7th
Edition

MODERN SYSTEMS ANALYSIS and DESIGN

Jeffrey A. **Hoffer** ▪ Joey F. **George** ▪ Joseph S. **Valacich**

Modern Systems Analysis and Design

SEVENTH EDITION

Jeffrey A. Hoffer
University of Dayton

Joey F. George
Iowa State University

Joseph S. Valacich
University of Arizona

PEARSON

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

Editor in Chief: Stephanie Wall
Executive Editor: Bob Horan
Editorial Project Manager: Kelly Loftus
Editorial Assistant: Ashlee Bradbury
Director of Marketing: Maggie Moylan
Executive Marketing Manager: Anne Fahlgren
Marketing Assistant: Gianna Sandri
Senior Managing Editor: Judy Leale
Production Project Manager: Jane Bonnell
Operations Specialist: Maura Zaldivar-Garcia
Creative Director: Blair Brown
Art Director: Steve Frim
Cover and Interior Designer: Jill Lehan
Cover Art: Fotolia © sirylok
Media Project Manager, Editorial: Alana Coles
Media Project Manager, Production: Lisa Rinaldi
Composition/Full-Service Project Management:
PreMediaGlobal
Printer/Binder: Courier/Kendallville
Cover Printer: Lehigh-Phoenix Color/Hagerstown
Text Font: New BaskervilleStd

Credits and acknowledgments borrowed from other sources and reproduced, with permission, in this textbook appear on the appropriate page within text.

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.

Library of Congress Cataloging-in-Publication Data
Hoffer, Jeffrey A.
Modern systems analysis and design / Jeffrey A. Hoffer, University of Dayton, Joey F. George, Iowa State University, Joseph S. Valacich, University of Arizona. — SEVENTH EDITION.
pages cm
Includes bibliographical references and index.
ISBN-13: 978-0-13-299130-8 (alk. paper)
ISBN-10: 0-13-299130-6 (alk. paper)
1. System design. 2. System analysis. I. George, Joey F. II. Valacich, Joseph S. III. Title.
QA76.9.S88H6197 2014
005.1'2—dc23
2012033669

10 9 8 7 6 5 4 3 2 1
ISBN 10: 0-13-299130-6
ISBN 13: 978-0-13-299130-8

To Patty, for her sacrifices, encouragement, and support.
To my students, for being receptive and critical, and challenging me to be a better teacher.

—Jeff

In memory of Tom Clark, mentor and friend.

—Joey

To Jackie, Jordan, James, and the rest of my family. Your love and support are my greatest inspiration.

—Joe

BRIEF CONTENTS

CONTENTS

PART TWO PLANNING

PART THREE ANALYSIS

PART FOUR DESIGN

PART FIVE IMPLEMENTATION AND MAINTENANCE

PREFACE

DESCRIPTION

Modern Systems Analysis and Design, Seventh 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 75 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 SEVENTH EDITION

The following features are new to the Seventh Edition:

- *New material.* The most dramatic change in this edition is the introduction of a new end-of-chapter case. Gone is the Broadway Entertainment Company, which has served us well for many years. It has been replaced with Petrie Electronics (following all chapters except 1 and 14). Although the former case lent itself to a focus on in-house development, the new case focuses on finding an existing system and adapting it to the company's needs. A second completely new addition to this edition is an appendix on business process modeling (Appendix 7D). The appendix is based on the Business Process Modeling Notation (BPMN), which is maintained by the Object Management Group (although BPMN is not part of Unified Modeling Language [UML]). The current edition also includes a new section on Microsoft's Security Development Life Cycle (Chapter 1) and a new section on choosing the right enterprise resource planning (ERP) system (Chapter 2). 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 cloud computing (Chapter 2). Examples of updates in other chapters include revising the information on the information services (IS)/information technology job market in Chapter 1. In Chapter 2, we have updated information about the various aspects of the global software industry. Another example is Chapter 13, where we have updated the examples of system implementation failure and

the leading security risks companies have reported. 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 the section on service oriented architecture (SOA). Chapter 4 no longer mentions intranet and extranet, including instead a discussion of more current electronic-commerce terms such as business-to-business and business-to-consumer.
- *Organization.* We have retained the organization of the book first introduced in the Sixth Edition. We have 14 chapters and six 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 Edition, 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. 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*, Eleventh Edition, 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, Rapid Application Development [RAD], 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

PINE VALLEY FURNITURE

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

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

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
- 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 Inferfaces and Dialogues), but Chapters 10 and 11 should be taught in sequence.

THE SUPPLEMENT PACKAGE: WWW.PEARSONHIGHERED.COM/HOFFER

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.pearsonhighered.com/hoffer.

For Instructors

- An *Instructor's Resource Manual* provides chapter-by-chapter instructor objectives; teaching suggestions; and answers to all text review questions, problems, and exercises.
- The *Test Item File* and *TestGen* include a comprehensive set of test questions in multiple-choice, true/false, and short-answer format; questions are ranked according to level of difficulty and referenced with page numbers from the text. The Test Item File is available in Microsoft Word and as the computerized TestGen software. Pearson Education's test-generating software is available from www.pearsonhighered.com/irc. The software is PC/MAC compatible and preloaded with all of the Test Item File questions. You can manually or

randomly view test questions and drag and drop to create a test. You can add or modify test-bank questions as needed.

- *PowerPoint Presentation Slides* feature lecture notes that highlight key text terms and concepts. Professors can customize the presentation by adding their own slides or by editing the existing ones.
- The *Image Library* is a collection of the text art organized by chapter. This collection includes all of the figures, tables, and screenshots (as permission allows) from the book. These images can be used to enhance class lectures and PowerPoint slides.

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 proddings of others have greatly improved our manuscript. The people we recognize here all have a strong commitment to students, to the IS field, and to excellence. Their contributions have stimulated us, frequently prompting us to include new topics and innovative pedagogy.

We would like to recognize the efforts of the many faculty and practicing systems analysts who have been reviewers of this and its associated text, *Essentials of Systems Analysis and Design*. We have tried to deal with each reviewer comment, and although we did not always agree with specific points (within the approach we wanted to take with this book), all reviewers made us stop and think carefully about what and how we were writing. The reviewers were:

Eric Ackerman, *Nova Southeastern University*
Barbara Allen, *Douglas College*
Bay Arinze, *Drexel University*
Janine Aronson, *University of Georgia*
Susan Athey, *Colorado State University*
Sultan Bhimjee, *San Francisco State*
Bill Boroski, *Trident Technical College*
Nora Braun, *Augsburg College*
Penny Brunner, *University of North Carolina, Asheville*
Richard Burkhard, *San Jose State University*
Pedro Cabrejos, *Champlain College*
Donald Chand, *Bentley College*
Phyllis Chasser, *Nova Southeastern University*
Joselina Cheng, *University of Central Oklahoma*
Suzanne R. Clayton, *Drake University*
Amir Dabirian, *California State University, Fullerton*
Jason Deane, *Virginia Tech*
Thomas W. Dillon, *James Madison University*
Mark Dishaw, *University of Wisconsin at Oshkosh*
Jerry Dubyk, *Northern Alabama Institute of Technology*
Bob Foley, *DeVry Institute of Technology*
Barry Frew, *Naval Post-Graduate School*
Jim Gifford, *University of Wisconsin*
Richard Glass, *Bryant University*
Bonnie C. Glassberg, *University of Buffalo*
Mike Godfrey, *California State University, Long Beach*
Dave Groff, *Washington University*
Dale Gust, *Central Michigan University*
John Haney, *Walla Walla College*
Alexander Hars, *University of Southern California*
Kathleen Hartzel, *Duquesne University*
Ellen Hoadley, *Loyola College-Baltimore*
Monica Holmes, *Central Michigan University*
Yujong Hwang, *DePaul University*
Robert Jackson, *Brigham Young University*
Murray Jennex, *University of Phoenix*
Len Jessup, *Washington State University*
Robert Keim, *Arizona State University*

Gene Klawikowski, *Nicolet Area Technical College*

Mat Klempa, *California State University at Los Angeles*

Ned Kock, *Temple University*

Rebecca Koop, *Wright State University*

Sophie Lee, *University of Massachusetts at Boston*

Chang-Yang Lin, *Eastern Kentucky University*

Michael Martel, *Ohio University*

Nancy Martin, *USA Group, Indianapolis, Indiana*

Roger McHaney, *Kansas State University*

Nancy Melone, *University of Oregon*

David Paper, *Utah State University*

G. Premkumar, *Iowa State University*

Mary Prescott, *University of South Florida*

Stephen Priest, *Daniel Webster College*

Harry Reif, *James Madison University*

Steven Ross, *Western Washington University*

Terence Ryan, *Southern Illinois University*

Robert Saldarini, *Bergen Community College*

Arthur Santoianni, *University of Dayton*

Elaine Seeman, *Pitt Community College*

Leiser Silva, *University of Houston*

Taverekere Srikantaiah, *University of Maryland*

Eugene Stafford, *Iona College*

Gary Templeton, *Mississippi State University*

Bob Tucker, *Antares Alliance, Plano, Texas*

Merrill Warkentin, *Northeastern University*

Cheryl Welch, *Ashland University*

Connie Wells, *Nicholls State University*

Chris Westland, *University of Southern California*

Charles Winton, *University of North Florida*

Liang Yu, *San Francisco State University*

Terry Zuechow, *EDS Corporation, Piano, Texas*

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 Jeremy and Saonee Sarker of Washington State University for their assistance in designing installation guides and tutorials for using Oracle development tools in conjunction with prior versions of the book; Darren Nicholson for the help he provided with the Visual Basic and .NET-related materials in previous editions; and Ryan Wright, of the University of Massachusetts, Amherst, who helped with the .NET screen captures in previous editions and who also carefully reviewed each of the book's chapters before they were revised. Thanks to Jeff Jenkins, of the University of Arizona, 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 at the University of Dayton, Iowa State University, and the University of Arizona, 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: Bob Horan (Executive Editor), Kelly Loftus (Editorial Project Manager), Ashlee Bradbury

(Editorial Assistant), Jane Bonnell (Production Project Manager), Anne Fahlgren (Senior Marketing Manager), and Alana Coles (Media Project Manager). We would also like to thank Haylee Schwenk and the crew at PreMediaGlobal.

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.

Jeffrey A. Hoffer, Dayton, Ohio
Joey F. George, Ames, Iowa
Joseph S. Valacich, Tucson, Arizona

PART ONE

Foundations for Systems Development

OVERVIEW

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. 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 cooperate, 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 whatever environment you 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 to Petrie provides insights into the company, which will help you understand the company more completely when we look at the requirements and design for new systems in later case sections.

CHAPTER ONE

The Systems Development Environment

Learning Objectives

After studying this chapter, you should be able to:

- Define *information systems analysis and design.*
- Describe the information systems development life cycle (SDLC).
- Explain Rapid Application Development (RAD) and computer-aided software engineering (CASE) tools.
- Describe the Agile Methodologies and eXtreme Programming.
- 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 uses 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.

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

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 U.S. 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, is predicted to lead to the creation of almost 14 million technology and related jobs by 2015 (McDougall, 2012). Annual revenues from 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.

Application software
Computer software designed to support organizational functions or processes.

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.

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

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.

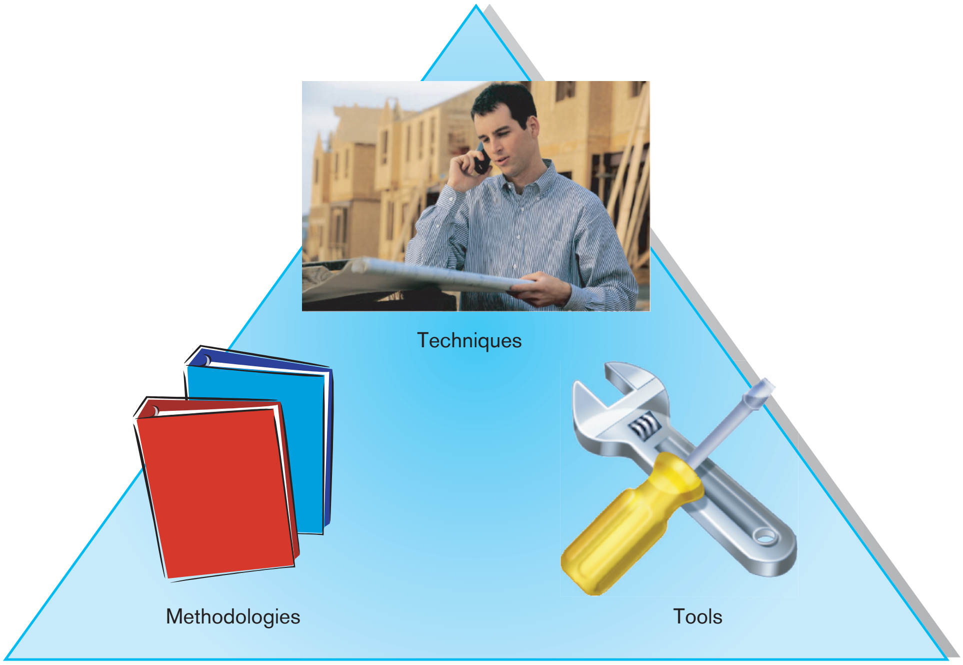

FIGURE 1-1
An organizational approach to systems analysis and design is driven by methodologies, techniques, and tools

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 for data storage as possible.

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 staffs. 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, in the first years of the new century, 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, such as ColdFusion, to speed and simplify the development of Web-based systems. Many CASE tools, such as those developed by Oracle, now 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 in the early years of the new century is the move to wireless system components. Wireless devices, such as cell phones and personal digital assistants (PDAs; e.g., Palm Pilots or Pocket PCs), 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 an application service provider (ASP).

DEVELOPING INFORMATION SYSTEMS AND THE SYSTEMS DEVELOPMENT LIFE CYCLE

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.

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.

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

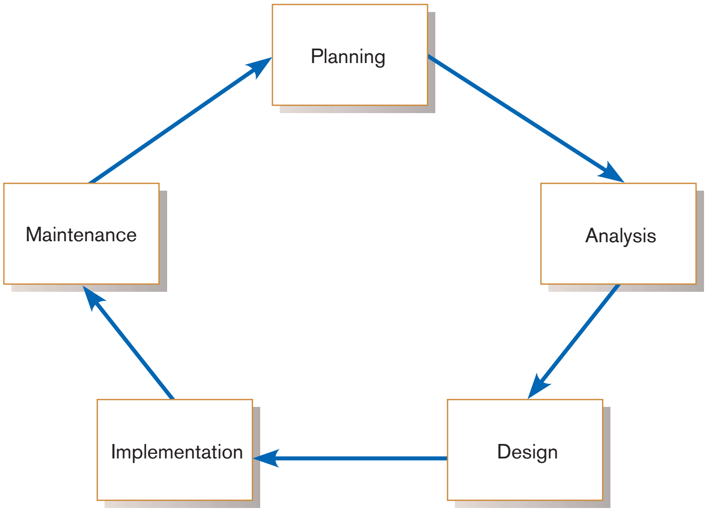

FIGURE 1-2
Systems development life cycle

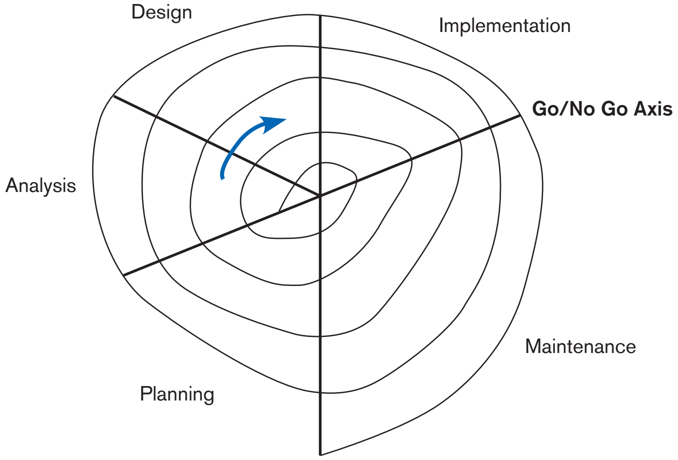

FIGURE 1-3
Evolutionary model

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.

Every medium to large corporation and every custom software producer will have its own specific life cycle or systems development methodology in place (see Figure 1-4). Even if a particular methodology does not look like a cycle, and Figure 1-4 does not, you will probably discover that many of the SDLC steps are performed and SDLC techniques and tools are used. Learning about systems analysis and design from the life cycle approach will serve you well no matter which systems development methodology you use.

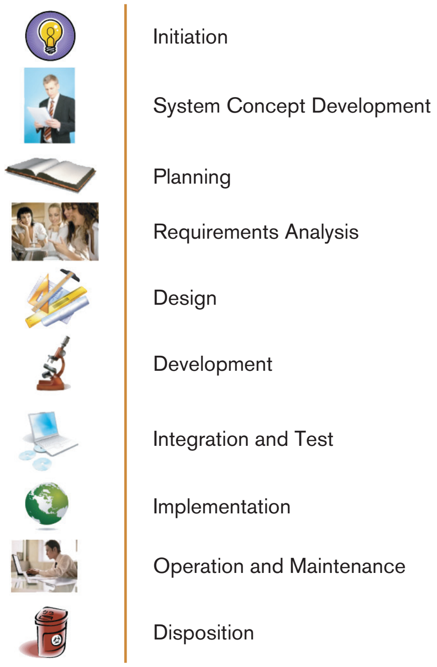

FIGURE 1-4
U.S. Department of Justice's systems development life cycle

(*Source:* Diagram based on *www.usdoj.gov/jmd/irm/lifecycle/ch1.htm#para1.2.*)

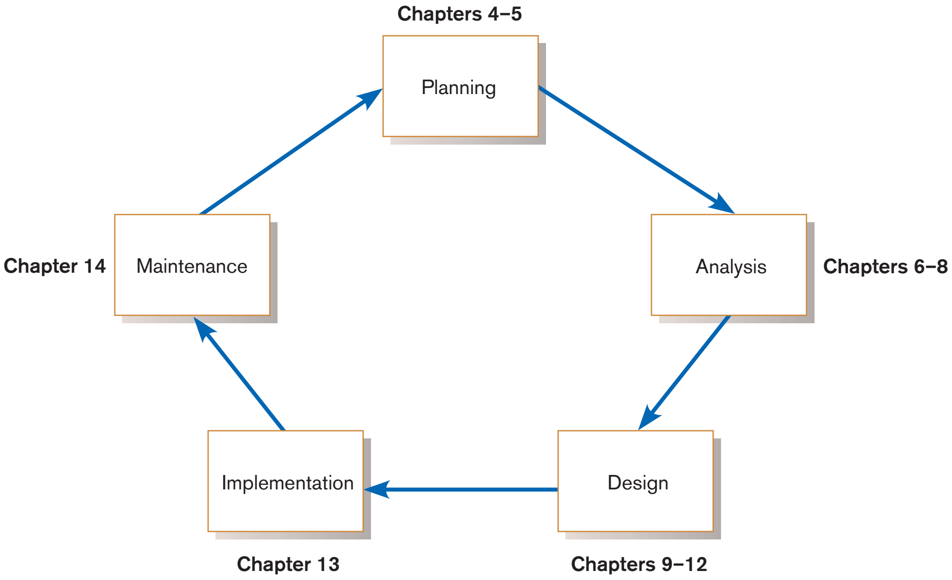

FIGURE 1-5
SDLC-based guide to this book

When you begin your first job, you will likely spend several weeks or months learning your organization's SDLC and its associated methodologies, techniques, and tools. In order to make this book as general as possible, we follow a rather generic life-cycle model, as described in more detail in Figure 1-5. Notice that our model is circular. We use this SDLC as one example of a methodology but, more important, as a way to arrange the topics of systems analysis and design. Thus, what you learn in this book, you can apply to almost any life cycle you might follow. As we describe this SDLC throughout the book, you will see that each phase has specific outcomes and deliverables that feed important information to other phases. At the end of each phase, a systems development project reaches a milestone and, as deliverables are produced, they are often reviewed by parties outside the project team. In the rest of this section, we provide a brief overview of each SDLC phase. At the end of the section, we summarize this discussion in a table that lists the main deliverables or outputs from each SDLC phase.

Planning
The first phase of the SDLC in which an organization's total information system needs are identified, analyzed, prioritized, and arranged.

The first phase in the SDLC is **planning**. In this phase, someone identifies the need for a new or enhanced system. In larger organizations, this recognition may be part of a corporate and systems planning process. Information needs of the organization as a whole are examined, and projects to meet these needs are proactively identified. The organization's information system needs may result from requests to deal with problems in current procedures, from the desire to perform additional tasks, or from the realization that information technology could be used to capitalize on an existing opportunity. These needs can then be prioritized and translated into a plan for the information systems department, including a schedule for developing new major systems. In smaller organizations (as well as in large ones), determination of which systems to develop may be affected by ad hoc user requests submitted as the need for new or enhanced systems arises, as well as from a formalized information planning process. In either case, during project identification and selection, an organization determines whether resources should be devoted to the development or enhancement of each information system under consideration. The outcome of the project identification and selection process is a determination of which systems development projects should be undertaken by the organization, at least in terms of an initial study.

Two additional major activities are also performed during the planning phase: the formal, yet still preliminary, investigation of the system problem or opportunity at hand and the presentation of reasons why the system should or should not be developed by the organization. A critical step at this point is determining the scope of the proposed system. The project leader and initial team of systems analysts also