

GLOBAL
EDITION




Modern Systems Analysis and Design

EIGHTH EDITION

Joseph S. Valacich • Joey F. George

ALWAYS LEARNING

PEARSON



Modern Systems Analysis and Design

8th Edition

Global Edition

Joseph S. Valacich

University of Arizona

Joey F. George

Iowa State University

PEARSON

Boston Columbus Indianapolis New York San Francisco
Amsterdam Cape Town Dubai London Madrid Milan Munich Paris Montréal Toronto
Delhi Mexico City São Paulo Sydney Hong Kong Seoul Singapore Taipei Tokyo

Vice President, Business Publishing: Donna Battista
Editor-in-Chief: Stephanie Wall
Senior Sponsoring Editor: Neera Bhalla
Acquisitions Editor: Samantha Lewis
Program Manager: Emily Bibberger
Editorial Assistants: Olivia Vignone, Michael Campbell
Editorial Assistant, Global Editions: Alice Dazeley
Associate Project Editor, Global Edition: Amrita Kar
Project Manager, Global Editions: Sudipto Roy
Manager, Media Production, Global Edition: Vikram Kumar
Senior Manufacturing Controller, Production, Global Edition: Trudy Kimber
Vice President, Product Marketing: Maggie Moylan
Director of Marketing, Digital Services and Products: Jeanette Koskinas
Executive Field Marketing Manager: Adam Goldstein

Field Marketing Manager: Lenny Ann Raper
Product Marketing Assistant: Jessica Quazza
Team Lead, Program Management: Ashley Santora
Team Lead, Project Management: Jeff Holcomb
Project Manager: Ilene Kahn
Operations Specialist: Diane Peirano
Creative Director: Blair Brown
Art Director: Janet Slowik
Vice President, Director of Digital Strategy and Assessment: Paul Gentile
Manager of Learning Applications: Paul DeLuca
Full-Service Project Management and Composition: Integra Software Services, Inc.
Interior Designer: Integra Software Services, Inc.
Cover Designer: Lumina Datamatics
Cover Art: echo3005/Shutterstock
Printer/Binder: Vivar, Malaysia
Cover Printer: Vivar Malaysia

Microsoft and/or its respective suppliers make no representations about the suitability of the information contained in the documents and related graphics published as part of the services for any purpose. All such documents and related graphics are provided “as is” without warranty of any kind. Microsoft and/or its respective suppliers hereby disclaim all warranties and conditions with regard to this information, including all warranties and conditions of merchantability, whether express, implied or statutory, fitness for a particular purpose, title and non-infringement. In no event shall Microsoft and/or its respective suppliers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of information available from the services.

The documents and related graphics contained herein could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Microsoft and/or its respective suppliers may make improvements and/or changes in the product(s) and/or the program(s) described herein at any time. Partial screen shots may be viewed in full within the software version specified.

Microsoft® and Windows® are registered trademarks of the Microsoft Corporation in the U.S.A. and other countries. This book is not sponsored or endorsed by or affiliated with the Microsoft Corporation.

Pearson Education Limited
Edinburgh Gate
Harlow
Essex CM20 2JE
England

and Associated Companies throughout the world

Visit us on the World Wide Web at: www.pearsonglobaleditions.com

© Pearson Education Limited 2017

The rights of Fraser P. Seitel to be identified as the author of this work have been asserted by them in accordance with the Copyright, Designs and Patents Act 1988.

Authorized adaptation from the United States edition, entitled Modern Systems Analysis and Design, Eight Edition, ISBN 978-0-13-420492-5 by Joseph S. Valacich and Joey F. George, published by Pearson Education © 2017.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without either the prior written permission of the publisher or a license permitting restricted copying in the United Kingdom issued by the Copyright Licensing Agency Ltd, Saffron House, 6–10 Kirby Street, London EC 1N 8TS.

All trademarks used herein are the property of their respective owners. The use of any trademark in this text does not vest in the author or publisher any trademark ownership rights in such trademarks, nor does the use of such trademarks imply any affiliation with or endorsement of this book by such owners.

ISBN 10: 1-292-15414-4
ISBN 13: 978-1-292-15414-5

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library

10 9 8 7 6 5 4 3 2 1

Typeset in New Baskerville ITC Pro Roman by Integra
Printed and bound by Vivar, Malaysia

To my mother, Mary Valacich. You are the best!

—Joe

To my mother, Loree George

—Joey

This page intentionally left blank

Brief Contents

Preface 19

PART ONE FOUNDATIONS FOR SYSTEMS DEVELOPMENT 25

1 The Systems Development Environment 27

2 The Origins of Software 50

3 Managing the Information Systems Project 68

Appendix: Object-Oriented Analysis and Design: Project Management 102

PART TWO PLANNING 109

4 Identifying and Selecting Systems Development Projects 111

5 Initiating and Planning Systems Development Projects 135

PART THREE ANALYSIS 169

6 Determining System Requirements 171

7 Structuring System Process Requirements 206

Appendix 7A: Object-Oriented Analysis and Design: Use Cases 241

Appendix 7B: Object-Oriented Analysis and Design: Activity Diagrams 256

Appendix 7C: Object-Oriented Analysis and Design: Sequence
Diagrams 261

Appendix 7D: Business Process Modeling 270

8 Structuring System Data Requirements 279

Appendix: Object-Oriented Analysis and Design: Object Modeling–Class
Diagrams 314

PART FOUR DESIGN 333

9 Designing Databases 335

10 Designing Forms and Reports 377

11 Designing Interfaces and Dialogues 405

12 Designing Distributed and Internet Systems 441

PART FIVE IMPLEMENTATION AND MAINTENANCE 475

13 System Implementation 477

14 Maintaining Information Systems 510

GLOSSARY OF TERMS 528

GLOSSARY OF ACRONYMS 535

INDEX 536

This page intentionally left blank



Contents

Preface 19

PART ONE FOUNDATIONS FOR SYSTEMS DEVELOPMENT

AN OVERVIEW OF PART ONE 26

1 The Systems Development Environment 27

Learning Objectives 27

Introduction 27

A Modern Approach to Systems Analysis and Design 29

Developing Information Systems and the Systems Development Life Cycle 30

A Specialized Systems Development Life Cycle 36

The Heart of the Systems Development Process 37

 The Traditional Waterfall SDLC 39

Different Approaches to Improving Development 40

 Case Tools 40

Agile Methodologies 41

 eXtreme Programming 43

Object-Oriented Analysis and Design 44

Our Approach to Systems Development 46

Summary 47

Key Terms 47

Review Questions 48

Problems and Exercises 48

Field Exercises 49

References 49

2 The Origins of Software 50

Learning Objectives 50

Introduction 50

Systems Acquisition 50

 Outsourcing 51

 Sources of Software 52

 Choosing Off-the-Shelf Software 58

 Validating Purchased Software Information 61

Reuse 61

Summary 64

Key Terms 64

Review Questions 65
Problems and Exercises 65
Field Exercises 65
References 65



BEC CASE: THE ORIGINS OF SOFTWARE 67
Case Questions 67

3 Managing the Information Systems Project 68

Learning Objectives 68
Introduction 68



Pine Valley Furniture Company Background 68

Managing the Information Systems Project 70
Initiating a Project 74
Planning the Project 77
Executing the Project 82
Closing Down the Project 86

Representing and Scheduling Project Plans 87
Representing Project Plans 88
Calculating Expected Time Durations Using PERT 89
Constructing a Gantt Chart and Network Diagram at Pine Valley Furniture 90



Using Project Management Software 93
Establishing a Project Start Date 94
Entering Tasks and Assigning Task Relationships 94
Selecting a Scheduling Method to Review Project Reports 95



Summary 96
Key Terms 97
Review Questions 98
Problems and Exercises 98
Field Exercises 100
References 100

Appendix: Object-Oriented Analysis and Design 102
Learning Objectives 102
Unique Characteristics of an OOSAD Project 102
Define the System as a Set of Components 102
Complete Hard Problems First 102
Using Iterations to Manage the Project 104
Don't Plan Too Much Up Front 104
How Many and How Long Are Iterations? 105
Project Activity Focus Changes Over the Life of a Project 107

Summary 107
Review Question 107
Problems and Exercises 107



BEC CASE: MANAGING THE INFORMATION SYSTEMS PROJECT 108
Case Questions 108

PART TWO PLANNING

AN OVERVIEW OF PART TWO 110

4 Identifying and Selecting Systems Development Projects 111

Learning Objectives 111

Introduction 111

Identifying and Selecting Systems Development Projects 112

The Process of Identifying and Selecting IS Development Projects 113

Deliverables and Outcomes 117

Corporate and Information Systems Planning 118

Corporate Strategic Planning 119

Information Systems Planning 121

Electronic Commerce Applications: Identifying and Selecting Systems Development Projects 128

Internet Basics 128

Pine Valley Furniture WebStore 129

Summary 130

Key Terms 130

Review Questions 131

Problems and Exercises 131

Field Exercises 132

References 132



BEC CASE: IDENTIFYING AND SELECTING SYSTEMS DEVELOPMENT PROJECTS 134

Case Questions 134



5 Initiating and Planning Systems Development Projects 135

Learning Objectives 135

Introduction 135

Initiating and Planning Systems Development Projects 135

The Process of Initiating and Planning Is Development Projects 136

Deliverables and Outcomes 137

Assessing Project Feasibility 138

Assessing Economic Feasibility 139

Assessing Technical Feasibility 147

Assessing Other Feasibility Concerns 150

Building and Reviewing the Baseline Project Plan 151

Building the Baseline Project Plan 151

Reviewing the Baseline Project Plan 156

Electronic Commerce Applications: Initiating and Planning Systems Development Projects 161

Initiating and Planning Systems Development Projects for Pine Valley Furniture's WebStore 161

Summary 163

Key Terms 163

Review Questions 164
Problems and Exercises 164
Field Exercises 165
References 165



BEC CASE: INITIATING AND PLANNING SYSTEMS DEVELOPMENT PROJECTS 167
Case Questions 167

PART THREE ANALYSIS

AN OVERVIEW OF PART THREE 170

6 Determining System Requirements 171

Learning Objectives 171
Introduction 171
Performing Requirements Determination 171
 The Process of Determining Requirements 172
 Deliverables and Outcomes 173
Traditional Methods for Determining Requirements 174
 Interviewing and Listening 174
 Interviewing Groups 178
 Directly Observing Users 179
 Analyzing Procedures and Other Documents 180
Contemporary Methods for Determining System Requirements 185
 Joint Application Design 186
 Using Prototyping During Requirements Determination 189
Radical Methods for Determining System Requirements 191
 Identifying Processes to Reengineer 192
 Disruptive Technologies 192
Requirements Determination Using Agile Methodologies 193
 Continual User Involvement 193
 Agile Usage-Centered Design 194
 The Planning Game from eXtreme Programming 195
Electronic Commerce Applications: Determining System Requirements 197
Determining System Requirements for Pine Valley Furniture's WebStore 197
Summary 200
Key Terms 200
Review Questions 201
Problems and Exercises 201
Field Exercises 202
References 203



BEC CASE: DETERMINING SYSTEM REQUIREMENTS 204
Case Questions 205

7 Structuring System Process Requirements 206

Learning Objectives 206

Introduction 206

Process Modeling 206

Modeling a System's Process for Structured Analysis 207

Deliverables and Outcomes 207

Data Flow Diagramming Mechanics 208

Definitions and Symbols 208

Developing DFDs: An Example 210

Data Flow Diagramming Rules 213

Decomposition of DFDs 214

Balancing DFDs 217

An Example DFD 219

Using Data Flow Diagramming in the Analysis Process 222

Guidelines for Drawing DFDs 222

Using DFDs as Analysis Tools 224

Using DFDs in Business Process Reengineering 225

Modeling Logic With Decision Tables 227

Electronic Commerce Application: Process Modeling Using Data Flow Diagrams 230

Process Modeling for Pine Valley Furniture's WebStore 231

Summary 232

Key Terms 233

Review Questions 234

Problems and Exercises 234

Field Exercises 240

References 240

Appendix 7A Object-Oriented Analysis and Design: Use Cases 241

Learning Objectives 241

Introduction 241

Use Cases 241

What Is a Use Case? 241

Use Case Diagrams 242

Definitions and Symbols 243

Written Use Cases 246

Level 247

The Rest of the Template 247

Electronic Commerce Application: Process Modeling Using Use Cases 249

Writing Use Cases for Pine Valley Furniture's Webstore 251

Summary 254

Key Terms 254

Review Questions 254

Problems and Exercises 254

Field Exercise 255

References 255



Appendix 7B: Object-Oriented Analysis and Design: Activity Diagrams 256
 Learning Objectives 256
 Introduction 256
 When to Use an Activity Diagram 259
 Problems and Exercises 259
 Reference 260

Appendix 7C: Object-Oriented Analysis and Design 261
 Learning Objectives 261
 Introduction 261
 Dynamic Modeling: Sequence Diagrams 261
 Designing a Use Case with a Sequence Diagram 263
 A Sequence Diagram for Hoosier Burger 266
 Summary 268
 Key Terms 268
 Review Questions 268
 Problems and Exercises 268
 Field Exercise 269
 References 269

Appendix 7D: Business Process Modeling 270
 Learning Objective 270
 Introduction 270
 Basic Notation 270
 Business Process Example 274
 Summary 275
 Key Terms 275
 Review Questions 275
 Problems and Exercises 275
 Field Exercises 276
 References 276



BEC CASE: STRUCTURING SYSTEM PROCESS REQUIREMENTS 277
Case Questions 278

8 Structuring System Data Requirements 279
 Learning Objectives 279
 Introduction 279
 Conceptual Data Modeling 280
 The Conceptual Data Modeling Process 281
 Deliverables and Outcomes 282
 Gathering Information for Conceptual Data Modeling 283

Introduction to E-R Modeling	285
Entities	285
Attributes	287
Candidate Keys and Identifiers	288
Other Attribute Types	289
Relationships	290
Conceptual Data Modeling and the E-R Model	291
Degree of a Relationship	292
Cardinalities in Relationships	294
Naming and Defining Relationships	295
Associative Entities	296
Summary of Conceptual Data Modeling with E-R Diagrams	298
Representing Supertypes and Subtypes	298
Business Rules	299
Domains	300
Triggering Operations	302
Role of Packaged Conceptual Data Models: Database Patterns	303
Universal Data Models	303
Industry-Specific Data Models	303
Benefits of Database Patterns and Packaged Data Models	303
Electronic Commerce Application: Conceptual Data Modeling	304
Conceptual Data Modeling for Pine Valley Furniture's WebStore	304
Summary	308
Key Terms	308
Review Questions	309
Problems and Exercises	310
Field Exercises	312
References	313



Appendix: Object-Oriented Analysis and Design: Object Modelling—Class Diagrams	314
Learning Objectives	314
Introduction	314
Representing Objects and Classes	314
Types of Operations	315
Representing Associations	316
Representing Associative Classes	318
Representing Stereotypes for Attributes	319
Representing Generalization	319
Representing Aggregation	322
An Example of Conceptual Data Modeling at Hoosier Burger	323
Summary	326
Key Terms	326

Review Questions 327
Problems and Exercises 327
References 328



BEC CASE: STRUCTURING SYSTEM DATA REQUIREMENTS 329
Case Questions 330

PART FOUR DESIGN

AN OVERVIEW OF PART FOUR 334

9 Designing Databases 335

Learning Objectives 335
Introduction 335
Database Design 335
 The Process of Database Design 336
 Deliverables and Outcomes 338
 The Relational Database Model 341
 Well-Structured Relations 341
Normalization 342
 Rules of Normalization 343
 Functional Dependence and Primary Keys 343
 Second Normal Form 344
 Third Normal Form 344
Transforming E-R Diagrams Into Relations 345
 Represent Entities 346
 Represent Relationships 346
 Summary of Transforming E-R Diagrams to Relations 350
Merging Relations 350
 An Example of Merging Relations 350
 View Integration Problems 351
Logical Database Design for Hoosier Burger 352
Physical File and Database Design 355
 Designing Fields 355
 Choosing Data Types 356
 Controlling Data Integrity 357
 Designing Physical Tables 358
 Arranging Table Rows 361
 Designing Controls for Files 365
Physical Database Design for Hoosier Burger 366
Electronic Commerce Application: Designing Databases 367
 Designing Databases for Pine Valley Furniture's WebStore 368
Summary 370
Key Terms 371
Review Questions 372
Problems and Exercises 372
Field Exercises 373



References 374

BEC CASE: DESIGNING DATABASES 375

Case Questions 376



10 Designing Forms and Reports 377

Learning Objectives 377

Introduction 377

Designing Forms and Reports 377

The Process of Designing Forms and Reports 379

Deliverables and Outcomes 380

Formatting Forms and Reports 384

General Formatting Guidelines 384

Highlighting Information 386

Color versus No Color 388

Displaying Text 389

Designing Tables and Lists 389

Paper versus Electronic Reports 393

Assessing Usability 395

Usability Success Factors 395

Measures of Usability 396

Electronic Commerce Applications: Designing Forms and Reports for Pine Valley Furniture's Webstore 397

General Guidelines 397

Designing Forms and Reports at Pine Valley Furniture 397

Lightweight Graphics 398

Forms and Data Integrity Rules 398

Stylesheet-Based HTML 399

Summary 399

Key Terms 400

Review Questions 400

Problems and Exercises 401

Field Exercises 401

References 402



BEC CASE: DESIGNING FORMS AND REPORTS 403

Case Questions 403

11 Designing Interfaces and Dialogues 405

Learning Objectives 405

Introduction 405

Designing Interfaces and Dialogues 405

The Process of Designing Interfaces and Dialogues 405

Deliverables and Outcomes 406

Interaction Methods and Devices 406

Methods of Interacting 406

Hardware Options for System Interaction 414

- Designing Interfaces 416
 - Designing Layouts 416
 - Structuring Data Entry 419
 - Controlling Data Input 421
 - Providing Feedback 422
 - Providing Help 424
- Designing Dialogues 427
 - Designing the Dialogue Sequence 428
 - Building Prototypes and Assessing Usability 429
- Designing Interfaces and Dialogues in Graphical Environments 431
 - Graphical Interface Design Issues 431
 - Dialogue Design Issues in a Graphical Environment 433
- Electronic Commerce Application: Designing Interfaces and Dialogues for Pine Valley Furniture's Webstore 433
 - General Guidelines 434
 - Designing Interfaces and Dialogues at Pine Valley Furniture 435
 - Menu-Driven Navigation with Cookie Crumbs 435
- Summary 436
- Key Terms 436
- Review Questions 437
- Problems and Exercises 437
- Field Exercises 438
- References 438
- BEC CASE: DESIGNING INTERFACES AND DIALOGUES 439**
 - Case Questions 440



12 Designing Distributed and Internet Systems 441

- Learning Objectives 441
- Introduction 441
- Designing Distributed and Internet Systems 441
 - The Process of Designing Distributed and Internet Systems 441
 - Deliverables and Outcomes 442
- Designing LAN and Client/Server Systems 443
 - Designing Systems for LANs 443
 - Designing Systems for a Client/Server Architecture 445
- Cloud Computing 449
 - What Is Cloud Computing? 449
 - Managing the Cloud 453
 - Service-Oriented Architecture 456
 - Web Services 457
- Designing Internet Systems 458
 - Internet Design Fundamentals 459
 - Site Consistency 460
 - Design Issues Related to Site Management 462



Electronic Commerce Application: Designing a Distributed Advertisement Server for Pine Valley Furniture's Webstore 465

Advertising on Pine Valley Furniture's WebStore 465

Designing the Advertising Component 466

Designing the Management Reporting Component 467

Summary 468

Key Terms 468

Review Questions 470

Problems and Exercises 470

Field Exercises 471

References 472



BEC CASE: DESIGNING DISTRIBUTED AND INTERNET SYSTEMS 473

Case Questions 473

PART FIVE IMPLEMENTATION AND MAINTENANCE

AN OVERVIEW OF PART FIVE 476

13 System Implementation 477

Learning Objectives 477

Introduction 477

System Implementation 478

Coding, Testing, and Installation Processes 479

Deliverables and Outcomes from Coding, Testing, and Installation 479

Deliverables and Outcomes from Documenting the System, Training Users, and Supporting Users 481

Software Application Testing 481

Seven Different Types of Tests 482

The Testing Process 485

Combining Coding and Testing 487

Acceptance Testing by Users 487

Installation 488

Direct Installation 488

Parallel Installation 489

Single-Location Installation 490

Phased Installation 490

Planning Installation 491

Documenting the System 492

User Documentation 492

Training and Supporting Users 494

Training Information Systems Users 494

Supporting Information Systems Users 495

Support Issues for the Analyst to Consider 497

Organizational Issues in Systems Implementation 498
 Why Implementation Sometimes Fails 499
 Security Issues 501

Electronic Commerce Application: System Implementation and Operation
for Pine Valley Furniture’s Webstore 502
 Developing Test Cases for the WebStore 502
 Alpha and Beta Testing the WebStore 504
 WebStore Installation 504

Project Closedown 505

Summary 505

Key Terms 506

Review Questions 507

Problems and Exercises 507

Field Exercises 508

References 508



BEC CASE: SYSTEM IMPLEMENTATION 509
Case Questions 509

14 Maintaining Information Systems 510

Learning Objectives 510

Introduction 510

Maintaining Information Systems 510
 The Process of Maintaining Information Systems 511
 Deliverables and Outcomes 512

Conducting Systems Maintenance 513
 Types of Maintenance 513
 The Cost of Maintenance 514
 Managing Maintenance 516
 Role of Automated Development Tools in Maintenance 521

Website Maintenance 521

Electronic Commerce Application: Maintaining an Information System for Pine Valley
Furniture’s Webstore 523
 Maintaining Pine Valley Furniture’s WebStore 523
 Cannot Find Server 523

Summary 524

Key Terms 525

Review Questions 526

Problems and Exercises 526

Field Exercises 526

References 527



GLOSSARY OF TERMS 528

GLOSSARY OF ACRONYMS 535

INDEX 536



Preface

DESCRIPTION

Modern Systems Analysis and Design, Eighth Edition, covers the concepts, skills, methodologies, techniques, tools, and perspectives essential for systems analysts to successfully develop information systems. The primary target audience is upper-division undergraduates in a management information systems (MIS) or computer information systems curriculum; a secondary target audience is MIS majors in MBA and MS programs. Although not explicitly written for the junior college and professional development markets, this book can also be used by these programs.

We have over 55 years of combined teaching experience in systems analysis and design and have used that experience to create this newest edition of *Modern Systems Analysis and Design*. We provide a clear presentation of the concepts, skills, and techniques that students need to become effective systems analysts who work with others to create information systems for businesses. We use the systems development life cycle (SDLC) model as an organizing tool throughout the book to provide students with a strong conceptual and systematic framework. The SDLC in this edition has five phases and a circular design.

With this text, we assume that students have taken an introductory course on computer systems and have experience designing programs in at least one programming language. We review basic system principles for those students who have not been exposed to the material on which systems development methods are based. We also assume that students have a solid background in computing literacy and a general understanding of the core elements of a business, including basic terms associated with the production, marketing, finance, and accounting functions.

NEW TO THE EIGHTH EDITION

The following features are new to the Eighth Edition:

- *New material.* To keep up with the changing environment for systems development, Chapter 12 has undergone a complete and thorough revision. While cloud computing is introduced in Chapter 2, it is given extensive coverage in the revised Chapter 12. Service-oriented architecture has been reintroduced to the book in the version of Chapter 12. Other new material includes expansions of two of the appendices to Chapter 7. The appendices on activity diagrams and on Business Process Management Notation now include additional text and figures. Throughout the book figures, tables, and related content have been updated and refreshed.
- *Updated content.* Throughout the book, the content in each chapter has been updated where appropriate. We have expanded our coverage of multiple topics in Chapter 2. Examples of updates in other chapters include revising the information on the information services (IS)/information technology job market in Chapter 1. Another example is Chapter 13, where we have updated and extended the section on information systems security. All screenshots come from current versions of leading software products. We have also made a special effort to update our reference lists, purging out-of-date material and including current references.

- *Dropped material.* In our efforts to keep the book current and to streamline it, the coverage of some things was dropped from this edition. Chapter 1 no longer includes Rapid Application Development. Chapter 12 no longer covers data warehouses or data marts. Chapter 13 no longer includes a section on Electronic Performance Support Systems.
- *Organization.* We have retained the organization of the book first introduced in the Sixth Edition. We have 14 chapters and 6 appendices. The first appendix follows Chapter 1. Four appendices follow Chapter 7, including the new one on business process modeling. The sixth appendix follows Chapter 8. This streamlined organization worked well in the Sixth and Seventh Editions, so we decided to continue with it.
- *Approach to presentation of object-oriented material.* We retain our approach to object-orientation (OO) from the last edition. Brief appendices related to the object-oriented approach continue to appear immediately after related chapters. The OO appendices appear as follows: Chapter 3 features a special OO section on IS project management. Chapter 7 now has three OO appendices: one on use cases; one on sequence diagrams; and one about activity diagrams. (The fourth appendix to Chapter 7 is about Business Process Management Notation, which is not part of UML, although it is governed by the Object Management Group (OMG).) Chapter 8 has a special section on object-oriented database design. The rationale for this organization is the same as in the past: to cleanly separate out structured and object-oriented approaches so that instructors not teaching OO can bypass it. On the other hand, instructors who want to expose their students to object-orientation can now do so with minimal effort devoted to finding the relevant OO material.
- *Updated illustrations of technology.* Screen captures have been updated throughout the text to show examples using the latest versions of programming and Internet development environments (including the latest versions of .NET, Visio, and Microsoft Office) and user interface designs. Many references to websites are provided for students to stay current with technology trends that affect the analysis and design of information systems.

Themes of Modern Systems Analysis and Design

1. Systems development is firmly rooted in an organizational context. The successful systems analyst requires a broad understanding of organizations, organizational culture, and organizational operations.
2. Systems development is a practical field. Coverage of current practices as well as accepted concepts and principles is essential in a textbook.
3. Systems development is a profession. Standards of practice, a sense of continuing personal development, ethics, and a respect for and collaboration with the work of others are general themes in the textbook.
4. Systems development has significantly changed with the explosive growth in databases, data-driven systems architectures, rapid development, the Internet, and Agile Methodologies. Systems development and database management can be and should be taught in a highly coordinated fashion. The text is compatible with the Hoffer, Ramesh, and Topi database text, *Modern Database Management*, Eleventh Edition, also published by Pearson. The proper linking of these two textbooks is a strategic opportunity to meet the needs of the IS academic field.

5. Success in systems analysis and design requires not only skills in methodologies and techniques, but also project management skills for managing time, resources, and risks. Thus, learning systems analysis and design requires a thorough understanding of the process as well as the techniques and deliverables of the profession.

Given these themes, this textbook emphasizes the following:

- A business, rather than a technology, perspective
- The role, responsibilities, and mind-set of the systems analyst as well as the systems project manager, rather than those of the programmer or business manager
- The methods and principles of systems development, rather than the specific tools or tool-related skills of the field

DISTINCTIVE FEATURES

The following are some of the distinctive features of *Modern Systems Analysis and Design*:

1. This book is organized in parallel to the Hoffer, Ramesh, and Topi database text, *Modern Database Management*, Twelfth Edition (2016), which will facilitate consistency of frameworks, definitions, methods, examples, and notations to better support systems analysis and design and database courses adopting both texts. Even with the strategic compatibilities between this text and *Modern Database Management*, each of these books is designed to stand alone as a market leader.
2. The grounding of systems development in the typical architecture for systems in modern organizations, including database management and web-based systems.
3. A clear linkage of all dimensions of systems description and modeling—process, decision, and data modeling—into a comprehensive and compatible set of systems analysis and design approaches. Such a broad coverage is necessary so that students understand the advanced capabilities of the many systems development methodologies and tools that are automatically generating a large percentage of code from design specifications.
4. Extensive coverage of oral and written communication skills, including systems documentation, project management, team management, and a variety of systems development and acquisition strategies (e.g., life cycle, prototyping, object orientation, Joint Application Development [JAD], systems reengineering, and Agile Methodologies).
5. Consideration of standards for the methodologies of systems analysis and the platforms on which systems are designed.
6. Discussion of systems development and implementation within the context of change management, conversion strategies, and organizational factors in systems acceptance.
7. Careful attention to human factors in systems design that emphasize usability in both character-based and graphical user interface situations.
8. Visual development products are illustrated and the current limitations technologies are highlighted.
9. The text includes a separate chapter on systems maintenance. Given the type of job many graduates first accept and the large installed base of systems, this chapter covers an important and often neglected topic in systems analysis and design texts.

PEDAGOGICAL FEATURES

The pedagogical features of *Modern Systems Analysis and Design* reinforce and apply the key content of the book.

Three Illustrative Fictional Cases

The text features three fictional cases, described below.



Pine Valley Furniture (PVF): In addition to demonstrating an electronic business-to-consumer shopping website, several other systems development activities from PVF are used to illustrate key points. PVF is introduced in Chapter 3 and revisited throughout the book. As key systems development life cycle concepts are presented, they are applied and illustrated with this descriptive case. For example, in Chapter 5 we explore how PVF plans a development project for a customer tracking system. A margin icon identifies the location of the case segments.



Hoosier Burger (HB): This second illustrative case is introduced in Chapter 7 and revisited throughout the book. HB is a fictional fast-food restaurant in Bloomington, Indiana. We use this case to illustrate how analysts would develop and implement an automated food-ordering system. A margin icon identifies the location of the case segments.



Petrie Electronics: This fictional retail electronics company is used as an extended project case at the end of 12 of the 14 chapters, beginning with Chapter 2. Designed to bring the chapter concepts to life, this case illustrates how a company initiates, plans, models, designs, and implements a customer loyalty system. Discussion questions are included to promote critical thinking and class participation. Suggested solutions to the discussion questions are provided in the Instructor's Manual.

End-of-Chapter Material

We developed an extensive selection of end-of-chapter materials that are designed to accommodate various learning and teaching styles.

- *Chapter Summary.* Reviews the major topics of the chapter and previews the connection of the current chapter with future ones.
- *Key Terms.* Designed as a self-test feature, students match each key term in the chapter with a definition.
- *Review Questions.* Test students' understanding of key concepts.
- *Problems and Exercises.* Test students' analytical skills and require them to apply key concepts.
- *Field Exercises.* Give students the opportunity to explore the practice of systems analysis and design in organizations.
- *Margin Term Definitions.* Each key term and its definition appear in the margin. Glossaries of terms and acronyms appear at the back of the book.
- *References.* References are located at the end of each chapter. The total number of references in this text amounts to over 100 books, journals, and websites that can provide students and faculty with additional coverage of topics.

USING THIS TEXT

As stated earlier, this book is intended for mainstream systems analysis and design courses. It may be used in a one-semester course on systems analysis and design or over two quarters (first in a systems analysis and then in a systems design course). Because this book text parallels *Modern Database Management*, chapters from this book and from *Modern Database Management* can be used in various sequences suitable for your curriculum. The book will be adopted typically in business schools or departments, not in computer science programs. Applied computer science or computer technology programs may also adopt the book.

The typical faculty member who will find this book most interesting is someone who

- has a practical, rather than technical or theoretical, orientation;
- has an understanding of databases and the systems that use databases; and
- uses practical projects and exercises in their courses.

More specifically, academic programs that are trying to better relate their systems analysis and design and database courses as part of a comprehensive understanding of systems development will be especially attracted to this book.

The outline of the book generally follows the systems development life cycle, which allows for a logical progression of topics; however, it emphasizes that various approaches (e.g., prototyping and iterative development) are also used, so what appears to be a logical progression often is a more cyclic process. Part One provides an overview of systems development and previews the remainder of the book. Part One also introduces students to the many sources of software that they can draw on to build their systems and to manage projects. The remaining four parts provide thorough coverage of the five phases of a generic systems development life cycle, interspersing coverage of alternatives to the SDLC as appropriate. Some chapters may be skipped depending on the orientation of the instructor or the students' background. For example, Chapter 3 (Managing the Information Systems Project) can be skipped or quickly reviewed if students have completed a course on project management. Chapter 4 (Identifying and Selecting Systems Development Projects) can be skipped if the instructor wants to emphasize systems development once projects are identified or if there are fewer than 15 weeks available for the course. Chapters 8 (Structuring System Data Requirements) and 9 (Designing Databases) can be skipped or quickly scanned (as a refresher) if students have already had a thorough coverage of these topics in a previous database or data structures course. The sections on object orientation in Chapters 3, 7, and 8 can be skipped if faculty wish to avoid object-oriented topics. Finally, Chapter 14 (Maintaining Information Systems) can be skipped if these topics are beyond the scope of your course.

Because the material is presented within the flow of a systems development project, it is not recommended that you attempt to use the chapters out of sequence, with a few exceptions: Chapter 9 (Designing Databases) can be taught after Chapters 10 (Designing Forms and Reports) and 11 (Designing Interfaces and Dialogues), but Chapters 10 and 11 should be taught in sequence.

THE SUPPLEMENT PACKAGE: WWW.PEARSONGLOBALEDITIONS.COM/VALACICH

A comprehensive and flexible technology support package is available to enhance the teaching and learning experience. All instructor supplements are available on the text website: www.pearsonglobaleditions.com/Valacich.

Instructor Resources

At the Instructor Resource Center, www.pearsonglobaleditions.com/Valacich, instructors can easily register to gain access to a variety of instructor resources available with this text in downloadable format. If assistance is needed, our dedicated technical support team is ready to help with the media supplements that accompany this text. Visit <https://support.pearson.com/getsupport/s/> for answers to frequently asked questions and toll-free user support phone numbers.

The following supplements are available with this text:

- Instructor's Manual
- Test Bank
- TestGen® Computerized Test Bank
- PowerPoint Presentation

ACKNOWLEDGMENTS

The authors have been blessed by considerable assistance from many people on all aspects of preparation of this text and its supplements. We are, of course, responsible for what eventually appears between the covers, but the insights, corrections, contributions, and prodding of others have greatly improved our manuscript. Over the years, dozens of people have reviewed the various editions of this textbook. Their contributions have stimulated us, frequently prompting us to include new topics and innovative pedagogy. We greatly appreciate the efforts of the many faculty and practicing systems analysts who have reviewed this text.

We extend a special note of thanks to Jeremy Alexander, who was instrumental in conceptualizing and writing the PVF WebStore feature that appears in Chapters 4 through 14. The addition of this feature has helped make those chapters more modern and innovative. We would also like to thank Jeff Jenkins, of Brigham Young University, for his help with the Visual Basic screenshots in the current edition.

We also wish to thank Atish Sinha of the University of Wisconsin–Milwaukee for writing the original version of some of the object-oriented analysis and design material. Dr. Sinha, who has been teaching this topic for several years to both undergraduates and MBA students, executed a challenging assignment with creativity and cooperation.

We are also indebted to our undergraduate and MBA students, who have given us many helpful comments as they worked with drafts of this text, and our thanks go to Fred McFadden (University of Colorado, Colorado Springs), Mary Prescott (University of South Florida), Ramesh Venkataraman (Indiana University), and Heikki Topi (Bentley University) for their assistance in coordinating this text with its companion book, *Modern Database Management*, also by Pearson Education.

Finally, we have been fortunate to work with a large number of creative and insightful people at Pearson, who have added much to the development, format, and production of this text. We have been thoroughly impressed with their commitment to this text and to the IS education market. These people include: Nicole Sam (Acquisitions Editor), Neeraj Bhalla (Senior Sponsoring Editor), Olivia Vignone (Editorial Assistant), Ilene Kahn (Project Manager). We would also like to thank George Jacobs and the crew at Integra Software Services, Inc.

The writing of this text has involved thousands of hours of time from the authors and from all of the people listed previously. Although our names will be visibly associated with this book, we know that much of the credit goes to the individuals and organizations listed here for any success it might achieve. It is important for the reader to recognize all the individuals and organizations that have been committed to the preparation and production of this book.

Joseph S. Valacich, Tucson, Arizona

Joey F. George, Ames, Iowa

Pearson would like to thank and acknowledge Petter Dessne (University of Borås), Simon Wu (University of Macau), Athary Alwasel (King Saud University) and Laura Lappalainen (University of Vaasa) for contributing to the Global Edition, and Bernd Schenk (University of Liechtenstein), Sven Rehm (WHU – Otto Beisheim School of Management), Jennifer Wilby (University of Hull), Fabian Ng (Ngee Ann Polytechnic) and Joao Baptista (The University of Warwick) for reviewing the Global Edition.

PART ONE

Foundations for Systems Development



Chapter 1

The Systems Development Environment

Chapter 2

The Origins of Software

Chapter 3

Managing the Information Systems Project

Foundations for Systems Development

You are beginning a journey that will enable you to build on every aspect of your education and experience. Becoming a systems analyst is not a goal; it is a path to a rich and diverse career that will allow you to exercise and continue to develop a wide range of talents. We hope that this introductory part of the text helps open your mind to the opportunities of the systems analysis and design field and to the engaging nature of systems work.

Chapter 1 shows you what systems analysis and design is all about and how it has evolved over the past several decades. As businesses and systems have become more sophisticated and more complex, there has been an increasing emphasis on speed in systems analysis and design. Systems development began as an art, but most businesspeople soon realized this was not a tenable long-term solution to developing systems to support business processes. Systems development became more structured and more like engineering, and managers stressed the importance of planning, project management, and documentation. Now we are witnessing a reaction against excesses in all three of these areas, and the focus has shifted to agile development. The evolution of systems analysis and design and the current focus on agility are explained in Chapter 1. It is also important, however, that you remember that systems analysis and design exists within a multifaceted organizational context that involves other organizational members and external parties. Understanding systems development requires an understanding not only of each technique, tool, and method, but also of how these elements complement and support each other within an organizational setting.

As you read this book, you'll also discover that the systems analysis and design field is constantly adapting to new situations due to a strong commitment to constant improvement. Our goal in this book is to provide you with a mosaic of the skills needed to work effectively in any environment where you may find yourself, armed

with the knowledge to determine the best practices for that situation and argue for them effectively.

Chapter 2 presents an introduction to the many sources from which software and software components can be obtained. Back when systems analysis and design was an art, all systems were written from scratch by in-house experts. Businesses had little choice. Now there is little excuse for in-house development, so it becomes crucial that systems analysts understand the software industry and the many different sources of software. Chapter 2 provides an initial map of the software industry landscape and explains most of the many choices available to systems analysts.

Chapter 3 addresses a fundamental characteristic of life as a systems analyst: working within the framework of projects with constrained resources. All systems-related work demands attention to deadlines, working within budgets, and coordinating the work of various people. The very nature of the systems development life cycle (SDLC) implies a systematic approach to a project, which is a group of related activities leading to a final deliverable. Projects must be planned, started, executed, and completed. The planned work of the project must be represented so that all interested parties can review and understand it. In your job as a systems analyst, you will have to work within the schedule and other project plans, and thus it is important to understand the management process controlling your work.

Finally, Part I introduces the Petrie Electronics case. The Petrie case helps demonstrate how what you learn in each chapter might fit into a practical organizational situation. The case begins after Chapter 2; the remaining book chapters through Chapter 13 each have an associated case installment. The first section introduces the company and its existing information systems. This introduction provides insights into Petrie, which will help you understand the company more completely when we look at the requirements and design for new systems in later case sections.

The Systems Development Environment

Learning Objectives

After studying this chapter, you should be able to

- 1.1 define *information systems analysis and design*,
- 1.2 describe the information systems development life cycle (SDLC),
- 1.3 explain computer-aided software engineering (CASE) tools,
- 1.4 describe the Agile Methodologies and eXtreme Programming, and
- 1.5 explain object-oriented analysis and design and the Rational Unified Process (RUP).

Introduction

Information systems analysis and design is a complex, challenging, and stimulating organizational process that a team of business and systems professionals uses to develop and maintain computer-based information systems. Although advances in information technology continually give us new capabilities, the analysis and design of information systems is driven from an organizational perspective. An organization might consist of a whole enterprise, specific departments, or individual work groups. Organizations can respond to and anticipate problems and opportunities through innovative use of information technology. Information systems analysis and design is therefore an organizational improvement process. Systems are built and rebuilt for organizational benefits. Benefits result from adding value during the process of creating, producing, and supporting the organization's products and services. Thus the analysis and design of information systems is based on your understanding of the organization's objectives, structure, and processes, as well as your knowledge of how to exploit information technology for advantage.

In the current business environment, the Internet, especially the World Wide Web, has been firmly integrated into an organization's way of doing business. Although you are probably most familiar with marketing done on the web and web-based retailing sites, such as eBay or Amazon.com, the overwhelming majority of business use of the web is business-to-business applications.

These applications run the gamut of everything businesses do, including transmitting orders and payments to suppliers, fulfilling orders and collecting payments from customers, maintaining business relationships, and establishing electronic marketplaces where businesses can shop online for the best deals on resources they need for assembling their products and services. Although the Internet seems to pervade business these days, it is important to remember that many of the key aspects of business—offering a product or service for sale, collecting payment, paying employees, maintaining supplier and client relationships—have not changed. Understanding the business and how it functions is still the key to successful systems development, even in the fast-paced, technology-driven environment that organizations find themselves in today.

Careers in information technology (IT) present a great opportunity for you to make a significant and visible impact on business. The demand for skilled information technology workers is growing. According to the US Bureau of Labor Statistics, the professional IT workforce will grow by more than 22 percent between 2010 and 2020 (Thibodeau, 2012). The fastest growth will come for software developers (32 percent) and database administrators (31 percent). One particular aspect of the information technology industry, cloud computing, created almost 14 million technology and related jobs between 2012 and 2015 (McDougall, 2012). Annual revenues from

Information systems analysis and design

The complex organizational process whereby computer-based information systems are developed and maintained.

Application software

Computer software designed to support organizational functions or processes.

Systems analyst

The organizational role most responsible for the analysis and design of information systems.

cloud computing will be over \$1.1 trillion (USD) starting that year. And the growth will be global, with the number of cloud computing jobs in Brazil increasing by 186 percent, the number of jobs in China and India almost doubling, and growth in cloud-related jobs increasing by 66 percent in the United States. (See more about cloud computing in Chapter 2.) With the challenges and opportunities of dealing with rapid advances in technology, it is difficult to imagine a more exciting career choice than information technology, and systems analysis and design is a big part of the IT landscape. Furthermore, analyzing and designing information systems will give you the chance to understand organizations at a depth and breadth that might take many more years to accomplish in other careers.

An important (but not the only) result of systems analysis and design is **application software**, software designed to support a specific organizational function or process, such as inventory management, payroll, or market analysis. In addition to application software, the total information system includes the hardware and systems software on which the application software runs, documentation and training materials, the specific job roles associated with the overall system, controls, and the people who use the software along with their work methods. Although we will address all of these various dimensions of the overall system, we will emphasize application software development—your primary responsibility as a systems analyst.

In the early years of computing, analysis and design was considered an art. Now that the need for systems and software has become so great, people in industry and academia have developed work methods that make analysis and design a disciplined process. Our goal is to help you develop the knowledge and skills needed to understand and follow such software engineering processes. Central to software engineering processes (and to this book) are various methodologies, techniques, and tools that have been developed, tested, and widely used over the years to assist people like you during systems analysis and design.

Methodologies are comprehensive, multiple-step approaches to systems development that will guide your work and influence the quality of your final product—the information system. A methodology adopted by an organization will be consistent with its general management style (e.g., an organization's orientation toward consensus management will influence its choice of systems development methodology). Most methodologies incorporate several development techniques.

Techniques are particular processes that you, as an analyst, will follow to help ensure that your work is well thought out, complete, and comprehensible to others on your project team. Techniques provide support for a wide range of tasks, including conducting thorough interviews to determine what your system should do, planning and managing the activities in a systems development project, diagramming the system's logic, and designing the reports your system will generate.

Tools are typically computer programs that make it easy to use and benefit from techniques and to faithfully follow the guidelines of the overall development methodology. To be effective, techniques and tools must both be consistent with an organization's systems development methodology. Techniques and tools must make it easy for systems developers to conduct the steps called for in the methodology. These three elements—methodologies, techniques, and tools—work together to form an organizational approach to systems analysis and design (see Figure 1-1).

Although many people in organizations are responsible for systems analysis and design, in most organizations the **systems analyst** has the primary responsibility. When you begin your career in systems development, you will most likely begin as a systems analyst or as a programmer with some systems analysis responsibilities. The primary role of a systems analyst is to study the problems and needs of an organization in order to determine how people, methods, and information technology can best be combined to bring about improvements in the organization. A systems analyst helps system users and other business managers define their requirements for new or enhanced information services. As such, a systems analyst is an agent of change and innovation.

**FIGURE 1-1**

An organizational approach to systems analysis and design is driven by methodologies, techniques, and tools

Sources: Top: Mitarart/Fotolia; Left: Lev/Fotolia; Right: PaulPaladin/Fotolia

In the rest of this chapter, we will examine the systems approach to analysis and design. You will learn how systems analysis and design has changed over the decades as computing has become more central to business. You will learn about the systems development life cycle, which provides the basic overall structure of the systems development process and of this book. This chapter ends with a discussion of some of the methodologies, techniques, and tools created to support the systems development process.

A MODERN APPROACH TO SYSTEMS ANALYSIS AND DESIGN

The analysis and design of computer-based information systems began in the 1950s. Since then, the development environment has changed dramatically, driven by organizational needs as well as by rapid changes in the technological capabilities of computers. In the 1950s, development focused on the processes the software performed. Because computer power was a critical resource, efficiency of processing became the main goal. Computers were large, expensive, and not very reliable. Emphasis was placed on automating existing processes, such as purchasing or paying, often within single departments. All applications had to be developed in machine language or assembly language, and they had to be developed from scratch because there was no software industry. Because computers were so expensive, computer memory was also at a premium, so system developers conserved as much memory as possible for data storage.

The first procedural, or third-generation, computer programming languages did not become available until the beginning of the 1960s. Computers were still large and expensive, but the 1960s saw important breakthroughs in technology that enabled the development of smaller, faster, less expensive computers—minicomputers—and the beginnings of the software industry. Most organizations still developed their applications from scratch using their in-house development staff. Systems development was more an art than a science. This view of systems development began to change in the 1970s, however, as organizations started to realize how expensive it was to develop customized information systems for every application. Systems development came to be more

disciplined as many people worked to make it more like engineering. Early database management systems, using hierarchical and network models, helped bring discipline to the storage and retrieval of data. The development of database management systems helped shift the focus of systems development from processes first to data first.

The 1980s were marked by major breakthroughs in computing in organizations, as microcomputers became key organizational tools. The software industry expanded greatly as more and more people began to write off-the-shelf software for microcomputers. Developers began to write more and more applications in fourth-generation languages, which, unlike procedural languages, instructed a computer on what to do instead of how to do it. Computer-aided software engineering (CASE) tools were developed to make systems developers' work easier and more consistent. As computers continued to get smaller, faster, and cheaper, and as the operating systems for computers moved away from line prompt interfaces to windows- and icon-based interfaces, organizations moved to applications with more graphics. Organizations developed less software in-house and bought relatively more from software vendors. The systems developer's job went through a transition from builder to integrator.

The systems development environment of the late 1990s focused on systems integration. Developers used visual programming environments, such as PowerBuilder or Visual Basic, to design the user interfaces for systems that run on client/server platforms. The database, which may be relational or object-oriented, and which may have been developed using software from firms such as Oracle, Microsoft, or Ingres, resided on the server. In many cases, the application logic resided on the same server. Alternatively, an organization may have decided to purchase its entire enterprise-wide system from companies such as SAP AG or Oracle. Enterprise-wide systems are large, complex systems that consist of a series of independent system modules. Developers assemble systems by choosing and implementing specific modules. Starting in the middle years of the 1990s, more and more systems development efforts focused on the Internet, especially the web.

Today there is continued focus on developing systems for the Internet and for firms' intranets and extranets. As happened with traditional systems, Internet developers now rely on computer-based tools to speed and simplify the development of web-based systems. Many CASE tools directly support web application development. More and more, systems implementation involves a three-tier design, with the database on one server, the application on a second server, and client logic located on user machines. Another important development is the move to wireless system components. Wireless devices can access web-based applications from almost anywhere. Finally, the trend continues toward assembling systems from programs and components purchased off the shelf. In many cases, organizations do not develop the application in-house. They don't even run the application in-house, choosing instead to use the application on a per-use basis by accessing it through the cloud.

DEVELOPING INFORMATION SYSTEMS AND THE SYSTEMS DEVELOPMENT LIFE CYCLE

Systems development methodology

A standard process followed in an organization to conduct all the steps necessary to analyze, design, implement, and maintain information systems.

Systems development life cycle (SDLC)

The traditional methodology used to develop, maintain, and replace information systems.

Most organizations find it beneficial to use a standard set of steps, called a **systems development methodology**, to develop and support their information systems. Like many processes, the development of information systems often follows a life cycle. For example, a commercial product follows a life cycle in that it is created, tested, and introduced to the market. Its sales increase, peak, and decline. Finally, the product is removed from the market and replaced by something else. The **systems development life cycle (SDLC)** is a common methodology for systems development in many organizations; it features several phases that mark the progress of the systems analysis and design effort. Every textbook author and information systems development organization uses a slightly different life-cycle model, with anywhere from 3 to almost 20 identifiable phases.