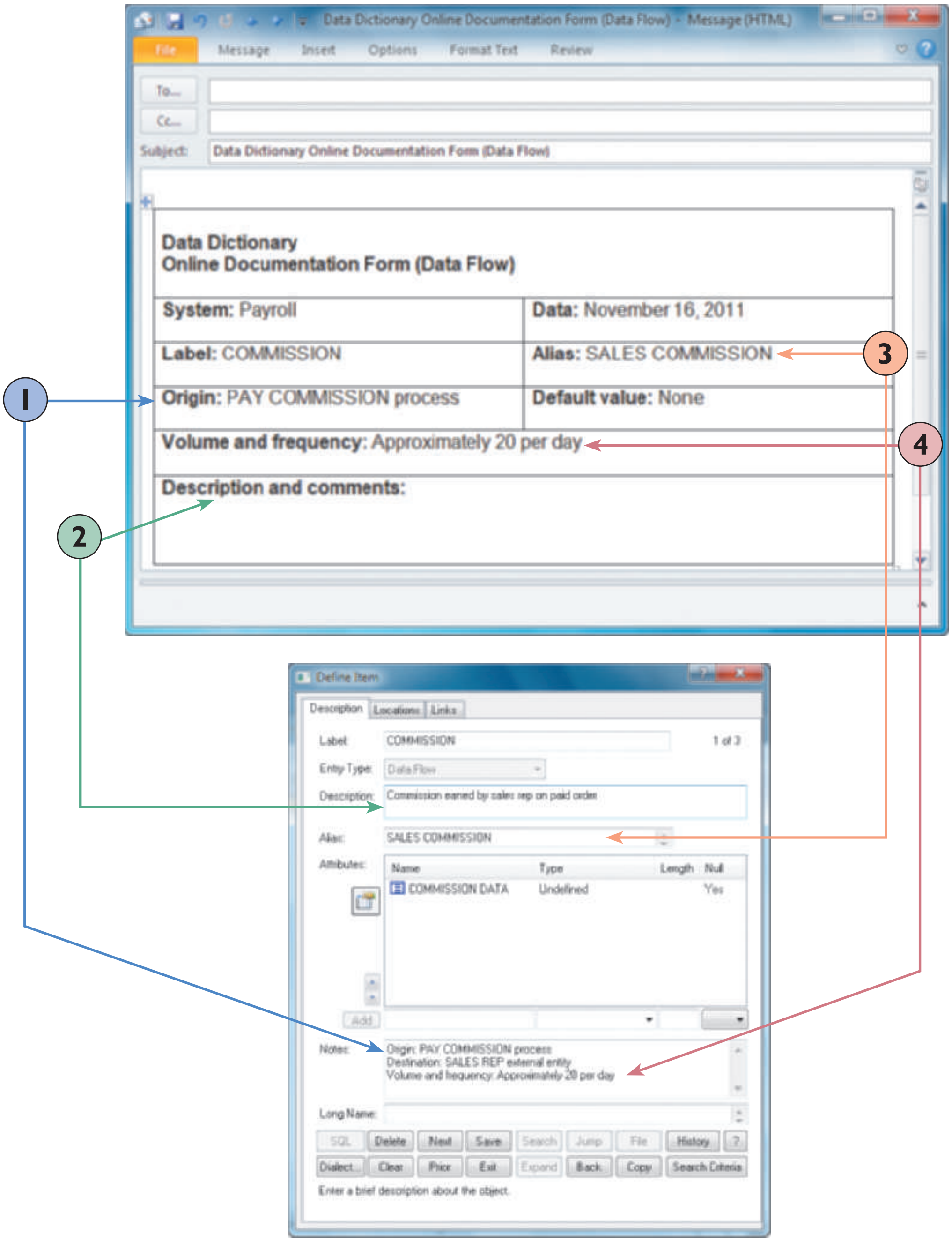


FIGURE 5-25 In the upper screen, an analyst has entered four items of information in an online documentation form. The lower screen shows the same four items entered into a Visible Analyst data dictionary form.

Documenting the Data Stores

You must document every DFD data store in the data dictionary. Figure 5-26 on the next page shows the definition of a data store named IN STOCK.

1. This data store has an alternative name, or alias.
2. For consistency, data flow names are standardized throughout the data dictionary.
3. It is important to document these estimates, because they will affect design decisions in subsequent SDLC phases.

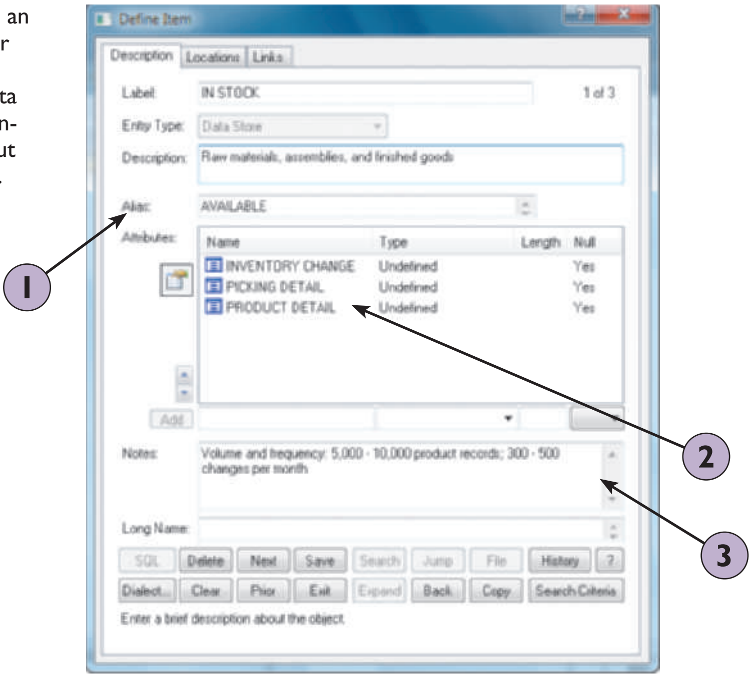


FIGURE 5-26 Visible Analyst screen that documents a data store named IN STOCK.

Typical characteristics of a data store are as follows:

Data store name or label. The data store name as it appears on the DFDs.

Description. Describes the data store and its purpose.

Alternate name(s). Aliases for the DFD data store name.

Attributes. Standard DFD names that enter or leave the data store.

Volume and frequency. Describes the estimated number of records in the data store and how frequently they are updated.

Documenting the Processes

You must document every process, as shown in Figure 5-27. Your documentation includes a description of the process's characteristics and, for functional primitives, a process description, which is a model that documents the processing steps and business logic.

The following are typical characteristics of a process:

Process name or label. The process name as it appears on the DFDs.

Description. A brief statement of the process's purpose.

Process number. A reference number that identifies the process and indicates relationships among various levels in the system.

Process description. This section includes the input and output data flows. For functional primitives, the process description also documents the processing steps and business logic. You will learn how to write process descriptions in the next section.

1. The process number identifies this process. Any subprocesses are numbered 1.1, 1.2, 1.3, and so on.
2. These data flows will be described specifically elsewhere in the data dictionary.

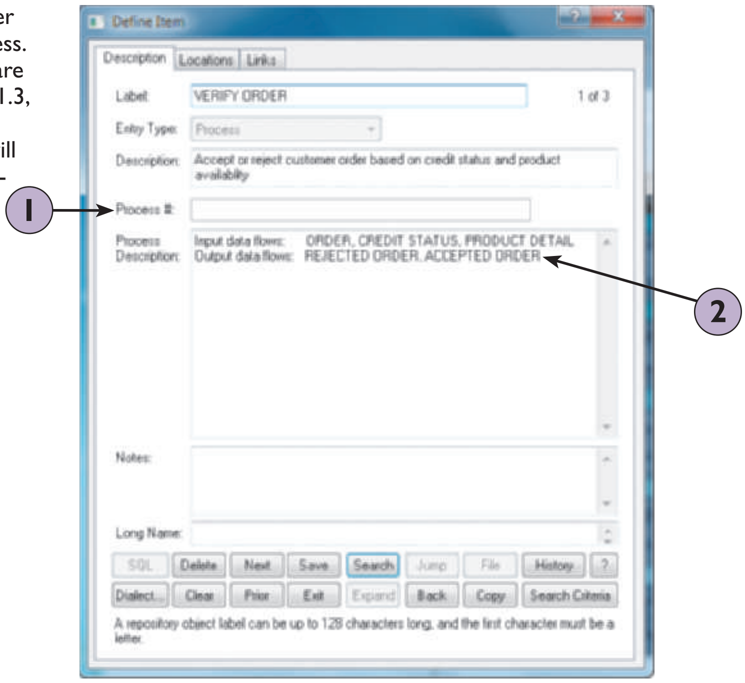


FIGURE 5-27 Visible Analyst screen that describes a process named VERIFY ORDER.

Documenting the Entities

By documenting all entities, the data dictionary can describe all external entities that interact with the system. Figure 5-28 shows a definition for an external entity named WAREHOUSE.

Typical characteristics of an entity include the following:

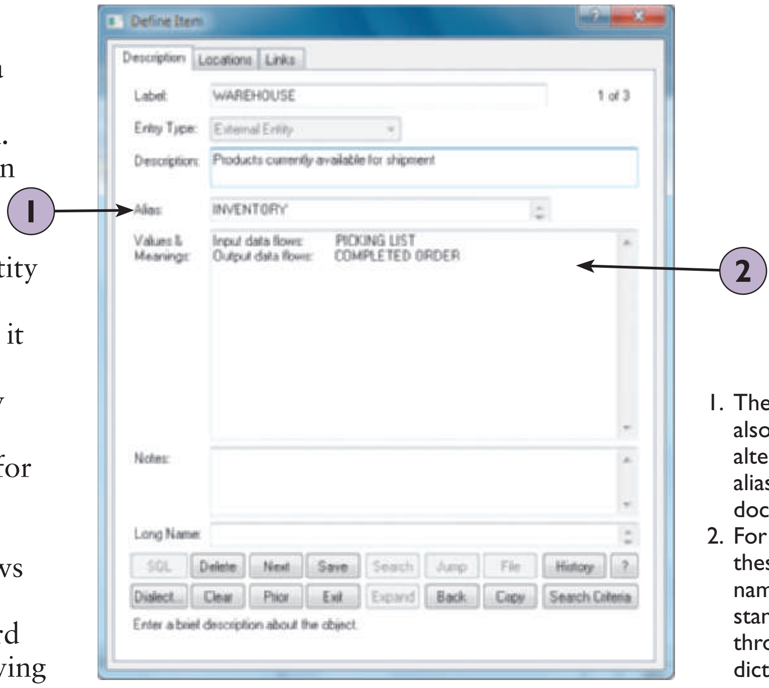
Entity name. The entity name as it appears on the DFDs.

Description. Describe the entity and its purpose.

Alternate name(s). Any aliases for the entity name.

Input data flows. The standard DFD names for the input data flows to the entity.

Output data flows. The standard DFD names for the data flows leaving the entity.



1. The external entity also can have an alternative name, or alias, if properly documented.
2. For consistency, these data flow names are standardized throughout the data dictionary.

FIGURE 5-28 Visible Analyst screen that documents an external entity named WAREHOUSE.

Documenting the Records

A record is a data structure that contains a set of related data elements that are stored and processed together. Data flows and data stores consist of records that you must document in the data dictionary. You define characteristics of each record, as shown in Figure 5-29.

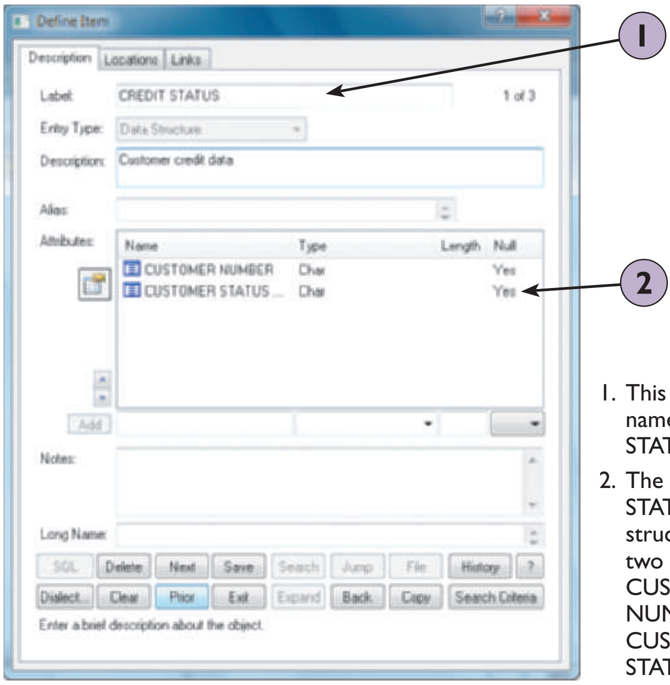
Typical characteristics of a record include the following:

Record or data structure name. The record name as it appears in the related data flow and data store entries in the data dictionary.

Definition or description. A brief definition of the record.

Alternate name(s). Any aliases for the record name.

Attributes. A list of all the data elements included in the record. The data element names must match exactly what you entered in the data dictionary.



1. This data structure is named CREDIT STATUS.
2. The CREDIT STATUS data structure consists of two data elements: CUSTOMER NUMBER and CUSTOMER STATUS CODE.

FIGURE 5-29 Visible Analyst screen that documents a record, or data structure named CREDIT STATUS.

Data Dictionary Reports

The data dictionary serves as a central storehouse of documentation for an information system. A data dictionary is created when the system is developed, and is updated constantly as the system is implemented, operated, and maintained. In addition to describing each data element, data flow, data store, record, entity, and process, the data dictionary documents the relationships among these components. You can obtain many valuable reports from a data dictionary, including the following:

- An alphabetized list of all data elements by name
- A report describing each data element and indicating the user or department that is responsible for data entry, updating, or deletion
- A report of all data flows and data stores that use a particular data element
- Detailed reports showing all characteristics of data elements, records, data flows, processes, or any other selected item stored in the data dictionary

PROCESS DESCRIPTION TOOLS

A **process description** documents the details of a functional primitive, and represents a specific set of processing steps and business logic. Using a set of process description tools, you create a model that is accurate, complete, and concise. Typical process description tools include structured English, decision tables, and decision trees. When you analyze a functional primitive, you break the processing steps down into smaller units in a process called modular design.

It should be noted that this chapter deals with structured analysis, but the process description tools also can be used in object-oriented development, which is described in Chapter 6. You learned in Chapter 1 that O-O analysis combines data and the processes that act on the data into things called objects, that similar objects can be grouped together into classes, and that O-O processes are called methods. Although O-O programmers use different terminology, they create the same kind of modular coding structures, except that the processes, or methods, are stored inside the objects, rather than as separate components.

FIGURE 5-30 Sequence structure.

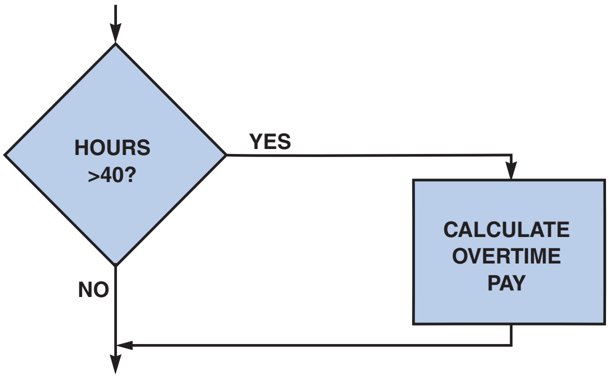


FIGURE 5-31 Selection structure.

Modular Design

Modular design is based on combinations of three **logical structures**, sometimes called **control structures**, which serve as building blocks for the process. Each logical structure must have a single entry and exit point. The three structures are called sequence, selection, and iteration. A rectangle represents a step or process, a diamond shape represents a condition or decision, and the logic follows the lines in the direction indicated by the arrows.

1. **Sequence.** The completion of steps in sequential order, one after another, as shown in Figure 5-30. One or more of the steps might represent a subprocess that contains additional logical structures.
2. **Selection.** The completion of one of two or more process steps based on the results of a test or condition. In the example shown in Figure 5-31, the system tests the input, and if the hours are greater than 40, it performs the CALCULATE OVERTIME PAY process.

3. **Iteration**. The completion of a process step that is repeated until a specific condition changes, as shown in Figure 5-32. An example of iteration is a process that continues to print paychecks until it reaches the end of the payroll file. Iteration also is called **looping**.

Sequence, selection, and iteration structures can be combined in various ways to describe processing logic.

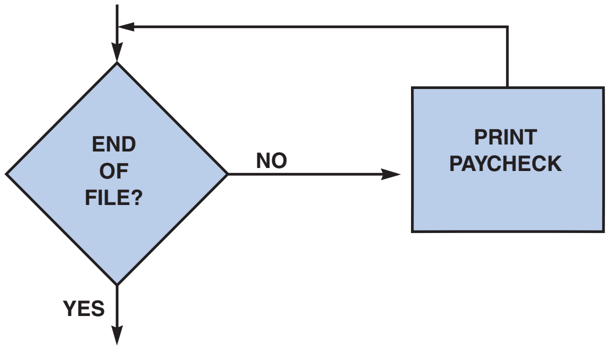


FIGURE 5-32 Iteration structure.

Structured English

Structured English is a subset of standard English that describes logical processes clearly and accurately. When you use structured English, you must conform to the following rules:

- Use only the three building blocks of sequence, selection, and iteration.
- Use indentation for readability.
- Use a limited vocabulary, including standard terms used in the data dictionary and specific words that describe the processing rules.

An example of structured English appears in Figure 5-33, which shows the VERIFY ORDER process that was illustrated earlier. Notice that the structured English version documents the actual logic that will be coded into the system. Structured English can help make your process descriptions accurate and understandable to users and system developers.

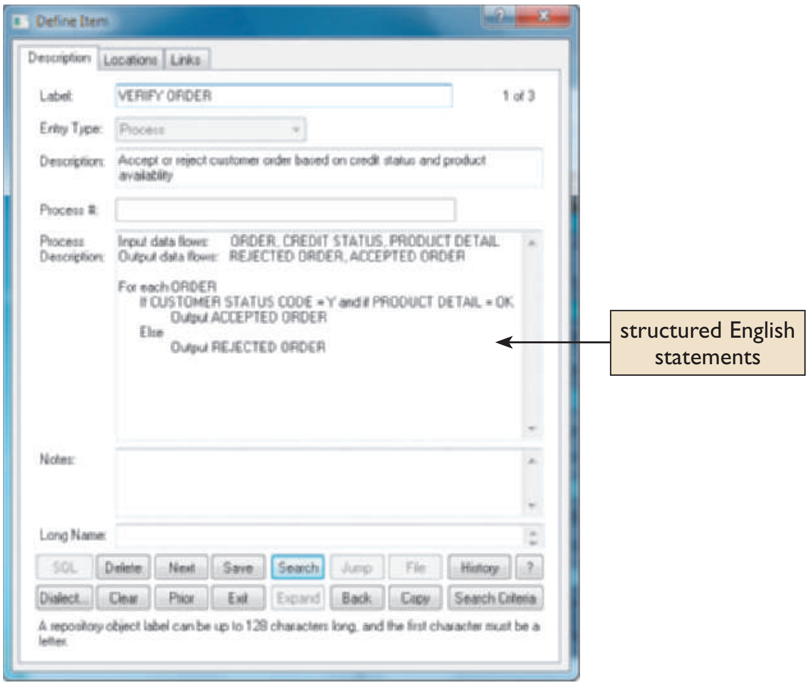


FIGURE 5-33 The VERIFY ORDER process description includes logical rules and a structured English version of the policy. Notice the alignment and indentation of the logic statements.

Structured English might look familiar to programming students because it resembles **pseudocode**, which is used in program design. Although the techniques are similar, the primary purpose of structured English is to describe the underlying business logic, while programmers, who are concerned with coding, mainly use pseudocode as a shorthand notation for the actual code.

Figure 5-34 shows another example of structured English. After you study the sales promotion policy, notice that the structured English version describes the processing logic that the system must apply. Following structured English rules ensures that your process descriptions are understandable to users who must confirm that the process is correct, as well as to other analysts and programmers who must design the information system from your descriptions.

VIDEO LEARNING SESSION: DECISION TABLES

Video Learning Sessions can help you understand key concepts, practice your skills, and check your work. To access the sessions, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and navigate to the **Video Learning Sessions** for this book. This session is about decision tables, why they are important process description tools, how to create decision tables, and how to analyze conditions and outcomes in a decision table.

Decision Tables

A **decision table** is a logical structure that shows every combination of conditions and outcomes. Analysts often use decision tables to describe a process and ensure that they have considered all possible situations. You can create decision tables using Microsoft PowerPoint, Word, or Excel.

ON THE WEB

To learn more about structured English, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Structured English link.

TABLES WITH ONE CONDITION If a process has a single condition, there only are two possibilities – *yes* or *no*. Either the condition is present or it is not, so there are only two rules. For example, to trigger an overtime calculation, the process condition might be: *Are the hours greater than 40?* If so, the calculation is made. Otherwise, it is not.

TABLES WITH TWO CONDITIONS Suppose you want to create a decision table based on the Verify Order business process shown in Figure 5-34. When documenting a process, it is important to ensure that you list every possibility. In this example, the process description contains two conditions: product stock status and customer credit status. If *both* conditions are met, the order is accepted. Otherwise the order is rejected.

VERIFY ORDER Business Process with Two Conditions

- An order will be accepted only if the product is in stock and the customer's credit status is OK.
- All other orders will be rejected.

FIGURE 5-34 The Verify Order business process has two conditions. For an order to be accepted, the product must be in stock and the customer must have an acceptable credit status.

After you identify all the conditions and outcomes, you are ready to create a decision table similar to the one shown in Figure 5-35. To create the table, follow the four steps listed in the margin.

1. Place the name of the process in a heading at the top left.
2. Enter the conditions under the heading, with one condition per line, to represent the customer status and availability of products.
3. Enter all potential combinations of Y/N (for yes and no) for the conditions. Each column represents a numbered possibility called a rule.
4. Place an X in the action entries area for each rule to indicate whether to accept or reject the order.

1 **VERIFY ORDER Process**

	1	2	3	4	
2 Credit status is OK	Y	Y	N	N	3
Product is in stock	Y	N	Y	N	
Accept order	X				4
Reject order		X	X	X	

FIGURE 5-35 Example of a simple decision table showing the processing logic of the VERIFY ORDER process.

Notice that each condition has two possible values, so the number of rules doubles each time you add another condition. For example, one condition creates two rules, two conditions create four rules, three conditions create eight rules, and so on. In the two-condition example in Figure 5-35, four possibilities exist, but Rule 1 is the *only* combination that will accept an order.

TABLES WITH THREE CONDITIONS Suppose the company now decides that the credit manager can waive the customer credit requirement, as shown in Figure 5-36. That creates a third condition, so there will be eight possible rules. The new decision table might resemble the one shown in Figure 5-37.

ON THE WEB

To learn more about decision tables, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Decision Tables link.

VERIFY ORDER Business Process with Three Conditions
• An order will be accepted only if the product is in stock and the customer's credit status is OK.
• The credit manager can waive the credit status requirement.
• All other orders will be rejected.

FIGURE 5-36 A third condition has been added to the Verify Order business process. For an order to be accepted, the product must be in stock and the customer must have an acceptable credit status. However, the credit manager now has the authority to waive the credit status requirement.

VERIFY ORDER Process with Credit Waiver (initial version)

	1	2	3	4	5	6	7	8
Credit status is OK	Y	Y	Y	Y	N	N	N	N
Product is in stock	Y	Y	N	N	Y	Y	N	N
Waiver from credit manager	Y	N	Y	N	Y	N	Y	N
Accept order	X	X			X			
Reject order			X	X		X	X	X

FIGURE 5-37 This table is based on the Verify Order conditions shown in Figure 5-36. With three conditions, there are eight possible combinations, or rules.

First, you must fill in the Y-N patterns, as shown in Figure 5-37. The best way to assure that all combinations appear is to use patterns like these. The first condition uses a pattern of Y-Y-Y-Y followed by N-N-N-N; the second condition uses a repeating Y-Y-N-N pattern; and the pattern in the third condition is a series of Y-Ns.

The next step is very important, regardless of the number of conditions. Each numbered column, or rule, represents a different set of conditions. You must analyze the logic carefully and show the outcome for each rule. Before going further, study the table in Figure 5-37 and be sure you understand the logical outcome for each of the eight rules.

When all the outcomes have been determined, you are ready to simplify the table. In a multi-condition table, some rules might be duplicates, redundant, or unrealistic. To simplify the table, follow these steps:

1. Study each combination of conditions and outcomes. When you have rules with three conditions, only one or two of them may control the outcome, and the other conditions simply do not matter.
2. If you identify conditions that do not affect the outcome, mark them with dashes (-) as shown in the first table in Figure 5-38.
3. Now combine and renumber the rules, as shown in the second table in Figure 5-38.

VERIFY ORDER Process with Credit Waiver (with rules marked for combination)

	1	2	3	4	5	6	7	8
Credit status is OK	Y	Y	-	-	N	N	-	-
Product is in stock	Y	Y	N	N	Y	Y	N	N
Waiver from credit manager	-	-	-	-	Y	N	-	-
Accept order	X	X			X			
Reject order			X	X		X	X	X

1. Because the product is not in stock, the other conditions do not matter.
2. Because the other conditions are met, the waiver does not matter.

VERIFY ORDER Process with Credit Waiver (after rule combination and simplification)

	1 (COMBINES PREVIOUS 1, 2)	2 (PREVIOUS 5)	3 (PREVIOUS 6)	4 (COMBINES PREVIOUS 3, 4, 7, 8)
Credit status is OK	Y	N	N	-
Product is in stock	Y	Y	Y	N
Waiver from credit manager	-	Y	N	-
Accept order	X	X		
Reject order			X	X

FIGURE 5-38 In the first table, dashes have been added to indicate that a condition is not relevant. In the second version, rules have been combined. Notice that in final version, only four rules remain. These rules document the logic, and will be transformed into program code when the system is developed.

If you follow these steps, you will see that Rules 1 and 2 can be combined because credit status is OK in both rules, so the waiver would not matter. Rules 3, 4, 7, and 8 also can be combined because the product is not in stock, so other conditions do not matter. The result is that instead of eight possibilities, only four logical rules control the Verify Order process.

MULTIPLE OUTCOMES In addition to multiple conditions, decision tables can have more than two possible outcomes. For example, the sales promotion policy shown in Figure 5-39 includes three conditions: Was the customer a preferred customer, did the customer order $1,000 or more, and did the customer use our company charge card? Based on these conditions, four possible actions can occur, as shown in the decision table in Figure 5-40.

SALES PROMOTION POLICY – Holiday Season, 2011

- Preferred customers who order $1,000 or more are entitled to a 5% discount, and an additional 5% discount if they use our charge card.
- Preferred customers who do not order $1,000 or more will receive a $25 bonus coupon.
- All other customers will receive a $5 bonus coupon.

FIGURE 5-39 A sales promotion policy with three conditions. Notice that the first statement contains two *separate* conditions – one for the 5% discount, and another for the additional discount.

Sales Promotion Policy (initial version)

	1	2	3	4	5	6	7	8
Preferred customer	Y	Y	Y	Y	N	N	N	N
Ordered $1,000 or more	Y	Y	N	N	Y	Y	N	N
Used our charge card	Y	N	Y	N	Y	N	Y	N
5% discount	X	X						
Additional 5% discount	X							
$25 bonus coupon			X	X				
$5 bonus coupon					X	X	X	X

FIGURE 5-40 This decision table is based on the sales promotion policy in Figure 5-39. This is the initial version of the table, before simplification.

As explained in the preceding section, most tables can be simplified, and this one is no exception. When you study the conditions and outcomes, you realize that:

- In Rule 1, all three conditions are met, so *both* 5% discounts apply.
- In Rule 2, a preferred customer orders $1,000 or more, but does not use our charge card, so only *one* 5% discount applies.
- Rules 3 and 4 can be combined into a single rule. Why? If preferred customers do not order $1,000 or more, it does not matter whether they use our charge card – either way, they only earn a $25 bonus coupon. Therefore, Rules 3 and 4 really are a single rule.
- Rules 5, 6, 7, and 8 also can be combined into a single rule – because if the person is *not* a preferred customer, he or she can *only* receive a $5 bonus coupon, and the other conditions simply do not matter. So, you insert a dash if a condition is irrelevant, as shown in Figure 5-41.

If you add dashes for rules that are not relevant, your table should resemble the one shown in Figure 5-41. When you combine and simplify the results, only four rules remain: Rule 1, Rule 2, Rule 3 (a combination of initial Rules 3 and 4), and Rule 4 (a combination of initial Rules 5, 6, 7, and 8).

Sales Promotion Policy (final version)

	1	2	3	4	5	6	7	8
Preferred customer	Y	Y	Y	Y	N	N	N	N
Ordered $1,000 or more	Y	Y	N	N	-	-	-	-
Used our charge card	Y	N	-	-	-	-	-	-
5% discount	X	X						
Additional 5% discount	X							
$25 bonus coupon			X	X				
$5 bonus coupon					X	X	X	X

FIGURE 5-41 In this version, dashes have been added to indicate that a condition is not relevant. At this point, it appears that several rules can be combined.

Decision tables often are the best way to describe a complex set of conditions. Many analysts use decision tables because they are easy to construct and understand, and programmers find it easy to work from a decision table when developing code.

CASE IN POINT 5.2: Rock Solid Outfitters (Part 1)

Leah Jones is the IT manager at Rock Solid Outfitters, a medium-sized supplier of outdoor climbing and camping gear. Steve Allen, the marketing director, has asked Leah to develop a special Web-based promotion. As Steve described it to Leah, Rock Solid will provide free shipping for any customer who either completes an online survey form or signs up for the Rock Solid online newsletter. Additionally, if a customer completes the survey *and* signs up for the newsletter, Rock Solid will provide a $10 merchandise credit for orders of $100 or more. Leah has asked you to develop a decision table that will reflect the promotional rules that a programmer will use. She wants you to show all possibilities, then to simplify the results to eliminate any combinations that would be unrealistic or redundant.

Decision Trees

A **decision tree** is a graphical representation of the conditions, actions, and rules found in a decision table. Decision trees show the logic structure in a horizontal form that resembles a tree with the roots at the left and the branches to the right. Like flowcharts, decision trees are useful ways to present the system to management. Decision trees and decision tables provide the same results, but in different forms. In many situations, a graphic is the most effective means of communication.

Figure 5-42 is based on the sales promotion policy shown in Figure 5-39 on page 228. A decision tree is read from left to right, with the conditions along the various branches and the actions at the far right. Because the example has two conditions with four resulting sets of actions, the example has four terminating branches at the right side of the tree.

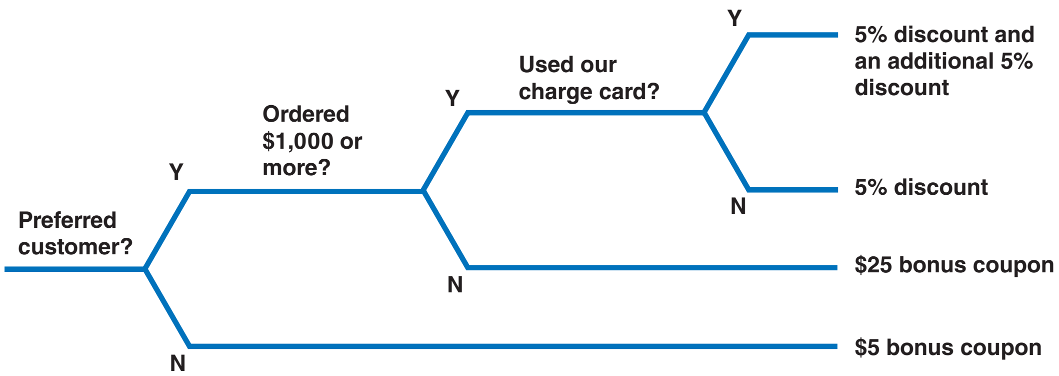


FIGURE 5-42 This example is based on the same Sales Promotion Policy shown in the decision tables in Figures 5-40 and 5-41 on the previous page. Like a decision table, a decision tree shows all combinations of conditions and outcomes. The main difference is the graphical format, which many viewers find easier to interpret.

Whether to use a decision table or a decision tree often is a matter of personal preference. A decision table might be a better way to handle complex combinations of conditions. On the other hand, a decision tree is an effective way to describe a relatively simple process.

CASE IN POINT 5.3: Rock Solid Outfitters (Part 2)

Leah Jones, the IT manager at Rock Solid Outfitters, thinks you did a good job on the decision table task she assigned to you. Now she wants you to use the same data to develop a decision tree that will show all the possibilities for the Web-based promotion described in Part 1 of the case. She also wants you to discuss the pros and cons of decisions tables versus decision trees.

LOGICAL VERSUS PHYSICAL MODELS

While structured analysis tools are used to develop a logical model for a new information system, such tools also can be used to develop physical models of an information system. A physical model shows how the system's requirements are implemented. During the systems design phase, you create a physical model of the new information system that follows from the logical model and involves operational tasks and techniques.

Sequence of Models

What is the relationship between logical and physical models? Think back to the beginning of the systems analysis phase, when you were trying to understand the existing system. Rather than starting with a logical model, you first studied the physical operations of the existing system to understand how the current tasks were carried out. Many systems analysts create a physical model of the current system and then develop a logical model of the current system before tackling a logical model of the new system. Performing that extra step allows them to understand the current system better.

Four-Model Approach

Many analysts follow a **four-model approach**, which means that they develop a physical model of the current system, a logical model of the current system, a logical model of the new system, and a physical model of the new system. The major benefit of the four-model approach is that it gives you a clear picture of current system functions before you make any modifications or improvements. That is important because mistakes made early in systems development will affect later SDLC phases and can result in unhappy users and additional costs. Taking additional steps to avoid these potentially costly mistakes can prove to be well worth the effort. Another advantage is that the requirements of a new information system often are quite similar to those of the current information system, especially where the proposal is based on new computer technology rather than a large number of new requirements. Adapting the current system logical model to the new system logical model in these cases is a straightforward process.

The only disadvantage of the four-model approach is the added time and cost needed to develop a logical and physical model of the current system. Most projects have very tight schedules that might not allow time to create the current system models. Additionally, users and managers want to see progress on the new system — they are much less concerned about documenting the current system. As a systems analyst, you must stress the importance of careful documentation and resist the pressure to hurry the development process at the risk of creating serious problems later.

CASE IN POINT 5.4: TIP TOP STAFFING

Tip Top Staffing supplies employees to hundreds of IT firms that require specialized skills for specific projects. Systems analysts Lisa Nuevo and Bill Goodman are working on the logical model of Tip Top's billing and records system, using DFDs, a data dictionary, and process descriptions. At some point while working on the logical model of the system, Lisa felt that some improvements should be made in the data forms that Tip Top uses to obtain information about job applicants. Was the subject of improving the forms a physical implementation issue? Is Lisa going off on a tangent by considering *how* something will be done, instead of sticking to *what* will be done?

A QUESTION OF ETHICS

This is your first week in your new job at Safety Zone, a leading producer of IT modeling software. Your prior experience with a smaller competitor gave you an edge in landing the job, and you are excited about joining a larger company in the same field.

So far, all is going well and you are getting used to the new routine. However, you are concerned about one issue. In your initial meeting with the IT manager, she seemed very interested in the details of your prior position, and some of her questions made you a little uncomfortable. She did not actually ask you to reveal any proprietary information, but she made it clear that Safety Zone likes to know as much as possible about its competitors.

Thinking about it some more, you try to draw a line between information that is OK to discuss, and topics such as software specifics or strategy that should be considered private. This is the first time you have ever been in a situation like this. How will you handle it?

CHAPTER SUMMARY

During data and process modeling, a systems analyst develops graphical models to show how the system transforms data into useful information. The end product of data and process modeling is a logical model that will support business operations and meet user needs. Data and process modeling involves three main tools: data flow diagrams, a data dictionary, and process descriptions.

Data flow diagrams (DFDs) graphically show the movement and transformation of data in the information system. DFDs use four symbols: The process symbol transforms data; the data flow symbol shows data movement; the data store symbol shows data at rest; and the external entity symbol represents someone or something connected to the information system. Various rules and techniques are used to name, number, arrange, and annotate the set of DFDs to make them consistent and understandable.

A set of DFDs is like a pyramid with the context diagram at the top. The context diagram represents the information system's scope and its external connections but not its internal workings. Diagram 0 displays the information system's major processes, data stores, and data flows and is the exploded version of the context diagram's process symbol, which represents the entire information system. Lower-level DFDs show additional detail of the information system through the leveling technique of numbering and partitioning. Leveling continues until you reach the functional primitive processes, which are not decomposed further and are documented with process descriptions. All diagrams must be balanced to ensure their consistency and accuracy.

The data dictionary is the central documentation tool for structured analysis. All data elements, data flows, data stores, processes, entities, and records are documented in the data dictionary. Consolidating documentation in one location allows you to verify the information system's accuracy and consistency more easily and generate a variety of useful reports.

Each functional primitive process is documented using structured English, decision tables, and decision trees. Structured English uses a subset of standard English that defines each process with combinations of the basic building blocks of sequence, selection, and iteration. You also can document the logic by using decision tables or decision trees.

Structured analysis tools can be used to develop a logical model during one systems analysis phase, and a physical model during the systems design phase. Many analysts use a four-model approach, which involves a physical model of the current system, a logical model of the current system, a logical model of the new system, and a physical model of the new system.

Key Terms and Phrases

alias *219*
balancing *212*
black box *201*
black hole *203*
business logic *201*
business rules *201*
child diagram *211*
context diagram *208*
control structures *224*
data dictionary *217*
data element *217*
data flow *202*
data flow diagram (DFD) *200*
data item *217*
data repository *217*
data store *203*
data structures *217*
decision table *226*
decision tree *230*
decomposing *213*
diagram 0 *209*
diverging data flow *211*
domain *220*
entity *205*
exploding *213*
field *217*
four-model approach *231*
functional primitive *211*
Gane and Sarson *200*
gray hole *203*
iteration *225*
length *219*
leveling *212*
logical model *198*
logical structures *224*
looping *225*
modular design *224*
parent diagram *211*
partitioning *213*
physical model *198*
process *200*
process 0 *208*
process description *224*
pseudocode *225*
records *217*
selection *224*
sequence *224*
sink *205*
source *205*
spontaneous generation *203*
structured English *225*
terminators *205*
type *219*
validity rules *220*
Yourdon *200*

Learn It Online

Instructions: To complete the Learn It Online exercises, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to the resources for this chapter, and click the link for the exercise you want to complete.

1 Chapter Reinforcement

TF, MC, and SA

Click one of the Chapter Reinforcement links for Multiple Choice, True/False, or Short Answer. Answer each question and submit to your instructor.

2 Flash Cards

Click the Flash Cards link and read the instructions. Type 20 (or a number specified by your instructor) in the Number of playing cards text box, type your name in the Enter your Name text box, and then click the Flip Card button. When the flash card is displayed, read the question and then click the ANSWER box arrow to select an answer. Flip through the Flash Cards. If your score is 15 (75%) correct or greater, click Print on the File menu to print your results. If your score is less than 15 (75%) correct, then redo this exercise by clicking the Replay button.

3 Practice Test

Click the Practice Test link. Answer each question, enter your first and last name at the bottom of the page, and then click the Grade Test button. When the graded practice test is displayed on your screen, click Print on the File menu to print a hard copy. Continue to take practice tests until you score 80% or better.

4 Who Wants To Be a Computer Genius?

Click the Computer Genius link. Read the instructions, enter your first and last name at the bottom of the page, and then click the Play button. When your score is displayed, click the PRINT RESULTS link to print a hard copy.

5 Wheel of Terms

Click the Wheel of Terms link. Read the instructions, and then enter your first and last name and your school name. Click the PLAY button. When your score is displayed on the screen, right-click the score and then click Print on the shortcut menu to print a hard copy.

6 Crossword Puzzle Challenge

Click the Crossword Puzzle Challenge link. Read the instructions, and then click the Continue button. Work the crossword puzzle. When you are finished, click the Submit button. When the crossword puzzle is redisplayed, submit it to your instructor.

SCR Associates Case Simulation Session 5: Data and Process Modeling

Overview

The SCR Associates case study is a Web-based simulation that allows you to practice your skills in a real-world environment. The case study transports you to SCR's intranet, where you complete 12 work sessions, each aligning with a chapter. As you work on the case, you will receive e-mail and voice mail messages, obtain information from SCR's online libraries, and perform various tasks.

How do I use the case?

- Review the SCR background material in Chapter 1.
- Read the Preview for this session and study the Task List.
- Visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to the **SCR Case Simulation**, and locate the intranet link.
- Enter your name and the password **sad9e**. An opening screen will display the 12 sessions.
- Select this session. Check your e-mail and voice mail carefully, and then work on the tasks.

Preview: Session 5

You recently completed requirements modeling tasks for the new Training Information Management System (TIMS). Now you are ready to begin data and process modeling, which will produce a logical model of the new system. You will create DFDs, develop a data dictionary, and use decision tables and trees.

Task List

1. *Jesse wants to see a context diagram and a diagram 0 DFD for TIMS.*
2. *Need to review the JAD session summary again! Try to identify four main TIMS functions and draw a lower-level DFD for each process.*
3. *Prepare a reply to Jesse's message about CASE tools. Search the Internet to find two more alternatives.*
4. *Prepare a decision table and a decision tree that show the logical rules described in Jesse's message about fees and discounts.*

FIGURE 5-43 Task list: Session 5.

Chapter Exercises

Review Questions

1. Describe data and process modeling, and name the main data and process modeling techniques.
2. Describe the Gane and Sarson symbols used for processes, data flows, data stores, and entities. Give four examples of typical names for processes, data flows, data stores, and entities.
3. What is the relationship between a context diagram and diagram 0, and which symbol is not used in a context diagram?
4. What is meant by an exploded DFD?
5. Describe a data dictionary and give examples of how and when it is used.
6. Explain the DFD leveling technique.
7. What is a balanced DFD?
8. Describe the steps in creating a decision table.
9. Discuss the pros and cons of decision tables versus decision trees.
10. What is structured English?

Discussion Topics

1. Suppose you were assigned to develop a logical model of the registration system at a school or college. Would you be better off using a top-down approach, or would a bottom-up strategy be better? What would influence your decision?
2. Some systems analysts find it better to start with a decision table, then construct a decision tree. Others believe it is easier to do it in the reverse order. Which do you prefer? Why?
3. A systems analyst attended a weeklong workshop on structured analysis. When she returned to her job, she told her boss that structured analysis was not worth the time to learn and use on the job. Her view was that it was too academic and had too much new terminology to be useful in a practical setting. Do you agree or disagree? Defend your position.
4. This chapter describes a black box concept that allows more detail to be shown as a process is exploded. Can the concept be applied in business management generally, or is it limited to information systems design? Provide reasons and examples with your answer.

Projects

1. Draw a context diagram and a diagram 0 DFD that represent the registration system at your school or an imaginary school.
2. On the Internet, locate at least three firms that offer CASE tools. Write e-mail messages to the companies to find out whether they offer demonstration copies or student versions of their products.
3. Suppose that you want to demonstrate a decision table to someone who has never seen one. Think of an example, with two or three conditions, from everyday life. Draw a decision table that captures all possible outcomes.
4. The data flow symbols shown on page 201 were designed by Ed Yourdon, a well-known IT author, lecturer, and consultant. Many IT professionals consider him to be among the most influential men and women in the software field. Learn more about Mr. Yourdon by visiting his Web site at www.yourdon.com, and write a brief review of his accomplishments.

Apply Your Knowledge

The Apply Your Knowledge section contains four mini-cases. Each case describes a situation, explains your role in the case, and asks you to respond to questions. You can answer the questions by applying knowledge you learned in the chapter.

1 Digital Consulting

Situation:

You are a senior systems analyst at Digital Consulting, a growing IT consulting firm. You are leading the development team for a major client. You need to explain structured analysis to your two newly hired junior analysts (Sara and Mike) before meeting with the client tomorrow afternoon.

1. Describe the rules for creating a context diagram.
2. Make a basic list of dos and don'ts when developing DFDs.
3. Explain the importance of leveling and balancing.
4. Ask Sara and Mike to review the order system context diagram on page 208, and compare it with the order system diagram 0 DFD on page 212. Then ask them to answer the following questions: (a) How many external entities are shown in each diagram? (b) In each diagram, how many data flows connect to the external entities? (c) How many subprocesses are identified in the diagram 0 DFD? (d) Could the data store have been shown in the context diagram? Why or why not?

2 Precision Tools

Situation:

Precision Tools sells a line of high-quality woodworking tools. When customers place orders on the company's Web site, the system checks to see if the items are in stock, issues a status message to the customer, and generates a shipping order to the warehouse, which fills the order. When the order is shipped, the customer is billed. The system also produces various reports.

1. Draw a context diagram for the order system.
2. Draw a diagram 0 DFD for the order system.
3. Name four attributes that you can use to define a process in the order system.
4. Name four attributes that you can use to define an entity in the order system.

3 Claremont School

Situation:

The Claremont School course catalog reads as follows: "To enroll in CIS 288, which is an advanced course, a student must complete two prerequisites: CIS 110 and CIS 286. A student who completes either one of these prerequisites and obtains the instructor's permission, however, will be allowed to take CIS 288."

1. Create a decision table that describes the Claremont School course catalog regarding eligibility for CIS 288. Show all possible rules.
2. Simplify the table you just created. Describe the results.
3. Draw a simplified decision tree to represent the Claremont School catalog. Describe the results.
4. Why might you use a decision tree rather than a decision table?

4 City Bus Lines

Situation:

City Bus Lines is developing an information system that will monitor passenger traffic, peak travel hours, and equipment requirements. The IT manager wants you to document a process called BALANCE that determines whether extra buses currently are needed on a particular route. The BALANCE process automatically assigns additional buses to that route, but *only* if all other routes are operating on schedule. In any case, a supervisor can override the automatic BALANCE process if he or she so desires.

1. Create a decision table that describes the bus transfer process.
2. Draw a decision tree that describes the bus transfer process.
3. Name four attributes that you can use to define a data flow in the bus information system.
4. Name four attributes that you can use to define a data store in the bus information system.

Case Studies

Case studies allow you to practice specific skills learned in the chapter. Each chapter contains several case studies that continue throughout the textbook, and a chapter capstone case.

New Century Health Clinic

New Century Health Clinic offers preventive medicine and traditional medical care. In your role as an IT consultant, you will help New Century develop a new information system.

Background

You began the systems analysis phase at New Century Health Clinic by completing a series of interviews, reviewing existing reports, and observing office operations. (Your instructor may provide you with a sample set of interview summaries.)

As you learned, the doctors, nurses, and physical therapists provide services and perform various medical procedures. All procedures are coded according to Current Procedure Terminology, which is published by the American Medical Association. The procedure codes consist of five numeric digits and a two-digit suffix, and are used for all billing and insurance claims.

From your fact-finding, you determined that seven reports are required at the clinic. The first report is the daily appointment list for each provider. The list shows all scheduled appointment times, patient names, and services to be performed, including the procedure code and description. A second daily report is the call list, which shows the patients who are to be reminded of their next day's appointments. The call list includes the patient name, telephone number, appointment time, and provider name. The third report is the weekly provider report that lists each of the providers and the weekly charges generated, plus a month-to-date (MTD) and a year-to-date (YTD) summary.

The fourth report is the statement — a preprinted form that is produced monthly and mailed in a window envelope. Statement header information includes the statement date, head of household name and address, the previous month's balance, the total household charges MTD, the total payments MTD, and the current balance. The bottom section of the statement lists all activity for the month in date order. For each service performed, a line shows the patient's name, the service date, the procedure code and description, and the charge. The statement also shows the date and amount of all payments and insurance claims. When an insurance payment is received, the source and amount are noted on the form. If the claim is denied or only partially paid, a code is used to explain the reason. A running balance appears at the far right of each activity line.

The associates also require two insurance reports: the weekly Insurance Company Report and the monthly Claim Status Summary. In addition to these six reports, the office staff would like to have mailing labels and computer-generated postcards for sending reminders to patients when it is time to schedule their next appointment. Reminders usually are mailed twice monthly. Now you are ready to organize the facts you gathered and prepare a system requirements document that represents a logical model of the proposed system. Your tools will include DFDs, a data dictionary, and process descriptions.

Assignments

1. Prepare a context diagram for New Century's information system.
2. Prepare a diagram 0 DFD for New Century. Be sure to show numbered processes for handling appointment processing, payment and insurance processing, report processing, and records maintenance. Also, prepare lower-level DFDs for each numbered process.

3. Prepare a list of data stores and data flows needed for the system. Under each data store, list the data elements required.
4. Prepare a data dictionary entry and process description for one of the system's functional primitives.

PERSONAL TRAINER, INC.

Personal Trainer, Inc., owns and operates fitness centers in a dozen Midwestern cities. The centers have done well, and the company is planning an international expansion by opening a new "supercenter" in the Toronto area. Personal Trainer's president, Cassia Umi, hired an IT consultant, Susan Park, to help develop an information system for the new facility. During the project, Susan will work closely with Gray Lewis, who will manage the new operation.

Background

Susan Park has completed a preliminary investigation and performed the fact-finding tasks that were described in Chapters 2 and 4. Now, she will use the results to develop a logical model of the proposed information system.

Assignments

Before you perform the following tasks, you should review the information provided in Chapters 2 and 4 of the case.

1. Prepare a context diagram for the new system.
2. Prepare a diagram 0 DFD for the new system.
3. Write a brief memo that explains the importance of leveling a set of DFDs.
4. Write a brief memo that explains the importance of balancing a set of DFDs.

CHAPTER CAPSTONE CASE: SoftWear, Limited

SoftWear, Limited (SWL), is a continuing case study that illustrates the knowledge and skills described in each chapter. In this case study, the student acts as a member of the SWL systems development team and performs various tasks.

Background

Rick Williams, a systems analyst, and Carla Moore, a programmer/analyst, continued their work on the SWL payroll system project. After completing detailed interviews and other fact-finding activities, Rick and Carla now understand how the current system operates and the new requirements desired by users. They are ready to organize and document their findings by preparing a logical model of the payroll system.

Data Flow Diagrams

After they completed the preliminary investigation, Rick and Carla felt that they knew more about the system entities and how they interacted.

The two analysts knew that the payroll department issues paychecks based on timesheet data submitted by department heads, and that each employee receives a W-2 form at the end of the year. They also knew that the human resources department prepares employee status changes, and the payroll department enters the pay data. The diagram also noted the output of state and federal government reports and internal reports to SWL's finance and payroll departments. The credit union and the SWL stock transfer department reports and fund transfers also were included.

Using this information, Rick and Carla prepared a sketch of a context diagram and scheduled a meeting for the next day with Amy Calico, director of the payroll department, to discuss the diagram. At the meeting, Amy made several comments:

- The human resources department would be setting up additional ESIP deduction choices for employees under a new 401(k) plan. Human resources also would receive ESIP reports from the payroll system.
- The payroll department enters timesheet data received from department heads, who do not interact directly with the system. Rather than showing the department head entity symbol on the context diagram, the input data flow from the payroll department should be expanded and called PAY DETAIL.
- State and federal reporting requirements differ, so they should be treated as two separate entities. Also, periodic changes in government tax rates should be shown as inputs to the payroll system.
- All accounting reports, except for an overall financial summary, should be distributed to the accounting department instead of to the finance department. The accounting department also should receive a copy of the payroll report.
- The bank returns cleared payroll checks to the payroll department once a month. Amy reminded the analysts that the payroll system handles the reconciliation of payroll checks.

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

After discussing Amy's comments, Rick and Carla prepared the final version of the payroll system context diagram shown in Figure 5-44.

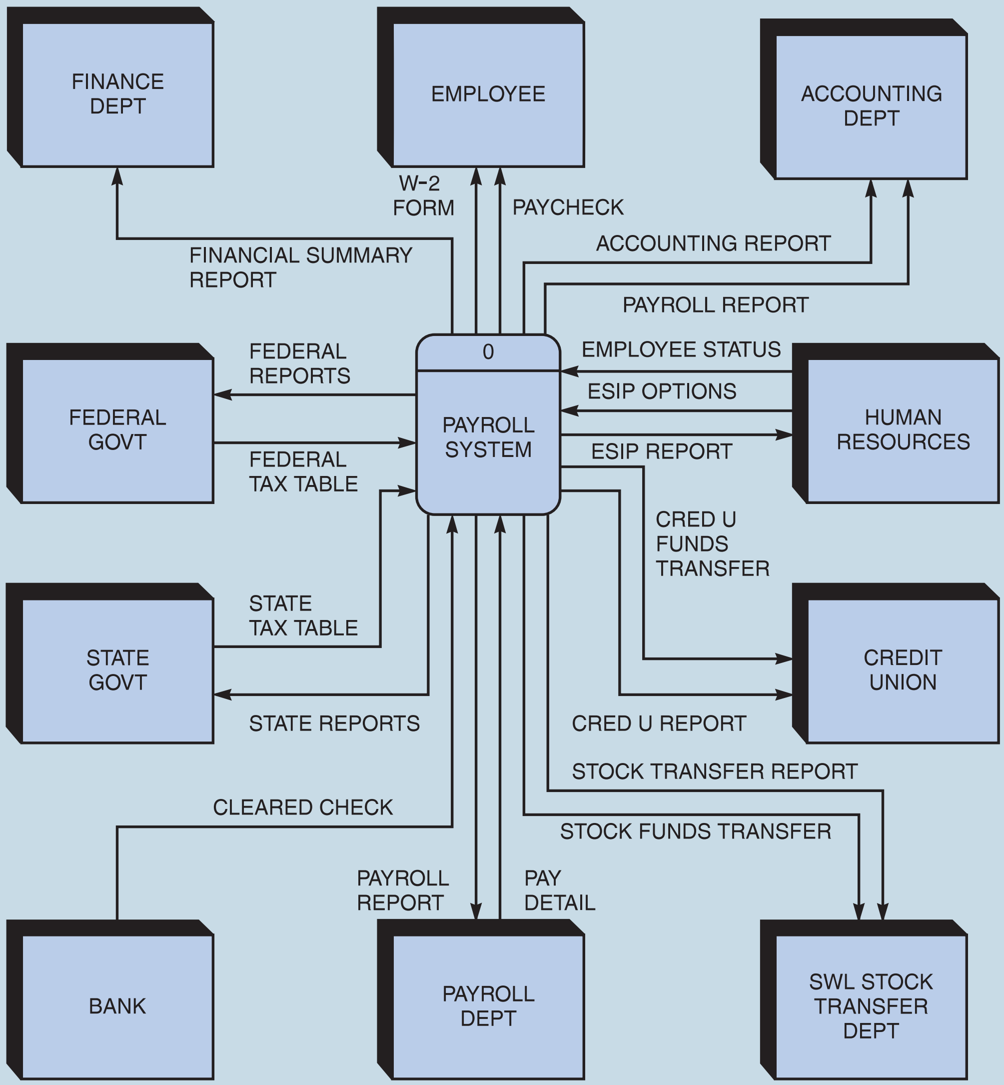


FIGURE 5-44 Final context diagram for SoftWear, Limited's payroll system.

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

While their conversation with Amy Calico still was fresh in her mind, Carla proposed that they construct the diagram 0 DFD. After going through several draft versions, they completed the diagram 0 shown in Figure 5-45. They identified four processes: the check reconciliation subsystem, the pay employee subsystem, the payroll accounting subsystem, and a subsystem that would handle all voluntary deductions, which they called the ESIP deduction subsystem.

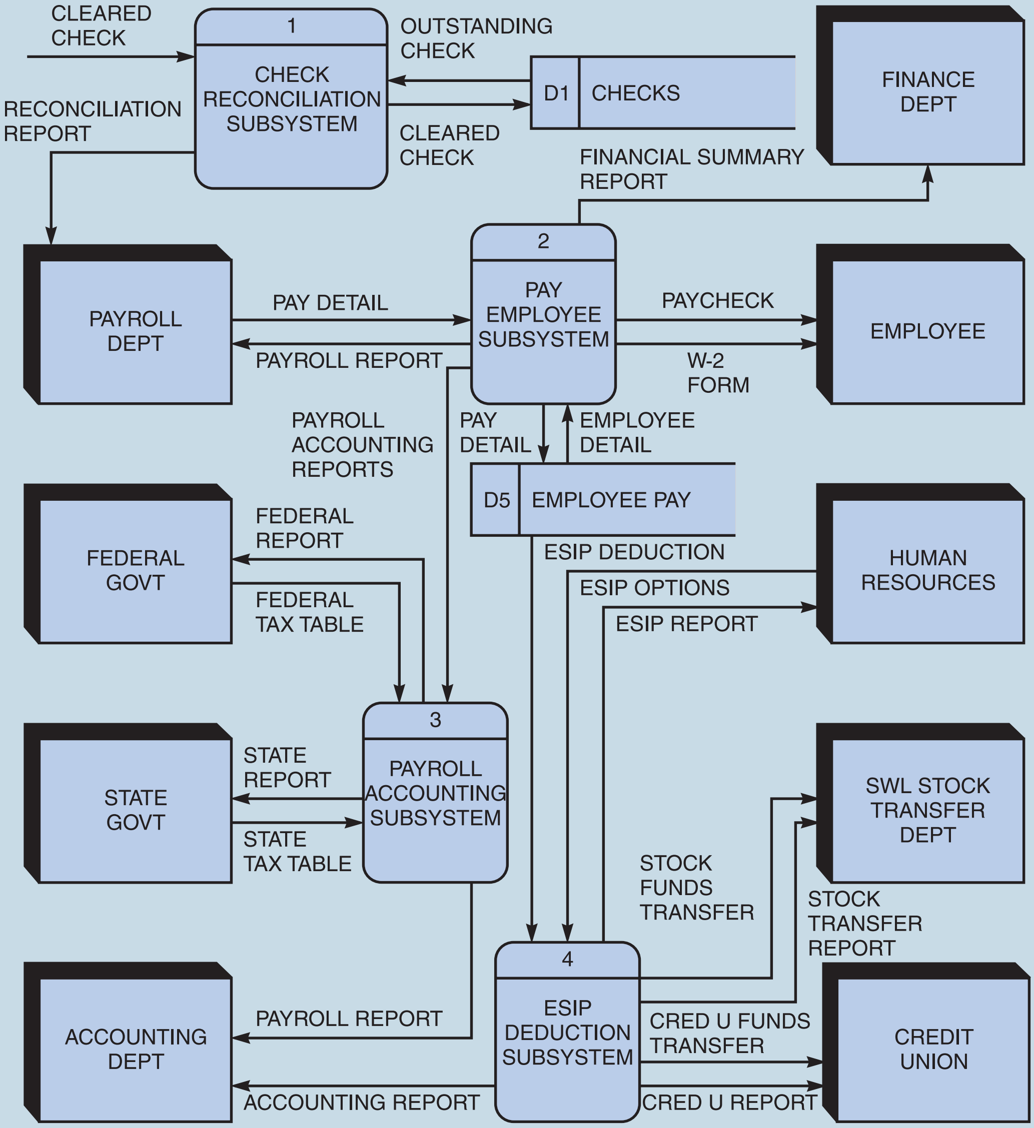


FIGURE 5-45 Diagram 0 DFD for SoftWear, Limited's payroll system.

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

Over the next few days, Rick concentrated on partitioning the pay employee subsystem and the ESIP subsystem, while Carla developed the lower-level diagrams for the other two subsystems.

At that point, Rick considered the problem of applying certain deductions on a monthly cycle, even though the deductions were made weekly. To provide flexibility, he decided to use two separate processes, as shown in Figure 5-46. When he finished, his diagram 4 DFD contained the two processes EXTRACT DEDUCTION and APPLY DEDUCTION, as well as a local data store, UNAPPLIED DEDUCTIONS. Several local data flows also were included. The first process, EXTRACT DEDUCTION, would deduct the proper amount in each pay period. The deductions would be held in the temporary data store

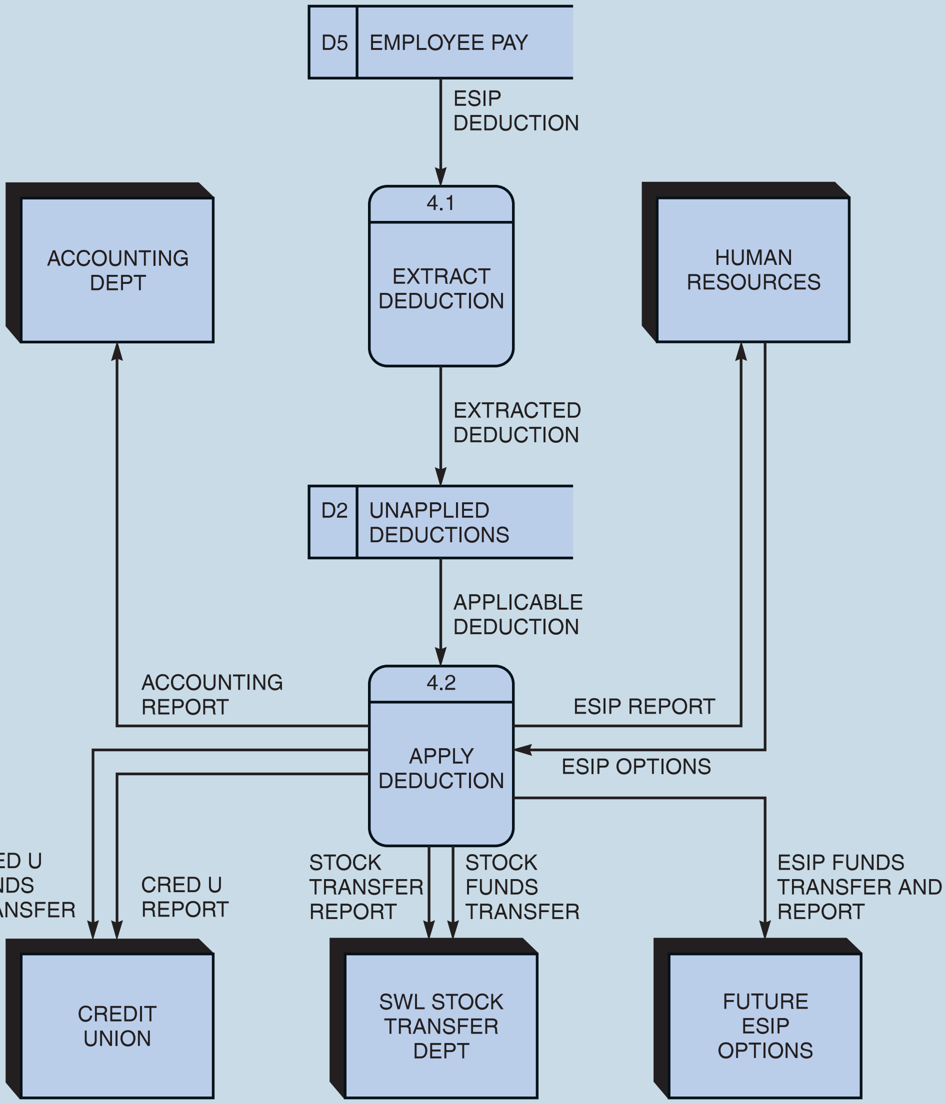


FIGURE 5-46 Diagram 4 DFD for SoftWear, Limited's payroll system shows the detail of process 4, the ESIP DEDUCTION SUBSYSTEM.

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

and then applied in the APPLY DEDUCTION process on a weekly or monthly basis, depending on the type of deduction. Rick decided that those processes were functional primitives and he did not need to partition them further. That task completed the logical model of the new SWL payroll system.

Rick and Carla also considered the physical design of the ESIP deduction subsystem that would be completed later. They knew that it would be necessary to add some new forms and to redesign others. They saw that the human resources department would need a new form for enrollments or deduction changes for the credit union, SWL stock purchase plan, or any new ESIP choices that might be offered in the future. The payroll department then could use the form as its official notification. To provide for future expansion and add flexibility, the human resources department also would need a form to notify payroll of any new type of deduction, with a deduction code, the name of the deduction, and the payroll cycle involved. Rick anticipated that the new system would eliminate problems with improper deductions, while adding flexibility and reducing maintenance costs.

Data Dictionary and Process Descriptions

As they constructed the DFDs for the payroll system, Rick and Carla also developed the data dictionary entries with supporting process descriptions. After completing the documentation of the ESIP deduction subsystem, Carla and Rick met with Amy to review the logical model for the subsystem. After a thorough discussion of all proposed changes and processing, Amy approved the model.

Rick and Carla continued their analysis and documentation of the payroll system over the next several days. As they completed a model of a portion of the information system, they would meet with the appropriate users at SWL to review the model, obtain user input, make necessary adjustments to the model, and obtain the users' approval. After Rick and Carla finished the complete payroll information system logical model, they turned their attention to completing the rest of the system requirements document.

SWL Team Tasks

Suppose that you are working with Rick and Carla when a new systems request comes in. SWL's vice president of marketing, Amy Neal, wants to change the catalog mailing program and provide a reward for customers who use the Internet.

Amy's plan specifies that customers will remain on SWL's mailing list if they either requested a catalog, ordered from SWL in the last two years, or signed the guest register on SWL's new Web site. To encourage Internet visitors, customers who register on the Web site also will receive a special discount certificate.

To document the requirements, Rick wants you to design a decision table. Initially, it appears to have eight rules, but you notice that some of those rules are duplicates, or might not be realistic combinations.

1. Design the decision table with all possibilities.
2. Simplify the table by combining rules where appropriate.
3. Draw a decision tree that reflects Amy Neal's policy.
4. Create a set of structured English statements that accurately describes the policy.

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

Manage the SWL Project

You have been asked to manage SWL's new information system project. One of your most important activities will be to identify project tasks and determine when they will be performed. Before you begin, you should review the SWL case in this chapter. Then list and analyze the tasks, as follows:

LIST THE TASKS Start by listing and numbering at least 10 tasks that the SWL team needs to perform to fulfill the objectives of this chapter. Your list can include SWL Team Tasks and any other tasks that are described in this chapter. For example, Task 3 might be to Identify the system entities, and Task 6 might be to Draw a context diagram.

ANALYZE THE TASKS Now study the tasks to determine the order in which they should be performed. First identify all concurrent tasks, which are not dependent on other tasks. In the example shown in Figure 5-47, Tasks 1, 2, 3, 4, and 5 are concurrent tasks, and could begin at the same time if resources were available.

Other tasks are called dependent tasks, because they cannot be performed until one or more earlier tasks have been completed. For each dependent task, you must identify specific tasks that need to be completed before this task can begin. For example, you would want to identify the system entities before you could draw a context diagram, so Task 6 cannot begin until Task 3 is completed, as Figure 5-47 shows.

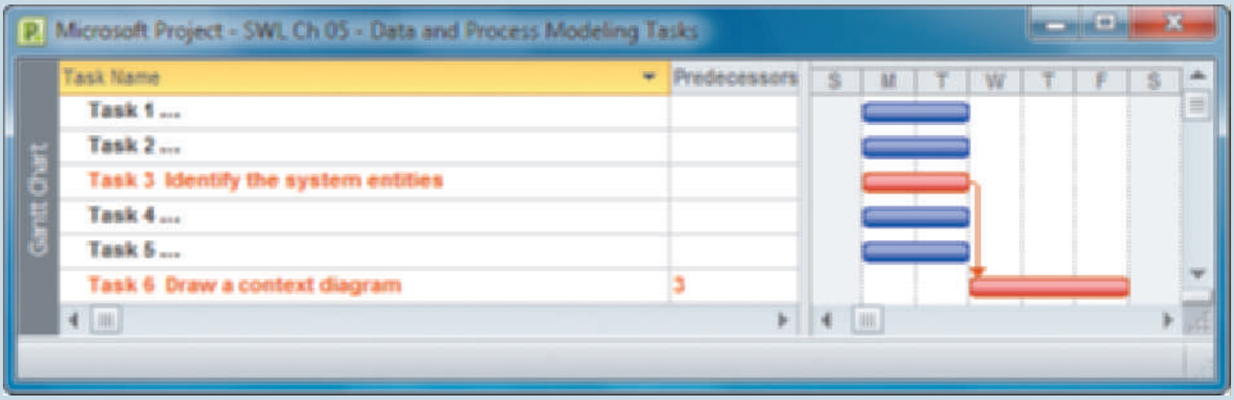


FIGURE 5-47 Tasks 1, 2, 3, 4, and 5 are concurrent tasks that could be performed at the same time. Task 6 is a dependent task that cannot be performed until Task 3 has been completed.

Chapter 3 describes project management tools, techniques, and software. To learn more, you can use the Features section on your Student Study Tool CD-ROM, or visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and locate the project management resources library for this book. On the Web, Microsoft offers demo versions, training, and tips for using Project 2010. You also can visit the OpenWorkbench.org site to learn more about this free, open-source software.

Ready for a Challenge?

In additional to technical skills, IT professionals need critical thinking skills such as perception, organization, analysis, problem-solving, and decision-making. The Ready for a Challenge feature can help you learn, practice, and apply critical thinking skills that you can take to the workplace.

The IT team at Game Technology is moving forward with the new Customer Contact Care information system, or C^3.

Your next assignment is to develop a set of data flow diagrams (DFDs). To be sure you can handle the tasks, you decide to review Chapter 5 of your systems analysis textbook. Based on previous requirements modeling, you know that the new C^3 system will have three external entities, with the following data flows:

Entity	Data Flow from the Entity	Data Flow to the Entity
Customer	Contact Information	Sales Specials
Sales Records System	Customer Sales History	None
Marketing Rep	Suggested Contact Plan	Sales Feedback

You also know that the C^3 system will have three subsystems: **Analyze Customer Data, Analyze Sales History**, and **Manage Contact Plan**, and a data store called **Customer Profile.** The data flows are as follows:

Data Flow	From	To
Contact Information	Customer entity	Analyze Customer Data process
Sales Specials	Manage Contact Plan process	Customer entity
Customer Sales History	Sales Records System entity	Analyze Sales History process
Suggested Contact Plan	Marketing Rep entity	Develop Contact Plan process
Sales Feedback	Analyze Sales History process	Marketing Rep entity
Customer Data (two-way)	Analyze Customer Data process	Customer Profile data store
Customer Data (two-way)	Customer Profile data store	Analyze Customer Data process
Profile Data	Customer Profile data store	Manage Contact Plan process
Sales Data	Analyze Sales History process	Customer Profile data store

Practice Tasks

A. Draw a context diagram for the new C^3 system.
B. Draw a DFD diagram 0 that shows the three processes, the data store, and the data flows.

After you complete the Practice Tasks, to check your work and view sample answers, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to the resources for this chapter, and locate Ready for a Challenge?.

The Challenge

Your context diagram and DFD diagram 0 were accurate, but there have been some design changes. Management has decided to connect the external **Accounts Receivable System** directly to the C^3 system as a fourth entity. A two-way data flow called **AR Data** will connect this entity to the C^3 system. Inside the C^3 system, **AR Data** will connect to the **Analyze Sales History** process.

Also, another new two-way data flow called **Billing Data** will connect to the **Analyze Sales History** process and the **Customer Profile** data store.

Challenge Tasks

A. Draw a context diagram for the new C^3 system that shows the revised design.
B. Draw a DFD diagram 0 that shows the revised design.

CHAPTER 6 Object Modeling

Chapter 6 is the third of four chapters in the systems analysis phase of the SDLC. This chapter discusses object modeling techniques that analysts use to create a logical model. In addition to structured analysis, object-oriented analysis is another way to represent and design an information system.

INTRODUCTION

OBJECTIVES

When you finish this chapter, you will be able to:

- Explain how object-oriented analysis can be used to describe an information system
- Define object modeling terms and concepts, including objects, attributes, methods, messages, classes, and instances
- Explain relationships among objects and the concept of inheritance
- Draw an object relationship diagram
- Describe Unified Modeling Language (UML) tools and techniques, including use cases, use case diagrams, class diagrams, sequence diagrams, state transition diagrams, and activity diagrams
- Explain the advantages of using CASE tools in developing the object model
- Explain how to organize an object model

In Chapter 5, you learned how to use structured analysis techniques to develop a data and process model of the proposed system. Now, in Chapter 6, you learn about object-oriented analysis, which is another way to view and model system requirements. In this chapter, you use object-oriented techniques to document, analyze, and model the information system. In Chapter 7, which concludes the systems analysis phase, you will evaluate alternatives, develop the system requirements document, learn about prototyping, and prepare for the systems design phase of the SDLC.

Chapter Introduction Case: Mountain View College Bookstore

Background: Wendy Lee, manager of college services at Mountain View College, wants a new information system that will improve efficiency and customer service at the three college bookstores.

In this part of the case, Tina Allen (systems analyst) and David Conroe (student intern) are talking about object-oriented concepts, tools, and techniques.

Participants: Tina and David
Location: Mountain View College Cafeteria, Wednesday afternoon, November 2, 2011.
Project status: Tina and David have completed data and processing modeling, and are discussing object-oriented techniques that they can use to develop an object model of the new system.
Discussion topics: Object-oriented concepts, tools, and techniques

Tina: Hi, David. I want to chat with you about object-oriented analysis before we finish the systems analysis phase. Would this be a good time to talk?

David: *Sure. I know that object-oriented analysis is another way of viewing the system, but I don't know much about it.*

Tina: Well, object-oriented analysis describes an information system by identifying things called objects. An object represents a real person, place, event, or transaction. For example, in the bookstore, when a student purchases a textbook, the student is an object, the textbook is an object, and the purchase transaction itself is an object.

David: *That sounds a little like the entities we identify in structured analysis.*

Tina: Yes, but there's a major difference! In structured analysis we treat data and the processes that affect the data separately. Objects, on the other hand, contain the data *and* the processes, called methods, that can add, modify, or change the data. To make it even more interesting, one object can send a message to another object to request some action or response. For example, a driver object adjusts the cruise control, which sends one or more messages to the car object telling it to maintain a steady speed.

David: *I get it. So objects can be people, things, or events?*

Tina: Yes. To show how the system works, we use a special modeling language called the UML. We even show human actors as stick figures that interact with business functions called use cases.

David: *I'd like to give it a try.*

Tina: No problem. Although we'll still use structured analysis, it will be interesting to model the system in object-oriented terms. Let's get started on our task list:

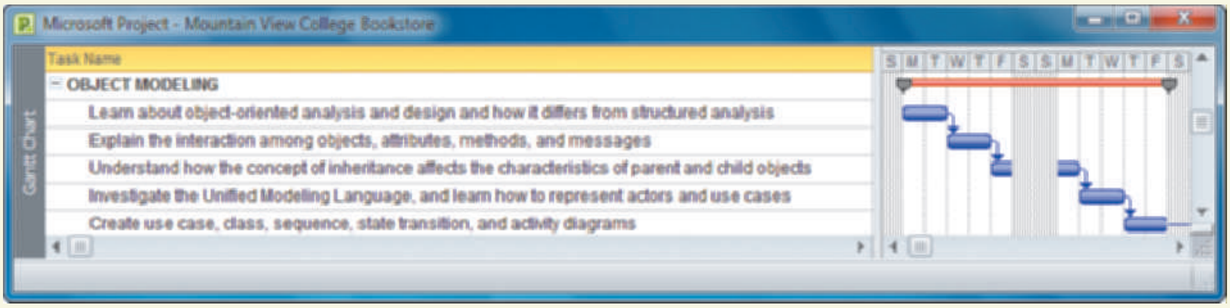


FIGURE 6-1 Typical object modeling task list.

ON THE WEB

To learn more about object-oriented analysis, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Object-Oriented Analysis link.

OVERVIEW OF OBJECT-ORIENTED ANALYSIS

As you learned in Chapter 1, the most popular systems development options are structured analysis, object-oriented analysis (O-O), and agile methods, also called adaptive methods. The table in Figure 1-25 on page 21 shows the three alternatives and describes some pros and cons for each approach. As the table indicates, O-O methodology is popular because it integrates easily with object-oriented programming languages such as Java, Smalltalk, C++, and Perl. Programmers also like O-O code because it is modular, reusable, and easy to maintain.

Object-oriented (**O-O**) **analysis** describes an information system by identifying things called objects. An **object** represents a real person, place, event, or transaction. For example, when a patient makes an appointment to see a doctor, the patient is an object, the doctor is an object, and the appointment itself is an object.

Object-oriented analysis is a popular approach that sees a system from the viewpoint of the objects themselves as they function and interact. The end product of object-oriented analysis is an **object model**, which represents the information system in terms of objects and object-oriented concepts.

VIDEO LEARNING SESSION: OBJECT-ORIENTED TERMS AND CONCEPTS

Video Learning Sessions can help you understand key concepts, practice your skills, and check your work. To access the sessions, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and navigate to the **Video Learning Sessions** for this book. In this session you'll learn basic object modeling terms and concepts, how to use symbols to create object models, and how you can use a CASE tool to create object models.

ON THE WEB

To learn more about the Unified Modeling Language, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Unified Modeling Language link.

Object-Oriented Terms and Concepts

In Chapter 4, you learned that the **Unified Modeling Language** (**UML**) is a widely used method of visualizing and documenting an information system. In this chapter, you use the UML to develop object models. Your first step is to understand basic object-oriented terms, including objects, attributes, methods, messages, classes, and instances. In this chapter, you will learn how systems analysts use those terms to describe an information system.

An object represents a person, place, event, or transaction that is significant to the information system. In Chapter 5, you created DFDs that treated data and processes separately. An object, however, includes data *and* the processes that affect that data. For example, a customer object has a name, an address, an account number, and a current balance. Customer objects also can perform specific tasks, such as placing an order, paying a bill, and changing their address.

An object has certain **attributes**, which are characteristics that describe the object. For example, if you own a car, it has attributes such as make, model, and color. An object also has **methods**, which are tasks or functions that the object performs when it receives a **message**, or command, to do so. For example, your car performs a method called OPERATE WIPERS when you send a message by moving the proper control. Figure 6-2 shows examples of attributes, methods, and messages for a car object.

A **class** is a group of similar objects. For example, Ford Fiestas belong to a class called CAR. An **instance** is a specific member of a class. *Your* Ford Fiesta, therefore, is an instance of the CAR class. At an auto dealership, like the one shown in Figure 6-3, you might observe many instances of the CAR class, the TRUCK class, the MINIVAN class, and the SPORT UTILITY VEHICLE class, among others. Although the term "object" usually refers to a particular instance, systems analysts sometimes use the term to refer to a class of objects. Usually the meaning is understood from the context and the way the term is used.

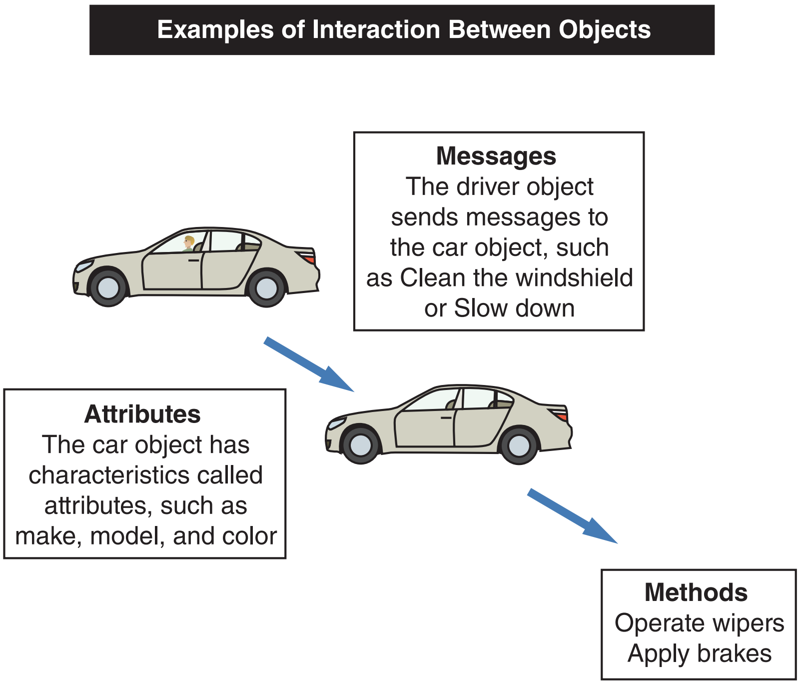


FIGURE 6-2 Objects have attributes, can send and receive messages, and perform actions called methods.

Objects

Consider how the UML describes a family with parents and children. The UML represents an object as a rectangle with the object name at the top, followed by the object's attributes and methods.

Figure 6-4 shows a PARENT object with certain attributes such as name, age, sex, and hair color. If there are two parents, then there are two instances of the PARENT object. The PARENT object can perform methods, such as reading a bedtime story, driving the car pool van, or preparing a school lunch. When a PARENT object receives a message, it performs an action, or method.

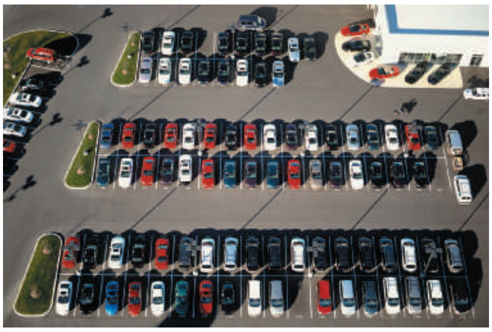


FIGURE 6-3 At an auto dealership, you can observe the CAR class, the TRUCK class, the MINIVAN class, and the SPORT UTILITY VEHICLE class.

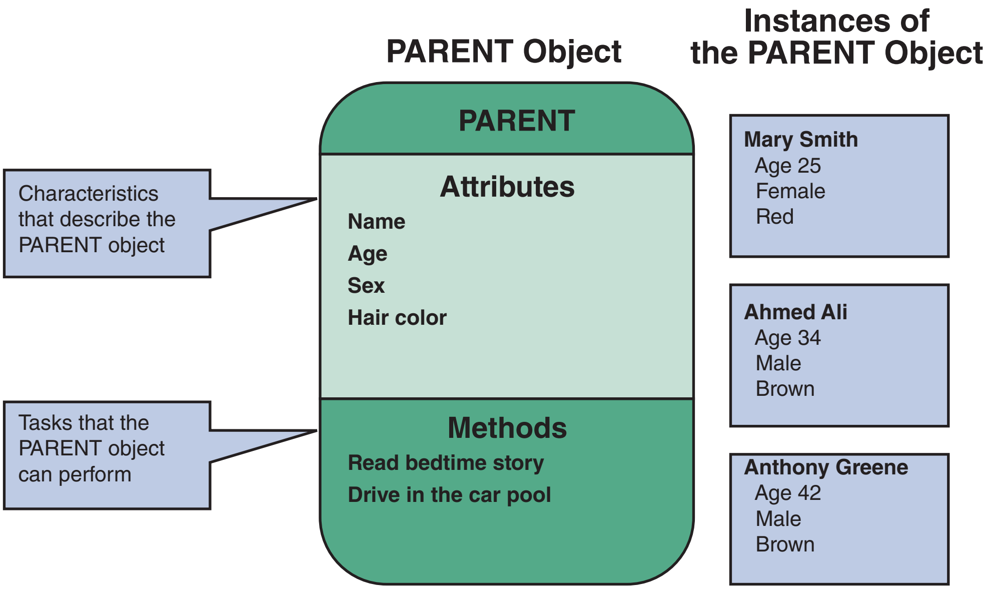


FIGURE 6-4 The PARENT object includes four attributes and two methods. Mary Smith, Ahmed Ali, and Anthony Greene are instances of the PARENT object.

For example, the message GOOD NIGHT from a child might tell the PARENT object to read a bedtime story, while the message DRIVE from another parent signals that it is the PARENT object's turn to drive in the car pool.

Continuing with the family example, the CHILD object in Figure 6-5 possesses the same attributes as the PARENT object and an additional attribute that shows the number of siblings. A CHILD object performs certain methods, such as picking up toys, eating dinner, playing, cooperating, and getting ready for bed. To signal the CHILD object to perform those tasks, a parent can send certain messages that the CHILD object will understand. For example, the DINNER'S READY message tells a CHILD object to come to the table, while the SHARE WITH YOUR BROTHER/SISTER message tells a CHILD object to cooperate with other CHILD objects.

The family also might have a DOG object, as shown in Figure 6-6. That object can have attributes such as name, breed, age, color, sex, and favorite food. The DOG object can perform methods such as wagging its tail, eating, fetching, and sleeping. The message GOOD DOG, when directed to the DOG object, signals it to wag its tail. Similarly, the DINNER'S READY message signals the DOG object to run to its food bowl.

Now consider an example of a fitness center, as shown in Figure 6-7, and the objects that interact with the fitness center's enrollment system. A typical fitness center might have students, instructors, fitness-class schedules, and a registration process.

STUDENT and INSTRUCTOR objects are shown in Figure 6-8. Each STUDENT object has the following attributes: student number, name, address, telephone, date of birth, fitness record, and status. In addition, a STUDENT can add a fitness-class; drop a fitness-class; change an address, telephone, or status; and update his or her fitness record.

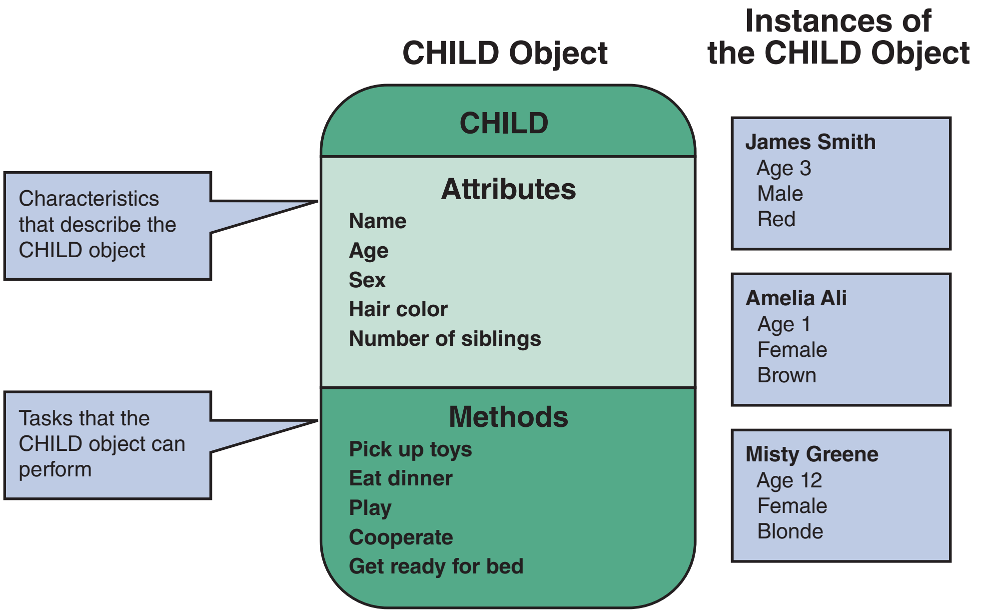


FIGURE 6-5 The CHILD object includes five attributes and five methods. James Smith, Amelia Ali, and Misty Greene are instances of the CHILD object.

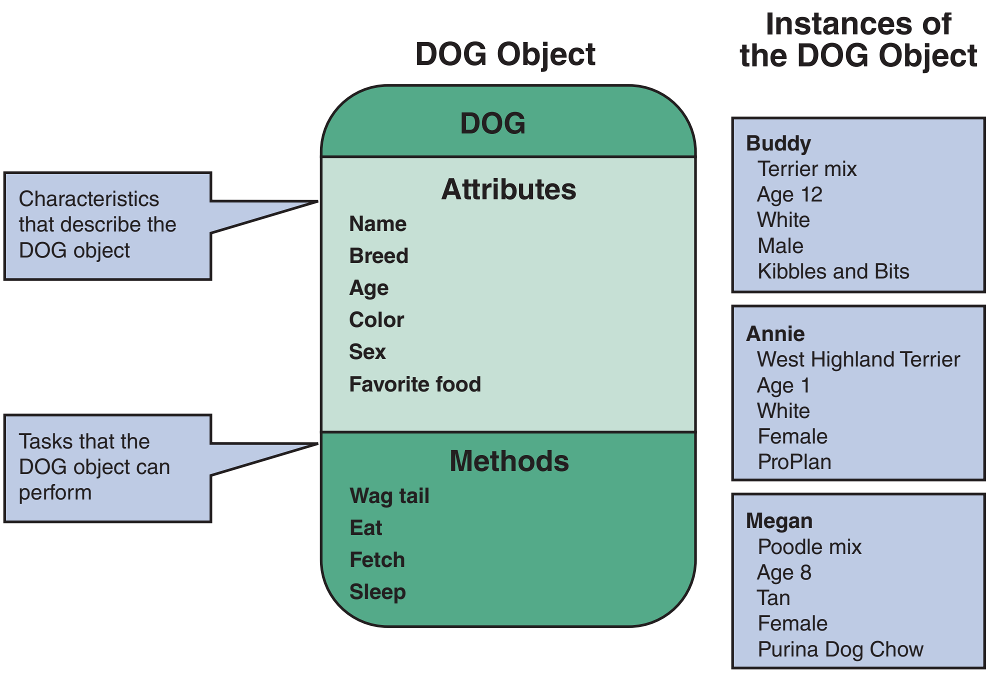


FIGURE 6-6 The DOG object includes six attributes and four methods. Buddy, Annie, and Megan are instances of the DOG object.

FIGURE 6-7 A typical fitness center might have students, instructors, fitness-class schedules, and a registration process.

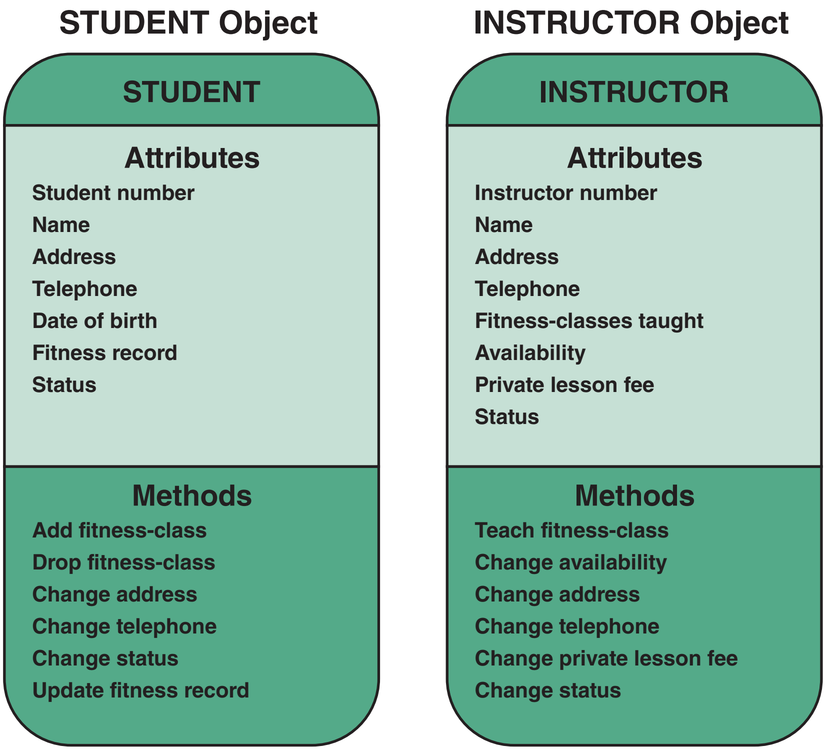


FIGURE 6-8 The STUDENT object includes seven attributes and six methods. The INSTRUCTOR object includes eight attributes and six methods.

The INSTRUCTOR object in Figure 6-8 has the following attributes: instructor number, name, address, telephone, fitness-classes taught, availability, private lesson fee, and status. An INSTRUCTOR object can teach a fitness-class, and change his or her availability, address, telephone, private lesson fee, or status.

The FITNESS-CLASS SCHEDULE object shown in Figure 6-9 on the next page includes data about fitness classes, including fitness-class number, date, time, type, location, instructor number, and maximum enrollment. The FITNESS-CLASS SCHEDULE object includes the methods that can add or delete a fitness class, or change a fitness-class date, time, instructor, location, or enrollment.

The REGISTRATION RECORD object shown in Figure 6-10 on the next page includes the student number, fitness-class number, registration date, fee, and status. The REGISTRATION RECORD object includes methods to add a REGISTRATION instance when a student enrolls, or drop a REGISTRATION instance if the fitness class is canceled or for nonpayment. Notice that if a student registers for three fitness classes, the result is three instances of the REGISTRATION RECORD object. The REGISTRATION RECORD object also includes a method of notifying students and instructors of information.

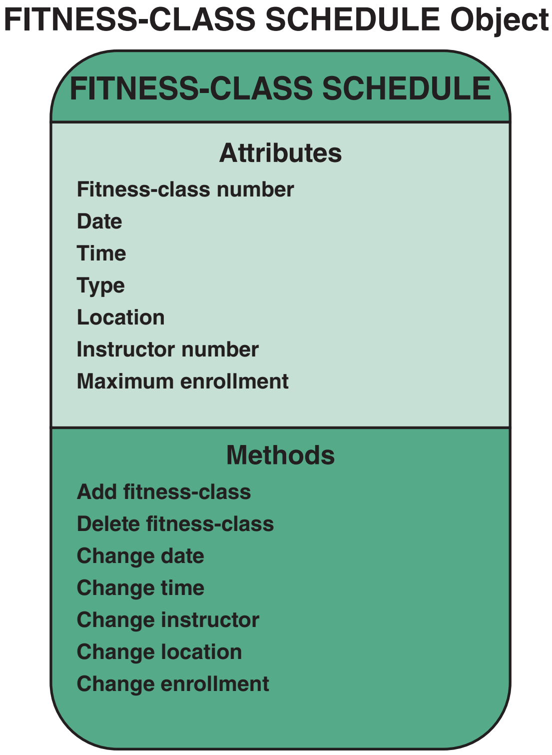


FIGURE 6-9 The FITNESS-CLASS SCHEDULE object includes seven attributes and seven methods.

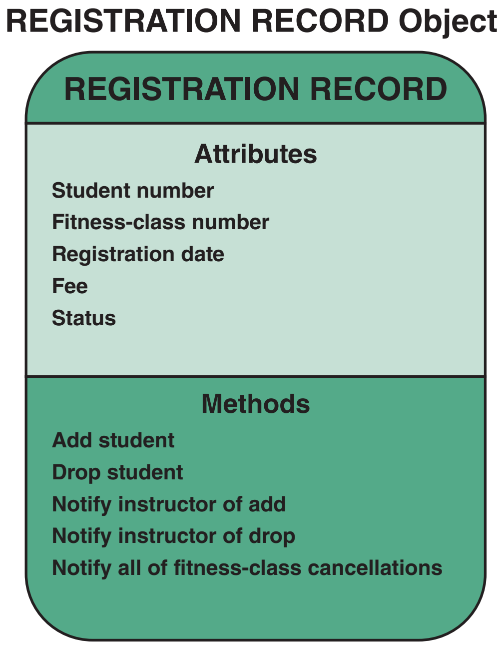


FIGURE 6-10 The REGISTRATION object includes five attributes and five methods.

Attributes

If objects are similar to nouns, attributes are similar to adjectives that describe the characteristics of an object. How many attributes are needed? The answer depends on the business requirements of the information system and its users. Even a relatively simple object, such as an inventory item, might have a part number, description, supplier, quantity on hand, minimum stock level, maximum stock level, reorder time, and so on. Some objects might have a few attributes; others might have dozens.

Systems analysts define an object's attributes during the systems design process. In an object-oriented system, objects can inherit, or acquire, certain attributes from other objects. When you learn about relationships between objects and classes, you will understand how that occurs.

Objects can have a specific attribute called a **state**. The state of an object is an adjective that describes the object's current status. For example, depending on the state, a student can be a future student, a current student, or a past student. Similarly, a bank account can be active, inactive, closed, or frozen.

FIGURE 6-11 In a fast-food restaurant, preparing more fries is a common task.

Methods

A method defines specific tasks that an object can perform. Just as objects are similar to nouns and attributes are similar to adjectives, methods resemble verbs that describe *what* and *how* an object does something.

Consider a server who prepares fries in a fast-food restaurant, as shown in Figure 6-11. A systems analyst might describe the operation as a method called MORE FRIES, as shown in

Method: MORE FRIES

Steps:
1. Heat oil
2. Fill fry basket with frozen potato strips
3. Lower basket into hot oil
4. Check for readiness
5. When ready raise basket and let drain
6. Pour fries into warming tray
7. Add salt

FIGURE 6-12 The MORE FRIES method requires the server to perform seven specific steps.

Method: ADD STUDENT

Steps:
1. Add a new student instance
2. Record student number
3. Record student name
4. Record student address
5. Record student telephone number
6. Record student date of birth
7. Record sex of student
8. Record state of student
9. Save new student data

FIGURE 6-13 In the fitness center example, the ADD STUDENT method requires the STUDENT object to perform nine specific steps.

Figure 6-12. The MORE FRIES method includes the steps required to heat the oil, fill the fry basket with frozen potato strips, lower it into the hot oil, check for readiness, remove the basket when ready and drain the oil, pour the fries into a warming tray, and add salt.

Figure 6-13 shows another example of a method. At the fitness center, an ADD STUDENT method adds a new instance of the STUDENT class. Notice that nine steps are required to add the new instance and record the necessary data.

Messages

A message is a command that tells an object to perform a certain method. For example, the message ADD STUDENT directs the STUDENT class to add a STUDENT instance. The STUDENT class understands that it should add the student number, name, and other data about that student, as shown in Figure 6-14. Similarly, a message named DELETE STUDENT tells the STUDENT class to delete a STUDENT instance.

The same message to two different objects can produce different results. The concept that a message gives different meanings to different objects is called **polymorphism**. For example, in Figure 6-15, the message GOOD NIGHT signals the PARENT object to

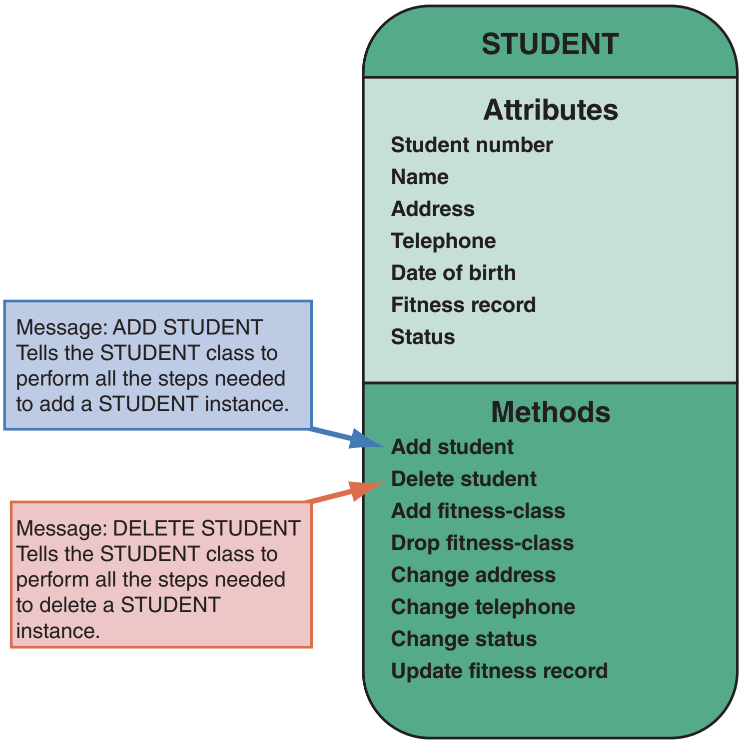


FIGURE 6-14 The message ADD STUDENT signals the STUDENT class to perform the ADD STUDENT method. The message DELETE STUDENT signals the STUDENT class to perform the DELETE STUDENT method.

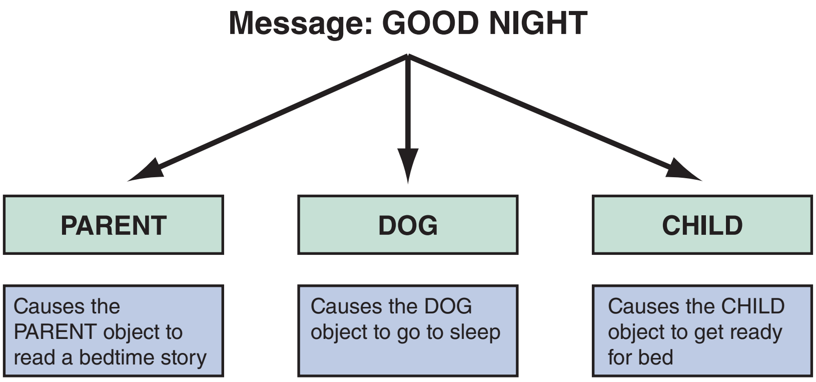


FIGURE 6-15 In an example of polymorphism, the message GOOD NIGHT produces different results, depending on which object receives it.

ON THE WEB

To learn more about polymorphism, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Polymorphism link.

read a bedtime story, but the same message to the DOG object tells the dog to sleep. The GOOD NIGHT message to the CHILD object signals it to get ready for bed.

You can view an object as a **black box**, because a message to the object triggers changes within the object without specifying how the changes must be carried out. A gas pump is an example of a black box. When you select the economy grade at a pump, you do not need to think about how the pump determines the correct price and selects the right fuel, as long as it does so properly.

The black box concept is an example of **encapsulation**, which means that all data and methods are self-contained. A black box does not want or need outside interference. By limiting access to internal processes, an object prevents its internal code from being altered by another object or process. Encapsulation allows objects to be used as modular components anywhere in the system, because objects send and receive messages but do not alter the internal methods of other objects.

FIGURE 6-16 In a school information system, an INSTRUCTOR object sends an ENTER GRADE message to an instance of the STUDENT RECORD class.

Object-oriented designs typically are implemented with object-oriented programming languages. A major advantage of O-O designs is that systems analysts can save time and avoid errors by using modular objects, and programmers can translate the designs into code, working with reusable program modules that have been tested and verified. For example, in Figure 6-16, an INSTRUCTOR object sends an ENTER GRADE message to an instance of the STUDENT RECORD class. Notice that the INSTRUCTOR object and STUDENT RECORD class could be reused, with minor modifications, in other school information systems where many of the attributes and methods would be similar.

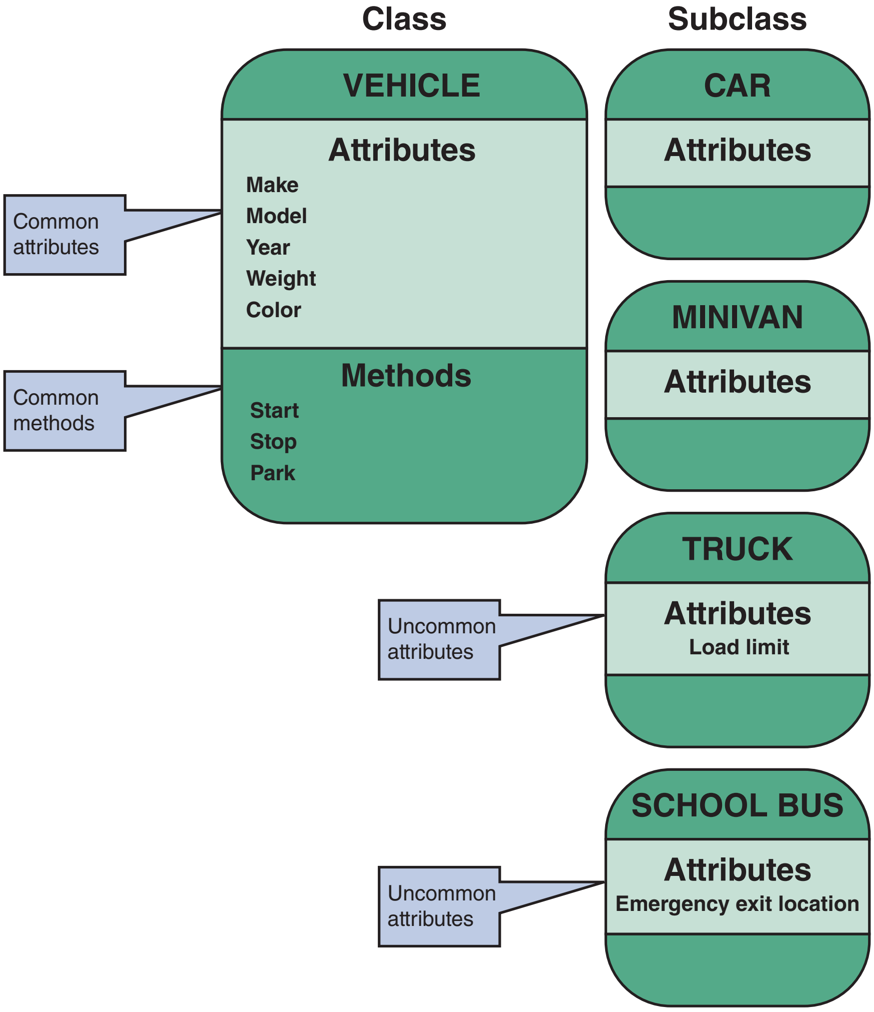


FIGURE 6-17 The VEHICLE class includes common attributes and methods. CAR, TRUCK, MINIVAN, and SCHOOL BUS are instances of the VEHICLE class.

Classes

An object belongs to a group or category called a class. All objects within a class share common attributes and methods, so a class is like a blueprint, or template for all the objects within the class. Objects within a class can be grouped into **subclasses**, which are more specific categories within a class. For example, TRUCK objects represent a subclass within the VEHICLE class, along with other subclasses called CAR, MINIVAN, and SCHOOL BUS, as shown in Figure 6-17. Notice that all four subclasses share common traits of the VEHICLE class, such as make, model, year, weight, and color. Each subclass also can possess traits that are uncommon, such as a load limit for the TRUCK or an emergency exit location for the SCHOOL BUS.

In the fitness center example shown in Figure 6-18, INSTRUCTOR objects represent a subclass within the EMPLOYEE class. The EMPLOYEE class also can contain MANAGER and OFFICE STAFF subclasses, because a manager and staff members are employees. All INSTRUCTOR, MANAGER, and OFFICE STAFF objects contain similar information (such as employee name, title, and pay rate) and perform similar tasks (such as getting hired and changing an address or telephone number).

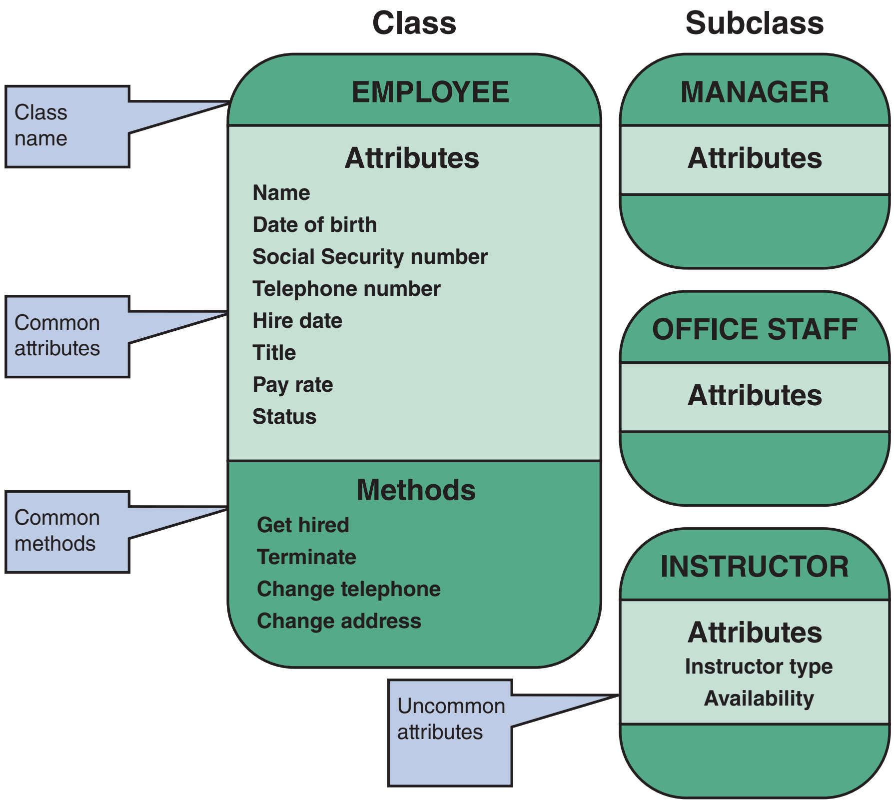


FIGURE 6-18 The fitness center EMPLOYEE class includes common attributes and methods. INSTRUCTOR, MANAGER, and OFFICE STAFF are subclasses within the EMPLOYEE class.

A class can belong to a more general category called a **superclass**. For example, a NOVEL class belongs to a superclass called BOOK, because all novels are books. The NOVEL class can have subclasses called HARDCOVER and PAPERBACK. Similarly, as shown in Figure 6-19, the EMPLOYEE class belongs to the PERSON superclass, because every employee is a person, and the INSTRUCTOR class is a subclass of EMPLOYEE.

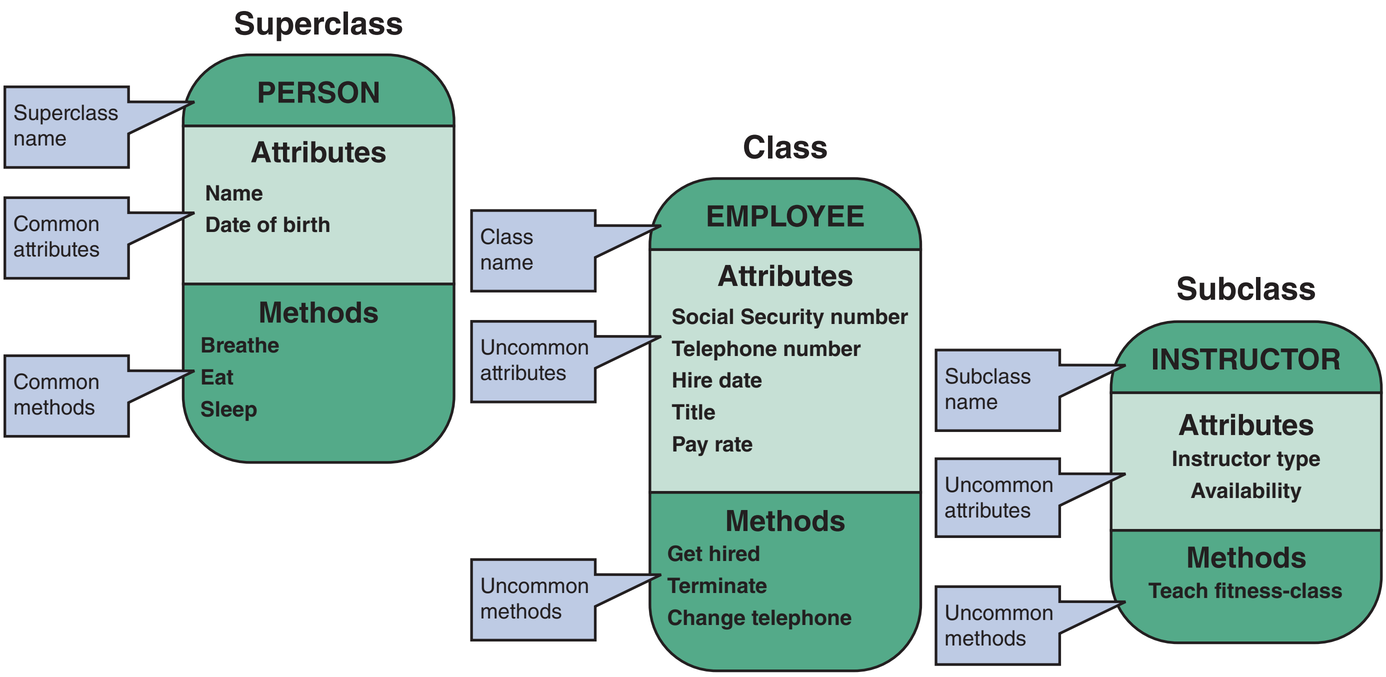


FIGURE 6-19 At the fitness center, the PERSON superclass includes common attributes and methods. EMPLOYEE is a class within the PERSON superclass. INSTRUCTOR is a subclass within the EMPLOYEE class.

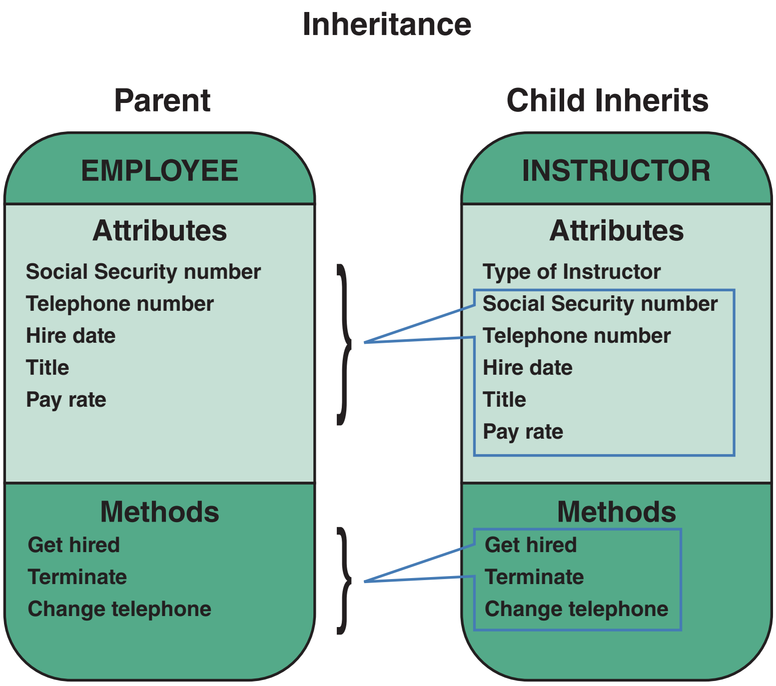


FIGURE 6-20 An inheritance relationship exists between the INSTRUCTOR and EMPLOYEE objects. The INSTRUCTOR (child) object inherits characteristics from the EMPLOYEE (parent) class and can have additional attributes of its own.

Relationships Among Objects and Classes

Relationships enable objects to communicate and interact as they perform business functions and transactions required by the system. Relationships describe what objects need to know about each other, how objects respond to changes in other objects, and the effects of membership in classes, superclasses, and subclasses. Some relationships are stronger than others (just as a relationship between family members is stronger than one between casual acquaintances). The strongest relationship is called inheritance. **Inheritance** enables an object, called a **child**, to derive one or more of its attributes from another object, called a **parent**. In the example in Figure 6-20, the INSTRUCTOR object (child) inherits many traits from the EMPLOYEE object (parent), including SOCIAL SECURITY NUMBER, TELEPHONE NUMBER, and HIRE DATE. The INSTRUCTOR object also can possess additional attributes, such as TYPE OF INSTRUCTOR. Because all employees share certain attributes, those attributes are assumed through inheritance and do not need to be repeated in the INSTRUCTOR object.

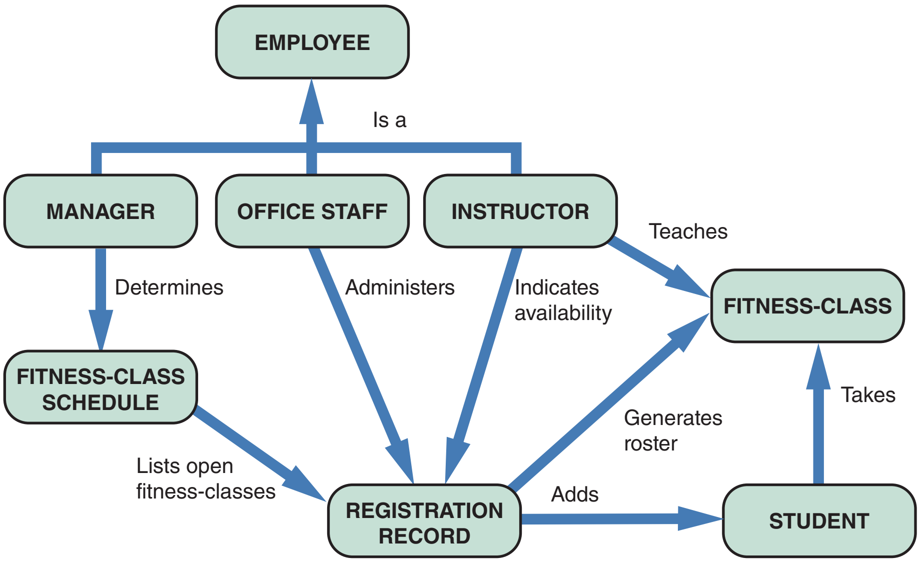


FIGURE 6-21 Object relationship diagram for the fitness center.

Object Relationship Diagram

After you identify the objects, classes, and relationships, you are ready to prepare an object relationship diagram that will provide an overview of the system. You will use that model as a guide as you continue to develop additional diagrams and documentation. Figure 6-21 shows an object relationship diagram for a fitness center. Notice that the model shows the objects and how they interact to perform business functions and transactions.

Object Modeling with the Unified Modeling Language

Just as structured analysis uses DFDs to model data and processes, systems analysts use the Unified Modeling Language (UML) to describe object-oriented systems.

In Chapter 4, you learned that the UML is a popular technique for documenting and modeling a system. The UML uses a set of symbols to represent graphically the various components and relationships within a system. Although the UML can be used for business process modeling and requirements modeling, it mainly is used to support object-oriented system analysis and to develop object models.

VIDEO LEARNING SESSION: Object-Oriented Diagrams and Models

Video Learning Sessions can help you understand key concepts, practice your skills, and check your work. To access the sessions, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and navigate to the **Video Learning Sessions** for this book. In this session you'll learn how to create various types of object diagrams, including use-case, class, sequence, state-transition, and activity diagrams, and how you can use a CASE tool to create object models.

Use Case Modeling

A **use case** represents the steps in a specific business function or process. An external entity, called an **actor**, initiates a use case by requesting the system to perform a function or process. For example, in a medical office system, a PATIENT (actor) can MAKE APPOINTMENT (use case), as shown in Figure 6-22.

Notice that the UML symbol for a use case is an oval with a label that describes the action or event. The actor is shown as a stick figure, with a label that identifies the actor's role. The line from the actor to the use case is called an association, because it links a particular actor to a use case. Figure 6-23 shows use case examples of a passenger making an airline reservation, a customer placing an order, and a bus dispatcher changing a student's pickup address.

Use cases also can interact with other use cases. When the outcome of one use case is incorporated by another use case, we say that the second case *uses* the first case. The UML

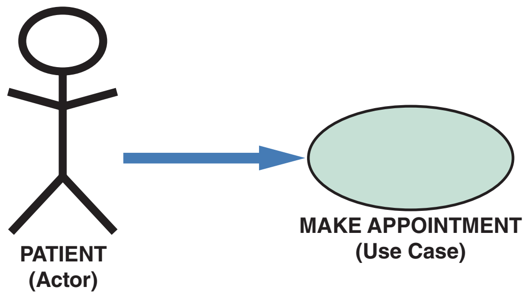


FIGURE 6-22 In a medical office system, a PATIENT (actor) can MAKE APPOINTMENT (use case).

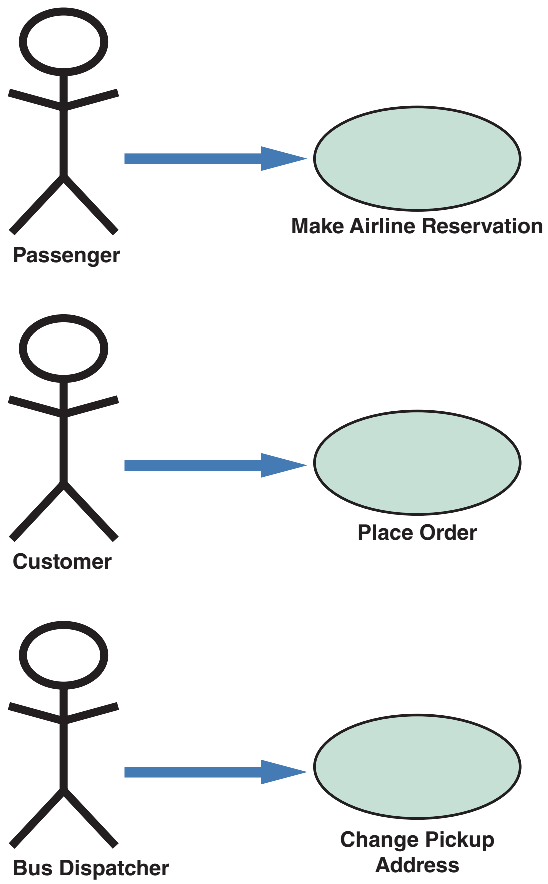


FIGURE 6-23 Three use case examples. The UML symbol for a use case is an oval. The actor is shown as a stick figure.

indicates the relationship with a hollow-headed arrow that *points at* the use case being used. Figure 6-24 shows an example where a student adds a fitness class and PRODUCE FITNESS-CLASS ROSTER *uses* the results of ADD FITNESS-CLASS to generate a new fitness-class roster. Similarly, if an instructor changes his or her availability, UPDATE INSTRUCTOR INFORMATION *uses* the CHANGE AVAILABILITY use case to update the instructor's information.

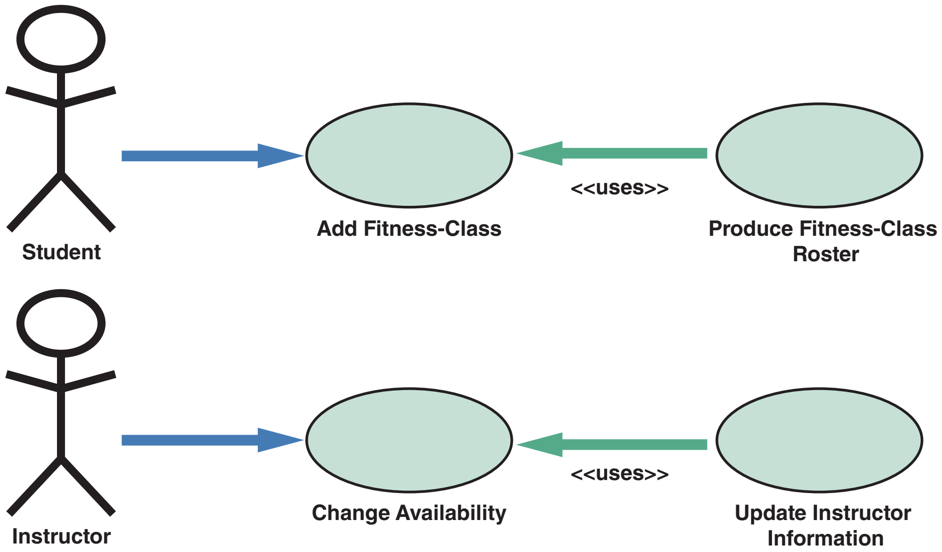


FIGURE 6-24 When a student adds a class, PRODUCE FITNESS-CLASS ROSTER uses the results of ADD CLASS to generate a new class roster. When an instructor changes his or her availability, UPDATE INSTRUCTOR INFORMATION uses the CHANGE AVAILABILITY use case to update the instructor's information.

To create use cases, you start by reviewing the information that you gathered during the requirements modeling phase. Your objective is to identify the actors and the functions or transactions they initiate. For each use case, you also develop a **use case description** in the form of a table. A use case description documents the name of the use case, the actor, a description of the use case, a step-by-step list of the tasks and actions required for successful completion, a description of alternative courses of action, preconditions, postconditions, and assumptions. Figure 6-25 shows an example of the ADD NEW STUDENT use case.

ON THE WEB

To learn more about use case modeling, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Use Case Modeling link.

ADD NEW STUDENT Use Case — Add New Student

Name:	Add New Student
Actor:	Student/Manager
Description:	Describes the process used to add a student to a fitness-class
Successful completion:	1. Manager checks FITNESS-CLASS SCHEDULE object for availability 2. Manager notifies student 3. Fitness-class is open and student pays fee 4. Manager registers student
Alternative:	1. Manager checks FITNESS-CLASS SCHEDULE object for availability 2. Fitness-class is full 3. Manager notifies student
Precondition:	Student requests fitness-class
Postcondition:	Student is enrolled in fitness-class and fees have been paid
Assumptions:	None

FIGURE 6-25 The ADD NEW STUDENT use case description documents the process used to add a current student into an existing class.

When you identify use cases, try to group all the related transactions into a single use case. For example, when a hotel customer reserves a room, the reservation system blocks a room, updates the occupancy forecast, and sends the customer a confirmation. Those events are all part of a single use case called RESERVE ROOM, and the specific actions are step-by-step tasks within the use case.

CASE IN POINT 6.1: Hilltop Motors

You have been hired by Hilltop Motors as a consultant to help the company plan a new information system. Hilltop is an old-line dealership, and the prior owner was slow to change. A new management team has taken over, and they are eager to develop a first-class system. Right now, you are reviewing the service department, which is going though a major expansion. You decide to create a model of the service department in the form of a use case diagram. The main actors in the service operation are customers, service writers who prepare work orders and invoices, and mechanics who perform the work. You are meeting with the management team tomorrow morning. Create an initial draft of the diagram to present to them at that time.

Use Case Diagrams

A **use case diagram** is a visual summary of several related use cases within a system or subsystem. Consider a typical auto service department, as shown in Figure 6-26. The service department involves customers, service writers who prepare work orders and invoices, and mechanics who perform the work. Figure 6-27 on the next page shows a possible use case diagram for the auto service department.

ON THE WEB

To learn more about use case diagrams, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Use Case Diagrams link.

FIGURE 6-26 A typical auto service department might involve customers, service writers who prepare work orders, and mechanics who perform the work.

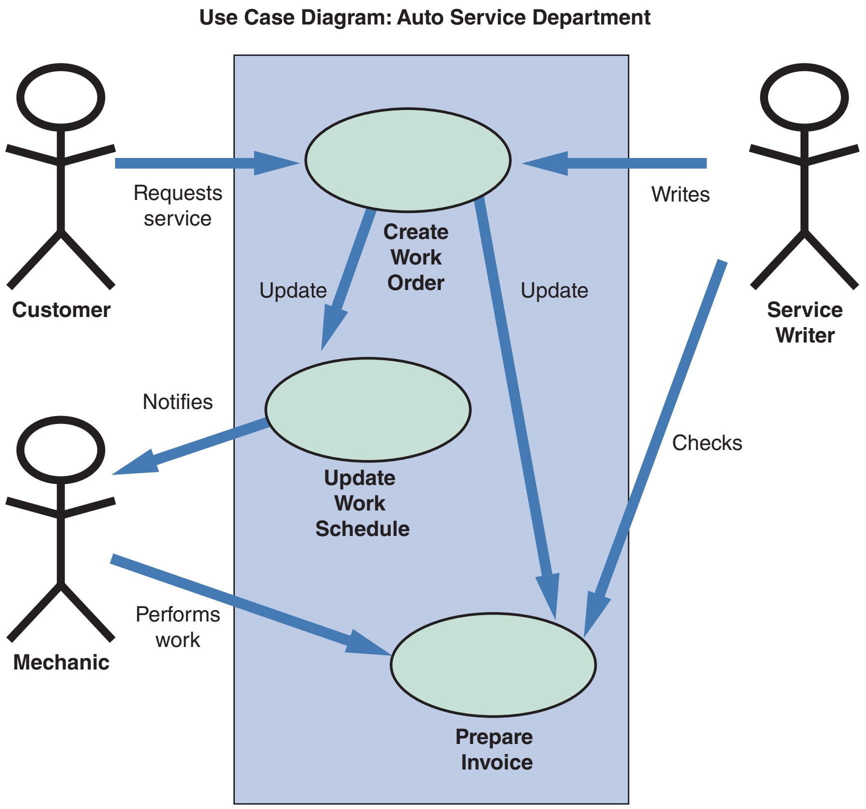


FIGURE 6-27 A use case diagram to handle work at an auto service department.

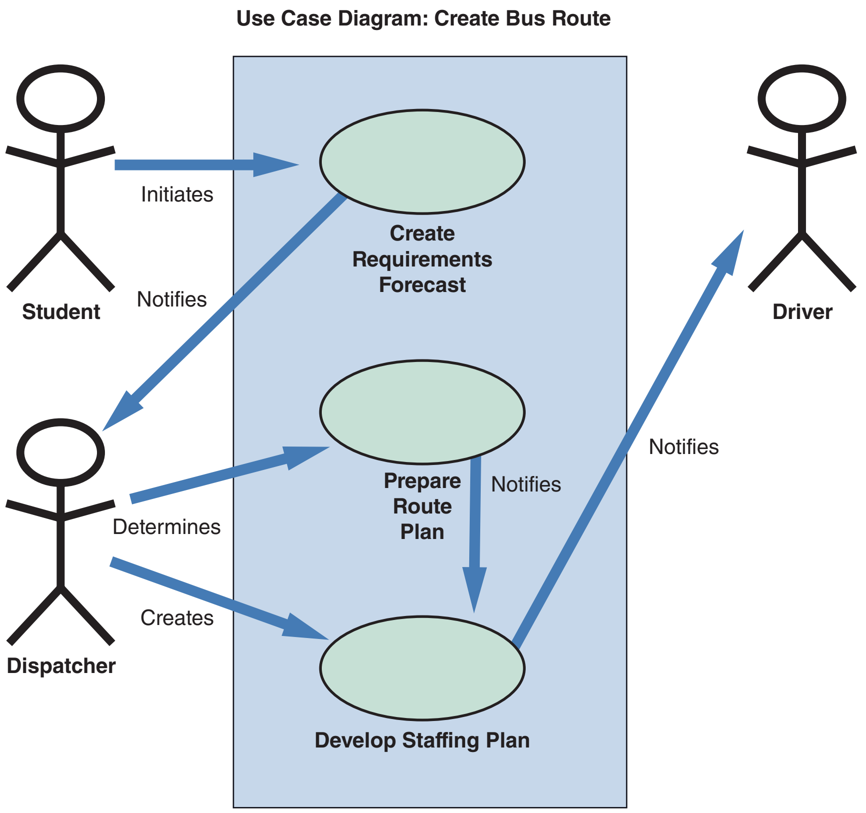


FIGURE 6-28 A use case diagram to create a school bus route.

When you create a use case diagram, the first step is to identify the system boundary, which is represented by a rectangle. The **system boundary** shows what is included in the system (inside the rectangle) and what is not included in the system (outside the rectangle). After you identify the system boundary, you place the use cases on the diagram, add the actors, and show the relationships. Figure 6-28 shows a use case diagram for a school bus system that creates a new bus route.

Class Diagrams

A **class diagram** shows the object classes and relationships involved in a use case. Like a DFD, a class diagram is a logical model, which evolves into a physical model and finally becomes a functioning information system. In structured analysis, entities, data stores, and processes are transformed into data structures and program code. Similarly, class diagrams evolve into code modules, data objects, and other system components.

In a class diagram, each class appears as a rectangle, with the class name at the top, followed by the class's attributes and methods. Lines show relationships between classes and have labels identifying the action that relates the two classes. To create a class diagram, you review the use case and identify the classes that participate in the underlying business process.

The class diagram also includes a concept called **cardinality**, which describes how instances of one class relate to instances of another class. For example, an employee might have earned no vacation days or one vacation day or many vacation days. Similarly, an employee might have no spouse or one spouse. Figure 6-29 shows various UML notations and cardinality examples. Notice that in Figure 6-29, the first column shows a UML notation symbol that identifies the relationship shown in the second column. The third column provides a typical example of the relationship, which is described in the last column. In the first row of the figure, the UML notation *0..** identifies a *zero or many* relation. The example is that an employee can have no payroll deductions or many deductions.

UML Notation	Nature of the Relationship	Example	Description
0..*	Zero or many	Employee (1) — Payroll Deduction (0..*)	An employee can have no payroll deductions or many deductions.
0..1	Zero or one	Employee (1) — Spouse (0..1)	An employee can have no spouse or one spouse.
1	One and only one	Office Manager (1) — Sales Office (1)	An office manager manages one and only one office.
1..*	One or many	Order (1) — Item Ordered (1..*)	One order can include one or many items ordered.

FIGURE 6-29 Examples of UML notations that indicate the nature of the relationship between instances of one class and instances of another class.

You will learn more about cardinality in Chapter 9, which discusses data design.

Figure 6-30 shows a class diagram for a sales order use case. Notice that the sales office has one sales manager who can have anywhere from zero to many sales reps. Each sales rep can have anywhere from zero to many customers, but each customer has only one sales rep.

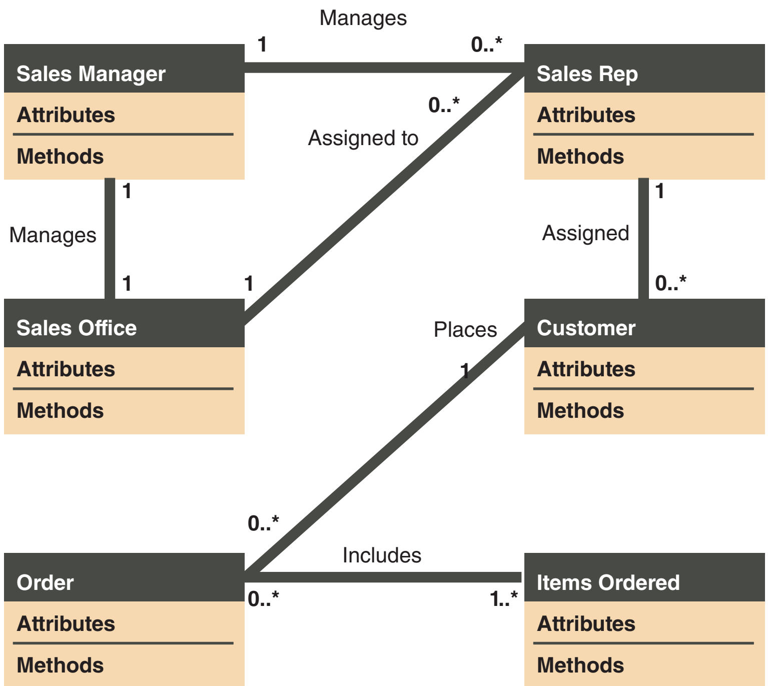


FIGURE 6-30 Class diagram for a sales order use case (attributes and methods omitted for clarity).

CASE IN POINT 6.2: TRAIN THE TRAINER, INC.

Train the Trainer develops seminars and workshops for corporate training managers, who in turn train their employees. Your job at Train the Trainer is to put together the actual training materials. Right now, you are up against a deadline. The new object modeling seminar has a chapter on cardinality, and the client wants you to come up with at least three more examples for each of the four cardinality categories listed in Figure 6-29 on the previous page. The four categories are *zero or many*, *zero or one*, *one and only one*, and *one or many*. Even though you are under pressure, you are determined to use examples that are realistic and familiar to the students. What examples will you submit?

Sequence Diagrams

A **sequence diagram** is a dynamic model of a use case, showing the interaction among classes during a specified time period. A sequence diagram graphically documents the use case by showing the classes, the messages, and the timing of the messages. Sequence diagrams include symbols that represent classes, lifelines, messages, and focuses. These symbols are shown in Figure 6-31.

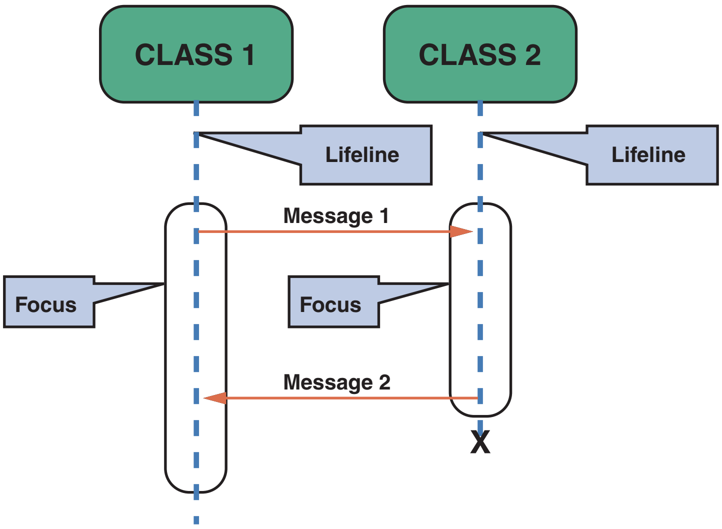


FIGURE 6-31 A sequence diagram with two classes. Notice the *X* that indicates the end of the CLASS 2 lifeline. Also notice that each message is represented by a line with a label that describes the message, and that each class has a focus that shows the period when messages are sent or received.

CLASSES A class is identified by a rectangle with the name inside. Classes that send or receive messages are shown at the top of the sequence diagram.

LIFELINES A lifeline is identified by a dashed line. The **lifeline** represents the time during which the object above it is able to interact with the other objects in the use case. An *X* marks the end of the lifeline.

MESSAGES A message is identified by a line showing direction that runs between two objects. The label shows the name of the message and can include additional information about the contents.

FOCUSES A focus is identified by a narrow vertical shape that covers the lifeline. The **focus** indicates when an object sends or receives a message.

The fitness center example shown in Figure 6-32 shows a sequence diagram for the ADD NEW STUDENT use case. Notice that the vertical position of each message indicates the timing of the message.

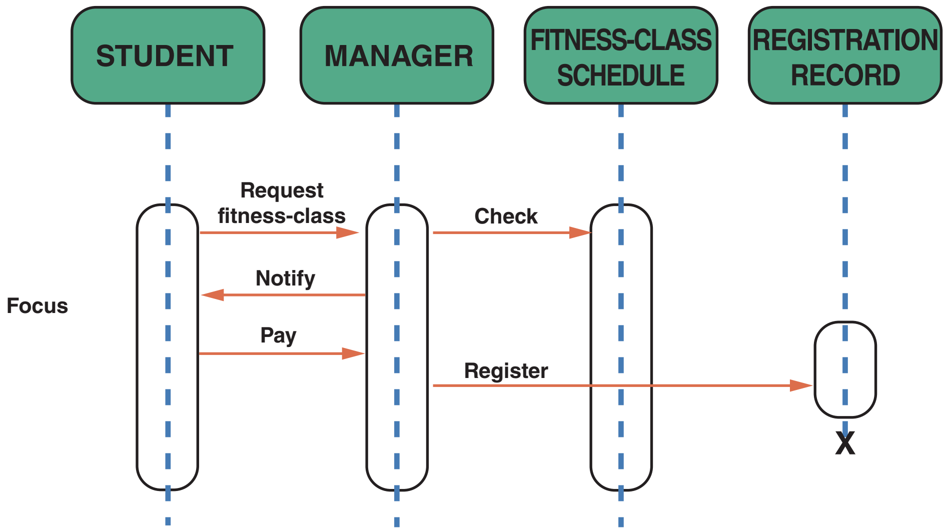


FIGURE 6-32 The sequence diagram for the ADD NEW STUDENT use case. The use case description for ADD NEW STUDENT is shown in Figure 6-25 on page 260.

State Transition Diagrams

Earlier in this chapter you learned that state refers to an object's current status. A **state transition diagram** shows how an object changes from one state to another, depending on events that affect the object. All possible states must be documented in the state transition diagram, as shown in Figure 6-33. A bank account, for example, could be opened as a NEW account, change to an ACTIVE or EXISTING account, and eventually become a CLOSED or FORMER account. Another possible state for a bank account could be FROZEN, if the account's assets are legally attached.

In a state transition diagram, the states appear as rounded rectangles with the state names inside. The small circle to the left is the initial state, or the point where the object first interacts with the system. Reading from left to right, the lines show direction and describe the action or event that causes a transition from one state to another. The circle at the right with a hollow border is the final state.

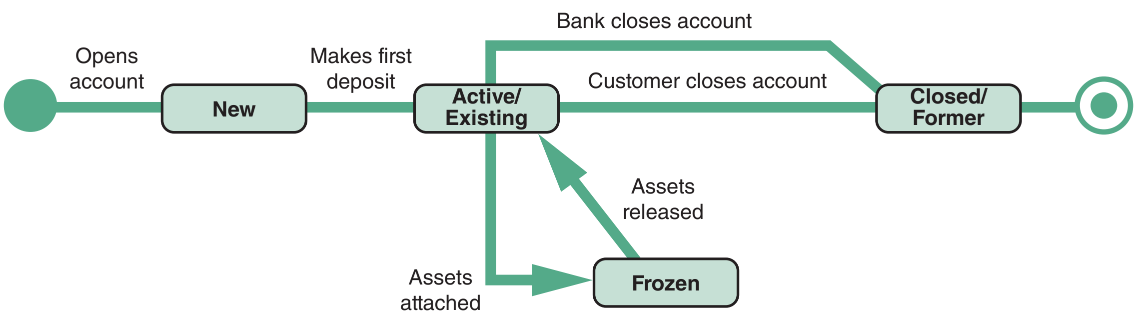


FIGURE 6-33 An example of a state transition diagram for a bank account.

Activity Diagrams

An **activity diagram** resembles a horizontal flowchart that shows the actions and events as they occur. Activity diagrams show the order in which the actions take place and identify the outcomes. Figure 6-34 shows an activity diagram for a cash withdrawal at an ATM machine. Notice that the customer initiates the activity by inserting an ATM card and requesting cash. Activity diagrams also can display multiple use cases in the form of a grid, where classes are shown as vertical bars and actions appear as horizontal arrows.

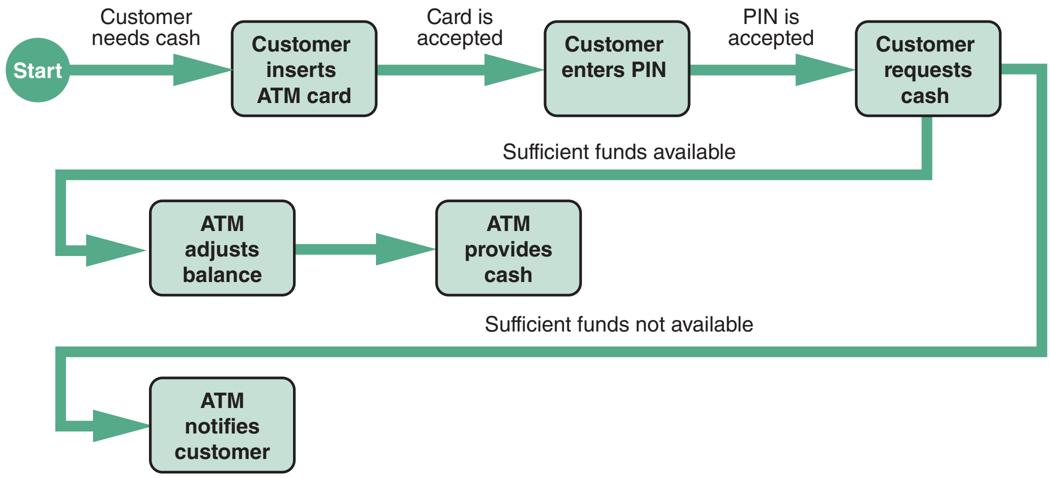


FIGURE 6-34 An activity diagram shows the actions and events involved in withdrawing cash from an ATM machine.

Sequence diagrams, state transition diagrams, and activity diagrams are dynamic modeling tools that can help a systems analyst understand how objects behave and interact with the system.

CASE IN POINT 6.3: TravelBiz

Jack Forester and Lisa Turner are systems analysts in the IT department of TravelBiz, a nationwide travel agency that specializes in business travel. TravelBiz has decided to expand into the vacation travel market by launching a new business division called TravelFun. The IT director assigned Jack and Lisa to create a flexible, efficient information system for the new division. Jack wants to use traditional analysis and modeling techniques for the project. Lisa, on the other hand, wants to use an object-oriented methodology. Which approach would you suggest and why?

TOOLKIT TIME

The CASE tools in Part B of the Systems Analyst's Toolkit can help you develop and maintain complex information systems. To learn more about these tools, turn to Part B of the four-part Toolkit that follows Chapter 12.

CASE Tools

Object modeling requires many types of diagrams to represent the proposed system. Creating the diagrams by hand is time consuming and tedious, so systems analysts rely on CASE tools to speed up the process and provide an overall framework for documenting the system components. In addition, CASE tools ensure consistency and provide common links so that once objects are described and used in one part of the design, they can be reused multiple times without further effort.

ORGANIZING THE OBJECT MODEL

In this chapter, you learned how to use object-oriented tools and techniques to build a logical model of the information system. Now you are ready to organize your diagrams and documentation so the object model is easily read and understood. If you used a CASE tool to develop the design, much of this work will be performed automatically by the CASE software.

There are many ways to proceed with the task of organizing the object model, and experience will be your best teacher. After you identify the system's objects, classes, and relationships, you should develop an object relationship diagram that provides an overview of the system. If you do not use a CASE-generated model, each diagram or object definition should be supported by clear, relevant documentation that can be accessed easily by anyone who reviews the object model. For example, you should organize your use cases and use case diagrams so they can be linked to the appropriate class, state transition, sequence, and activity diagrams. Your diagrams and documentation are the foundation for the system's design, so accuracy is important. Remember that it is much easier to repair a diagram now than to change the software later.

CASE IN POINT 6.4: CYBER ASSOCIATES

One of your responsibilities at Cyber Associates, an IT consulting firm, is to assign new systems analysts to various tasks and projects. Some of the senior people believe that inexperienced analysts should start with object-oriented techniques, which are easier to learn and apply. Others think that an analyst should learn structured analysis first, and then proceed to object-oriented skills. What is your viewpoint, and why?

A QUESTION OF ETHICS

Last month, your company launched a peer review process for IT projects. At the end of each project, team members rate the performance of the overall team, and his or her co-workers individually. The stated goal was to obtain honest, peer-based feedback. Unfortunately, like many good ideas, there was a downside. Although the input is anonymous, the results are submitted to the entire team. Some members, including you, are uncomfortable with the new process because it could encourage cliques and actually undermine a team-based culture. Others see it as an opportunity for honest input.

One team member, who is a close friend of yours, is not very popular with her teammates. To make matters worse, she recently had some personal problems that affected her work, and she is worried that her ratings will be quite negative. She has not specifically asked you about your feedback, but you know she is hoping for a favorable review from you. Even though her work was not great, you don't want to see her get hurt by a process that you yourself are not comfortable with.

Is this a question of ethics versus friendship? Would it be wrong to tilt the scales in her favor just a bit?

Chapter Summary

This chapter introduces object modeling, which is a popular technique that describes a system in terms of objects. Objects represent real people, places, events, and transactions. Unlike structured analysis, which treats data and processes separately, objects include data and processes that can affect the data. During the implementation process, systems analysts and programmers transform objects into program code modules that can be optimized, tested, and reused as often as necessary.

Object-oriented terms include classes, attributes, instances, messages, and methods. Classes include objects that have similar attributes, or characteristics. Individual members of a class are called object instances. Objects within a class can be grouped into subclasses, which are more specific categories within the class. A class also can belong to a more general category called a superclass.

Objects can send messages, or commands, that require other objects to perform certain methods, or tasks. The concept that a message gives different meanings to different objects is called polymorphism. An object resembles a black box, with encapsulated, or self-contained, data and methods. The strongest relationship between objects is inheritance.

After you identify the objects, classes, and relationships, you prepare an object relationship diagram that shows the objects and how they interact to perform business functions and transactions.

The Unified Modeling Language (UML) is a widely used method of visualizing and documenting an information system. UML techniques include use cases, use case diagrams, class diagrams, sequence diagrams, state transition diagrams, and activity diagrams.

A use case describes a business situation initiated by an actor, who interacts with the information system. Each use case represents a specific transaction, or scenario. A use case diagram is a visual summary of related use cases within a system or subsystem. A class diagram represents a detailed view of a single use case, showing the classes that participate in the underlying business transaction, and the relationship among class instances, which is called cardinality. A sequence diagram is a dynamic model of a use case, showing the interaction among classes during a specified time period. Sequence diagrams include lifelines, messages, and focuses. A state transition diagram shows how an object changes from one state to another, depending on events that affect the object. An activity diagram resembles a horizontal flowchart that shows actions and events as they occur in a system.

CASE tools provide an overall framework for system documentation. CASE tools can speed up the development process, ensure consistency, and provide common links that enable objects to be reused.

At the end of the object modeling process, you organize your use cases and use case diagrams and create class, sequence, state transition, and activity diagrams.

Key Terms and Phrases

activity diagram *266*
actor *259*
attributes *250*
black box *256*
cardinality *263*
child *258*
class *251*
class diagram *262*
encapsulation *256*
focus *265*
inheritance *258*
instance *251*
lifeline *264*
message *250*
methods *250*
object *250*
object model *250*
object-oriented (O-O) analysis *250*
parent *258*
polymorphism *255*
relationships *258*
sequence diagram *264*
state *254*
state transition diagram *265*
subclass *256*
superclass *257*
system boundary *262*
Unified Modeling Language (UML) *250*
use case *259*
use case description *260*
use case diagram *261*

Learn It Online

Instructions: To complete the Learn It Online exercises, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to the resources for this chapter, and click the link for the exercise you want to complete.

1 Chapter Reinforcement

TF, MC, and SA

Click one of the Chapter Reinforcement links for Multiple Choice, True/False, or Short Answer. Answer each question and submit to your instructor.

2 Flash Cards

Click the Flash Cards link and read the instructions. Type 20 (or a number specified by your instructor) in the Number of Playing Cards text box, type your name in the Enter your Name text box, and then click the Flip Card button. When the flash card is displayed, read the question and then click the ANSWER box arrow to select an answer. Flip through the Flash Cards. If your score is 15 (75%) correct or greater, click Print on the File menu to print your results. If your score is less than 15 (75%) correct, then redo this exercise by clicking the Replay button.

3 Practice Test

Click the Practice Test link. Answer each question, enter your first and last name at the bottom of the page, and then click the Grade Test button. When the graded practice test is displayed on your screen, click Print on the File menu to print a hard copy. Continue to take practice tests until you score 80% or better.

4 Who Wants To Be a Computer Genius?

Click the Computer Genius link. Read the instructions, enter your first and last name at the bottom of the page, and then click the Play button. When your score is displayed, click the PRINT RESULTS link to print a hard copy.

5 Wheel of Terms

Click the Wheel of Terms link. Read the instructions, and then enter your first and last name and your school name. Click the PLAY button. When your score is displayed on the screen, right-click the score and then click Print on the shortcut menu to print a hard copy.

6 Crossword Puzzle Challenge

Click the Crossword Puzzle Challenge link. Read the instructions, and then click the Continue button. Work the crossword puzzle. When you are finished, click the Submit button. When the crossword puzzle is redisplayed, submit it to your instructor.

SCR Associates Case Simulation Session 6: Object Modeling

Overview

The SCR Associates case study is a Web-based simulation that allows you to practice your skills in a real-world environment. The case study transports you to SCR's intranet, where you complete 12 work sessions, each aligning with a chapter. As you work on the case, you will receive e-mail and voice mail messages, obtain information from SCR's online libraries, and perform various tasks.

How do I use the case?

- Review the SCR background material in Chapter 1.
- Read the Preview for this session and study the Task List.
- Visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to the **SCR Case Simulation**, and locate the intranet link.
- Enter your name and the password **sad9e**. An opening screen will display the 12 sessions.
- Select this session. Check your e-mail and voice mail carefully, and then work on the tasks.

Preview: Session 6

In the last session, you used data and process modeling techniques to develop a logical model of the new system. Now you will apply your object modeling skills to create various diagrams and documentation for the new TIMS system. You will review the background material and develop an object-oriented model that includes several types of diagrams.

Task List

1. I need to review Jesse's e-mail message regarding object modeling and the JAD session summary. Then she wants me to identify possible use cases and actors, and create a use case diagram for the TIMS system.
2. She also wants me to select one of the use cases and create a class diagram.
3. I will need a sequence diagram for the selected use case.
4. Jesse asked for a state transition diagram that describes typical student states and how they change based on certain actions and events.

FIGURE 6-35 Task list: Session 6.

Chapter Exercises

Review Questions

1. What is object-oriented analysis, and what are some advantages of using this technique?
2. Define an object, and give an example.
3. Define an attribute, and give an example.
4. Define a method, and give an example.
5. Define encapsulation, and explain the benefits it provides.
6. Define polymorphism, and give an example.
7. Define a class, subclass, and superclass, and give examples.
8. Define an actor, and give an example.
9. Define a use case and a use case diagram, and give examples.
10. Define the term *black box*, and explain why it is an important concept in object-oriented analysis.

Discussion Topics

1. The chapter mentioned that systems analysts and programmers transform objects into program code modules that can be optimized, tested, and reused. Modular design is a very popular design concept in many industries. What other examples of modular design can you suggest?
2. You are an IT consultant, and you are asked to create a new system for a small real estate brokerage firm. Your only experience is with traditional data and process modeling techniques. This time, you decide to try an object-oriented approach. How will you begin? How are the tasks different from traditional structured analysis?
3. You are creating a system for a bowling alley to manage information about its leagues. During the modeling process, you create a state transition diagram for an object called LEAGUE BOWLERS. What are the possible states of a league bowler, and what happens to a bowler who quits the league and rejoins the following season?
4. A debate is raging at the IT consulting firm where you work. Some staff members believe that it is harder for experienced analysts to learn object-modeling techniques, because the analysts are accustomed to thinking about data and processes as separate entities. Others believe that solid analytical skills are easily transferable and do not see a problem in crossing over to the newer approach. What do you think, and why?

Projects

1. Search the Internet for information about the history and development of UML.
2. Contact the IT staff at your school or at a local business to learn whether the organization uses object-oriented programming languages. If so, determine what languages and versions are used and why they were selected.
3. Search the Internet for information about groups and organizations that support and discuss object-oriented methods and issues.
4. Search the Internet for information about CASE tools that provide UML support.

Apply Your Knowledge

The Apply Your Knowledge section contains four mini-cases. Each case describes a situation, explains your role in the case, and asks you to respond to questions. You can answer the questions by applying knowledge you learned in the chapter.

1 Hertford Post Office

Situation:

Hertford has a typical small town post office that sells stamps, rents post office boxes, and delivers mail to postal customers.

1. Identify possible actors and use cases involved in the post office functions.
2. Create a use case diagram for the post office operation.
3. Select one of the use cases and create a class diagram.
4. Create a sequence diagram for the use case you selected.

2 New Branch School District

Situation:

The New Branch School District operates a fleet of 40 buses that serve approximately 1,000 students in grades K–12. The bus operation involves 30 regular routes, plus special routes for activities, athletic events, and summer sessions. The district employs 12 full-time drivers and 25 to 30 part-time drivers. A dispatcher coordinates the staffing and routes and relays messages to drivers regarding students and parents who call about pickup and drop-off arrangements.

1. Identify possible actors and use cases involved in school bus operations.
2. Create a use case diagram for the school bus system.
3. Create a sequence diagram for the use case you selected.
4. Create a state transition diagram that describes typical student states and how they change based on specific actions and events.

3 Pleasant Creek Community College Registration System

Situation:

Pleasant Creek Community College has a typical school registration process. Student support services include faculty advisors and tutors. The administration has asked you, as IT manager, to develop an object-oriented model for a new registration system.

1. List possible objects in the new registration system, including their attributes and methods.
2. Identify possible use cases and actors.
3. Create a use case diagram that shows how students register.
4. Create a state transition diagram that describes typical student states and how they change based on specific actions and events.

4 Student Bookstore at Pleasant Creek Community College

Situation:

The bookstore staff at Pleasant Creek Community College works hard to satisfy students, instructors, and the school's business office. Instructors specify textbooks for particular courses, and the bookstore orders the books and sells them to students. The bookstore wants you to develop an object-oriented model for a new bookstore information management system.

1. List possible objects in the bookstore operation, including their attributes and methods.
2. Identify possible use cases and actors.
3. Select one of the use cases that you identified in step 2 and create a sequence diagram.
4. Create an object relationship diagram that provides an overview of the system, including how textbooks are selected by instructors, approved by a department head, and sold to students by the bookstore.

Case Studies

Case studies allow you to practice specific skills learned in the chapter. Each chapter contains several case studies that continue throughout the textbook, and a chapter capstone case.

New Century Health Clinic

New Century Health Clinic offers preventive medicine and traditional medical care. In your role as an IT consultant, you will help New Century develop a new information system.

Background

You began the systems analysis phase at New Century Health Clinic by completing a series of interviews, reviewing existing reports, and observing office operations. Then, in Chapter 5, you acquired more information and developed a set of DFDs, process descriptions, and a data dictionary.

Now you decide to practice the object modeling skills you learned in this chapter. Before you begin, go back to Chapter 5 and review the New Century background material and fact-finding results. Also, your instructor may provide you with a complete set of interview summaries that you can use to perform your assignments. Then complete the following tasks.

Assignments

1. Identify possible use cases and actors, and create a use case diagram for the New Century Health Clinic system.
2. Select one of the use cases and create a class diagram.
3. Create a sequence diagram for the use case that you selected.
4. Create a state transition diagram that describes typical patient states and how they change based on specific actions and events.

PERSONAL TRAINER, INC.

Personal Trainer, Inc., owns and operates fitness centers in a dozen Midwestern cities. The centers have done well, and the company is planning an international expansion by opening a new "supercenter" in the Toronto area. Personal Trainer's president, Cassia Umi, hired an IT consultant, Susan Park, to help develop an information system for the new facility. During the project, Susan will work closely with Gray Lewis, who will manage the new operation.

Background

Working as an IT consultant for Personal Trainer, Susan Park used data and process modeling tools to create a logical model of the proposed information system. Now she wants to build an object-oriented view of the system using O-O tools and techniques.

Assignments

Before you perform the following tasks, you should review the information and background in Chapters 1 and 2, and the fact-finding summary of the case provided in Chapter 4.

1. Identify possible use cases and actors, and create a use case diagram for the Personal Trainer information system.
2. Select one of the use cases and create a class diagram.
3. Create an object relationship diagram for the system.
4. Create a state transition diagram that describes typical member states and how they change based on specific actions and events.

CHAPTER CAPSTONE CASE: SoftWear, Limited

SWL

SoftWear, Limited (SWL), is a continuing case study that illustrates the knowledge and skills described in each chapter. In this case study, the student acts as a member of the SWL systems development team and performs various tasks.

Background

Rick Williams, a systems analyst, and Carla Moore, a programmer/analyst, completed a set of DFDs representing a data and process model of the SWL payroll system project. Rick had recently attended a workshop on object modeling techniques, and suggested that he and Carla should experiment with object-oriented analysis. After he explained the concepts and techniques to Carla, she agreed that it was a good opportunity to gain some experience, and they decided to give it a try.

Rick and Carla began by reviewing the data they had collected earlier, during requirements modeling. They studied the DFDs and the data dictionary to identify the people, events, and transactions that would show as classes. They identified employees, human resources transactions, time sheet entries, payroll actions, and stock transfers. They defined attributes and methods for each of those classes, as shown in Figure 6-36. When they were finished, they reviewed the results. They noticed that the structured DFDs did not show a department head as an entity. Rick remembered that department heads submitted time sheets to the payroll department, and the payroll clerks actually entered the data into the system. Because they were looking at the system in a different way, they decided to include department heads as a subclass of the EMPLOYEE class.

EMPLOYEE

Employee number
Employee name
Address
Telephone number
Date of birth
Sex
Title, rate of pay
Deductions
State

Add new
Change name
Change address
Change telephone
Change deductions
Change state

HR TRANSACTION

Employee number
Employee name
State

Add new
Change state
Notify

TIME SHEET ENTRY

Employee number
Week ending date
Hours worked

Add new
Correct hours
Generate report

PAYROLL ACTION

Employee number
Hours worked
Overtime hours
Rate of pay
Overtime rate
Deductions
Contributions
Federal tax withheld
State tax withheld
Local tax withheld

Generate checks
Change deductions
Change contributions
Change federal rate
Change local rate
Change state rate
Change rate of pay
Generate W-2
Notify
Calculate

STOCK TRANSFER

Employee number
Stock holdings
Stock contribution

Purchase stock
Sell stock
Change contribution
Generate report

DEPT HEAD

Employee number
Employee name
Address
Telephone number
Date of birth
Sex
Title, rate of pay
Deductions
State

Add new
Change name
Change address
Change telephone
Change deductions
Change state
Manages work
Submits time sheets

FIGURE 6-36 SWL payroll system classes.

Use Cases

The next step was for Rick and Carla to define the use cases. They tried to think of all the situations that involve an EMPLOYEE object. For example, employees might get hired, promoted, receive a raise, terminate, retire, change their names, or change their payroll contributions.

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

They also decided to create use cases that involved the PAYROLL ACTION object. The examples included these: Change an employee's deductions, change an employee's contributions, change the federal tax rate, change the state tax rate, change the local tax rate, calculate weekly gross pay, calculate weekly taxes, calculate weekly contributions, generate weekly paychecks, and notify the stock transfer department of change in contributions.

After they defined the use cases and the actors, they created a description for each use case showing the use case name, actors, description, successful completion, alternatives, preconditions, postconditions, and assumptions.

Creating use case descriptions was hard work, and they found that they had to return frequently to their documentation and fact-finding results. First, they created descriptions for the RECEIVE RAISE and RECEIVE PROMOTION use cases, as shown in Figure 6-37.

RECEIVE RAISE Use Case

Name:	RECEIVE RAISE
Actor:	Employee
Description:	Describes the change to an employee's pay rate
Successful completion:	1. Employee gets a raise 2. Human resources department changes employee data to the employee object and the human resources records
Alternative:	None
Precondition:	Employee has been approved for a raise
Postcondition:	Employee's pay rate is changed on all records
Assumptions:	None

RECEIVE PROMOTION Use Case

Name:	RECEIVE PROMOTION
Actor:	Employee
Description:	Describes the change to employee title
Successful completion:	1. Employee gets promoted 2. Human resources department changes employee data and completes HR transaction
Alternative:	None
Precondition:	Employee has been approved for promotion
Postcondition:	Employee title is changed
Assumptions:	Employee accepts position

FIGURE 6-37 Descriptions for the RECEIVE RAISE and RECEIVE PROMOTION use cases.

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

Now they were ready to create a use case diagram to show several related use cases. To keep it simple, Carla suggested that they show no more than three use cases per diagram.

They decided to create a use case diagram to describe how the payroll is generated. The diagram, shown in Figure 6-38, includes three use cases: CREATE TIMESHEET, CALCULATE PAYROLL, and GENERATE PAYCHECK. In the diagram, the DEPARTMENT HEAD actor creates a new instance of the TIMESHEET ENTRY object, which notifies the CALCULATE PAYROLL use case, which is initiated by the PAYROLL CLERK. The GENERATE PAYCHECK use case then issues a paycheck to the EMPLOYEE actor.

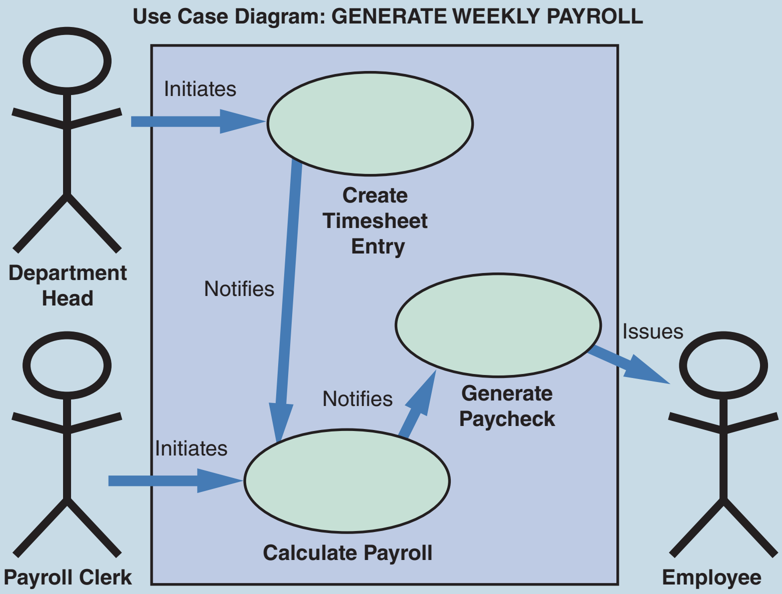


FIGURE 6-38 Use case diagram for the GENERATE WEEKLY PAYROLL scenario.

Class Diagrams

The use case diagram gave Rick and Carla a valuable snapshot of the business processes involved in GENERATE WEEKLY PAYROLL. To document the relationships and interaction in more detail, they created the class diagram shown in Figure 6-39. The diagram includes five different classes and various types of cardinality.

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

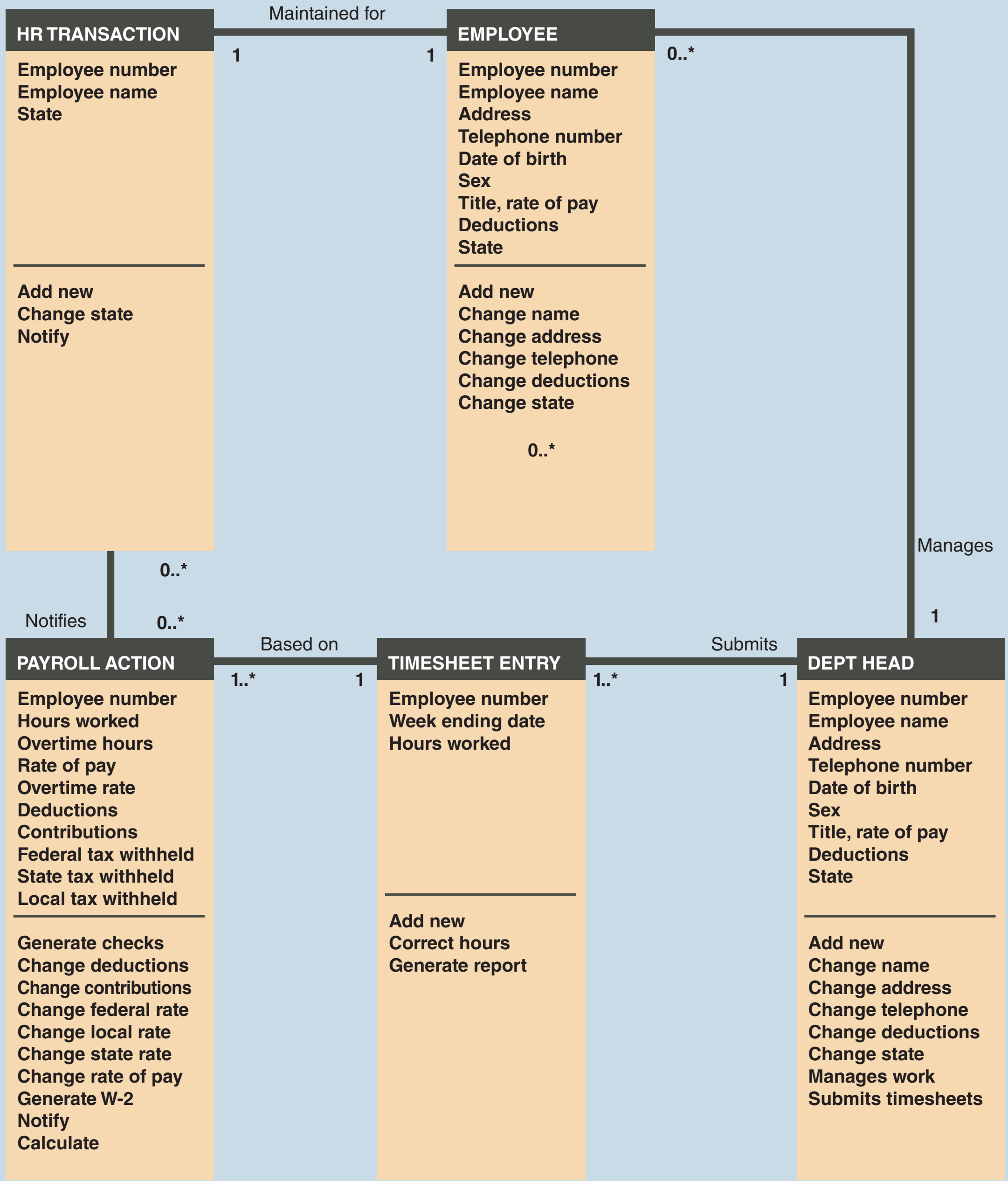


FIGURE 6-39 The GENERATE WEEKLY PAYROLL class diagram includes five classes and various types of relationships among the classes.

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

Sequence Diagrams

Next, the pair decided to create a sequence diagram. Carla was eager to see how a sequence diagram would help them visualize the time frame in which events occur. They created a diagram for the CHANGE CONTRIBUTIONS method in the EMPLOYEE object. The sequence diagram in Figure 6-40 shows the steps that occur when an employee changes benefits contributions. Notice that the diagram includes the messages sent and the lifeline of the objects. Rick and Carla were satisfied that they could create sequence diagrams easily, and they decided to move on to the state transition diagram.

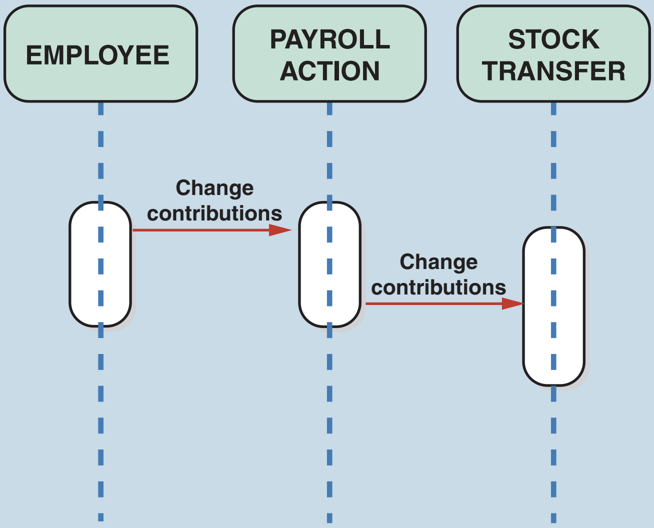


FIGURE 6-40 Sequence diagram for the CHANGE CONTRIBUTIONS scenario.

State Transition Diagram

Rick explained that a state transition diagram shows how an object's state, or status, changes as a result of various actions or events. The state transition diagram they created in Figure 6-41 shows the status of an employee from the time the employee is hired to the time he or she quits, is fired, or retires. Notice that the employee is a PROSPECTIVE employee until all physicals are passed and all paperwork is processed, and then he or she becomes a CURRENT employee. Once employment ends for any reason, the individual becomes a PAST employee. At this point, even if the employee returns to the company later on he or she will come in as a new instance of the EMPLOYEE object. Rick and Carla were surprised at how easy that was, and they decided to try an activity diagram.

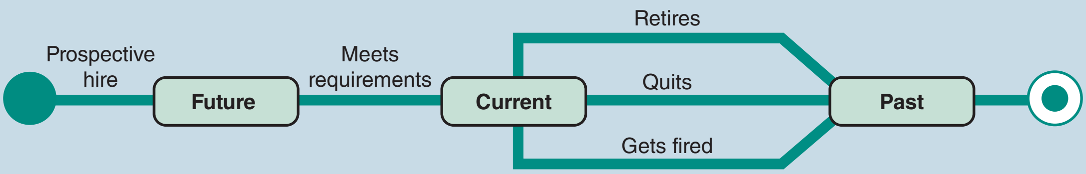


FIGURE 6-41 State transition diagram shows changes in employee status caused by actions and events.

Activity Diagram

Rick suggested that they create an activity diagram showing some of the situations they had explored in detail. Their diagram showed the interaction between objects during certain scenarios and enabled them to visualize system activity, as shown in Figure 6-42. They agreed that the technique gave them additional object modeling experience that would be valuable in the future. At that point, they packaged all the diagrams in a folder and saved the overall object model for future reference.

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

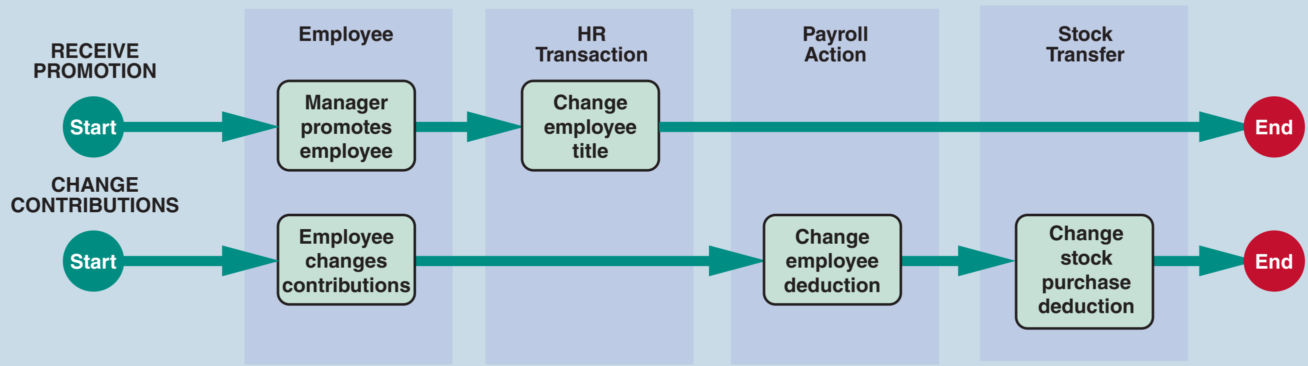


FIGURE 6-42 Activity diagram shows the RECEIVE PROMOTION scenario and the CHANGE CONTRIBUTIONS scenario.

Team Tasks

1. Rick is interested in your views on the future of object-oriented analysis and design. He is scheduled to make a presentation on the topic next week at a meeting of IT professionals. He asked you to do some research, using the Internet and industry publications, and send him an e-mail message describing the current use of object-oriented analysis and trends for the future.
2. As a team member, you know how important it can be to have a well-organized object model. The team has asked you to handle this task. How will you go about it?
3. When you worked on the class diagrams, you had to understand and apply the concept of cardinality. How would you explain this concept to a new team member?
4. List all the different types of diagrams you used to create the object model, with a brief explanation of each diagram.

Manage the SWL Project

You have been asked to manage SWL's new information system project. One of your most important activities will be to identify project tasks and determine when they will be performed. Before you begin, you should review the SWL case in this chapter. Then list and analyze the tasks, as follows:

LIST THE TASKS Start by listing and numbering at least 10 tasks that the SWL team needs to perform to fulfill the objectives of this chapter. Your list can include SWL Team Tasks and any other tasks that are described in this chapter. For example, Task 3 might be to Identify the actors, and Task 6 might be to Draw a use case diagram.

ANALYZE THE TASKS Now study the tasks to determine the order in which they should be performed. First identify all concurrent tasks, which are not dependent on other tasks. In the example shown in Figure 6-43 on the next page, Tasks 1, 2, 3, 4, and 5 are concurrent tasks, and could begin at the same time if resources were available.

Other tasks are called dependent tasks, because they cannot be performed until one or more earlier tasks have been completed. For each dependent task, you must identify specific tasks that need to be completed before this task can begin. For example, you would want to identify the actors before you could draw a use case diagram, so Task 6 cannot begin until Task 3 is completed, as Figure 6-43 shows.

Chapter Capstone Case: SoftWear, Limited (continued)

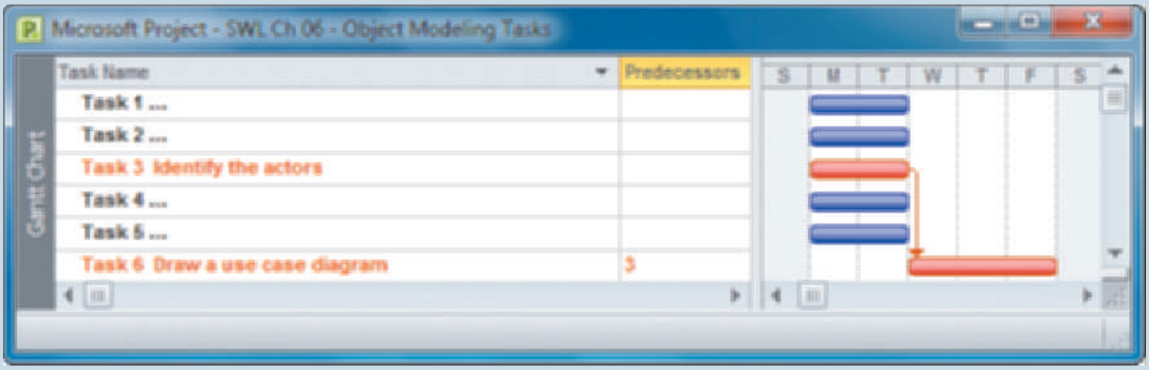


FIGURE 6-43 Tasks 1, 2, 3, 4, and 5 are concurrent tasks that could be performed at the same time. Task 6 is a dependent task that cannot be performed until Task 3 has been completed.

Chapter 3 describes project management tools, techniques, and software. To learn more, you can use the Features section on your Student Study Tool CD-ROM, or visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and locate the project management resources library for this book. On the Web, Microsoft offers demo versions, training, and tips for using Project 2010. You also can visit the OpenWorkbench.org site to learn more about this free, open-source software.

Ready for a Challenge?

In additional to technical skills, IT professionals need critical thinking skills such as perception, organization, analysis, problem-solving, and decision-making. The Ready for a Challenge feature can help you learn, practice, and apply critical thinking skills that you can take to the workplace.

The IT team at Game Technology prepared DFDs for the new Customer Contact Care system (C^3). Now, you will help the team develop a set of object models. To be sure you can handle the tasks you decide to review Chapter 6 of your systems analysis textbook.

The C^3 system model will include use cases that represent various business processes and functions. The customer and sales rep entities will be shown as actors, with the following methods:

Entity	Methods
Customer	• Request product information • Place order • Respond to Web promotion • Sign up on game blog
Sales Rep	• Initiate sales call to customer • Respond to customer inquiry • Follow up after prior order • Follow up after prior information request

You also learned that customers will fall into one of four categories. Customers move from one category to another when certain conditions occur, as shown:

Customer Category	Conditions
Potential	Has not placed an order in the last 24 months, but *has* • Responded to a Web promotion, or • Requested product information, or • Signed up on our game blog
Active	Placed at least one order in the last 24 months
Preferred	Placed at least three orders in the last 24 months
Former	Any **Active** or **Preferred** customer who placed no orders in the last 24 months

Practice Tasks

A. Draw a use case diagram that includes the entities and the use cases.
B. Draw a state transition diagram that shows the customer categories and the conditions for each category.

After you complete the Practice Tasks, to check your work and view sample answers, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to the resources for this chapter, and locate Ready for a Challenge?.

The Challenge

The marketing group decided to add a new marketing technician position to provide IT support for sales reps. Specifically, the marketing tech will track customer response times, prepare contact management plans for sales reps, and develop surveys to measure customer satisfaction.

Marketing also decided to update the customer categories. The **Potential** and **Former** categories are not affected. The new categories are **Silver**, which replaces the **Active** category; **Gold**, which replaces the **Preferred** category; and **Platinum**, which includes any customer whose orders total at least $1,000 in the last 24 months.

Challenge Tasks

A. Draw a new version of the use case diagram that adds the new marketing technician.
B. Draw a state transition diagram that shows the new customer categories and the conditions for each category.

CHAPTER 7 Development Strategies

Chapter 7 is the final chapter in the systems analysis phase of the SDLC. This chapter describes software trends, acquisition and development strategies, traditional versus Web-based development, outsourcing versus in-house development, the system requirements document, prototyping, and preparing for the transition to the next SDLC phase — systems design.

INTRODUCTION

OBJECTIVES

When you finish this chapter, you will be able to:

- Describe the concept of Software as a Service
- Define Web 2.0 and cloud computing
- Explain software acquisition alternatives, including traditional and Web-based software development strategies
- Describe software outsourcing options, including offshore outsourcing and the role of service providers
- Explain advantages and disadvantages of in-house software development
- Discuss cost-benefit analysis and financial analysis tools
- Describe a request for proposal (RFP) and a request for quotation (RFQ)
- Describe the system requirements document
- Explain the transition from systems analysis to systems design
- Discuss systems design guidelines
- Describe software development trends

The main objective of the systems analysis phase is to build a logical model of the new information system. In Chapters 4, 5, and 6, you learned about requirements modeling, data and process modeling, and object modeling. Chapter 7 describes the remaining activities in the systems analysis phase, which include evaluation of alternative solutions, preparation of the system requirements document, and presentation of the system requirements document to management. The chapter also describes the transition to systems design, prototyping, and systems design guidelines. The chapter concludes with a discussion of trends in software development.

CHAPTER INTRODUCTION CASE: Mountain View College Bookstore

Background: Wendy Lee, manager of college services at Mountain View College, wants a new information system that will improve efficiency and customer service at the three college bookstores.

In this part of the case, Tina Allen (systems analyst) and David Conroe (student intern) are talking about development strategies for the new system.

Participants: Tina and David

Location: Tina's office, Wednesday morning, November 16, 2011

Project status: Tina and David developed a logical model that includes data flow diagrams, a data dictionary, and process descriptions. They also created an object model. Now they are ready to discuss development strategies for the new bookstore system.

Discussion topics: Web-based versus traditional development, cost-benefit analysis, steps in purchasing a software package, transition to systems design, and systems design guidelines

Tina: Good morning, David. Are you ready for the next step?

David: *Sure. Now that we have a logical model of the bookstore system, what comes next?*

Tina: We're at a transition point between the logical design, which describes what the new system will do, and the physical design phase, which describes how it will be done, including the user interface and physical components. Before we start the physical design, we have to study various systems development options and make a recommendation to Wendy.

David: *What are the options?*

Tina: Well, some large organizations use Web-based systems hosted by outside vendors who supply and maintain the software. In a sense, the customer rents the application. I checked with our IT director, and she feels we're not ready for that approach. She wants us to implement a system on the college network and migrate to a Web-based system later. That brings us to the next set of questions.

David: *Such as?*

Tina: We need to consider our role in the development process. We can build the system ourselves, which is called in-house development. Or we can purchase a software package, which might need some degree of modification to meet our needs. Or we could consider outsourcing options, including hiring an IT consultant to help with development tasks. Either way, we need to do a cost-benefit study.

David: *What about the transition from logical to physical design that you mentioned?*

Tina: The idea is to take our logical design, which is similar to an architect's proposal, and translate it into a physical design, which is more like a working blueprint. If we decide to develop the system in-house, we'll probably build a prototype, or working model of the system. If we decide to purchase a package, we'll follow a series of steps that will help us select the best product. We'll also talk about systems design guidelines.

David: *When you mention the idea of a blueprint, it sounds like we're getting ready to pick up our tools and go to work.*

Tina: We sure are. Here's a task list to get us started:

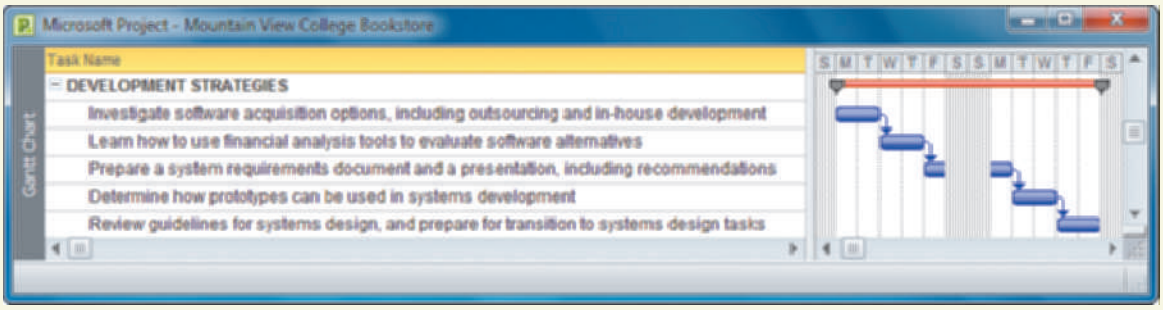


FIGURE 7-1 Typical development strategies task list.

Development Strategies Overview

Just a few years ago, a typical company either developed software itself, purchased a software package (which might need some modification), or hired consultants or outside resources to perform the work. Today, a company has many more choices, including application service providers, Web-hosted software options, and firms that offer a variety of enterprise-wide software solutions.

Selecting the best development path is an important decision that requires companies to consider three key topics: the impact of the Internet, software outsourcing options, and in-house software development alternatives. These topics are reviewed in the following sections.

The Impact of the Internet

The Internet has triggered enormous changes in business methods and operations, and software acquisition is no exception. This section examines a trend that views Software as a Service, the changing marketplace for software, and how Web-based development compares to traditional methods. The section concludes with a description of Internet-related trends, including Web 2.0 and cloud computing.

Software as a Service

In the traditional model, software vendors develop and sell application packages to customers. Typically, customers purchase licenses that give them the right to use the software under the terms of the license agreement. Although this model still accounts for most software acquisition, a new model, called **Software as a Service** (**SaaS**), is changing the picture dramatically.

SaaS is a model of software deployment where an application is hosted as a service provided to customers over the Internet. SaaS reduces the customer's need for software maintenance, operation, and support.

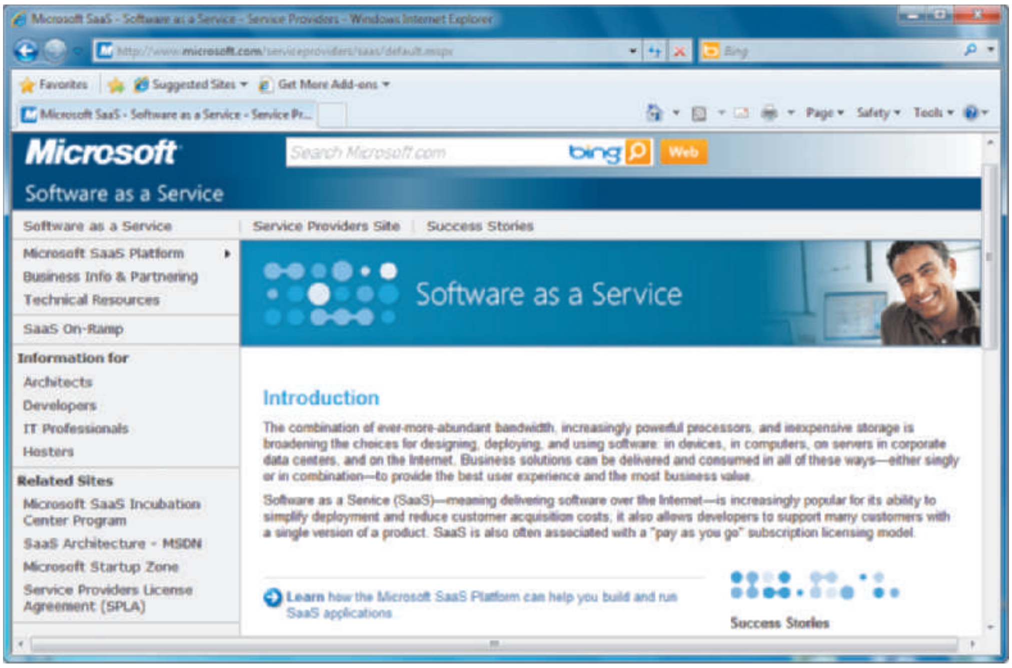


FIGURE 7-2 Microsoft's SaaS platform promises better value and a new concept called Software + Services.

In a highly competitive marketplace, major vendors constantly strive to deliver new and better solutions. For example, as shown in Figure 7-2, Microsoft claims that its SaaS platform offers the best solution and business value. Microsoft also promotes a broader vision, called Software + Services. The term refers to Microsoft's strategy for cloud computing, which integrates software applications, platforms and infrastructure. According to Microsoft, Software + Services will allow seamless connection of smart devices to the tremendous power of the Web.

The Web Host Industry Review shown in Figure 7-3 is an online source of information about SaaS products, trends, and events. In a published report, the Review quoted a Gartner, Inc. prediction that 25% of all new business software will be deployed as a service by 2011, while the value of the SaaS industry will grow to $40 billion.

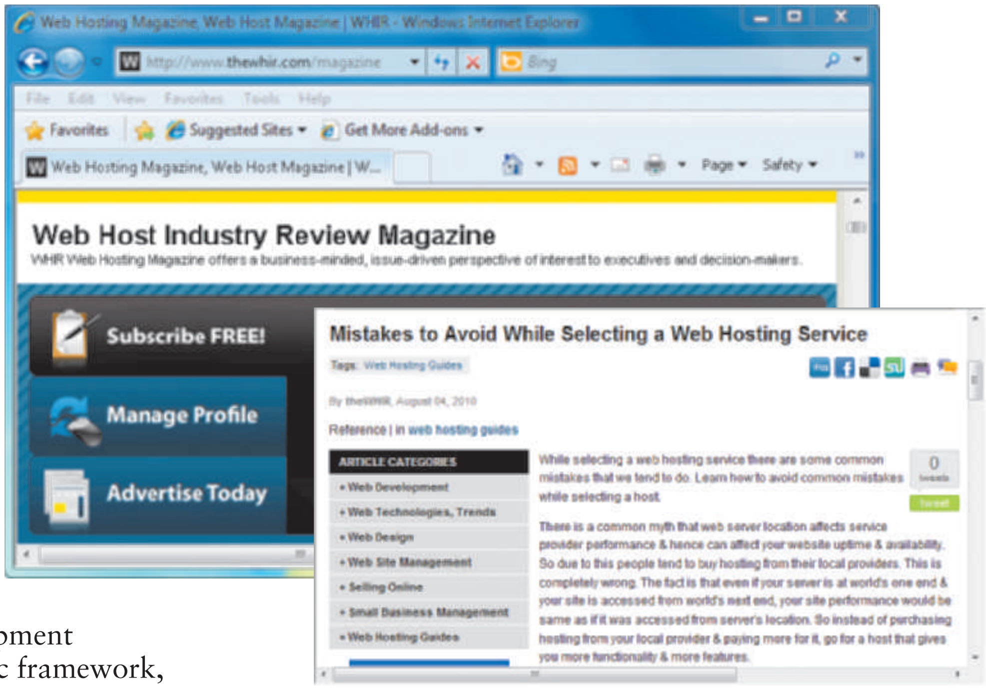


FIGURE 7-3 The Web Host Industry Review (WHIR) is a clearinghouse for SaaS information.

Traditional vs. Web-Based Systems Development

As a systems analyst, you must consider whether development will take place in a Web-centric framework, or in a traditional environment. This section provides an overview of some of the similarities and differences.

In an Internet-based system, the Web becomes an integral part of the application, rather than just a communication channel, and systems analysts need new application development tools and solutions to handle the new systems. Two major Web-based development environments are Microsoft's **.NET** and IBM's **WebSphere,** which are shown in Figure 7-4 on the next page. Microsoft regards .NET as a platform-independent software environment. IBM describes WebSphere as a set of products specifically designed to support e-business applications across multiple computing platforms.

Although there is a major trend toward Web-based architecture, many firms rely on traditional systems, either because they are legacy applications that are not easily replaced, or because they do not require a Web component to satisfy user needs. If you need to choose, you should consider some key differences between traditional and Web-based system development. Building the application in a Web-based environment can offer greater benefits, and sometimes greater risks, compared to a traditional environment. The following sections list some characteristics of traditional versus Web-based development.

TRADITIONAL DEVELOPMENT In a traditional systems development environment:

- Systems design is influenced by compatibility issues, including existing hardware and software platforms and legacy system requirements.
- Systems are designed to run on local and wide-area company networks.
- Systems often utilize Internet links and resources, but Web-based features are treated as enhancements rather than core elements of the design.
- Development typically follows one of three main paths: in-house development, purchase of a software package with possible modification, or use of outside consultants.

- Scalability can be affected by network limitations and constraints.
- Many applications require substantial desktop computing power and resources.
- Security issues usually are less complex than with Web-based systems, because the system operates on a private company network, rather than the Internet.

FIGURE 7-4 Microsoft's .Net Framework and IBM's WebSphere are comprehensive software development environments.

WEB-BASED DEVELOPMENT In a Web-based systems development environment:

- Systems are developed and delivered in an Internet-based framework such as .NET or WebSphere.
- Internet-based development treats the Web *as* the platform, rather than just a communication channel.
- Web-based systems are easily scalable, and can run on multiple hardware environments.
- Large firms tend to deploy Web-based systems as enterprise-wide software solutions for applications such as customer relationship management, order processing, and materials management.
- Web-based software treats the software application as a service that is less dependent on desktop computing power and resources.
- When companies acquire Web-based software as a *service* rather than a *product* they purchase, they can limit in-house involvement to a minimum and have the vendor install, configure, and maintain the system by paying agreed-upon fees.
- Web-based software usually requires additional layers, called **middleware**, to communicate with existing software and legacy systems.

Looking to the Future: Web 2.0 and Cloud Computing

In the constantly changing world of IT, no area is more dynamic than Internet technology. Two examples of evolving trends are Web 2.0 and cloud computing. Systems analysts should be aware of these concepts and consider them as they plan large-scale systems. Web 2.0 and cloud computing are discussed in more detail in Chapter 10, System Architecture.

Many IT professionals use the term **Web 2.0** to describe a second generation of the Web that will enable people to collaborate, interact, and share information much more effectively. This new environment is based on continuously available user applications rather than static HTML Web pages, without limitations regarding the number of users or how they will be able to access, modify, and exchange data. The Web 2.0 platform will enhance interactive experiences, including wikis and blogs, and social-networking applications such as Twitter, MySpace and Facebook.

The term **cloud computing** refers to the cloud symbol that indicates a network, or the Internet. Some industry leaders predict that cloud computing will offer an overall online software and data environment supported by supercomputer technology. If so, cloud computing would be an ultimate form of SaaS, delivering services and data to users who would need only an Internet connection and a browser. However, as the *InfoWorld* article shown in Figure 7-5 points out, no standard definition of cloud computing exists, and the concept means different things to different people.

FIGURE 7-5 Cloud computing holds the promise of a new generation of powerful Web applications and services. This is an evolving technology, and at this point, the term means different things to different people.

ON THE WEB

To learn more about Web 2.0, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Web 2.0 link.

ON THE WEB

To learn more about cloud computing, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Cloud Computing link.

Outsourcing

Outsourcing is the transfer of information systems development, operation, or maintenance to an outside firm that provides these services, for a fee, on a temporary or long-term basis. Outsourcing can refer to relatively minor programming tasks, the rental of software from a service provider, the outsourcing of a basic business process (often called **business process outsourcing**, or **BPO**), or the handling of a company's entire IT function. Numerous firms and organizations offer information about outsourcing topics and issues. For example, the Outsourcing Center, shown in Figure 7-6, provides free research, case studies, database directories, market intelligence, and updates on trends and best practices in outsourcing as a strategic business solution.

FIGURE 7-6 The Outsourcing Center is dedicated to providing information about outsource trends and practices.

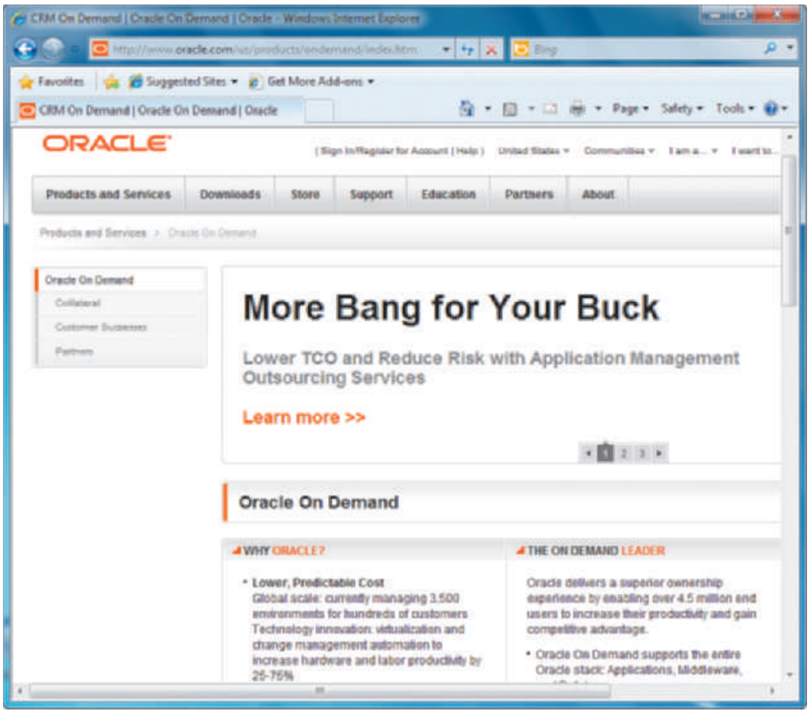


FIGURE 7-7 Oracle Corporation offers a fixed-fee outsourcing plan called Oracle On Demand.

The Growth of Outsourcing

Traditionally, firms outsourced IT tasks as a way of controlling costs and dealing with rapid technological change. While those reasons still are valid, outsourcing has become part of an overall IT strategy for many organizations. The outsourcing trend also has affected software vendors, who have adjusted their marketing accordingly. For example, Oracle Corporation offers a service called Oracle On Demand, which provides e-business applications, as shown in Figure 7-7. Oracle also cites data that shows that businesses spend up to 80% of their IT budgets maintaining existing software and systems, which forces IT managers "... to spend time managing tedious upgrades instead of revenue-generating IT projects."

A firm that offers outsourcing solutions is called a **service provider**. Some service providers concentrate on specific software applications; others offer business services such as order processing and customer billing. Still others offer enterprise-wide software solutions that integrate and manage functions such as accounting, manufacturing, and inventory control.

Two popular outsourcing options involve application service providers and firms that offer Internet business services. These terms are explained in the following sections.

APPLICATION SERVICE PROVIDERS An **application service provider** (ASP) is a firm that delivers a software application, or access to an application, by charging a usage or subscription fee. An ASP provides more than a license to use the software; it *rents* an operational package to the customer. ASPs typically provide commercially available software such as databases and accounting packages. If a company uses an ASP to supply a data management package, for example, the company does not have to design, develop, implement, or maintain the package. ASPs represent a rapidly growing trend, using the Internet as the primary delivery channel.

ON THE WEB

To learn more about application service providers, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Application Service Providers link.

INTERNET BUSINESS SERVICES Some firms offer **Internet business services** (IBS), which provide powerful Web-based support for transactions such as order processing, billing, and customer relationship management. Another term for IBS is **managed hosting**, because system operations are managed by the outside firm, or host.

An IBS solution is attractive to customers because it offers online data center support, mainframe computing power for mission-critical functions, and universal access via the Internet. Many firms, such as Rackspace, compete in the managed hosting market, as shown in Figure 7-8.

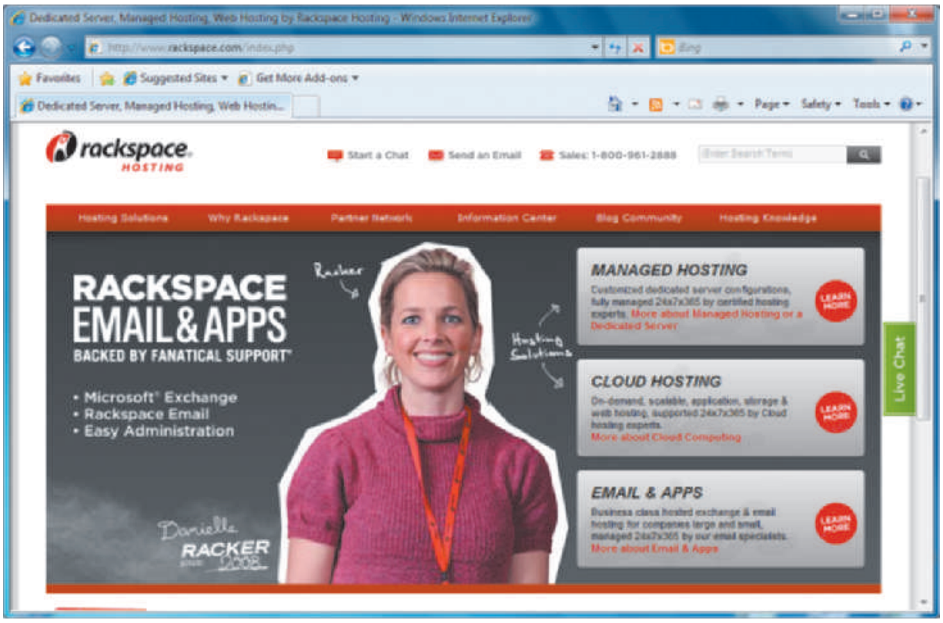


FIGURE 7-8 Rackspace Corporation offers multi-platform managed hosting and IBS services.

Outsourcing Fees

Firms that offer Software as a Service, rather than a product, have developed fee structures that are based on how the application is used by customers during a specific time period. Several models exist, including fixed fee, subscription, and usage or transaction. A **fixed fee model** uses a set fee based on a specified level of service and user support. An example of a fixed fee model is Oracle's On Demand service. A **subscription model** has a variable fee based on the number of users or workstations that have access to the application. Finally, a **usage model** or **transaction model** charges a variable fee based on the volume of transactions or operations performed by the application.

When a company considers outsourcing, it should estimate usage characteristics to determine which fee structure would be most desirable, and then attempt to negotiate a service provider contract based on that model.

Outsourcing Issues and Concerns

When a company decides to outsource IT functions, it takes an important step that can affect the firm's resources, operations, and profitability. Mission-critical IT systems should be outsourced only if the result is a cost-attractive, reliable, business solution that fits the company's long-term business strategy and involves an acceptable level of risk. Moving IT work overseas raises even more issues, including potential concerns about control, culture, communication, and security.

In addition to long-term strategic consequences, outsourcing also can raise some concerns. For example, a company must turn over sensitive data to an external service provider and trust the provider to maintain security, confidentiality, and quality. Also, before outsourcing, a company must carefully review issues relating to insurance, potential liability, licensing and information ownership, warranties, and disaster recovery.

Most important, a company considering outsourcing must realize that the solution can be only as good as the outsourcing firm that provides the service. A dynamic economy can give rise to business failures and uncertainty about the future. In this climate, it is especially important to review the history and financial condition of an outsourcing firm before making a commitment.

Mergers and acquisitions also can affect outsourcing clients. For example, after their merger, Compaq and Hewlett-Packard restructured and streamlined the products and services offered by the new company. Even with large, financially healthy firms such as these, a merger or acquisition can have some impact on clients and customers. If stability is important, an outsourcing client should consider these issues.

ON THE WEB

To learn more about outsourcing, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Outsourcing link.

Outsourcing can be especially attractive to a company whose volume fluctuates widely, such as a defense contractor. In other situations, a company might decide to outsource application development tasks to an IT consulting firm if the company lacks the time or expertise to handle the work on its own. Outsourcing relieves a company of the responsibility of adding IT staff in busy times and downsizing when the workload lightens. A major disadvantage of outsourcing is that it raises employee concerns about job security. Talented IT people usually prefer positions where the firm is committed to in-house IT development — if they do not feel secure, they might decide to work directly for the service provider.

Offshore Outsourcing

Offshore outsourcing, or **global outsourcing**, refers to the practice of shifting IT development, support, and operations to other countries. In a trend similar to the outflow of manufacturing jobs over a several-decade period, many firms are sending IT work overseas at an increasing rate.

For example, Dartmouth professor Matthew Slaughter has noted that IT work will move offshore even faster than manufacturing, because it is easier to ship work across networks and telephone lines and put consultants on airplanes than it is to ship bulky raw materials, build factories, and deal with tariffs and transportation issues. Several years ago, the IT consulting firm Gartner, Inc., accurately forecast the steady growth of offshore outsourcing, and predicted that outsourcing would evolve from labor-intensive maintenance and support to higher-level systems development and software design.

In addition to exporting IT jobs, many large multinational firms, including Microsoft and IBM, have opened technical centers in India and other countries. Some observers believe that India might gain as many as 2 million IT jobs in the next decade.

The main reason for offshore outsourcing is the same as domestic outsourcing: lower bottom-line costs. Offshore outsourcing, however, involves some unique risks and concerns. For example, workers, customers, and shareholders in some companies have protested this trend, and have raised public awareness of possible economic impact. Even more important, offshore outsourcing involves unique concerns regarding project control, security issues, disparate cultures, and effective communication with critical functions that might be located halfway around the globe.

CASE IN POINT 7.1: Turnkey Services

Turnkey Services is an application service provider that offers payroll and tax preparation services for hundreds of businesses in the Midwest. The firm is considering a major expansion into accounting and financial services, and is looking into the possibility of supporting this move by hiring IT subcontractors in several foreign countries. Peter Belmont, Turnkey's president, has asked you to help him reach a decision. Specifically, he wants you to cite the pros and cons of offshore outsourcing. He expects you to perform Internet research on this topic, and he wants you to present your views at a meeting of Turnkey managers next week.

In-House Software Development Options

In addition to numerous outsourcing options, a company can choose to develop its own systems, or purchase, possibly customize, and implement a software package. These development alternatives are shown in Figure 7-9. Although many factors influence this decision, the most important consideration is the total cost of ownership (TCO), which was explained in Chapter 4. In addition to these options, companies also develop user applications designed around commercial software packages, such as Microsoft Office, to improve user productivity and efficiency.

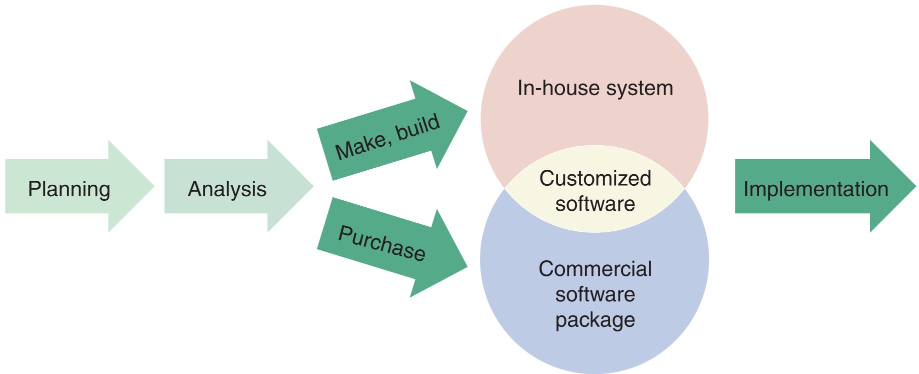


FIGURE 7-9 Instead of outsourcing, a company can choose to develop a system in-house, or purchase and possibly customize a commercial package.

Make or Buy Decision

The choice between developing versus purchasing software often is called a **make or buy**, or **build or buy** decision. The company's IT department makes, builds, and develops **in-house software**. A **software package** is obtained from a vendor or application service provider.

The package might be a standard commercial program or a customized package designed specifically for the purchaser. Companies that develop software for sale are called **software vendors**. A firm that enhances a commercial package by adding custom features and configuring it for a particular industry is called a **value-added reseller** (**VAR**).

Software packages are available for every type of business activity. A software package that can be used by many different types of organizations is called a **horizontal application**. An accounting package is a good example of a horizontal application because it can be utilized by many different businesses, or separate divisions that exist in large, diversified companies.

ON THE WEB

To learn more about value-added resellers, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Value-Added Resellers link.

FIGURE 7-10 Hotel chains require vertical applications to support reservation systems and information needs that are unique to the hotel industry.

In contrast, a software package developed to handle information requirements for a specific type of business is called a **vertical application**. For example, organizations with special system requirements include colleges, banks, hospitals, insurance companies, construction companies, real estate firms, and airlines. A hotel chain might require a vertical application for its guest reservation system, as shown in Figure 7-10, and use horizontal applications for basic business needs, such as payroll processing and accounts payable.

Of the in-house software acquisition options — developing a system, buying a software package, or customizing a software package — each has advantages, disadvantages, and cost considerations, as shown in Figure 7-11. These software acquisition options are described in detail in the following sections.

Developing Software In-House

With an enormous variety of software packages available to handle horizontal and vertical business operations, why would a firm choose to develop its own software? Typically, companies choose in-house development to satisfy unique business requirements, to minimize changes in business procedures and policies, to meet constraints of existing systems and existing technology, and to develop internal resources and capabilities.

SATISFY UNIQUE BUSINESS REQUIREMENTS Companies often decide to develop software in-house because no commercially available software package can meet their unique business requirements. A college, for example, needs a course scheduling system based on curriculum requirements, student demand, classroom space, and available instructors. A package delivery company needs a system to identify the best combination of routes and loading patterns for the company's fleet of delivery trucks. If existing software packages cannot handle those requirements, then in-house developed software might be the only choice.

REASONS FOR IN-HOUSE DEVELOPMENT	REASONS FOR PURCHASING A SOFTWARE PACKAGE
Satisfy unique business requirements	Lower costs
Minimize changes in business procedures and policies	Requires less time to implement
Meet constraints of existing systems	Proven reliability and performance benchmarks
Meet constraints of existing technology	Requires less technical development staff
Develop internal resources and capabilities	Future upgrades provided by the vendor
Satisfy unique security requirements	Obtain input from other companies

FIGURE 7-11 Companies consider various factors when comparing in-house development with the purchase of a software package.

MINIMIZE CHANGES IN BUSINESS PROCEDURES AND POLICIES A company also might choose to develop its own software if available packages will require changes in current business operations or processes. Installing a new software package almost always requires some degree of change in how a company does business; however, if the installation of a purchased package will be too disruptive, the organization might decide to develop its own software instead.

MEET CONSTRAINTS OF EXISTING SYSTEMS Any new software installed must work with existing systems. For example, if a new budgeting system must interface with an existing accounting system, finding a software package that works correctly with the existing accounting system might prove difficult. If so, a company could develop its own software to ensure that the new system will interface with the old system.

MEET CONSTRAINTS OF EXISTING TECHNOLOGY Another reason to develop software in-house is that the new system must work with existing hardware and legacy systems. That could require a custom design not commercially available. Some companies have older microcomputer workstations that cannot handle graphics-intensive software or high-speed Internet access. In that situation, the company either must upgrade the environment or must develop in-house software that can operate within the constraints of the existing hardware. As a systems analyst, you addressed the issue of technical feasibility during the preliminary investigation. Now, in the systems analysis phase, you must examine the advantages and disadvantages of in-house software development to decide whether it is justifiable.

DEVELOP INTERNAL RESOURCES AND CAPABILITIES By designing a system in-house, companies can develop and train an IT staff that understands the organization's business functions and information support needs. Many firms feel that in-house IT resources and capabilities provide a competitive advantage because an in-house team can respond quickly when business problems or opportunities arise. For example, if a company lacks internal resources, it must depend on an outside firm for vital business support. Also, outsourcing options might be attractive, but a series of short-term solutions would not necessarily translate into lower TCO over the long term. Top managers often feel more comfortable with an internal IT team to provide overall guidance and long-term stability.

Purchasing a Software Package

If a company decides not to outsource, a commercially available software package might be an attractive alternative to developing its own software. Advantages of purchasing a software package over developing software in-house include lower costs, less time to implement a system, proven reliability and performance benchmarks, less technical development staff, future upgrades that are provided by the vendor, and the ability to obtain input from other companies who already have implemented the software.

LOWER COSTS Because many companies use software packages, software vendors spread the development costs over many customers. Compared with software developed in-house, a software package almost always is less expensive, particularly in terms of initial investment.

REQUIRES LESS TIME TO IMPLEMENT When you purchase a package, it already has been designed, programmed, tested, and documented. The in-house time normally spent on those tasks, therefore, is eliminated. Of course, you still must install the software and integrate it into your systems environment, which can take a significant amount of time.

PROVEN RELIABILITY AND PERFORMANCE BENCHMARKS If the package has been on the market for any length of time, any major problems probably have been detected already and corrected by the vendor. If the product is popular, it almost certainly has been rated and evaluated by independent reviewers.

REQUIRES LESS TECHNICAL DEVELOPMENT STAFF Companies that use commercial software packages often are able to reduce the number of programmers and systems analysts on the IT staff. Using commercial software also means that the IT staff can concentrate on systems whose requirements cannot be satisfied by software packages.

FUTURE UPGRADES PROVIDED BY THE VENDOR Software vendors regularly upgrade software packages by adding improvements and enhancements to create a new version or release. A new release of a software package, for example, can include drivers to support a new laser printer or a new type of data storage technology. In many cases, the vendor receives input and suggestions from current users when planning future upgrades.

INPUT FROM OTHER COMPANIES Using a commercial software package means that you can contact users in other companies to obtain their input and impressions. You might be able to try the package or make a site visit to observe the system in operation before making a final decision.

Customizing a Software Package

If the standard version of a software product does not satisfy a company's requirements, the firm can consider adapting the package to meet its needs. Three ways to customize a software package are:

1. You can purchase a basic package that vendors will customize to suit your needs. Many vendors offer basic packages in a standard version with add-on components that are configured individually. A vendor offers options when the standard application will not satisfy all customers. A human resources information system is a typical example, because each company handles employee compensation and benefits differently. If you need assistance in making a determination, firms such as Ideas International offer services to help you select and configure a system, as shown in Figure 7-12.

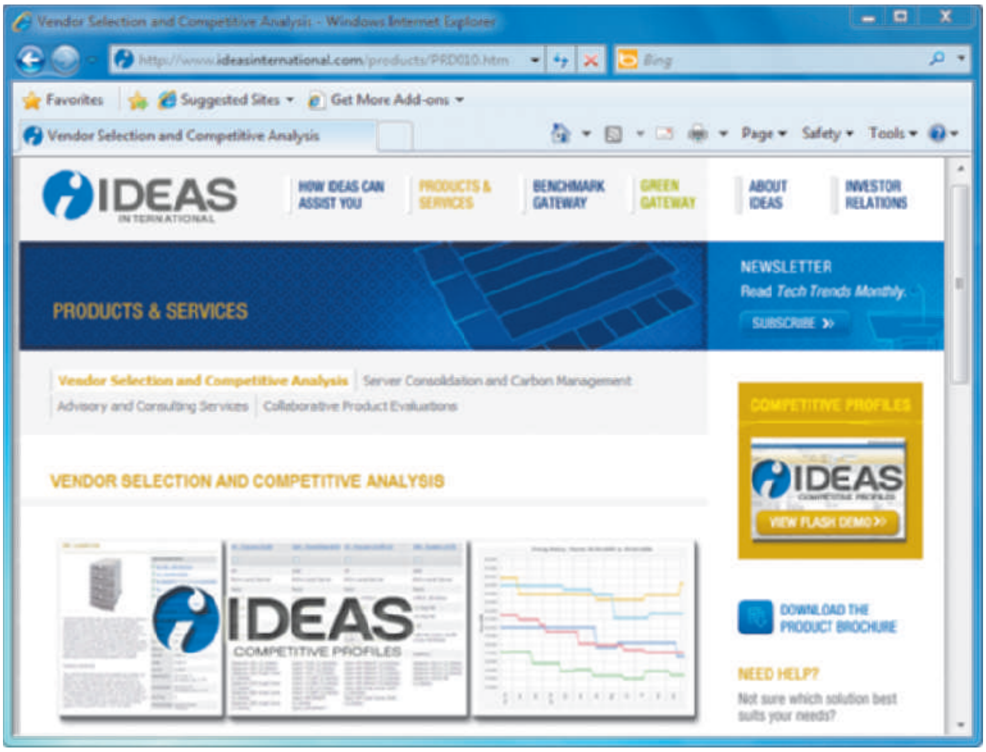


FIGURE 7-12 Firms such as Ideas International offer services to help customers select and configure a system.

2. You can negotiate directly with the software vendor to make enhancements to meet your needs by paying for the changes.
3. You can purchase the package and make your own modifications, if this is permissible under the terms of the software license. A disadvantage of this approach is that systems analysts and programmers might be unfamiliar with the software and will need time to learn the package and make the modifications correctly.

Additionally, some advantages of purchasing a standard package disappear if the product must be customized. If the vendor does the customizing, the modified package probably will cost more and take longer to obtain. Another issue is future support: Although vendors regularly upgrade their standard software packages, they might not upgrade a customized version. In addition, if the modifications are done by the company purchasing the software, when a new release of the package becomes available, the company might have to modify the new version.

Creating User Applications

Business requirements sometimes can be fulfilled by a user application, rather than a formal information system or commercial package. User applications are examples of user productivity systems, which were discussed in Chapter 1.

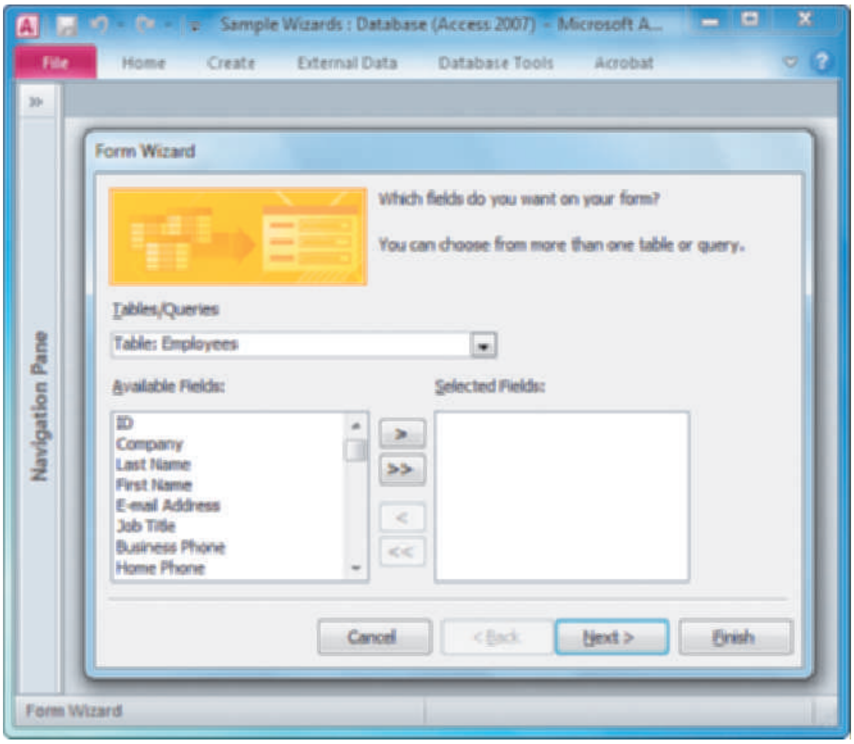
A **user application** utilizes standard business software, such as Microsoft Word or Microsoft Excel, which has been configured in a specific manner to enhance user productivity. For example, to help a sales rep respond rapidly to customer price requests, an IT support person can set up a form letter with links to a spreadsheet that calculates incentives and discounts. In addition to configuring the software, the IT staff can create a **user interface**, which includes screens, commands, controls, and features that enable users to interact more effectively with the application. User interface design is described in Chapter 8.

In some situations, user applications offer a simple, low-cost solution. Most IT departments have a backlog of projects, and IT solutions for individuals or small groups do not always receive a high priority. At the same time, application software is more powerful, flexible, and user-friendly than ever. Companies such as Microsoft and Corel offer software suites and integrated applications that can exchange data with programs that include tutorials, wizards, and Help features to guide less experienced users who know what they need to do but do not know how to make it happen.

Many companies empower lower-level employees by providing more access to data and more powerful data management tools. The main objective is to allow lower-level employees more access to the data they require to perform their jobs, with no intervention from the IT department. This can be accomplished by creating effective user interfaces for company-wide applications such as accounting, inventory, and sales systems. Another technique is to customize standard productivity software, such as Microsoft Word or Microsoft Excel, to create user applications. In either case, empowerment makes the IT department more productive because it can spend less time responding to the daily concerns and data needs of users and more time on high-impact systems development projects that support strategic business goals.

Empowerment reduces costs and makes good business sense, but companies that adopt this approach must provide the technical support that empowered users require. In most large and medium-sized companies, a **help desk**, or **information center** (IC), within the IT department is responsible for providing user support. The IC staff offers services such as hotline assistance, training, and guidance to users who need technical help.

Once they learn an application, many users can perform tasks that once required a programmer. Some user applications have powerful **screen generators** and **report generators** that allow users to design their own data entry forms and reports. For



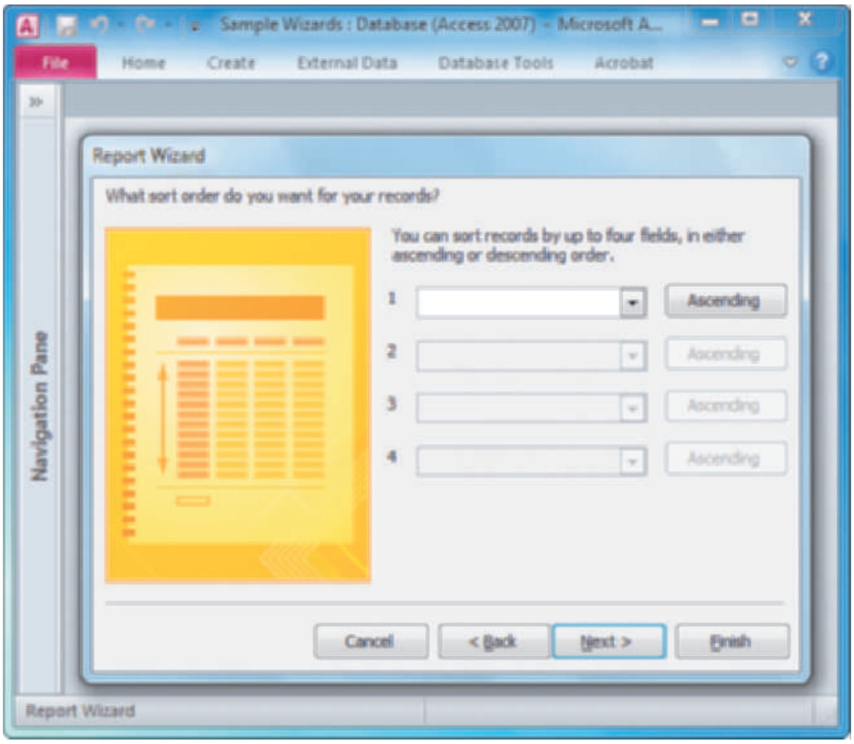


FIGURE 7-13 Microsoft Access includes Form Wizard and Report Wizard tools that ask a series of questions, and then create the form or report.

example, as shown in Figure 7-13, Microsoft Access includes a Form Wizard and a Report Wizard, which are menu-driven tools that can create screen forms and reports. These design tools allow users to design specific input and output views that meet their operational needs — with little or no assistance required from the IT staff.

Users typically require spreadsheets, database management programs, and other software packages to meet their information needs. If user applications access corporate data, you must provide appropriate controls to ensure data security and integrity. For example, some files should be hidden totally from view; others should have **read-only properties** so users can view, but not change, the data.

Role of the Systems Analyst

At some point in the systems development process, the company must decide whether to use an outsourcing option, develop software in-house, acquire a software package, develop user applications, or select some combination of these solutions. The decision will affect the remaining SDLC phases and your involvement as a systems analyst. The decision to develop software in-house, for example, will require more participation from the systems analyst than outsourcing or choosing a commercial package. Management usually makes a determination after receiving written recommendations from the IT staff and a formal presentation, which is described later in this chapter.

Even a single system can use a mix of software alternatives. For example, a company might purchase a standard software package to process its payroll, and then develop its own software to handle the interface between the payroll package and the company's in-house manufacturing cost analysis system.

The evaluation and selection of alternatives is not a simple process. The objective is to obtain the product with the lowest total cost of ownership, but actual cost and performance can be difficult to forecast. With a large number of choices, how do you select the best alternative?

When selecting hardware and software, systems analysts often work as an **evaluation and selection team.** A team approach ensures that critical factors are not overlooked and that a sound choice is made. The evaluation and selection team also must include users, who will participate in the selection process and feel a sense of ownership in the new system.

The primary objective of the evaluation and selection team is to eliminate system alternatives that will not meet requirements, rank the alternatives that are feasible, and present the viable alternatives to management for a final decision. The process begins with a careful study of the costs and benefits of each alternative, as explained in the following section.

ANALYZING COST AND BENEFITS

In Chapter 2, you learned that economic feasibility is one of the four feasibility measurements that are made during the preliminary investigation of a systems request. Now, at the end of the systems analysis phase of the SDLC, you must apply financial analysis tools and techniques to evaluate development strategies and decide how the project will move forward. Part C of the Systems Analyst's Toolkit describes three popular tools, which are payback analysis, return on investment (ROI), and net present value (NPV). These tools, and others, can be used to determine total cost of ownership (TCO), which was described in Chapter 4. At this stage, you will identify specific systems development strategies and choose a course of action. For example, a company might find that its total cost of ownership will be higher if it develops a system in-house, compared with outsourcing the project or using an ASP.

An accurate forecast of TCO is critical, because nearly 80% of total costs occur *after* the purchase of the hardware and software, according to Gartner, Inc. An IT department can develop its own TCO estimates, or use TCO calculation tools offered by vendors. For example, as shown in Figure 7-14 on the next page, HP and Oracle offer an online TCO calculator that includes a questionnaire and a graphical display of results.

ON THE WEB

To learn more about financial analysis tools, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Financial Analysis Tools link.

VIDEO LEARNING SESSIONS

To learn more about financial analysis tools, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and navigate to the Video Learning Sessions for this book. These sessions can help you understand key concepts, practice your skills, and check your work.

Financial Analysis Tools

Part C of the Systems Analyst's Toolkit explains how to use three main cost analysis tools: payback analysis, return on investment (ROI), and net present value (NPV). **Payback analysis** determines how long it takes an information system to pay for itself through reduced costs and increased benefits. **Return on investment (ROI)** is a percentage rate that compares the total net benefits (the return) received from a project to the total costs (the investment) of the project. The **net present value (NPV)** of a project is the total value of the benefits minus the total value of the costs, with both costs and benefits adjusted to reflect the point in time at which they occur.

CASE IN POINT 7.2: STERLING ASSOCIATES

Joan Sterling is CEO and principal stockholder of Sterling Associates, which specializes in advising clients on IT projects and information systems development. Joan is creating a brochure for prospective new clients. She wants you to develop a section that describes payback analysis, ROI, and NPV in simple terms, and mentions the pros and cons of each financial analysis tool. She suggested that you start by viewing the Video Learning Sessions on financial analysis tools, and reviewing the material in Part C of the Systems Analyst's Toolkit.

FIGURE 7-14 HP and Oracle offer an online TCO calculator.

Cost-Benefit Analysis Checklist

Companies use all three financial analysis tools to evaluate various development strategies. The best way to apply the tools is to develop a cost-benefit checklist with the following steps:

- List each development strategy being considered.
- Identify all costs and benefits for each alternative. Be sure to indicate when costs will be incurred and benefits realized.
- Consider future growth and the need for scalability.
- Include support costs for hardware and software.
- Analyze various software licensing options, including fixed fees and formulas based on the number of users or transactions.
- Apply the financial analysis tools to each alternative.
- Study the results and prepare a report to management.

THE SOFTWARE ACQUISITION PROCESS

Although each situation is different, the following section describes a typical example of the issues and tasks involved in software acquisition.

Step 1: Evaluate the Information System Requirements

Based on your analysis of the system requirements, you must identify the system's key features; consider network and Web-related issues; estimate volume and future growth; specify any hardware, software, or personnel constraints; and prepare a request for proposal or quotation.

IDENTIFY KEY FEATURES Whether you are considering in-house development or outsourcing options, you must develop a clear, detailed list of features that can serve as an overall specification for the system. Using the data you gathered during fact-finding, which was discussed in Chapter 4, you must list all system requirements and critical features. This information will be included in the system requirements document, which is the end product of the SDLC systems analysis phase.

CONSIDER NETWORK AND WEB-RELATED ISSUES As you evaluate the system requirements, you must consider network and Web-related issues. You must decide whether the system will run on a network, the Internet, or a company intranet, and build these requirements into the design. Also, you must determine whether the system will exchange data with vendor or customer systems, and ensure that the system will be compatible.

ESTIMATE VOLUME AND FUTURE GROWTH You need to know the current volume of transactions and forecast future growth. Figure 7-15 shows volume estimates for an order processing system. In addition to current levels, the figure displays two forecasts, one based on the existing order processing procedures and another that assumes a new Web site is operational.

Online Order Processing System Estimated Activity During Next 12-Month Period

	CURRENT LEVEL	FUTURE GROWTH (based on existing procedures)	FUTURE GROWTH (assuming new Web site is operational)
Customers	36,500	40,150	63,875
Daily Orders	1,435	1,579	2,811
Daily Order Lines	7,715	7,893	12,556
Sales Reps	29	32	12
Order Processing Support Staff	2	4	3
Products	600	650	900

FIGURE 7-15 Volume estimate for an order processing system showing current activity levels and two forecasts: one based on the existing order processing procedures and another that assumes a new Web site is operational.

A comparison of the two forecasts shows that the Web site will generate more new customers, process almost 80% more orders, and substantially reduce the need for sales reps and support staff. If you are considering in-house development, you must make sure that your software and hardware can handle future transaction volumes and data storage requirements. Conversely, if you are considering outsourcing, volume and usage data is essential to analyze ASP fee structures and develop cost estimates for outsourcing options.

SPECIFY HARDWARE, SOFTWARE, OR PERSONNEL CONSTRAINTS You must determine whether existing hardware, software, or personnel issues will affect the acquisition decision. For example, if the firm has a large number of legacy systems or if an ERP strategy has been adopted, these factors will have an impact on the decision. Also, you must investigate the company's policy regarding outsourcing IT functions, and whether outsourcing is part of a long-term strategy. With regard to personnel issues, you must define in-house staffing requirements to develop, acquire, implement, and maintain the system — and determine whether the company is willing to commit to those staffing levels versus an outsourcing option.

PREPARE A REQUEST FOR PROPOSAL OR QUOTATION To obtain the information you need to make a decision, you should prepare a request for proposal or a request for quotation. The two documents are similar but used in different situations, based on whether or not you have selected a specific software product.

A **request for proposal** (**RFP**) is a document that describes your company, lists the IT services or products you need, and specifies the features you require. An RFP helps ensure that your organization's business needs will be met. An RFP also spells out the service and support levels you require. Based on the RFP, vendors can decide if they have a product that will meet your needs. RFPs vary in size and complexity, just like the systems they describe. An RFP for a large system can contain dozens of pages with unique requirements and features. You can use an RFP to designate some features as essential and others as desirable. An RFP also requests specific pricing and payment terms.

Figure 7-16 shows an example of a ready-made RFP template offered by Infotivity Technologies. Notice that the vendor can choose from a range of responses, and also add comments. Figure 7-17 shows the RFP Evaluation Centers site. This organization offers samples of RFP templates, cover letters, and other resources.

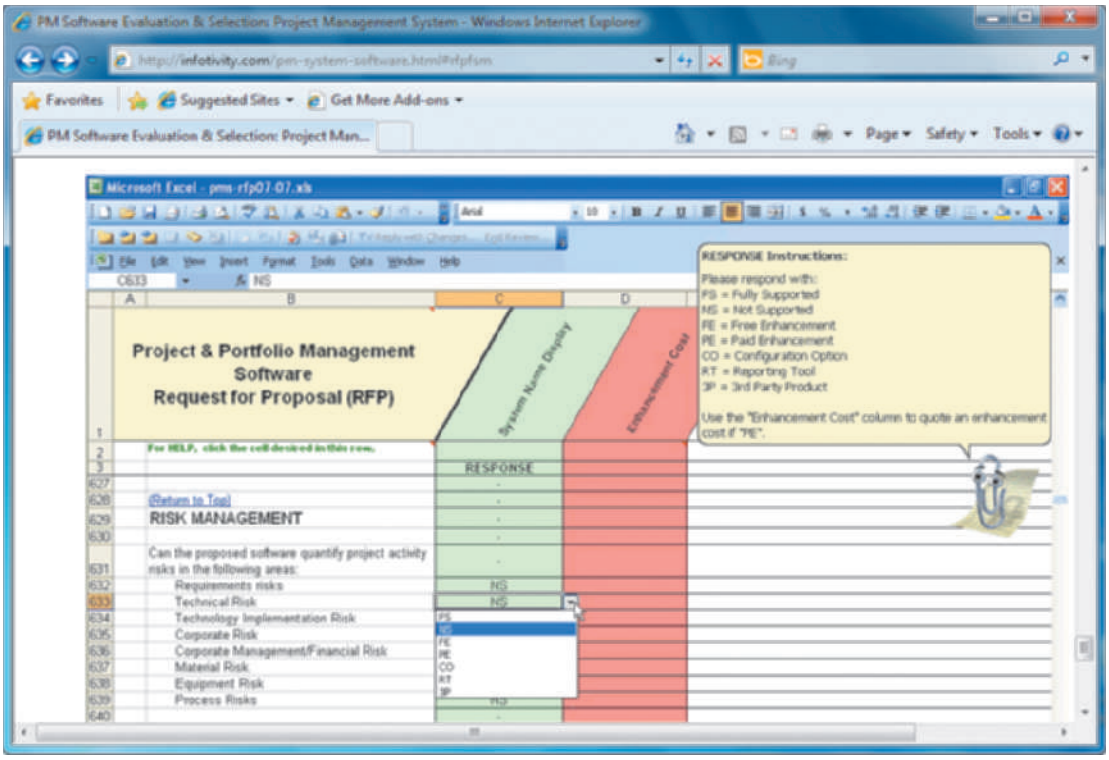


FIGURE 7-16 Infotivity Technologies offers a ready-made RFP template that allows a wide range of reponses and comments.

When you evaluate several responses to an RFP, you might find it helpful to use an evaluation model. An **evaluation model** is a technique that uses a common yardstick to measure and compare vendor ratings.

Figure 7-18 on the next page shows two evaluation models for a network project. The evaluation model at the top of the figure simply lists the key elements and each vendor's score. The model at the bottom of the figure adds a weight factor. In this example, each element receives a rating based on its relative importance. Although the initial scores are the same in both models, notice that vendor A has the highest point total in the top example, but vendor C emerges as the best in the weighted model.

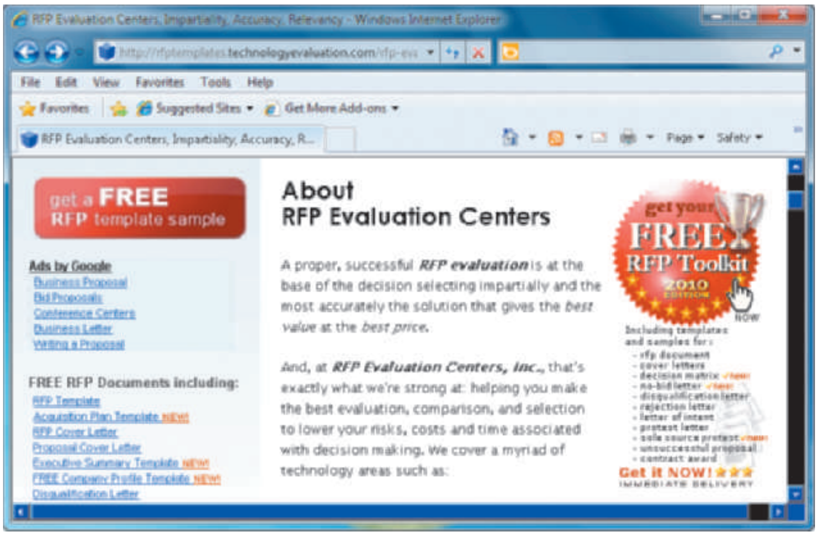


FIGURE 7-17 The RFP Evaluation Centers site offers samples of RFP templates, cover letters, and other resources.

Unweighted Evaluation Model for a Network Project

Instructions: Rate each vendor on a scale from 1(low) to 10 (high), then add vendor scores to calculate total points.

	VENDOR A	VENDOR B	VENDOR C
Price	6	5	9
Completion Date	2	5	8
Layout/Design	8	8	5
References	10	6	3
TOTAL POINTS	26	24	25

Weighted Evaluation Model for a Network Project

Instructions: Rate each vendor on a scale from 1(low) to 10 (high), then multiply the vendor's score by the weight factor. Add vendor scores to calculate total points.

	WEIGHT FACTOR	VENDOR A	VENDOR B	VENDOR C
Price	25	6 * 25 = 150	5 * 25 = 125	9 * 25 = 225
Completion Date	25	2 * 25 = 50	5 * 25 = 125	8 * 25 = 200
Layout/Design	35	8 * 35 = 280	8 * 35 = 280	5 * 35 = 175
References	15	10 * 15 = 150	6 * 15 = 90	3 * 15 = 45
TOTAL POINTS	100	630	620	645

FIGURE 7-18 The three vendors have the same initial ratings, but the two evaluation models produce different results. In the unweighted model at the top of the figure, vendor A has the highest total points. However, after applying weight factors, vendor C is the winner, as shown in the model at the bottom of the figure.

Evaluation models can be used throughout the SDLC, and you will find them a valuable tool. You can use a spreadsheet program to build an evaluation model, experiment with different weighting factors, and graph the results.

A **request for quotation (RFQ)** is more specific than an RFP. When you use an RFQ, you already know the specific product or service you want and you need to obtain price quotations or bids. RFQs can involve outright purchase or a variety of leasing options and can include maintenance or technical support terms. Many vendors provide convenient RFQ forms on their Web sites, as shown in Figure 7-19. RFPs and RFQs have the same objective: to obtain vendor replies that are clear, comparable, and responsive so you can make a well-informed selection decision.

In today's fast-paced IT marketplace, traditional methods for obtaining RFPs often are too slow. The Web site shown in Figure 7-20 offers an online meeting place where customers can post RFPs and vendors can reply with solutions and bids.

Step 2: Identify Potential Vendors or Outsourcing Options

The next step is to identify potential vendors or outsourcing providers. The Internet is a primary marketplace for all IT products and services, and you can find descriptive information on the Web about all major products and acquisition alternatives.

If you need to locate vertical applications for specific industries, you can research industry trade journals or Web sites to find reviews for industry-specific software. Industry trade groups often can direct you to companies that offer specific software solutions.

Another approach is to work with a consulting firm. Many IT consultants offer specialized services that help companies select software packages. A major advantage of using a consultant is that you can tap into broad experience that is difficult for any one company to acquire. Consultants can be located by contacting professional organizations or industry sources, or simply by searching the Internet. Using a consultant involves additional expense but can prevent even more costly mistakes.

Another valuable resource is the Internet bulletin board system that contains thousands of forums, called **newsgroups**, that cover every imaginable topic. Newsgroups are excellent sources of information and good places to exchange ideas with other analysts and IT professionals. You can search the Web for newsgroups that interest you, or you can visit the sites of specific companies, such as Microsoft, that provide a valuable source of information for IT professionals, including blogs, technical chats, newsgroups, Webcasts, and other resources, as shown in Figure 7-21 on the next page.

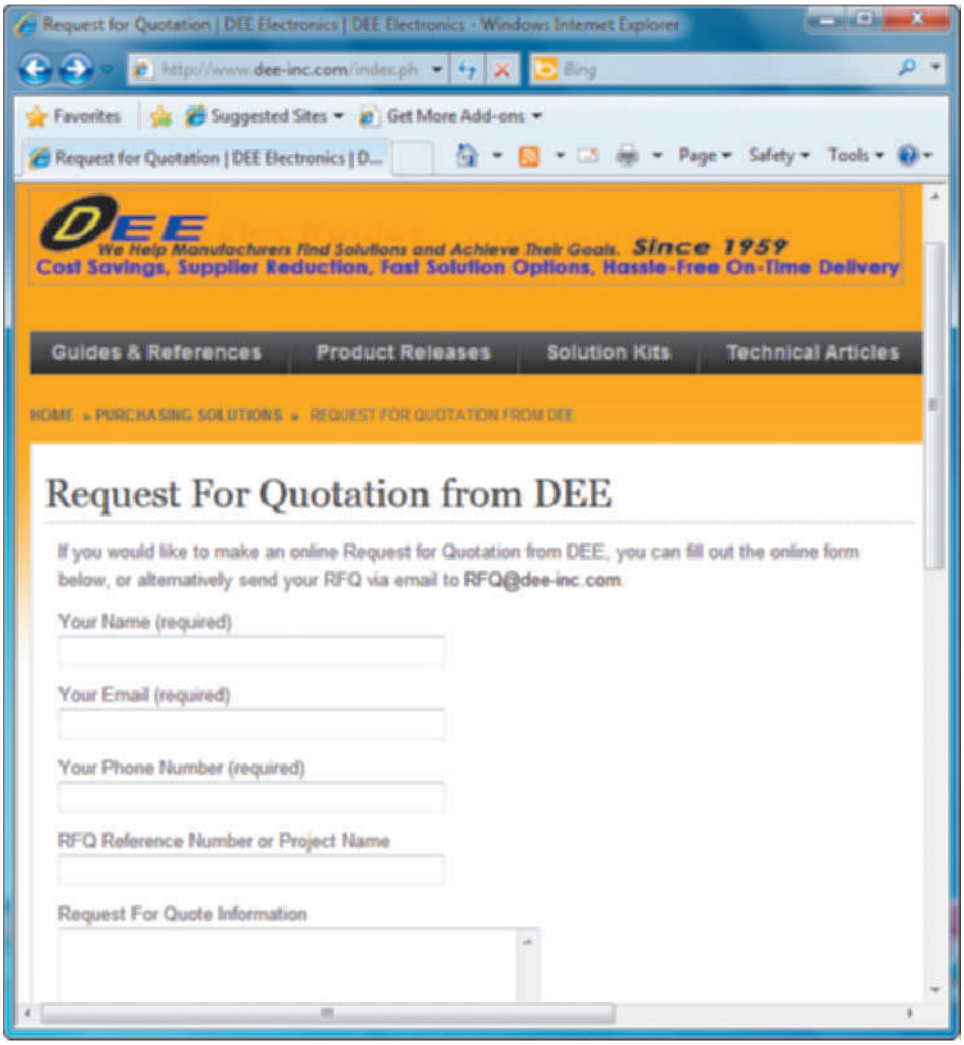


FIGURE 7-19 Many vendors provide convenient RFQ forms on their Web sites, as shown in this example.

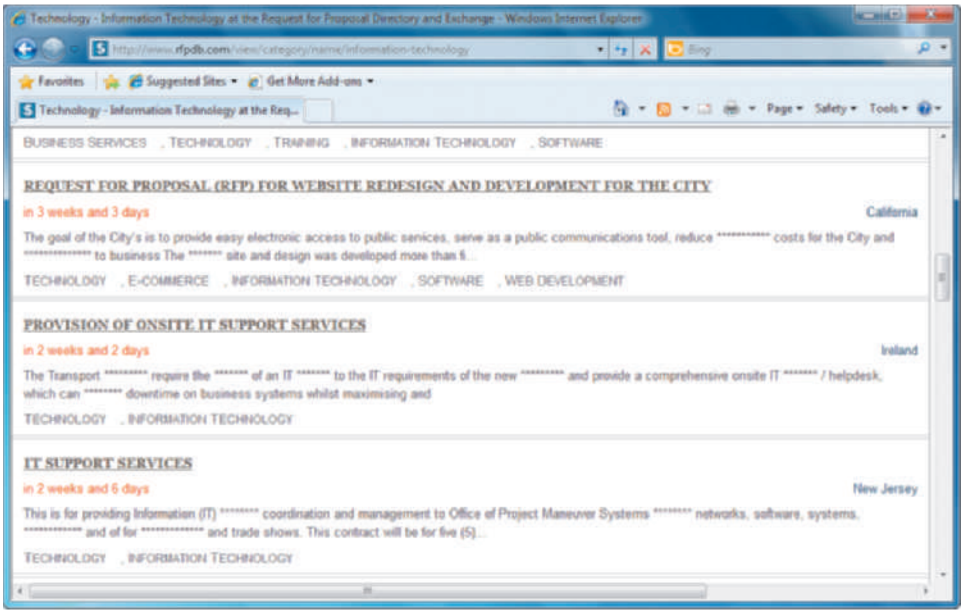


FIGURE 7-20 The rfpDB site offers an online meeting place where customers post RFPs and vendors can respond.

FIGURE 7-21 Microsoft Communities is an excellent resource for IT professionals.

Step 3: Evaluate the Alternatives

After identifying the alternatives, you must select the one that best fits the company's needs. You should obtain information about the options from as many sources as possible, including vendor presentations and literature, product documentation, trade publications, and companies that perform software testing and evaluation. To learn more about particular software packages, search the Internet using keywords that describe the application. Web sites maintained by consultants and software publishers often include product references and links to vendors. As part of the evaluation process, you should try to obtain information from existing users, test the application, and benchmark the package.

EXISTING USERS You can contact existing users to obtain feedback and learn about their experiences. For large-scale software packages, ASPs and vendors typically supply user references. User references are important because you need to know whether the software package has worked well for companies like yours. Be aware that some vendors limit their reference lists to satisfied clients, so you can expect mostly positive feedback from those firms.

APPLICATION TESTING If a software package is one of the options, find out if it is possible for users in your organization to try the product. For horizontal applications or small systems, using a demo copy to enter a few sample transactions could be an acceptable test. For vertical applications or large systems, a team of IT staff and users might need several days or weeks to perform tests.

ON THE WEB

To learn more about benchmark tests, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Benchmark Tests link.

BENCHMARKING To determine whether a package can handle a certain transaction volume efficiently, you can perform a benchmark test. A **benchmark** measures the time a package takes to process a certain number of transactions. For example, a benchmark test can measure the time needed to post 1,000 sales transactions.

If you use benchmarks, remember that a benchmark test is conducted in a controlled environment, which might not resemble the actual day-to-day situation at your company. Although benchmarking cannot predict your specific results, benchmark testing is a good way to measure relative performance of two or more competing products in a standard environment.

Many IT publications publish regular reviews of individual packages, including benchmark tests, and often have annual surveys covering various categories of software. Some of the publications shown in Figure 7-22 also offer online versions and additional Web-based features, search capability, and IT links.

You also can obtain information from independent firms that benchmark various software packages and sell comparative analyses of the results, as shown in Figure 7-23 on the next page. The Transaction Processing Performance Council (TPC) is an example of a non-profit organization that publishes standards and reports for its members and the general public, while InfoSizing is an IT consulting firm that offers analysis of performance benchmarks.

Finally, you should match each package against the RFP features and rank the choices. If some features are more important than others, give them a higher weight using an evaluation model similar to the one shown in Figure 7-18 on page 304.

FIGURE 7-22 Many IT publications test and review software packages. Although an enormous amount of information is available on the Web, many IT professionals enjoy reading traditional magazines like these.

Step 4: Perform Cost-Benefit Analysis

Review the suggestions in this chapter and in Part C of the Systems Analyst's Toolkit, and develop a spreadsheet to identify and calculate TCO for each option you are considering. Be sure to include all costs, using the volume forecasts you prepared. If you are considering outsourcing options, carefully study the alternative fee structure models described earlier. If possible, prepare charts to show the results graphically, and build in what-if capability so you can gauge the impact if one or more variables change.

If you are considering a software package, be sure to consider acquisition options. When you purchase software, what you are buying is a **software license** that gives you the right to use the software under certain terms and conditions. For example, the license could allow you to use the software only on a single computer, a specified number of computers, a network, or an entire site, depending on the terms of the agreement. Other license restrictions could prohibit you from making the software available to others or modifying the program. For desktop applications, software license terms and conditions usually cannot be modified. For large-scale systems, license agreement terms often can be negotiated.

Also consider user support issues, which can account for a significant part of TCO. If you select an outsourcing alternative, the arrangement probably will include certain technical support and maintenance. If you choose in-house development, you must consider the cost of providing these services on your own. If you purchase a software package, consider a supplemental **maintenance agreement**, which offers additional support and assistance from the vendor. The agreement might provide full support for a period of time or list specific charges for particular services. Some software packages provide free technical support for a period of time. Afterward, support is offered with a charge per occurrence, or per minute or hour of technical support time. Some software vendors contact registered owners whenever a new release is available and usually offer the new release at a reduced price.

TOOLKIT TIME

The Financial Analysis tools in Part C of the Systems Analyst's Toolkit can help you analyze project costs, benefits, and economic feasibility. To learn more about these tools, turn to Part C of the four-part Toolkit that follows Chapter 12.

Step 5: Prepare a Recommendation

You should prepare a recommendation that evaluates and describes the alternatives, together with the costs, benefits, advantages, and disadvantages of each option. At this point, you may be required to submit a formal system requirements document and deliver a presentation. You should review the suggestions for presenting written

FIGURE 7-23 The Transaction Processing Performance Council is a non-profit organization that publishes standards and reports for its members and the general public, while InfoSizing is an IT consulting firm that offers analysis of performance benchmarks.

proposals and oral presentations in Part A of the Systems Analyst's Toolkit. Additional suggestions about preparing the system requirements document and the management presentation are contained in the following section.

Step 6: Implement the Solution

Implementation tasks will depend on the solution selected. In-house options will require more time and effort than outsourcing alternatives. For large systems or network installations, the process can require considerable time and effort. Your installation strategy should be planned well in advance, especially if any disruption of normal business operations is expected. If the software package is customized, then the task will be more complex and difficult.

Before the new software becomes operational, you must complete all implementation steps, including loading, configuring, and testing the software; training users; and converting data files to the new system's format. Chapter 11 discusses implementation strategies and techniques in more detail.

CASE IN POINT 7.3: DOUG'S SPORTING GOODS

Doug's Sporting Goods sells hiking and camping supplies. The company has grown considerably in the last two years. Doug Sawyer, the company's founder and president, wants to develop a customer order entry system and hired your IT consulting firm to advise him about software alternatives. Doug is leaning toward in-house development because he does not want to depend on outside vendors and suppliers for technical support and upgrades. Doug also says that he is not interested in selling on the Web, but that could change in the future.

Doug wants to meet with you tomorrow to make a decision. What will you say to Doug at the meeting?

Completion of Systems Analysis Tasks

To complete the systems analysis phase, you must prepare the system requirements document and your presentation to management.

System Requirements Document

The **system requirements document**, or **software requirements specification**, contains the requirements for the new system, describes the alternatives that were considered, and makes a specific recommendation to management. This important document is the starting point for measuring the performance, accuracy, and completeness of the finished system before entering the systems design phase.

The system requirements document is like a contract that identifies what the system developers must deliver to users. Recall that system requirements are identified during the fact-finding process, and a system requirements checklist is created at that time. Various examples of system requirements are listed on pages 153–155 in Chapter 4. You should write the system requirements document in language that users can understand so they can offer input, suggest improvements, and approve the final version.

Because the system requirements document can be lengthy, you should format and organize it so it is easy to read and use. The system requirements document should include a cover page and a detailed table of contents. You also can add an index and a glossary of terms to make the document easier to use. The content of the system requirements document will depend on the company and the complexity of the system.

TOOLKIT TIME

The Communication Tools in Part A of the Systems Analyst's Toolkit can help you develop better documents, reports, and presentations. To learn more about these tools, turn to Part A of the four-part Toolkit that follows Chapter 12.

Presentation to Management

The presentation to management at the end of the systems analysis phase is one of the most critical milestones in the systems development process. At this point, managers make key decisions that affect the future development of the system.

Prior to the management presentation, you might give two other presentations: one to the principal individuals in the IT department to keep them posted, and another presentation to users to answer their questions and invite feedback. The system requirements document is the basis for all three presentations, and you should distribute the document (or a summary) in advance so the recipients can review it.

When preparing your presentation, you should review the suggestions in Part A of the Systems Analyst's Toolkit, which will help you design and deliver a successful presentation. If you plan a slide presentation, you should review the Toolkit guidelines for effective presentations. In addition to the techniques found in the Toolkit, also keep the following suggestions in mind:

- Begin your presentation with a brief overview of the purpose and primary objectives of the system project, the objectives of this presentation, and what decisions need to be made.
- Summarize the primary viable alternatives. For each alternative, describe the costs, advantages, and disadvantages.
- Explain why the evaluation and selection team chose the recommended alternative.
- Allow time for discussion and for questions and answers.
- Obtain a final decision from management or agree on a timetable for the next step in the process.

The object of the management presentation is to obtain approval for the development of the system and to gain management's full support, including necessary financial resources. Management probably will choose one of five alternatives: develop an in-house system, modify a current system, purchase or customize a software package, perform additional systems analysis work, or stop all further work. Depending on their decision, your next task as a systems analyst will be one of the following:

1. Implement an outsourcing alternative. If outsourcing is selected, you will work with representatives of the service provider to achieve a smooth transition to the new environment.
2. Develop an in-house system. Begin systems design tasks, as described in Chapters 8, 9, and 10.
3. Purchase or customize a software package. Negotiate the purchase terms with the software vendor for management approval. Then, if the package will be used without modification, you can begin planning the systems implementation phase. If you must make modifications to the package, your next step is to start the systems design phase. If the vendor will make the modifications, then your next step is to start planning the testing and documentation of the modifications as part of the systems implementation phase, which is described in Chapter 11.
4. Perform additional systems analysis work. Management might want you to investigate certain alternatives further, explore alternatives not examined, develop a prototype, reduce the project scope because of cost constraints, or expand the project scope based on new developments. If necessary, you will perform the additional work and schedule a follow-up presentation.
5. Stop all further work. The decision might be based on your recommendation, a shift in priorities or costs, or for other reasons. Whatever the reason, if that is management's decision, then you have no additional tasks for the project other than to file all your research in a logical location so it can be retrieved if the project is reopened in the future.

After the presentation and management decision, you will begin a transition to the systems design phase of the SDLC. If you are developing an in-house system or modifying a package, you will build a model of the proposed system and start designing the system's output, input, files, and data structures. The following sections describe several tools and techniques that can assist you in that process, including prototyping, CASE tools, and alternative graphical tools.

TRANSITION TO SYSTEMS DESIGN

In a traditional SDLC environment, systems design usually started when the systems analysis phase was done. Using the system requirements specification as a blueprint, developers transformed the logical design into a working model that could be tested, reviewed by users, and implemented. Today, the process is much more dynamic. In general, systems development is faster, more flexible, and more user-oriented. The introduction of adaptive methods such as agile development and extreme programming has changed the landscape significantly. Depending on the project, system developers often blend traditional and cutting-edge development methods, because what works in one situation might not work in another.

This textbook discusses systems analysis in Chapters 4, 5, 6, and 7, and systems design in Chapters 8, 9, and 10. However, in a typical IT workplace, all these tasks — and more — are integrated and managed together.

This section discusses preparation for systems design and the relationship between logical and physical design. The chapter concludes with a description of systems design guidelines, prototyping methods, and software development trends.

Preparing for Systems Design

Regardless of the development method, systems design requires accurate documentation. Traditionally, a system requirements document provided detailed specifications for output, input, data, processes, and whatever else was needed. Although agile methods do not require a particular form of documentation, a successful development team must understand and record user requirements as they evolve during the project.

Logical and Physical Design

A **logical design** defines *what* must take place, not *how* it will be accomplished. Logical designs do not address the actual methods of implementation. In contrast, a **physical design** is like a set of blueprints for the actual construction of a building. Typically, a physical design describes the actual processes of entering, verifying, and storing data; the physical layout of data files and sorting procedures, the format of reports, and so on. Because logical and physical designs are related so closely, good systems design is impossible without careful, accurate systems analysis. For example, you might return to fact-finding if you discover that you overlooked an important issue, if users have significant new needs, or if legal or governmental requirements change.

SYSTEMS DESIGN GUIDELINES

Because the components of a system are interdependent, the design phase is not a series of clearly defined steps. Although you might start in one area, you could find yourself working with several different elements at the same time. For example, a decision to change a report format might require changes in data design or input screens. Your design checklist will include the user interface, input and output procedures, data design, and system architecture. As shown in Figure 7-24, the final steps are to prepare a systems design specification and present the results to management.

STEP	ACTIVITY	DESCRIPTION
1	Review system requirements.	Study the system requirements document carefully to understand the logical design.
2	Design the system.	
	• User interface, output, and input issues	Design an overall user interface, including screens, commands, controls, and features that enable users to interact with an application. Determine how data will enter the system, and design necessary source documents. Design the physical layout for input and output forms and reports, including screens and printed reports.
	• Data issues	Determine how data will be organized, stored, maintained, updated, accessed, and used.
	• System architecture issues	Determine processing strategies and methods, client/server interaction, network configuration, and Internet/intranet interface issues.
3	Create a system design specification.	Develop a systems design specification that describes and documents the proposed design.
4	Deliver a management presentation.	Include a progress report, a budget update, and a timetable for system implementation and operation.

FIGURE 7-24 Systems design tasks typically are performed in a four-step process.

Overview

The goal of **systems design** is to build a system that satisfies business requirements. A successful system must be effective, reliable, and maintainable:

- A system is effective if it supports business requirements and meets user needs.
- A system is reliable if it handles input errors, processing errors, hardware failures, or human mistakes. A good design will anticipate errors, detect them as early as possible, make it easy to correct them, and prevent them from damaging the system itself.
- A system is maintainable if it is flexible, scalable, and easily modified. Changes might be needed to correct problems, adapt to user requirements, or take advantage of new technology.

Although each project is different, design considerations usually involve users, data, and system architecture.

USER CONSIDERATIONS The most important goal is to make the system user-friendly. Here are some suggestions to keep in mind:

- **Carefully consider any point where users receive output or provide input.** The user interface must be easy to learn. Input processes should be easy to follow, intuitive, and forgiving of errors. Output should be attractive and easy to understand, with an appropriate level of detail.
- **Anticipate future needs.** Suppose that a parts inventory database contains a one-character field for category, such as electrical, mechanical, or hydraulic. The design works well, but what if the company decides to break these overall groups down into more specific segments? Why should there be a limitation of just one character? A better design would anticipate possible expansion to two or more characters. For example, many people recall the concern called the **Y2K issue**, when some older programs that used only two characters to store the year might not adjust properly to the new century.
- **Provide flexibility.** Suppose that a user wants a screen display of all customer balances that exceed $5,000 in an accounts receivable system. How should you design that feature? The program could be coded to check customer balances against a fixed value of 5000, which is a simple solution for both the programmer and the user because no extra keystrokes are required to produce the display. However, that approach is inflexible. For instance, if a user later needs a list of customers whose balances exceed $7,500 rather than $5,000, more programming would be needed. A better approach might be to allow the user to enter the amount. For example, if a user wants to display customers with balances of more than $7,500, he or she can enter that figure in a parameter query. A **parameter** is a value that the user enters whenever the query is run, which provides flexibility, enables users to access information easily, and costs less. A good systems design can combine both approaches. For example, you could design the program to accept a variable amount entered by the user, but start with a **default** value of 5000 that the system displays automatically. Users can press the ENTER key to accept the default value, or enter another value. Often the best design strategy is to come up with several alternatives, so users can decide what will work best for them.

CASE IN POINT 7.4: DOWNTOWN!

Downtown! is a rapidly growing Web-based retailer with about 100 management and technical support employees at its headquarters office in Florida. Mary Estrada, the firm's IT manager, is planning a new information system that will give users better access to sales and marketing data and trends. She has a concern, however. She knows that users often request reports but use only a small portion of the data. In many offices she sees inboxes filled with printed reports gathering dust. Mary asked for your opinion: What if new system users could design most of their own reports without assistance from the IT staff, by using a powerful, user-friendly report writer program? Do you think they would request as many reports or the same types of reports? What are the pros and cons of giving users total control over output?

DATA CONSIDERATIONS Data entry and storage are important in every system. Here are some suggestions to keep in mind:

- **Enter data as soon as possible.** For example, employees in the receiving department should enter data about incoming shipments when the shipments arrive, and sales clerks should enter data about new orders when they take the orders.
- **Verify data as it is entered.** The input design should specify a data type, such as alphabetic, numeric, or alphanumeric, and a range of acceptable values for each data item. If an incorrect value is entered, the system should recognize and flag it immediately. The system also should allow corrections at any time. Some errors, for example, can be easily corrected while the original source documents are at hand or the customer is on the phone. Other errors may need further investigation, so users must be able to correct errors at a later time.
- **Use automated methods of data entry whenever possible.** For example, receiving department employees can use scanners to capture data about merchandise received. Automated data entry methods, such as the RFID scanner shown in Figure 7-25, can reduce input errors and improve employee productivity.
- **Control data entry access and report all entries or changes to critical values.** Dollar fields and volume fields are critical data fields. Examples of critical volumes might include the number of checks processed, the number of medical prescriptions dispensed, or the number of insurance premium payments received. Reports that trace the data entry and changes to critical data values are called **audit trails** and are essential in every system.
- **Log every instance of data entry and changes.** For example, the system should record when a customer's credit limit was established, by whom, and any other information necessary to construct the history of a transaction.

FIGURE 7-25 Automated data entry methods, such as the RFID scanner shown above, reduce input errors and improve employee productivity.

- **Enter data once.** If input data for a payroll system also is needed for a human resources system, you should design a program interface between the systems so data can be transferred automatically. For example, an employee's date of birth should be entered only once, but the data should be accessible by multiple systems or authorized users.
- **Avoid data duplication.** In an inventory database, vendor addresses should not be stored with every part record. Otherwise, the address of a vendor who supplies 100 different parts will be repeated 100 times. Additionally, if the vendor's address changes, all 100 parts records must be updated. Data duplication also can produce inconsistencies. If the 100 stored addresses for the vendor are not identical, how would a user know which version is correct? In Chapter 9, you will learn about data design and a technique called normalization, which is a set of rules that can help you identify and avoid data design problems when you create a database.

ARCHITECTURE CONSIDERATIONS In addition to the issues affecting users and data, you should consider the overall architecture. Here are some suggestions to keep in mind:

- **Use a modular design.** In a modular design, you create individual components, called modules, which connect to a higher-level program or process. In a structured design, each module represents a specific process, which is shown on a DFD and documented in a process description. If you are using an object-oriented design, as described in Chapter 6, object classes are represented by code modules. You will learn more about modular design in Chapter 11, which describes systems implementation.
- **Design modules that perform a single function.** Independent modules provide greater flexibility because they can be developed and tested individually, and then combined or reused later in the development process. Modular design is especially important in designing large-scale systems, because separate teams of analysts and programmers can work on different areas and then integrate the results.

Design Trade-Offs

You will find that design goals often conflict, and you must consider alternatives and trade-offs. To make a system easier to use, for example, programming requirements might be more complex. Making a system more flexible might increase maintenance requirements. Meeting one user's requirements could make it harder to satisfy another user's needs.

Most design trade-off decisions that you will face come down to the basic issue of quality versus cost. Although every project has budget and financial constraints, you should avoid decisions that achieve short-term savings but might mean higher costs later. For example, if you try to reduce implementation costs by cutting back on system testing or user training, you can create higher operational costs in the future. If necessary, you should document and explain the situations carefully to management and discuss the possible risks. Each trade-off must be considered individually, and the final result must be acceptable to users, the systems staff, and company management.

PROTOTYPING

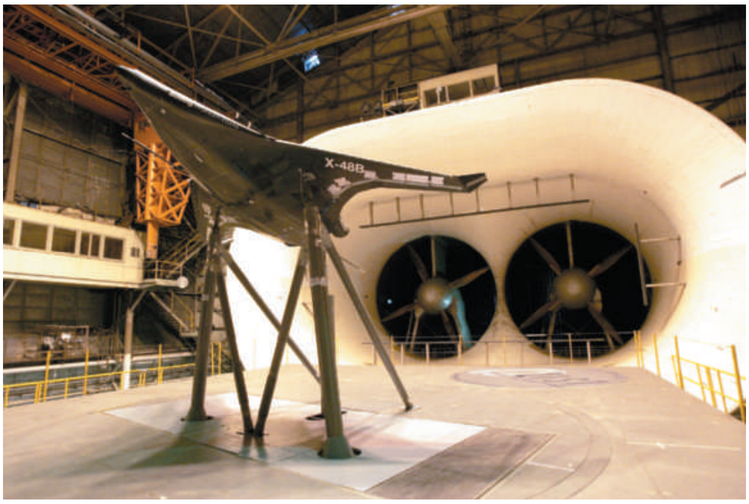


FIGURE 7-26 Wind tunnel testing is a typical example of prototyping.

Prototyping produces an early, rapidly constructed working version of the proposed information system, called a **prototype**. Prototyping, which involves a repetitive sequence of analysis, design, modeling, and testing, is a common technique that can be used to design anything from a new home to a computer network. For example, engineers use a prototype to evaluate an aircraft design before production begins, as shown in the wind tunnel testing in Figure 7-26.

User input and feedback is essential at every stage of the systems development process. Prototyping allows users to examine a model that accurately represents system outputs, inputs, interfaces, and processes. Users can "test-drive" the model in a risk-free environment and either approve it or request changes. In some situations, the prototype evolves into the final version of the information system; in other cases, the prototype is intended only to validate user requirements and is discarded afterward.

Perhaps the most intense form of prototyping occurs when agile methods are used. As you learned in Chapter 1, agile methods build a system by creating a series of prototypes and constantly adjusting them to user requirements. As the agile process continues, developers revise, extend, and merge earlier versions into the final product. An agile approach emphasizes continuous feedback, and each incremental step is affected by what was learned in the prior steps.

Prototyping Methods

Systems analysts use two different prototyping methods: system prototyping and design prototyping. **System prototyping** produces a full-featured, working model of the information system. As Figure 7-27 shows, a system prototype is ready for the implementation phase of the SDLC.

FIGURE 7-27 The end product of system prototyping is a working model of the information system, ready for implementation.

While agile methods represent the latest approach to system prototyping, rapid application development (RAD), which is described in Chapter 4, remains a popular strategy. Using RAD methods, a team of users, managers, and IT staff members works together to develop a model of the information system that evolves into the completed system. The RAD team defines, analyzes, designs, and tests prototypes using a highly interactive process, which is shown in Figure 4-5 on page 146.

Systems analysts also use prototyping to verify user requirements, after which the prototype is discarded and implementation continues, as shown in Figure 7-28 on the next page. The approach is called **design prototyping**, or **throwaway prototyping**. In this case, the prototyping objectives are more limited, but no less important. The end product of design prototyping is a user-approved model that documents and benchmarks the features of the finished system.

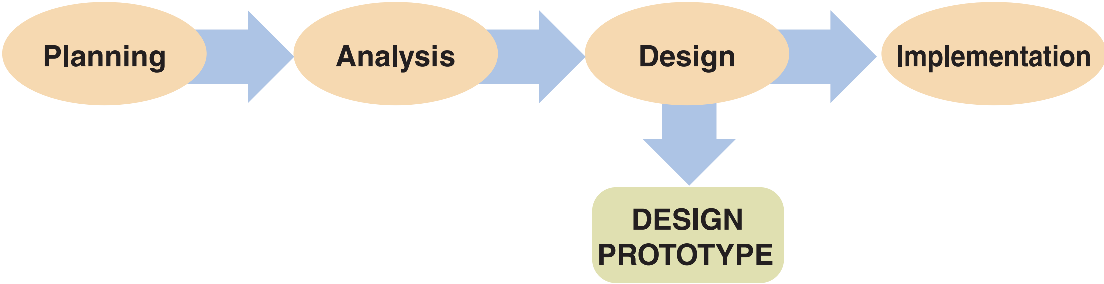


FIGURE 7-28 The end product of design prototyping is a user-approved model that documents and benchmarks the features of the finished system.

Design prototyping makes it possible to capture user input and approval while continuing to develop the system within the framework of the SDLC. Systems analysts typically use design prototyping as they construct outputs, inputs, and user interfaces, as discussed in Chapter 8.

Whenever possible, you should allow users to experiment with a prototype and provide feedback on how well it meets their needs. This approach can increase development costs, but the expense will be offset by lower costs during subsequent SDLC phases. Prototyping offers many benefits, including the following:

- Users and systems developers can avoid misunderstandings.
- System developers can create accurate specifications for the finished system based on the prototype.
- Managers can evaluate a working model more effectively than a paper specification.
- Systems analysts can use a prototype to develop testing and training procedures before the finished system is available.
- Prototyping reduces the risk and potential financial exposure that occur when a finished system fails to support business needs.

Although most systems analysts believe that the advantages of prototyping far outweigh any disadvantages, you should consider the following potential problems:

- The rapid pace of development can create quality problems, which are not discovered until the finished system is operational.
- Other system requirements, such as reliability and maintainability, cannot be tested adequately using a prototype.
- In very complex systems, the prototype becomes unwieldy and difficult to manage.

TOOLKIT TIME

The CASE Tools in Part B of the Systems Analyst's Toolkit can help you document business functions and processes, develop graphical models, and provide an overall framework for information system development. To learn more about these tools, turn to Part B of the four-part Toolkit that follows Chapter 12.

Prototyping Tools

Systems analysts can use powerful tools to develop prototypes. Most prototyping is done using CASE tools, application generators, report generators, screen generators, and fourth-generation languages (4GLs). In a **fourth-generation language (4GL)**, the commands tend to resemble natural statements that people use. For example, a 4GL statement might be PRINT ALL PRODUCTS WHERE CODE = IN STOCK AND STATUS = OK.

In combination, the tools provide a framework for rapid, efficient software development, called a **fourth-generation environment**.

Part B of the Systems Analyst's Toolkit describes CASE tools in more detail and explains how systems analysts can use them to speed the development process, reduce costs, and avoid design errors. In a fourth-generation environment, the development tools are highly interactive. For example, systems analysts use CASE tools to create a series of diagrams and definitions, which generate a data dictionary automatically. The data dictionary organizes and documents all data elements and interacts with application, screen, and report generators to produce a system prototype.

ON THE WEB

To learn more about software development trends, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to **On the Web Links** for this chapter, and locate the Software Development Trends link.

Limitations of Prototypes

The final version of the system typically demands higher-level performance than the prototype can provide. A prototype is a functioning system, but it is less efficient than a fully developed system. Because it is a model, rather than a completed system, the prototype will have slower processing speeds and response times. The prototype also might lack security requirements, exception and error-handling procedures, and other required functions. Despite those limitations, systems developers can upgrade the prototype into the final information system by adding the necessary capability. Otherwise, the prototype is discarded and the remaining SDLC phases are completed.

Even when it does not evolve into the finished system, a prototype helps to ensure that the final product will meet all requirements. Satisfying system requirements is the ultimate goal of systems development, and prototyping is an extremely valuable tool during the process.

Software Development Trends

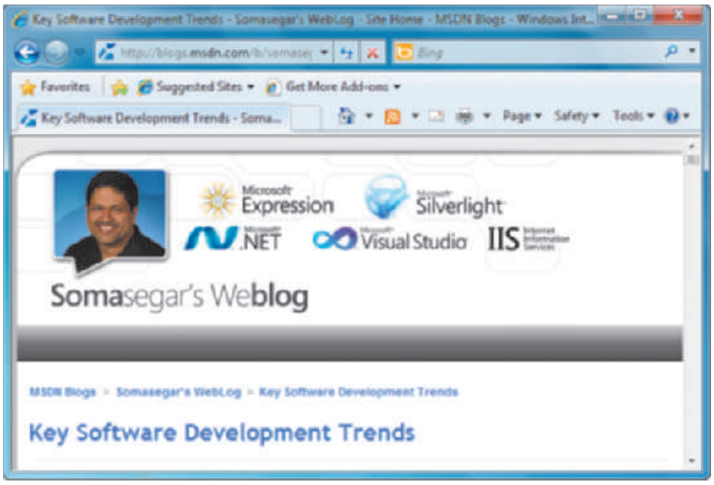
Whether you are a student, an entry-level IT professional, or a seasoned veteran, you need to be aware of technology trends and developments. Because the digital world changes so rapidly, if you have a personal development plan, you can take advantage of future changes and opportunities. This section describes software development trends to consider.

Views from the IT Community

As senior vice president of the Developer Division at Microsoft, Dr. S. Somasegar supports software developers and the technical computing community. His team handles Visual Studio, Expression Studio, the .NET Framework, Silverlight, and related development technologies. Dr. Somasegar mentions six key software development trends in his February, 2010 blog, which are summarized in Figure 7-29.

Many IT professionals try to stay in touch with key issues and trends. For most people, this is not a scientific research project, but rather an effort to keep up with developments and avoid surprises. A sampling of current articles and blogs includes some predictions and forecasts for the next few years:

- Software quality will be more important than ever, and intense modeling will support the quality assurance process. Software testing also will receive more emphasis. In the post-9/11 environment, many firms will



Cloud Computing	Powerful, scalable computing resources will level the playing field for smaller companies. Virtual computing will reduce costs and enable a whole new generation of cloud-based applications, tools and services.
The Web as a Platform	This concept will enable friction-less access to applications and a robust development environment that will allow developers to go beyond the limitations of HTML.
Parallel Computing	Supercomputing in the future will be achieved by computing centers that can harness the power of multiple processors and transform them into powerful computing engines.
Proliferation of Devices	Relatively inexpensive Internet-capable devices will enable a new generation of information access. Interface evolution will focus on touch-based technology, intuitive interfaces, and complete integration of multimedia content.
Agile Development	Agile methods will increase in popularity as IT developers seek flexible, team-based strategies.
Distributed Development	Cloud-based architecture will enable more collaboration and task-sharing as team members seamlessly integrate their efforts. This trend will take advantage of global talent, regardless of time zones or geographical borders.

FIGURE 7-29 In his blog, Dr. S. Somasegar mentions six key software development trends.

> **TOOLKIT TIME**
>
> The Internet Resource tools in Part D of the Systems Analyst's Toolkit can help you in using the Internet to stay abreast of current IT trends and to build your systems analysis skills. To learn more about these tools, turn to Part D of the four-part Toolkit that follows Chapter 12.

need to do more testing to ensure that their systems are not vulnerable to physical attack or electronic intrusion.

- Project management will be a major focus of IT managers. With increased pressure for quality software that meets budget, schedule, and quality requirements, project managers will be key players. In this environment, there will be even more emphasis on project management training and credentials.
- Service-oriented architecture (SOA) will become an important factor in future development. **Service-oriented architecture (SOA)** is an architectural style whose goal is to achieve loose coupling among interacting software objects that provide services. **Loose coupling** means that objects can interact, but are essentially independent. A common example is a DVD and a DVD player— if you want to watch your DVD, you put it into a DVD player and watch your video, because the player provides a DVD playing service. But loose coupling allows you to you to replace your DVD player with another, or to play your videos on more than one player.
- Growth in open-source software such as Linux has increased demand for powerful open-source development tools, while traditional development languages such as C and C++ are becoming less popular. There is a growing open-source community that supports and promotes vendor-neutral open-source development.
- Developers will use more **Web services**, which are modular applications such as currency converters or language translators. Most Web services are based on a combination of HTML and XML. You learned in Chapter 1 that HTML is a platform-independent language that controls the way information is presented on a browser, and that Extensible Markup Language (XML) provides a common data description language that allows easy Web-based communication between different types of hardware and software.
- Programmers will continue to use dynamic languages such as Java, Python, Perl, Ruby, and Visual Basic, among others, and new languages will evolve.

As a systems analyst, you will be affected by rapidly changing technology, and you will want to know about IT trends and developments. Part D of the Systems Analyst's Toolkit contains tips and techniques that you can use to access Web-based information and use it to help build your skills and success.

A QUESTION OF ETHICS

Sally works as a junior analyst for a medium-sized IT consulting firm. Her manager, Bob, has asked her to draft a response to an RFP from a large company that is seeking IT consulting services in connection with a new accounting system.

As Sally worked on the RFP, she noticed a specific question about her firm's recent experience on this type of system. To the best of her knowledge, the firm has only worked on one other accounting project in the last three years. When Bob saw Sally's draft response, he was upset about the way she answered the question. "You don't have to be quite that candid," he said. "Even though we only had one *formal* project, we do have several people who worked on accounting systems before they came here."

"Yes," Sally replied, "But that isn't what the question is asking." As he left her office, Bob's final comment was, "If we want that job, we'll have to come up with a better answer." Thinking about it, Sally isn't comfortable with anything but a straight answer. Is this an ethical question? What are Sally's options?

Chapter Summary

This chapter describes system development strategies, the preparation and presentation of the system requirements document, and the transition to the systems design phase of the SDLC.

An important trend that views Software as a Service (SaaS), rather than a product, has created new software acquisition options. Systems analysts must consider Web-based development environments such as .NET and WebSphere, and various outsourcing options, including application service providers and Internet business services. Application service providers (ASPs) charge subscription fees for providing application software packages. Internet business services (IBSs) offer powerful Web-based servers, software hosting, and IT support services to customers.

Traditional systems must function in various hardware and software environments, be compatible with legacy systems, and operate within the constraints of company networks and desktop computing capability. Such systems utilize Internet links and resources as enhancements. In contrast, Internet-based systems treat the Web *as* the platform, rather than just a communication channel. Many large companies use Web-based systems to handle enterprise-wide applications. Compared to traditional systems, Web-based systems are more scalable, less dependent on specific hardware and software, and more adaptable to outsourcing the operation and support of a software application.

The new Web generation is called Web 2.0, and it is fueling the expansion of information sharing, user collaboration, and social-networking applications such as Twitter, MySpace and Facebook. Another development, called cloud computing because of the commonly used cloud symbol for the Internet, describes an overall online software and data environment, powered by supercomputer technology, that will be an ultimate form of Software as a Service.

If a company chooses to handle its own software development needs, it can create in-house systems, or purchase (and possibly customize) commercially available software packages from a software vendor or value-added reseller (VAR).

Compared with developing an in-house system, an existing commercial software package can be an attractive alternative, because a package generally costs less, takes less time to implement, has a proven track record, and is upgraded frequently. In-house development or customizing a software package might be the best choice when a standard software package cannot meet specific business requirements or constraints. In addition to customizing software packages, companies can create user applications based on standard software that has been specially configured to enhance user productivity.

The systems analyst's role in the software development process depends on the specific development strategy. In-house development requires much more involvement than outsourcing or choosing a commercial package.

The most important factor in choosing a development strategy is total cost of ownership (TCO). Financial analysis tools include payback analysis, which determines how long it takes for a system to pay for itself through reduced costs and increased benefits; return on investment (ROI), which compares a project's total return with its total costs; and net present value (NPV), which analyzes the value of a project by adjusting costs and benefits to reflect the time that they occur.

The process of acquiring software involves a series of steps: evaluate the system requirements, consider network and Web-related issues, identify potential software vendors or outsourcing options, evaluate the alternatives, perform cost-benefit analysis, prepare a recommendation, and implement the solution. During software acquisition, a company can use a request for proposal (RFP) or a request for quotation (RFQ). An RFP invites vendors to respond to a list of system requirements and features; an RFQ seeks bids for a specific product or service.

The system requirements document is the deliverable, or end product, of the systems analysis phase. The document details all system requirements and constraints, recommends the best solution, and provides cost and time estimates for future development work. The system requirements document is the basis for the management presentation. At this point, the firm might decide to develop an in-house system, modify the current system, purchase or customize a software package, perform additional systems analysis work, or stop all further work.

As you prepared for the transition from the systems analysis to systems activities, you learned that a prototype is a working model of the proposed system that you can use to verify the system requirements with users or as a basis for the new system.

You learned that a set of interactive tools, called a fourth-generation environment, can help you construct the prototype. A fourth-generation environment includes screen generators, report writers, application or code generators, and fourth-generation languages, all of which interact with a data dictionary developed with CASE tools. You also reviewed a set of systems design guidelines and suggestions, including user considerations, data considerations, and processing considerations. Finally, you learned about trends in software development, including outsourcing, agile development, and various other topics.

Key Terms and Phrases

application service provider (ASP) *291*
audit trail *313*
benchmark *306*
build or buy *293*
business process outsourcing (BPO) *290*
cloud computing *289*
default *312*
design prototyping *315*
evaluation and selection team *299*
evaluation model *303*
fixed fee model *291*
fourth-generation environment *316*
fourth-generation language (4GL) *316*
global outsourcing *292*
help desk *297*
horizontal application *293*
in-house software *293*
information center (IC) *297*
Internet business services (IBS) *291*
logical design *311*
loose coupling *318*
maintenance agreement *307*
make or buy *293*
managed hosting *291*
middleware *289*
.NET *287*
net present value (NPV) *299*
newsgroup *305*
offshore outsourcing *292*
outsourcing *290*
parameter *312*
payback analysis *299*
physical design *311*
prototype *315*
prototyping *315*
read-only properties *298*
report generator *297*
request for proposal (RFP) *302*
request for quotation (RFQ) *304*
return on investment (ROI) *299*
screen generator *297*
service-oriented architecture (SOA) *318*
service provider *290*
Software as a Service (SaaS) *286*
software license *307*
software package *293*
software requirements specification *309*
software vendor *293*
subscription model *291*
system prototyping *315*
system requirements document *309*
systems design *312*
throwaway prototyping *315*
transaction model *291*
usage model *291*
user application *297*
user interface *297*
value-added reseller (VAR) *293*
vertical application *294*
Web 2.0 *289*
WebSphere *287*
Web services *318*
Y2K issue *312*

Learn It Online

Instructions: To complete the Learn It Online exercises, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to the resources for this chapter, and click the link for the exercise you want to complete.

1 Chapter Reinforcement

TF, MC, and SA

Click one of the Chapter Reinforcement links for Multiple Choice, True/False, or Short Answer. Answer each question and submit to your instructor.

2 Flash Cards

Click the Flash Cards link and read the instructions. Type 20 (or a number specified by your instructor) in the Number of playing cards text box, type your name in the Enter your Name text box, and then click the Flip Card button. When the flash card is displayed, read the question and then click the ANSWER box arrow to select an answer. Flip through the Flash Cards. If your score is 15 (75%) correct or greater, click Print on the File menu to print your results. If your score is less than 15 (75%) correct, then redo this exercise by clicking the Replay button.

3 Practice Test

Click the Practice Test link. Answer each question, enter your first and last name at the bottom of the page, and then click the Grade Test button. When the graded practice test is displayed on your screen, click Print on the File menu to print a hard copy. Continue to take practice tests until you score 80% or better.

4 Who Wants To Be a Computer Genius?

Click the Computer Genius link. Read the instructions, enter your first and last name at the bottom of the page, and then click the Play button. When your score is displayed, click the PRINT RESULTS link to print a hard copy.

5 Wheel of Terms

Click the Wheel of Terms link. Read the instructions, and then enter your first and last name and your school name. Click the PLAY button. When your score is displayed on the screen, right-click the score and then click Print on the shortcut menu to print a hard copy.

6 Crossword Puzzle Challenge

Click the Crossword Puzzle Challenge link. Read the instructions, and then click the Continue button. Work the crossword puzzle. When you are finished, click the Submit button. When the crossword puzzle is redisplayed, submit it to your instructor.

SCR Associates Case Simulation Session 7: Development Strategies

Overview

The SCR Associates case study is a Web-based simulation that allows you to practice your skills in a real-world environment. The case study transports you to SCR's intranet, where you complete 12 work sessions, each aligning with a chapter. As you work on the case, you will receive e-mail and voice mail messages, obtain information from SCR's online libraries, and perform various tasks.

How do I use the case?

- Review the SCR background material in Chapter 1.
- Read the Preview for this session and study the Task List.
- Visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to the **SCR Case Simulation**, and locate the intranet link.
- Enter your name and the password sad9e. An opening screen will display the 12 sessions.
- Select this session. Check your e-mail and voice mail carefully, and then work on the tasks.

Preview: Session 7

As you consider various development strategies for the TIMS system, you receive specific directions from your supervisor, Jesse Baker. She wants you to determine whether vertical software packages exist, and she wants you to explore outsourcing options for the new system. She also expects you to conduct a cost-benefit analysis of developing TIMS in-house, and she wants your input on outsourcing and prototyping.

Task List

1. *Determine whether vertical software packages exist for training operations management. Search the Internet and draft a message describing the results.*
2. *Investigate the possibility of outsourcing the TIMS system. List the options, together with advantages and disadvantages of each.*
3. *Follow Jesse's e-mail instructions about calculating payback, ROI, and NPV for the TIMS system.*
4. *Jesse wants my thoughts on how we can use prototyping for TIMS. She also wants me to prepare a system requirements document and a management presentation.*

FIGURE 7-30 Task list: Session 7.

Chapter Exercises

Review Questions

1. Describe the trend that views software as a service rather than a product. What effect has this trend had on software acquisition options?
2. Explain the difference between horizontal and vertical application software.
3. What is the most common reason for a company to choose to develop its own information system? Give two other reasons why a company might choose the in-house approach.
4. What is an RFP, and how does it differ from an RFQ?
5. What is the purpose of a benchmark test?
6. Explain software licenses and maintenance agreements.
7. What decisions might management reach at the end of the systems analysis phase, and what would be the next step in each case?
8. What is a prototype, and how do systems developers use prototyping?
9. What is a fourth-generation environment?
10. Explain the relationship between logical and physical design.

Discussion Topics

1. As more companies outsource systems development, will there be less need for in-house systems analysts? Why or why not?
2. Suppose you tried to explain the concept of throwaway prototyping to a manager, and she responded by asking, "So, is throwaway prototyping a waste of time and money?" How would you reply?
3. Select a specific type of vertical application software to investigate. Visit local computer stores and use the Internet to determine what software packages are available. Describe the features of those packages.
4. Select a specific type of horizontal application software to investigate. Visit local computer stores and use the Internet to determine what software packages are available. Describe the features of those packages.

Projects

1. The text mentions several firms and organizations that offer IT benchmarking. Locate another benchmarking firm on the Internet, and describe its services.
2. Turn to Part C of the Systems Analyst's Toolkit and review the concept of net present value (NPV). Determine the NPV for the following: An information system will cost $95,000 to implement over a one-year period and will produce no savings during that year. When the system goes online, the company will save $30,000 during the first year of operation. For the next four years, the savings will be $20,000 per year. Assuming a 12% discount rate, what is the NPV of the system?
3. Visit the IT department at your school or at a local company and determine whether the information systems were developed in-house or purchased as software packages. If packages were acquired, determine what customizing was done, if any. Write a brief memo describing the results of your visit.
4. To create user applications as described in this chapter, systems analysts often use macros. Microsoft defines a macro as "a series of commands and instructions that you group together as a single command to accomplish a task automatically." Learn more about macros by using the Help feature in Microsoft Word, and suggest three tasks that might be performed by macros.

Apply Your Knowledge

The Apply Your Knowledge section contains four mini-cases. Each case describes a situation, explains your role in the case, and asks you to respond to questions. You can answer the questions by applying knowledge you learned in the chapter.

1 Top Sail Realty

Situation:

Top Sail Realty is one of the largest time-sharing and rental brokers for vacation cottages along the North Carolina coast. After 10 successful years of matching up owners and renters, Top Sail decided to acquire a computerized reservation and booking system. Top Sail's owner read an article about software packages, and she asked you, as an IT consultant, for your advice.

1. Should Top Sail implement a Web-based system? Why or why not?
2. What software acquisition options are available to Top Sail?
3. Do you consider the reservations system to be a horizontal or a vertical application? Give reasons for your answer.
4. When you evaluate software packages, what steps will you follow?

2 One Way Movers, Inc.

Situation:

As IT manager at One Way, you scheduled a management presentation next week. You prepared and distributed a system requirements document, and you anticipate some intense questioning at the meeting.

1. When planning your presentation, what are some techniques you will use?
2. Based on the suggestions in Part A of the Systems Analyst's Toolkit, what visual aids could you use during your presentation?
3. In deciding on your proposal, what options does management have?
4. If management decides to purchase or customize a software package, what steps will you take?

3 Tangible Investments Corporation

Situation:

Tangible Investments Corporation needs a new customer billing system. As project leader, you decided to create a prototype that users can evaluate before the final design is implemented. You plan to use a traditional structured analysis methodology. To prepare for your meeting with top management tomorrow, you need to review the following topics.

1. Explain the main purpose of prototyping.
2. Explain why a prototype might or might not evolve into the final version of the system.
3. Describe the tools typically used in developing prototypes.
4. List three advantages and three disadvantages of prototyping.

4 IT Flash Magazine

Situation:

You are a staff writer at IT Flash Magazine, a popular online newsletter aimed at IT professionals. Your editor has asked you to prepare a special report for next week's edition. Specifically, she wants you to research the subject of software outsourcing, and other significant trends that might affect software development in the future. If possible, she wants you to cite specific sources for your information, including IT employment statistics and employment forecasts from the Bureau of Labor Statistics.

1. Search for information about software outsourcing generally, using the search techniques described in Part D of the Systems Analyst's Toolkit.
2. Visit the Bureau of Labor Statistics site at bls.gov and search for information about employment trends affecting systems analysts, computer programmers, and software engineers.
3. Does the Bureau of Labor Statistics offer any comments or insights into the subject of outsourcing generally? What conclusions does it reach?
4. In your report, comment on whether the offshore outsourcing of IT jobs is just another step in the progression that began with manufacturing jobs, or represents a whole new trend. Be sure to cite Web research sources and your own reasons.

Case Studies

Case studies allow you to practice specific skills learned in the chapter. Each chapter contains several case studies that continue throughout the textbook, and a chapter capstone case.

New Century Health Clinic

New Century Health Clinic offers preventive medicine and traditional medical care. In your role as an IT consultant, you will help New Century develop a new information system.

Background

Based on your earlier recommendations, New Century decided to continue the systems development process for a new information system that would improve operations, decrease costs, and provide better service to patients.

Now, at the end of the systems analysis phase, you are ready to prepare a system requirements document and give a presentation to the New Century associates. Many of the proposed system's advantages were described during the fact-finding process. Those include smoother operation, better efficiency, and more user-friendly procedures for patients and New Century staff.

You also must examine tangible costs and benefits to determine the economic feasibility of several alternatives. If New Century decides to go ahead with the development process, the main options are to develop the system in-house or purchase a vertical package and configure it to meet New Century's needs. You have studied those choices and put together some preliminary figures.

You know that New Century's current workload requires three hours of office staff overtime per week at a base rate of $8.50 per hour. In addition, based on current projections, New Century will need to add another full-time clerical position in about six months. Neither the overtime nor the additional job will be needed if New Century implements the new system. The current manual system also causes an average of three errors per day, and each error takes about 20 minutes to correct. The new system should eliminate those errors.

Based on your research, you estimate by working full-time you could complete the project in about 12 weeks. Your consulting rate, which New Century agreed to, is $30 per hour. If you design the new system as a database application, you can expect to spend about $2,500 for a networked commercial package. After the system is operational and the staff is trained, New Century should be able to handle routine maintenance tasks without your assistance.

As an alternative to in-house development, a vertical software package is available for about $9,000. The vendor offers a lease-purchase package of $3,000 down, followed by two annual installments of $3,000 each. If New Century buys the package, it would take you about four weeks to install, configure, and test it, working full-time. The vendor provides free support during the first year of operation, but then New Century must sign a technical support agreement at an annual cost of $500. Although the package contains many of the features that New Century wants, most of the reports are pre-designed and it would be difficult to modify their layouts.

No matter which approach is selected, New Century probably will need you to provide about 10 hours of initial training and support each week for the first three months of operation. After the new system is operational, it will need routine maintenance, file backups, and updating. These tasks will require about four hours per week and can be performed by a clinic staff member. In both cases, the necessary hardware and network installation will cost about $5,000.

In your view, the useful life of the system will be about five years, including the year in which the system becomes operational.

Assignments

You scheduled a presentation to New Century in one week, and you must submit a system requirements document during the presentation. Prepare both the written documentation and the presentation. (To give a successful presentation, you will need to learn the skills described in Part A of the Systems Analyst's Toolkit.) Your oral and written presentation must include the following tasks:

1. Provide an overview of the proposed system, including costs and benefits, with an explanation of the various cost-and-benefit types and categories.
2. Develop an economic feasibility analysis, using payback analysis, ROI, and present value (assume a discount rate of 10%).
3. Prepare a context diagram and diagram 0 for the new system.
4. Provide a brief explanation of the various alternatives that should be investigated if development continues, including in-house development and any other possible strategies.

You may wish to include other material to help your audience understand the new system and make a decision on the next step.

Presentation Rules

The following presentation rules should be considered:

- Use suitable visual aids.
- Use presentation software, if possible.
- Distribute handouts before, during, or after the presentation.
- Follow the guidelines in Part A of the Systems Analyst's Toolkit.
- Keep your presentation to 30 minutes, including 5 minutes for questions.

Rules for the System Requirements Document

Consider the following rules while preparing the system requirements document:

- Follow the guidelines in Part A of the Systems Analyst's Toolkit.
- Include charts, graphs, or other helpful visual information in the document.
- Spell check and carefully proofread the entire document.

PERSONAL TRAINER, INC.

Personal Trainer, Inc., owns and operates fitness centers in a dozen Midwestern cities. The centers have done well, and the company is planning an international expansion by opening a new "supercenter" in the Toronto area. Personal Trainer's president, Cassia Umi, hired an IT consultant, Susan Park, to help develop an information system for the new facility. During the project, Susan will work closely with Gray Lewis, who will manage the new operation.

Background

During data and process modeling, Susan Park developed a logical model of the proposed system. She drew an entity-relationship diagram and constructed a set of leveled and balanced DFDs. Now Susan is ready to consider various development strategies for the new system. She will investigate traditional and Web-based approaches and weigh the advantages and disadvantages of in-house development versus other alternatives. As she moves ahead to the systems design phase, she will review design guidelines, consider the use of prototypes, and analyze the possible use of codes.

Before You Begin . . .

Review the facts presented in the Personal Trainer case study in Chapters 2, 4, and 5. Use that information to complete the following tasks.

Assignments

1. Should the new system be designed as a Web-based system? Why or why not? What are some specific issues and options that Susan should consider in making a decision?
2. Assume that Cassia Umi, Personal Trainer's president, has asked Susan to prepare a system requirements document and deliver a presentation to the management team. What should be the main elements of the system requirements document? Also, based on the suggestions in Part A of the Systems Analyst's Toolkit, what visual aids should Susan use during her presentation?
3. Should Susan use a prototype during systems design? What options does she have, and how would you advise her?
4. Susan wants to prepare a presentation that will calculate the total cost of ownership for the system. What financial analysis tools are available to her, and what are the advantages (and possible disadvantages) of each tool?

Chapter Capstone Case: SoftWear, Limited

SWL

SoftWear, Limited (SWL), is a continuing case study that illustrates the knowledge and skills described in each chapter. In this case study, the student acts as a member of the SWL systems development team and performs various tasks.

Background

Systems analyst Rick Williams and programmer/analyst Carla Moore continued to work on a logical model of the payroll system. Meanwhile, the information systems department recently purchased and installed Visible Analyst, a CASE toolkit that supports logical and physical modeling. Rick and Carla traveled to Massachusetts to attend a one-week workshop to learn how to use the package.

After returning from their trip, Rick and Carla decided to create the logical model for the payroll system with Visible Analyst. They felt that the time spent now would pay off in later phases of the project. Rick and Carla used the manual DFDs they created in Chapter 5 to create computerized DFDs using Visible Analyst. Now all related items for the new system are stored in the CASE tool.

Over the next month, Rick and Carla looked at various alternatives and spent their time evaluating the potential solutions. They determined that the best solution was to purchase a payroll package, but the ESIP processing was so unique that none of the available software packages would handle SWL's specific requirements. They concluded that SWL should purchase a payroll package and develop the ESIP system in-house. Jane Rossman and Ann Hon agreed with their recommendation.

The systems analysts completed work on the logical model, alternative evaluations, and cost and time estimates and then prepared the system requirements document for the payroll system. The document was printed and distributed and a management presentation was scheduled at the end of the following week.

At this point, the IT team members were confident that they had done a good job. They had worked closely with SWL users throughout the development process and received user approval on important portions of the document as it was being prepared. They developed visual aids, rehearsed the presentation, and then tried to anticipate questions that management might ask.

Carla gave the management presentation. She recommended that SWL purchase a payroll package sold by Pacific Software Solutions and that ESIP processing be developed in-house to interface with the payroll package.

During the presentation, Carla and Rick answered questions on several points, including the economic analysis they had done. Michael Jeremy, vice president of finance, was especially interested in the method they used to calculate payback analysis, return on investment, and net present value for the new system.

Robert Lansing, SWL's president, arrived for the last part of the presentation. When the presentation ended, he asked the top managers how they felt about the project, and they indicated support for the proposal made by the IT department. The next step was to negotiate a contract with Pacific Software Solutions and for Rick and Carla to begin systems design for the ESIP processing component.

SWL Team Tasks

1. Although the presentation was successful, Rick and Carla ask you to create a checklist of presentation dos and don'ts that would be helpful for IT staff people who deliver presentations.
2. Rick and Carla also want you to review the DFDs that they prepared to see if you have any suggestions for improvement. If you have access to a copier, make a copy of the DFDs shown in Chapter 5 and then write your notes directly on the diagrams.

CHAPTER CAPSTONE CASE: SoftWear, Limited (continued)

3. Michael Jeremy, vice president of finance, was interested in the financial analysis tools that Rick and Carla used in the presentation. Rick has asked you to write a memo to Mr. Jeremy explaining each tool, with a specific description of how it is used, and what results can be obtained. Before you do this, you should review the material in Part C of the Systems Analyst's Toolkit.
4. Although SWL decided to develop the ESIP system in-house, Ann Hon, director of information technology, has requested a report on the trend toward outsourcing software development. Perform Internet research to get up-to-date information about this topic, and prepare a memo for Ms. Hon. Be sure to cite your sources of information.

Manage the SWL Project

You have been asked to manage SWL's new information system project. One of your most important activities will be to identify project tasks and determine when they will be performed. Before you begin, you should review the SWL case in this chapter. Then list and analyze the tasks, as follows:

LIST THE TASKS Start by listing and numbering at least ten tasks that the SWL team needs to perform to fulfill the objectives of this chapter. Your list can include SWL Team Tasks and any other tasks that are described in this chapter. For example, Task 3 might be to Evaluate system requirements, and Task 6 might be to Prepare an RFP.

ANALYZE THE TASKS Now study the tasks to determine the order in which they should be performed. First identify all concurrent tasks, which are not dependent on other tasks. In the example shown in Figure 7-31, Tasks 1, 2, 3, 4, and 5 are concurrent tasks, and could begin at the same time if resources were available.

Other tasks are called dependent tasks, because they cannot be performed until one or more earlier tasks have been completed. For each dependent task, you must identify specific tasks that need to be completed before this task can begin. For example, you would want to evaluate system requirements before you could prepare an RFP, so Task 6 cannot begin until Task 3 is completed, as Figure 7-31 shows.

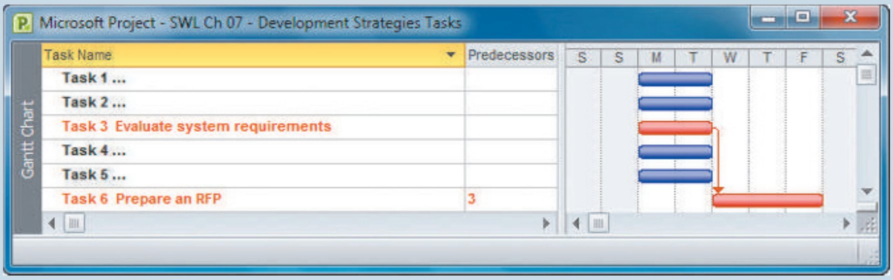


FIGURE 7-31 Tasks 1, 2, 3, 4, and 5 are concurrent tasks that could be performed at the same time. Task 6 is a dependent task that cannot be performed until Task 3 has been completed.

Chapter 3 describes project management tools, techniques, and software. To learn more, you can use the Features section on your Student Study Tool CD-ROM, or visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com** and locate the project management resources library for this book. On the Web, Microsoft offers demo versions, training, and tips for using Project 2010. You also can visit the OpenWorkbench.org site to learn more about this free, open-source software.

Ready for a Challenge?

In additional to technical skills, IT professionals need critical thinking skills such as perception, organization, analysis, problem-solving, and decision-making. The Ready for a Challenge feature can help you learn, practice, and apply critical thinking skills that you can take to the workplace.

You are helping the IT team at Game Technology study several software acquisition options for their Customer Contact Care system (C^3). First, you review Chapter 7 of your systems analysis textbook and the material on financial analysis tools, including the Video Learning Sessions. The IT team is considering two commercial software packages and an in-house development option. They prepared the following summary.

Option	Description	Costs	Benefits
Software Solutions, Inc.	Software package Four-year useful life Less flexible than in-house system Some customizing needed	\$8,000 to purchase \$1,500 to install and configure \$2,500 to load existing data \$1,000 additional hardware needed \$2,000 annual support fee after first year free	\$9,000/yr through positive customer response. Cannot predict other specific savings. Increased sales, improved customer care, and better productivity are expected.
CRM Corp.	Software package Five-year useful life Less flexible than in-house system Moderate customizing needed Runs slower than other options	\$7,000 to purchase \$2,500 to install and configure \$2,000 to load existing data \$4,000 additional hardware needed \$1,200 annual support fee in all five years	Same as above
Develop C^3 system in-house	In-house system Six-year useful life, can use our software and hardware Easiest to update and maintain	\$15,000 to develop, install, and configure \$1,000 to load existing data Existing staff can handle support	Same as above

Practice Tasks

Study the summary carefully and prepare spreadsheet templates, or navigate to the Forms Library at the MIS CourseMate for this book at **www.cengagebrain.com**. You should assume that all three options require a six-month period for acquisition or development, installation, configuration, and data loading. This period is called Year 0. Actual operation begins in Year 1. Now complete these tasks:

A. Perform ROI **and** NPV analysis of the Software Solutions package. Use an 8% discount factor.
B. Perform ROI analysis of the in-house development option.

After you complete the Practice Tasks, to check your work and view sample answers, visit the Management Information Systems CourseMate Web site at **www.cengagebrain.com**, navigate to the resources for this chapter, and locate Ready for a Challenge?.

The Challenge

Now you will work on your own to finish up the remaining financial analysis tasks. Again, you can prepare spreadsheet templates, or navigate to the Forms Library (see above).

Challenge Tasks

A. Perform ROI and NPV analysis of the CRM package. Use an 8% discount factor.
B. Perform NPV analysis of the in-house option. Use an 8% discount factor.