

GLOBAL
EDITION



Modern Systems Analysis and Design

NINTH EDITION

Joseph S. Valacich
Joey F. George





NINTH
EDITION
GLOBAL
EDITION

Modern Systems Analysis and Design

Joseph S. Valacich

University of Arizona

Joey F. George

Iowa State University



Pearson

Harlow, England • London • New York • Boston • San Francisco • Toronto • Sydney • Dubai • Singapore • Hong Kong
Tokyo • Seoul • Taipei • New Delhi • Cape Town • São Paulo • Mexico City • Madrid • Amsterdam • Munich • Paris • Milan

Please contact <https://support.pearson.com/getsupport/s/contactsupport> with any queries on this content.

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

KAO Two
KAO Park
Hockham Way
Harlow
Essex
CM17 9SR
United Kingdom

and Associated Companies throughout the world

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

© Pearson Education Limited, 2021

The rights of Joseph S. Valacich and Joey F. George to be identified as the authors 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, 9th Edition, ISBN 978-0-13-517275-9 by Joseph S. Valacich and Joey F. George, published by Pearson Education © 2020.

Acknowledgments of third-party content appear on the appropriate page within the text.

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 EC1N 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. For information regarding permissions, request forms, and the appropriate contacts within the Pearson Education Global Rights and Permissions department, please visit www.pearsoned.com/permissions/.

This eBook is a standalone product and may or may not include all assets that were part of the print version. It also does not provide access to other Pearson digital products like MyLab and Mastering. The publisher reserves the right to remove any material in this eBook at any time.

ISBN 10: 1-292-35162-4

ISBN 13: 978-1-292-35162-9

eBook ISBN 13: 978-1-292-35163-6

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library

Cover image: Ashalatha/Shutterstock

eBook formatted by SPi Global

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

—Joe

To my parents, John and 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 47
- 3 Managing the Information Systems Project 64
- Appendix:** Object-Oriented Analysis and Design: Project Management 98

PART TWO Planning 107

- 4 Identifying and Selecting Systems Development Projects 109
- 5 Initiating and Planning Systems Development Projects 133

PART THREE Analysis 167

- 6 Determining System Requirements 169
- 7 Structuring System Process Requirements 203
- Appendix 7A:** Object-Oriented Analysis and Design: Use Cases 237
- Appendix 7B:** Object-Oriented Analysis and Design: Activity Diagrams 252
- Appendix 7C:** Business Process Modeling 257
- 8 Structuring System Data Requirements 266
- Appendix:** Object-Oriented Analysis and Design: Object Modeling–Class Diagrams 301

PART FOUR Design 319

- 9 Designing Databases 321
- 10 Designing Forms and Reports 363
- 11 Designing Interfaces and Dialogues 391
- 12 Designing Distributed and Internet Systems 427

PART FIVE Implementation and Maintenance 461

- 13 System Implementation 463
- 14 Maintaining Information Systems 495

GLOSSARY OF TERMS 513

GLOSSARY OF ACRONYMS 520

INDEX 521

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

The Heart of the Systems Development Process 35

 The Traditional Waterfall SDLC 36

Agile Methodologies 38

 eXtreme Programming 39

 Scrum 40

 Agile in Practice 41

Object-Oriented Analysis and Design 42

Our Approach to Systems Development 44

Summary 45

Key Terms 45

Review Questions 45

Problems and Exercises 46

Field Exercises 46

References 46

2 The Origins of Software 47

Learning Objectives 47

Introduction 47

Systems Acquisition 47

 Outsourcing 48

 Sources of Software 49

 Choosing Off-the-Shelf Software 55

 Validating Purchased Software Information 57

Reuse 58

Summary 60

Key Terms 61

Review Questions 61

Problems and Exercises 61
Field Exercises 61
References 62



BEC CASE: THE ORIGINS OF SOFTWARE 63
Case Questions 63

3 Managing the Information Systems Project 64

Learning Objectives 64
Introduction 64



Pine Valley Furniture Company Background 64

Managing the Information Systems Project 66

Initiating a Project 70
Planning the Project 72
Executing the Project 79
Closing Down the Project 82

Representing and Scheduling Project Plans 83

Representing Project Plans 85
Calculating Expected Time Durations Using PERT 85
Constructing a Gantt Chart and Network Diagram at Pine Valley Furniture 86



Using Project Management Software 89

Establishing a Project Start Date 90
Entering Tasks and Assigning Task Relationships 90
Selecting a Scheduling Method to Review Project Reports 91



Summary 93

Key Terms 93

Review Questions 94

Problems and Exercises 94

Field Exercises 96

References 96

Appendix: Object-Oriented Analysis and Design: Project Management 98

Learning Objectives 98

Unique Characteristics of an OOSAD Project 98

Define the System as a Set of Components 98

Complete Hard Problems First 98
Using Iterations to Manage the Project 100
Don't Plan Too Much Up Front 100
How Many and How Long Are Iterations? 102
Project Activity Focus Changes Over the Life of a Project 102

Summary 104

Review Question 104

Problems and Exercises 104



BEC CASE: MANAGING THE INFORMATION SYSTEMS PROJECT 105

Case Questions 105

PART TWO Planning

AN OVERVIEW OF PART TWO 108

4 Identifying and Selecting Systems Development Projects 109

Learning Objectives 109

Introduction 109

Identifying and Selecting Systems Development Projects 110

The Process of Identifying and Selecting IS Development Projects 111

Deliverables and Outcomes 115

Corporate and Information Systems Planning 116

Corporate Strategic Planning 117

Information Systems Planning 119

Electronic Commerce Applications: Identifying and Selecting Systems Development Projects 126

Internet Basics 126

Pine Valley Furniture WebStore 127

Summary 128

Key Terms 128

Review Questions 129

Problems and Exercises 129

Field Exercises 130

References 130



BEC CASE: IDENTIFYING AND SELECTING SYSTEMS DEVELOPMENT PROJECTS 132

Case Questions 132



5 Initiating and Planning Systems Development Projects 133

Learning Objectives 133

Introduction 133

Initiating and Planning Systems Development Projects 133

The Process of Initiating and Planning is Development Projects 134

Deliverables and Outcomes 135

Assessing Project Feasibility 136

Assessing Economic Feasibility 137

Assessing Technical Feasibility 145

Assessing Other Feasibility Concerns 148

Building and Reviewing the Baseline Project Plan 149

Building the Baseline Project Plan 149

Reviewing the Baseline Project Plan 154

Electronic Commerce Applications: Initiating and Planning Systems Development Projects 159

Initiating and Planning Systems Development Projects for Pine Valley Furniture's WebStore 159

Summary 161

Key Terms 161



- Review Questions 162
- Problems and Exercises 162
- Field Exercises 163
- References 163



- BEC CASE: INITIATING AND PLANNING SYSTEMS DEVELOPMENT PROJECTS 165**
- Case Questions 165

PART THREE Analysis

AN OVERVIEW OF PART THREE 168

6 Determining System Requirements 169

- Learning Objectives 169
- Introduction 169
- Performing Requirements Determination 169
 - The Process of Determining Requirements 170
 - Deliverables and Outcomes 171
- Traditional Methods for Determining Requirements 172
 - Interviewing and Listening 172
 - Interviewing Groups 176
 - Directly Observing Users 177
 - Analyzing Procedures and Other Documents 178
- Contemporary Methods for Determining System Requirements 183
 - Joint Application Design 184
 - Using Prototyping During Requirements Determination 186
- Radical Methods for Determining System Requirements 188
 - Identifying Processes to Reengineer 189
 - Disruptive Technologies 190
- Requirements Determination Using Agile Methodologies 191
 - Continual User Involvement 191
 - Agile Usage-Centered Design 192
 - The Planning Game from eXtreme Programming 192
- Electronic Commerce Applications: Determining System Requirements 194
 - Determining System Requirements for Pine Valley Furniture's WebStore 195
- Summary 197
- Key Terms 198
- Review Questions 198
- Problems and Exercises 199
- Field Exercises 199
- References 200



- BEC CASE: DETERMINING SYSTEM REQUIREMENTS 201**
- Case Questions 202

7 Structuring System Process Requirements 203

Learning Objectives 203

Introduction 203

Process Modeling 203

Modeling a System's Process for Structured Analysis 204

Deliverables and Outcomes 204

Data Flow Diagramming Mechanics 205

Definitions and Symbols 205

Developing DFDs: An Example 207

Data Flow Diagramming Rules 210

Decomposition of DFDs 211

Balancing DFDs 214

An Example DFD 216

Using Data Flow Diagramming in the Analysis Process 219

Guidelines for Drawing DFDs 219

Using DFDs as Analysis Tools 221

Using DFDs in Business Process Reengineering 222

Modeling Logic with Decision Tables 223

Electronic Commerce Application: Process Modeling Using Data Flow Diagrams 227

Process Modeling for Pine Valley Furniture's WebStore 227

Summary 229

Key Terms 229

Review Questions 230

Problems and Exercises 230

Field Exercises 236

References 236



Appendix 7A: Object-Oriented Analysis and Design: Use Cases 237

Learning Objectives 237

Introduction 237

Use Cases 237

What Is a Use Case? 237

Use Case Diagrams 238

Definitions and Symbols 239

Written Use Cases 242

Level 243

The Rest of the Template 243

Electronic Commerce Application: Process Modeling Using Use Cases 245

Writing Use Cases for Pine Valley Furniture's WebStore 247

Summary 250

Key Terms 250

Review Questions 250

Problems and Exercises 250

Field Exercise 251

References 251

Appendix 7B: Object-Oriented Analysis and Design: Activity Diagrams 252

- Learning Objectives 252
- Introduction 252
- When to Use an Activity Diagram 255
- Problems and Exercises 255
- Reference 256

Appendix 7C: Business Process Modeling 257

- Learning Objective 257
- Introduction 257
- Basic Notation 257
- Business Process Example 261
- Summary 262
- Key Terms 262
- Review Questions 262
- Problems and Exercises 262
- Field Exercises 263
- References 263

**BEC CASE: STRUCTURING SYSTEM PROCESS REQUIREMENTS 264**

Case Questions 265

8 Structuring System Data Requirements 266

- Learning Objectives 266
- Introduction 266
- Conceptual Data Modeling 267
 - The Conceptual Data Modeling Process 268
 - Deliverables and Outcomes 269
- Gathering Information for Conceptual Data Modeling 270
- Introduction to E-R Modeling 272
 - Entities 272
 - Attributes 274
 - Candidate Keys and Identifiers 275
 - Other Attribute Types 276
 - Relationships 277
- Conceptual Data Modeling and the E-R Model 278
 - Degree of a Relationship 279
 - Cardinalities in Relationships 281
 - Naming and Defining Relationships 282
 - Associative Entities 283
 - Summary of Conceptual Data Modeling with E-R Diagrams 285
- Representing Supertypes and Subtypes 285
- Business Rules 286
 - Domains 287
 - Triggering Operations 289

Role of Packaged Conceptual Data Models: Database Patterns	290
Universal Data Models	290
Industry-Specific Data Models	290
Benefits of Database Patterns and Packaged Data Models	290
Electronic Commerce Application: Conceptual Data Modeling	291
Conceptual Data Modeling for Pine Valley Furniture's WebStore	291
Summary	295
Key Terms	295
Review Questions	296
Problems and Exercises	297
Field Exercises	299
References	300



Appendix: Object-Oriented Analysis and Design: Object Modeling—Class Diagrams 301

Learning Objectives	301
Introduction	301
Representing Objects and Classes	301
Types of Operations	302
Representing Associations	303
Representing Associative Classes	305
Representing Stereotypes for Attributes	306
Representing Generalization	306
Representing Aggregation	309
An Example of Conceptual Data Modeling at Hoosier Burger	310



Summary	313
Key Terms	313
Review Questions	314
Problems and Exercises	314
References	315



BEC CASE: STRUCTURING SYSTEM DATA REQUIREMENTS	316
Case Questions	317

PART FOUR Design

AN OVERVIEW OF PART FOUR 320

9 Designing Databases 321

Learning Objectives	321
Introduction	321
Database Design	321
The Process of Database Design	322
Deliverables and Outcomes	324
The Relational Database Model	327
Well-Structured Relations	327

- Normalization 328
 - Rules of Normalization 329
 - Functional Dependence and Primary Keys 329
 - Second Normal Form 330
 - Third Normal Form 330
- Transforming E-R Diagrams into Relations 331
 - Represent Entities 332
 - Represent Relationships 332
 - Summary of Transforming E-R Diagrams to Relations 336
- Merging Relations 336
 - An Example of Merging Relations 336
 - View Integration Problems 337
- Logical Database Design for Hoosier Burger 338
 - Physical File and Database Design 341
 - Designing Fields 341
 - Choosing Data Types 342
 - Controlling Data Integrity 343
 - Designing Physical Tables 344
 - Arranging Table Rows 347
 - Designing Controls for Files 351
 - Physical Database Design for Hoosier Burger 352
- Electronic Commerce Application: Designing Databases 353
 - Designing Databases for Pine Valley Furniture’s WebStore 354
- Summary 356
- Key Terms 357
- Review Questions 358
- Problems and Exercises 358
- Field Exercises 359
- References 360
- BEC CASE: DESIGNING DATABASES 361**
 - Case Questions 362



10 Designing Forms and Reports 363

- Learning Objectives 363
- Introduction 363
- Designing Forms and Reports 363
 - The Process of Designing Forms and Reports 365
 - Deliverables and Outcomes 366
- Formatting Forms and Reports 370
 - General Formatting Guidelines 370
 - Highlighting Information 372
 - Color versus No Color 374
 - Displaying Text 375
 - Designing Tables and Lists 375
 - Paper versus Electronic Reports 379

Assessing Usability 381
 Usability Success Factors 381
 Measures of Usability 382



Electronic Commerce Applications: Designing Forms and Reports for Pine Valley Furniture's WebStore 383
 General Guidelines 383
 Designing Forms and Reports at Pine Valley Furniture 383
 Lightweight Graphics 384
 Forms and Data Integrity Rules 384
 Stylesheet-Based HTML 385

Summary 385

Key Terms 385

Review Questions 386

Problems and Exercises 386

Field Exercises 387

References 387



BEC CASE: DESIGNING FORMS AND REPORTS 389
 Case Questions 389

11 Designing Interfaces and Dialogues 391

Learning Objectives 391

Introduction 391

Designing Interfaces and Dialogues 391
 The Process of Designing Interfaces and Dialogues 391
 Deliverables and Outcomes 392

Interaction Methods and Devices 392
 Methods of Interacting 392
 Hardware Options for System Interaction 400

Designing Interfaces 402
 Designing Layouts 402
 Structuring Data Entry 405
 Controlling Data Input 407
 Providing Feedback 408
 Providing Help 410

Designing Dialogues 412
 Designing the Dialogue Sequence 413
 Building Prototypes and Assessing Usability 415

Designing Interfaces and Dialogues in Graphical Environments 417
 Graphical Interface Design Issues 417
 Dialogue Design Issues in a Graphical Environment 419



Electronic Commerce Application: Designing Interfaces and Dialogues for Pine Valley Furniture's WebStore 419
 General Guidelines 420
 Designing Interfaces and Dialogues at Pine Valley Furniture 421
 Menu-Driven Navigation with Cookie Crumbs 421

Summary 422
Key Terms 422
Review Questions 423
Problems and Exercises 423
Field Exercises 424
References 424



BEC CASE: DESIGNING INTERFACES AND DIALOGUES 425
Case Questions 426

12 Designing Distributed and Internet Systems 427

Learning Objectives 427
Introduction 427
Designing Distributed and Internet Systems 427
 The Process of Designing Distributed and Internet Systems 427
 Deliverables and Outcomes 428
Designing LAN and Client/Server Systems 429
 Designing Systems for LANs 429
 Designing Systems for a Client/Server Architecture 431
Cloud Computing 435
 What Is Cloud Computing? 435
 Managing the Cloud 439
 Service-Oriented Architecture 442
 Web Services 443
Designing Internet Systems 444
 Internet Design Fundamentals 445
 Site Consistency 446
 Design Issues Related to Site Management 448



Electronic Commerce Application: Designing a Distributed Advertisement Server for Pine Valley Furniture's WebStore 451
 Advertising on Pine Valley Furniture's WebStore 451
 Designing the Advertising Component 452
 Designing the Management Reporting Component 453

Summary 454
Key Terms 454
Review Questions 456
Problems and Exercises 456
Field Exercises 457
References 458



BEC CASE: DESIGNING DISTRIBUTED AND INTERNET SYSTEMS 459
Case Questions 459

PART FIVE Implementation and Maintenance

AN OVERVIEW OF PART FIVE 462

13 System Implementation 463

Learning Objectives 463

Introduction 463

System Implementation 464

 Coding, Testing, and Installation Processes 465

 Deliverables and Outcomes from Coding, Testing, and Installation 465

 Deliverables and Outcomes from Documenting the System, Training Users, and Supporting Users 466

Software Application Testing 467

 Seven Different Types of Tests 468

 The Testing Process 470

 Combining Coding and Testing 472

 Acceptance Testing by Users 473

Installation 474

 Direct Installation 474

 Parallel Installation 474

 Single-Location Installation 475

 Phased Installation 476

 Planning Installation 476

Documenting the System 477

 User Documentation 478

Training and Supporting Users 479

 Training Information Systems Users 480

 Supporting Information Systems Users 481

Organizational Issues in Systems Implementation 482

 Why Implementation Sometimes Fails 483

 Security Issues 485

Electronic Commerce Application: System Implementation and Operation for Pine Valley Furniture's WebStore 487

 Developing Test Cases for the WebStore 487

 Alpha and Beta Testing the WebStore 488

 WebStore Installation 489

Project Closedown 489

Summary 490

Key Terms 490

Review Questions 491

Problems and Exercises 492

Field Exercises 492

References 493

BEC CASE: SYSTEM IMPLEMENTATION 494

Case Questions 494



14 Maintaining Information Systems 495

Learning Objectives 495

Introduction 495

Maintaining Information Systems 495

 The Process of Maintaining Information Systems 496

 Deliverables and Outcomes 497

Conducting Systems Maintenance 498

 Types of Maintenance 498

 The Cost of Maintenance 499

 Managing Maintenance 501

 Role of Automated Development Tools in Maintenance 506

Website Maintenance 506

Electronic Commerce Application: Maintaining an Information System for Pine Valley Furniture's WebStore 508

 Maintaining Pine Valley Furniture's WebStore 508

 Cannot Find Server 508

Summary 509

Key Terms 510

Review Questions 510

Problems and Exercises 511

Field Exercises 511

References 511

GLOSSARY OF TERMS 513

GLOSSARY OF ACRONYMS 520

INDEX 521





Preface

DESCRIPTION

Modern Systems Analysis and Design, Ninth 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 60 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 NINTH EDITION

The following features are new to the Ninth Edition:

- *New material.* To keep up with the changing environment for systems development, Chapter 1 has undergone a thorough revision, with a renewed focus on agile methodologies. While the book has long included material on eXtreme Programming, we now also include a section on Scrum.
- *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. Another example of an updated chapter is Chapter 13, where we have updated and extended the section on information systems security. Chapter 13 also includes new examples of systems implementation failure. 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. Throughout the book figures, tables, and related content have been updated and refreshed.
- *Dropped material.* In our efforts to keep the book current and to streamline it, the coverage of some things was dropped from this edition. Chapters 1, 6 and 7 no longer include computer assisted systems engineering (CASE) tools. We also made some changes in the appendices to Chapter 7. We deleted the appendix on UML sequential diagrams. Appendix 7A is still about use cases,

and Appendix 7B is still about activity diagrams, but Appendix 7C is now about Business Process Management Notation.

- *Organization.* We have retained the organization of the book first introduced in the Sixth Edition, with the only change being the deletion of the former Appendix 7C. We have 14 chapters and 5 appendices. The first appendix follows Chapter 1. Three appendices follow Chapter 7. The fifth appendix follows Chapter 8. This streamlined organization worked well in the Sixth, Seventh and Eighth Editions, so we decided to continue with it and improve on it.
- *Approach to presentation of object-oriented material.* We generally 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 two OO appendices: one on use cases and one about activity diagrams. (The third 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, 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*, Thirteenth 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; and
- 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*, Thirteenth Edition (2019), 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 re-engineering, 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 in the following text.

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: [HTTP://WWW.PEARSONGLOBALEDITIONS.COM](http://www.pearsonglobaleditions.com)

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: <http://www.pearsonglobaleditions.com>.

Instructor Resources

At the Instructor Resource Center, www.pearsonglobaleditions.com, 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 <http://support.pearson.com/getsupport> 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 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: Samantha Lewis (Executive Portfolio Manager), Madeline Houpt (Portfolio Management Assistant), Faraz Sharique Ali (Content Producer) at Pearson, and Freddie Domini and Sindhuja Vadlamani (Full-Service Project Management) at Pearson CSC.

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

GLOBAL EDITION ACKNOWLEDGMENTS

Pearson would like to thank the following people for their work on the Global Edition:

Contributors

Sahil Raj, Punjabi University

Reviewers

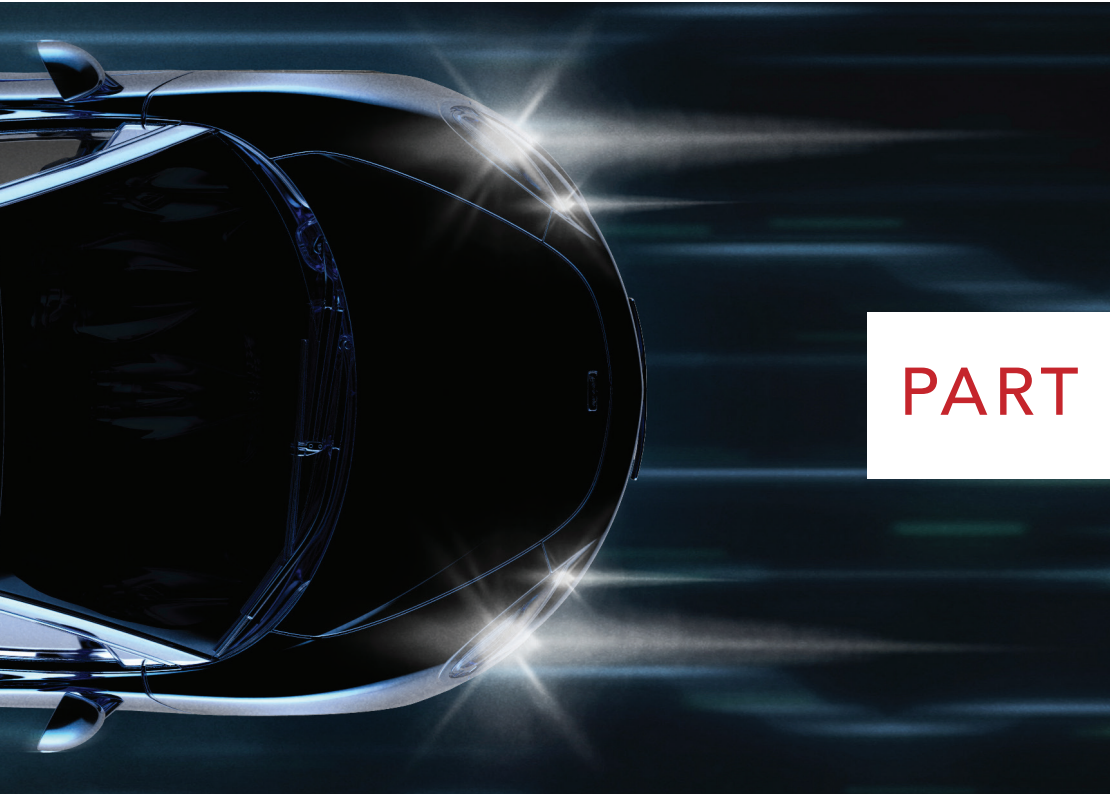
Muhammad Shakaib Akram, University of Essex

Nash Milic, American University of Sharjah

Upasana Singh, University of KwaZulu-Natal

Petter Terenius, Lancaster University and University of Borås

Yuen Yee Yen, Multimedia University



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

PART ONE

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. The focus of systems analysis and design 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 in-house development is much rarer, 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.



1

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 describe the agile methodologies, eXtreme Programming, and Scrum; and
- 1.4 explain object-oriented analysis and design and the Rational Unified Process (RUP).

Introduction

The world runs on information systems. Information systems form the foundation for every major organizational activity and industry, from retail to healthcare to manufacturing to logistics. Systems consist of computer hardware, software, networks, and the people who oversee their operation and the people who use them. **Information systems analysis and design** is the complex, challenging, and stimulating organizational process that a team of business and systems professionals uses to develop and maintain 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.

Information systems support almost everything organizations do, whether the systems are developed for internal use, for exchanges with business partners, or for interactions with customers. Networks, especially the Internet—especially the World Wide Web—are crucial for

connecting organizations with their partners and their customers. 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. Regardless of the technology involved, understanding the business and how it functions is the key to successful systems analysis and design, even in the fast-paced, technology-driven environment that organizations find themselves in today.

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 (IT), 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

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.

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 or a craft. Rapid growth in the need for systems in the 1970s resulted in a highly structured approach to systems analysis and design. While the structured approach is still in use, current approaches focus on rapid and constant software delivery, managed by small teams of talented developers. This approach, called agile development, has become standard for most organizations that develop systems. In fact, 94% of companies report that they practice agile in their systems development efforts (VersionOne, 2017). Our goal is to help you develop the knowledge and skills needed to understand and follow structured and agile processes. Central to analysis and design (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 gathering information 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 system’s interface and outputs.

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

FIGURE 1-1
An organizational approach to systems analysis and design is driven by methodologies, techniques, and tools.

(Sources: Top: Monkey Business Images/Shutterstock; Left: Benchart/Shutterstock; Right: Lifestyle Graphic/Shutterstock)



you begin your career in systems development, you will most likely begin as a systems analyst or as a business analyst. 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.

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. We consider both the structured and the agile approaches to systems analysis and design.

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 payroll, 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 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, 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. 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.

Whether they rely on structured or agile approaches, or on a hybrid, 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 text book author and information systems development organization uses a slightly different life-cycle model, with anywhere from 3 to almost 20 identifiable phases.

The life cycle can be thought of as a circular process in which the end of the useful life of one system leads to the beginning of another project that will develop a new version or replace an existing system altogether (see Figure 1-2). At first glance, the life cycle appears to be a sequentially ordered set of phases, but it is not. The specific steps and their sequence are meant to be adapted as required for a project, consistent with management approaches. For example, in any given SDLC phase, the project can return to an earlier phase if necessary. Similarly, if a commercial product does not perform well just after its introduction, it may be temporarily removed from the market and improved before being reintroduced. In the SDLC, it is also possible to complete some activities in one phase in parallel with some activities of another phase. Sometimes the life cycle is iterative; that is, phases are repeated as required until an acceptable system is found. Some people consider the life cycle to be a spiral, in which we constantly cycle through the phases at different levels of detail (see Figure 1-3). However conceived, the systems development life cycle used in an organization is an orderly set of activities conducted and planned for each development project. The skills required of a systems analyst apply to all life-cycle models. Software is the most obvious end product of the life cycle; other essential outputs include documentation about the system and how it was developed, as well as training for users.