ESSENTIALS OF
Systems Analysis and Design

Sixth Edition

JOSEPH S. VALACICH ▪ JOEY F. GEORGE ▪ JEFFREY A. HOFFER

Essentials of Systems Analysis and Design

This page intentionally left blank

Essentials of Systems Analysis and Design

SIXTH EDITION

Joseph S. Valacich
University of Arizona

Joey F. George
Iowa State University

Jeffrey A. Hoffer
University of Dayton

PEARSON

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

Editor in Chief: Stephanie Wall
[Executive, Senior as applicable] Acquisitions Editor: Nicole Sam
Program Manager Team Lead: Ashley Santora
Program Manager: Denise Vaughn
Editorial Assistant: Kaylee Rotella
Director of Marketing: Maggie Moylan
Executive Marketing Manager: Anne K. Fahlgren
Project Manager Team Lead: Judy Leale
Project Manager: Karalyn Holland
Procurement Specialist: Diane Peirano
Creative Director: Blair Brown
Interior Designer: S4Carlisle Publishing Services
Cover Designer: PreMedia Global
Cover Art: Iappenno/fotolia
Full-Service Project Management: S4Carlisle Publishing Services
Composition: S4Carlisle Publishing Services
Printer/Binder: Courier/Kendallville
Cover Printer: Courier/Kendallville
Text Font: ITC Century Book

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

Library of Congress Cataloging-in-Publication Data
Valacich, Joseph S.
Essentials of systems analysis and design / Joseph S. Valacich, University of Arizona, Joey F. George, Iowa State University, Jeffrey A. Hoffer, University of Dayton.—Sixth edition.
pages cm
Includes bibliographical references and index.
ISBN 978-0-13-354623-1 — ISBN 0-13-354623-3
1. System design. 2. System analysis. I. George, Joey F. II. Hoffer, Jeffrey A. III. Title.
QA76.9.S88V345 2015
003--dc23
2014017344

PEARSON

10 9 8 7 6 5 4 3 2 1
ISBN 10: 0-13-354623-3
ISBN 13: 978-0-13-354623-1

To my mother, Mary Valacich.

—Joe

To Karen, Evan, and Caitlin.

—Joey

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

—Jeff

Brief Contents

This page intentionally left blank

Contents

PART V SYSTEMS IMPLEMENTATION AND OPERATION 326

Preface

Our Approach

In today's information- and technology-driven business world, students need to be aware of three key factors. First, it is more crucial than ever to know how to organize and access information strategically. Second, success often depends on the ability to work as part of a team. Third, the Internet will play an important part in their work lives. *Essentials of Systems Analysis and Design, Sixth Edition*, addresses these key factors.

More than 50 years' combined teaching experience in systems analysis and design have gone into creating *Essentials of Systems Analysis and Design, Sixth Edition*, a text that emphasizes hands-on, experimental learning. We provide a clear presentation of the concepts, skills, and techniques students need to become effective systems analysts who work with others to create information systems for businesses. We use the systems development life cycle model as an organizing tool throughout the book to provide a strong conceptual and systematic framework.

Electronic commerce coverage is provided in each chapter via an integrated, extended illustrative case (Pine Valley Furniture WebStore) and an end-of-chapter case (Petrie's Electronics).

Many systems analysis and design courses involve lab work and outside reading. Lecture time can be limited. Based on market research and our own teaching experience, we understand the need for a book that combines depth of coverage with brevity. So we have created a ten-chapter book that covers key systems analysis and design content without overwhelming students with unnecessary detail.

New to the Sixth Edition

The following features are new to the Sixth Edition:

- *Expanded coverage of business processes.* Process modeling is at the heart of systems analysis and design. Data-flow diagrams have been a staple of this book since its first edition, but now they are framed in the context of business process diagramming. The beginning of Chapter 6 has been rewritten to show how data-flow diagrams are just one of many common methods for modeling business processes. Business processes are defined and illustrated before the discussion of data-flow diagrams begins.
- *Updates to the WebStore running case.* Since the advent of electronic commerce, this book has featured an end-of-chapter Pine Valley Furniture (PVF) case focused on the WebStore, an e-commerce application for PVF. In the current edition, the WebStore case has been expanded to include the analysis, design, and testing of a new mobile app for PVF. Development of the e-commerce application and the mobile app now go hand-in-hand in the revised case.
- *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, and user interface designs.
- *Updated content.* Throughout the book, the content in each chapter has been updated where appropriate.

Themes

Essentials of Systems Analysis and Design, Sixth Edition, is characterized by the following themes:

- *Systems development is firmly rooted in an organizational context.* The successful systems analyst requires a broad understanding of organizations, organizational culture, and operations.
- *Systems development is a practical field.* Coverage of current practices as well as accepted concepts and principles is essential for today's systems analyst.
- *Systems development is a profession.* The text presents standards of practice, and fosters a sense of continuing personal development, ethics, and a respect for and collaboration with the work of others.
- *Systems development has significantly changed with the explosive growth in databases, data-driven architecture for systems, and the Internet.* Systems development and database management can be taught in a highly coordinated fashion. The Internet has rapidly become a common development platform for database-driven electronic commerce systems.
- *Success in systems analysis and design requires not only skills in methodologies and techniques, but also in the management of time, resources, and risks.* 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, the text emphasizes these approaches:

- A business rather than a technology perspective
- The role, responsibilities, and mindset 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

Audience

The book assumes that students have taken an introductory course on computer systems and have experience writing 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.

Organization

The outline of the book follows the systems development life cycle:

- Part I, "Foundations for Systems Development," gives an overview of systems development and previews the remainder of the book.
- Part II, "Systems Planning and Selection," covers how to assess project feasibility and build the baseline project plan.
- Part III, "Systems Analysis," covers determining system requirements, process modeling, and conceptual data modeling.

- Part IV, “Systems Design,” covers how to design the human interface and databases.
- Part V, “Systems Implementation and Operation,” covers system implementation, operation, closedown, and system maintenance.
- Appendix A, “Object-Oriented Analysis and Design,” and Appendix B, “Agile Methodologies,” can be skipped or treated as advanced topics at the end of the course.

Distinctive Features

Here are some of the distinctive features of *Essentials of Systems Analysis and Design*, *Sixth Edition:*

1. The grounding of systems development in the typical architecture for systems in modern organizations, including database management and Web-based systems.
2. 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 broad coverage is necessary for students to understand the advanced capabilities of many systems development methodologies and tools that automatically generate a large percentage of code from design specifications.
3. 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, object orientation, joint application development, participatory design, and business process reengineering).
4. Coverage of rules and principles of systems design, including decoupling, cohesion, modularity, and audits and controls.
5. A discussion of systems development and implementation within the context of management of change, conversion strategies, and organizational factors in systems acceptance.
6. Careful attention to human factors in systems design that emphasize usability in both character-based and graphical user interface situations.

Pedagogical Features

The pedagogical features of *Essentials of Systems Analysis and Design*, *Sixth Edition*, reinforce and apply the key content of the book.

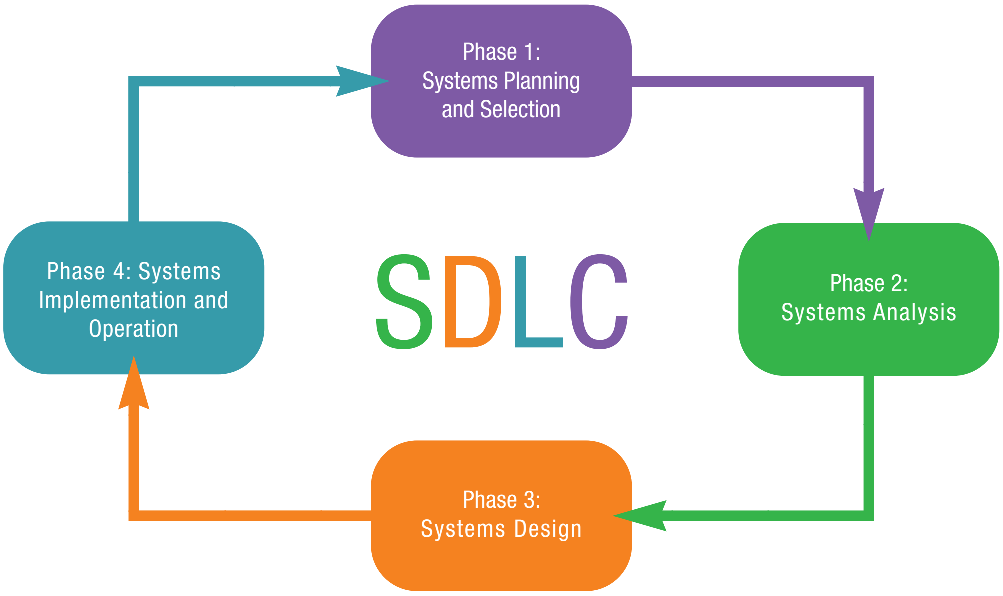
SDLC Framework

Although several conceptual processes can be used for guiding a systems development effort, the systems development life cycle (SDLC) is arguably the most widely applied method for designing contemporary information systems. We highlight four key SDLC steps (Figure P-1):

- Planning and selection
- Analysis
- Design
- Implementation and operation

We use the SDLC to frame the part and chapter organization of our book. Most chapters open with an SDLC figure with various parts highlighted to show

FIGURE P-1
The systems development life cycle (SDLC): management is necessary throughout.



students how these chapters, and each step of the SDLC, systematically build on the previous one.

Internet Coverage and Features

Pine Valley Furniture WebStore A furniture company founded in 1980 has decided to explore electronic commerce as an avenue to increase its market share. Should this company sell its products online? Should this system include a custom mobile app? How would a team of analysts work together to develop, propose, and implement a plan? Beginning in Chapter 4, we explore the step-by-step process.

Petrie's Electronics This end-of-chapter fictional case illustrates how a national electronics retailer develops a Web-based customer loyalty program to build and strengthen customer relationships. The case first appears at the end of Chapter 2 and concludes at the end of Chapter 10.

Three Illustrative Fictional Cases

Pine Valley Furniture (PVF) This case is introduced in Chapter 3 and revisited throughout the book. As key systems development life cycle concepts are presented, they are applied and illustrated. For example, in Chapter 3, we explore how PVF implements the purchasing fulfillment system, and in Chapter 4, we explore how PVF implements a customer tracking system. A margin icon identifies the location of the case segments. A case problem related to PVF is included in the end-of-chapter material.

Hoosier Burger (HB) This second illustrative case is introduced in Chapter 6 and revisited throughout the book. Hoosier Burger 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 these case segments. A case problem related to HB is included in the end-of-chapter material.

Petrie's Electronics This fictional electronics retailer is used as an extended case at the end of each chapter, 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 Web-based customer loyalty program. 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 have developed an extensive selection of end-of-chapter material designed to accommodate various learning and teaching styles.

Key Points Review This section repeats the learning objectives that appear at the opening of the chapter and summarizes the key points related to the objectives.

Key Terms Checkpoint In this self-test feature, students match each key term in the chapter with its definition.

Review Questions These questions test students' understanding of key concepts.

Problems and Exercises These exercises test students' analytical skills and require them to apply key concepts.

Discussion Questions These questions promote class participation and discussion.

Case Problems These problems require students to apply the concepts of the chapter to fictional cases from various industries. The two illustrative cases from the chapters are revisited—Pine Valley Furniture and Hoosier Burger. Other cases are from various fields such as medicine, agriculture, and technology. Solutions are provided in the Instructor's Manual.

Margin Term Definitions

Each key term and its definition appear in the margin. A glossary of terms appears at the back of the book.

References

Located at the end of the text, references are organized by chapter and list more than 200 books and journals that can provide students and faculty with additional coverage of topics.

The Supplement Package: www.pearsonhighered.com/valacich

A comprehensive and flexible technology support package is available to enhance the teaching and learning experience. Instructor supplements are available at www.pearsonhighered.com/valacich:

- 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 more than 1,500 test questions in multiple-choice, true-false, and short-answer format; questions are ranked according to level of difficulty and referenced with page numbers and topic headings from the text. The Test Item File is available in Microsoft Word and as the computerized Prentice Hall TestGen software. 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.

Materials for Your Online Course

Our TestGens are converted for use in BlackBoard and WebCT. These conversions can be found on the Instructor's Resource Center. Conversions to D2L or Angel can be requested through your local Pearson Sales Representative.

CourseSmart

CourseSmart eTextbooks were developed for students looking to save on required or recommended textbooks. Students simply select their eText by title or author and purchase immediate access to the content for the duration of the course using any major credit card. With a CourseSmart eText, students can search for specific keywords or page numbers, take notes online, print out reading assignments that incorporate lecture notes, and bookmark important passages for later review. For more information or to purchase a CourseSmart eTextbook, visit www.coursesmart.com.

Acknowledgments

The authors are fortunate to have had 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, and frequently rejuvenated us during periods of waning energy for this project.

We would like to recognize the efforts of the many faculty and practicing systems analysts who have been reviewers of the six editions of this text and its associated text, *Modern 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:

Richard Allen, *Richland Community College*
Charles Arbutina, *Buffalo State College*
Paula Bell, *Lock Haven University of Pennsylvania*
Sultan Bhimjee, *San Francisco State University*
Bill Boroski, *Trident Technical College*
Nora Braun, *Augsburg College*
Rowland Brengle, *Anne Arundel Community College*
Richard Burkhard, *San Jose State University*
Doloras Carlisle, *Western Oklahoma State College*
Pam Chapman, *Waubonsee Community College*
Edward Chen, *University of Massachusetts Lowell*
Suzanne Clayton, *Drake University*
Garry Dawdy, *Metropolitan State College of Denver*
Thomas Dillon, *James Madison University*
Brad Dyer, *Hazard Community and Technical College*
Veronica Echols-Noble, *DeVry University–Chicago*

Richard Egan, *New Jersey Institute of Technology*
Gerald Evans, *University of Montana*
Lawrence Feidelman, *Florida Atlantic University*
David Firth, *University of Montana*
John Fowler, *Walla Walla Community College*
Larry Fudella, *Erie Community College*
Carol Grimm, *Palm Beach Community College*
Carol Healy, *Drake University*
Lenore Horowitz, *Schenectady County Community College*
Daniel Ivancevich, *University of North Carolina–Wilmington*
Jon Jasperson, *University of Oklahoma*
Len Jessup, *Washington State University*
Rich Kepenach, *St. Petersburg College*
Lin Lin, *Lehigh University*
James Scott Magruder, *University of Southern Mississippi*
Diane Mayne-Stafford, *Grossmont College*
David McNair, *Maryville University*
Loraine Miller, *Cayuga Community College*
Klara Nelson, *University of Tampa*
Max North, *Southern Polytechnic State University*
Doncho Petkov, *Eastern Connecticut State University*
Lou Pierro, *Indiana University*
Selwyn Piramuthu, *University of Florida*
Mitzi Pitts, *University of Memphis*
Richard Platt, *University of West Florida*
James Pomykalski, *Susquehanna University*
Robin Poston, *University of Memphis*
Rao Prabhakar, *Amarillo College*
Mary Prescott, *University of Tampa*
Joseph Rottman, *University of Missouri, St. Louis*
Robert Saldarini, *Bergen Community College*
Howard Schuh, *Rockland Community College*
Elaine Seeman, *Pitt Community College*
Teresa Shaft, *The University of Oklahoma*
Thomas Shaw, *Louisiana State University*
Gary Templeton, *Mississippi State University*
Dominic Thomas, *University of Georgia*
Don Turnbul, *The University of Texas at Austin*
Kathleen Voge, *University of Alaska–Anchorage*
Erica Wagner, *Portland State University*
Sharon Walters, *Southern Illinois University*
Haibo Wang, *Texas A&M International University*
Mark Ward, *Southern Illinois University, Edwardsville*
Merrill Warkentin, *Northeastern University*
June Wei, *University of West Florida*
Mudasser Wyne, *University of Michigan–Flint*
Saeed Yazdain, *Lane College*
Liang Yu, *San Francisco State University*
Steven Zeltmann, *University of Central Arkansas*
Justin Zhang, *Eastern New Mexico University*
Wen-Bin "Vincent" Yu, *Missouri University of Science and Technology*
Gary Kappenman, *Southeast Technical Institute*

We extend a special note of thanks to Jeremy Alexander, who was instrumental in conceptualizing and writing the initial version of the Pine Valley Furniture WebStore feature that appears in Chapters 3 through 10. The addition of this feature has helped make those chapters more applied and innovative. We also want to thank Jeff Jenkins, Brigham Young University, for the help he provided with the Visual Basic and .NET related materials in Chapter 8.

In addition, we want to thank John Russo for his work on the Instructor's Resource Manual, Test Bank, and PowerPoint presentations of Essentials of Systems Analysis and Design.

We also wish to thank Atish Sinha of the University of Wisconsin–Milwaukee for writing the initial draft of Appendix A on object-oriented analysis and design. 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, MS, 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.

Thanks also go to V. Ramesh (Indiana University) and Heikki Topi (Bentley College) for their assistance in coordinating this text with its companion book—*Modern Database Management*, also by Pearson.

Finally, we have been fortunate to work with a large number of creative and insightful people at Pearson, who have added much to the development, format, and production of this text. We have been thoroughly impressed with their commitment to this text and to the IS education market. These people include Nicole Sam, Acquisitions Editor; Anne Fahlgren, Executive Marketing Manager; Denise Vaughn, Program Manager; Judy Leale, Project Manager Team Lead;

Karalyn Holland, Project Manager; and Janet Slowik, Senior Art Director. We especially thank our Executive Editor for the past twelve years, Bob Horan. Bob, thanks so much for your vision and support over all these years. Have a wonderful and well-deserved retirement.

The writing of this text has involved thousands of hours of time from the authors and from all of the people listed. 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 this book might achieve.

About the Authors

Joseph S. Valacich is an Eller Professor of Management Information Systems in the Eller College of Management at the University of Arizona. He has had visiting faculty appointments at Buskerud College (Norway), City University of Hong Kong, Norwegian University of Life Sciences, Riga Technical University (Latvia), and Helsinki School of Economics and Business. He received a Ph.D. degree from the University of Arizona (MIS), and MBA and BS (computer science) degrees from the University of Montana. His teaching interests include systems analysis and design, collaborative computing, project management, and management of information systems. Professor Valacich cochaired the national task forces to design *IS 2010: The Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems*. He also served on the Executive Committee, funded by the National Science Foundation, to define the *IS Program Accreditation Standards* and on the Board of Directors for CSAB (formally, the Computing Sciences Accreditation Board), representing the Association for Information Systems (AIS). He was the general conference co-chair for the 2003 International Conference on Information Systems (ICIS), and the co-chair for the Americas' Conference on Information Systems (AMCIS) in 2012.

Prior to his academic career, Dr. Valacich worked in the information systems field as a programmer, systems analyst, and technical product manager. He has conducted numerous corporate training and executive development programs for organizations, including AT&T, Boeing, Dow Chemical, EDS, Exxon, FedEx, General Motors, Microsoft, and Xerox.

Dr. Valacich is the co-Editor-in-Chief for *AIS Transactions on Human-Computer Interaction*, a senior editor at *MIS Quarterly*, and was formerly an associate editor for *Information Systems Research*. He has published more than 200 scholarly articles in numerous prestigious journals and conferences. His scholarly work has had a tremendous impact not only on the field of information systems, but also on a number of other disciplines, including computer science, cognitive and social psychology, marketing, and management. In February 2014, Google Scholar lists his citation counts at over 13,800, with an H-index of 54. He is also a coauthor of the leading *Modern Systems Analysis and Design* (Seventh Edition) and *Information Systems Today* (Seventh Edition).

Joey F. George is professor of information systems and the John D. DeVries Endowed Chair in Business at the Iowa State University College of Business. Dr. George earned his bachelor's degree at Stanford University in 1979 and his Ph.D. in management at the University of California at Irvine in 1986. He was previously the Edward G. Schlieder Chair of Information Systems in the E. J. Ourso College of Business Administration at Louisiana State University. He also served at Florida State University as Chair of the Department of Information and Management Sciences from 1995 to 1998.

Dr. George has published dozens of articles in such journals as *Information Systems Research*, *Communications of the ACM*, *MIS Quarterly*, *Journal of*

MIS, and *Communication Research*. His research interests focus on the use of information systems in the workplace, including computer-based monitoring, computer-mediated deceptive communication, and group support systems.

Dr. George is coauthor of the textbooks *Modern Systems Analysis and Design*, *Seventh Edition*, published in 2014, and *Object-Oriented Systems Analysis and Design*, *Second Edition*, published in 2007, both from Pearson. He has served as an associate editor and senior editor for both *MIS Quarterly* and *Information Systems Research*. He served three years as the editor-in-chief of the *Communications of the AIS*. Dr. George was the conference co-chair for the 2001 ICIS, held in New Orleans, Louisiana; conference chair for the 2012 ICIS, held in Orlando, Florida; and the doctoral consortium co-chair for the 2003 ICIS, held in Seattle, Washington. He is a Fellow of the Association for Information Systems (AIS) and served as President of AIS in 2010–11.

Jeffrey A. Hoffer is the Sherman–Standard Register Professor of Data Management for the Department of MIS, Operations Management, and Decision Sciences in the School of Business Administration at the University of Dayton. He also taught at Indiana University and Case Western Reserve University. Dr. Hoffer earned his BA from Miami University in 1969 and his Ph.D. from Cornell University in 1975.

Dr. Hoffer has coauthored all editions of three college textbooks: *Modern Systems Analysis and Design*, with George and Valacich; *Managing Information Technology: What Managers Need to Know*, with Brown, DeHayes, Martin, and Perkins; and *Modern Database Management*, with Ramesh and Topi, all published by Pearson Prentice Hall. His research articles have appeared in numerous journals, including the *MIS Quarterly–Executive*, *Journal of Database Management*, *Small Group Research*, *Communications of the ACM*, and *Sloan Management Review*. He has received research grants from Teradata (Division of NCR), IBM Corporation, and the U.S. Department of the Navy.

Dr. Hoffer is cofounder of the International Conference on Information Systems and Association for Information Systems and has served as a guest lecturer at the Catholic University of Chile, Santiago, and the Helsinki School of Economics and Business in Mikkeli, Finland.

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

Essentials of Systems Analysis and Design

one

The Systems Development Environment

Sam Edwards/Getty Images

Chapter Objectives

After studying this chapter, you should be able to:

- Define information systems analysis and design.
- Describe the role of the systems analyst in information systems development.
- Describe the information systems development life cycle (SDLC).
- List alternatives to the systems development life cycle, including a description of the role of computer-aided software engineering (CASE) tools in systems development.

Chapter Preview . . .

The key to success in business is the ability to gather, organize, and interpret information. Systems analysis and design is a proven methodology that helps both large and small businesses reap the rewards of utilizing information to its full capacity. As a systems analyst—the person in the organization most involved with systems analysis and design—you will enjoy a rich career path that will enhance both your computer and interpersonal skills.

The systems development life cycle (SDLC) is central to the development of an efficient information system. We will highlight four key SDLC steps: (1) planning and selection, (2) analysis, (3) design, and (4) implementation and operation. Be aware that these steps may vary in each organization, depending on its goals. The SDLC is illustrated in Figure 1-1.

This text requires that you have a general understanding of computer-based information systems as provided in an introductory information systems course. This chapter previews systems analysis and lays the groundwork for the rest of the book.

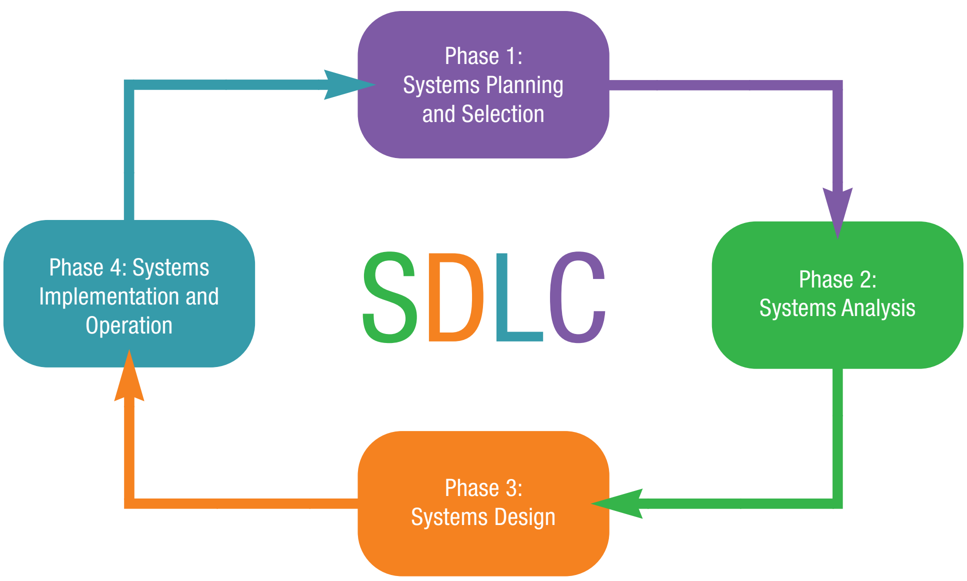


FIGURE 1-1
The four steps of the systems development life cycle (SDLC): (1) planning and selection, (2) analysis, (3) design, and (4) implementation and operation.

What Is Information Systems Analysis and Design?

Information systems analysis and design
The process of developing and maintaining an information system.

Information systems analysis and design is a method used by companies ranging from IBM to PepsiCo to Sony to create and maintain information systems that perform basic business functions such as keeping track of customer names and addresses, processing orders, and paying employees. The main goal of systems analysis and design is to improve organizational systems, typically through applying software that can help employees accomplish key business tasks more easily and efficiently. As a systems analyst, you will be at the center of developing this software. The analysis and design of information systems are based on:

- Your understanding of the organization's objectives, structure, and processes
- Your knowledge of how to exploit information technology for advantage

To be successful in this endeavor, you should follow a structured approach. The SDLC, shown in Figure 1-1, is a four-phased approach to identifying, analyzing, designing, and implementing an information system. Before we talk about the SDLC, we first describe what is meant by systems analysis and design.

Systems Analysis and Design: Core Concepts

Application software
Software designed to process data and support users in an organization. Examples include spreadsheets, word processors, and database management systems.

The major goal of systems analysis and design is to improve organizational systems. Often this process involves developing or acquiring **application software** and training employees to use it. Application software, also called a *system*, is designed to support a specific organizational function or process, such as inventory management, payroll, or market analysis. The goal of application software is to turn data into information. For example, software developed for the inventory department at a bookstore may keep track of the number of books in stock of the latest bestseller. Software for the payroll department may keep track of the changing pay rates of employees. A variety of off-the-shelf application software can be purchased, including TurboTax, Excel, and Photoshop. However, off-the-shelf software may not fit the needs of a particular organization, and so the organization must develop its own product.

In addition to application software, the information system includes:

- The hardware and systems software on which the application software runs. Note that the systems software helps the computer function, whereas the application software helps the user perform tasks such as writing a paper, preparing a spreadsheet, and linking to the Internet.
- Documentation and training materials, which are materials created by the systems analyst to help employees use the software they've helped create.
- The specific job roles associated with the overall system, such as the people who run the computers and keep the software operating.
- Controls, which are parts of the software written to help prevent fraud and theft.
- The people who use the software in order to do their jobs.

The components of a computer-based information system application are summarized in Figure 1-2. We address all the dimensions of the overall system, with particular emphasis on application software development—your primary responsibility as a systems analyst.

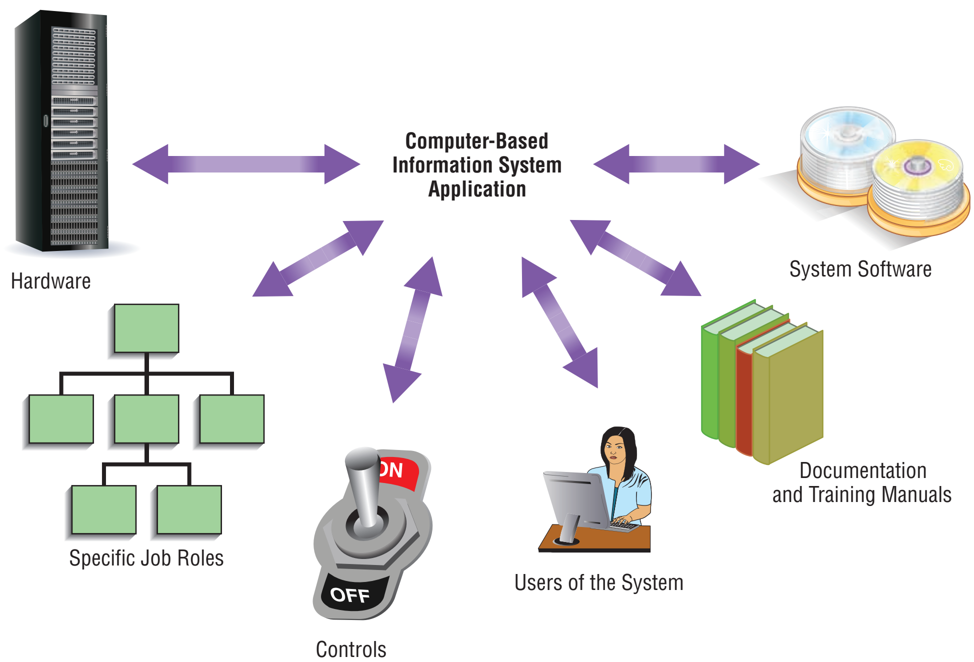


FIGURE 1-2
Components of a computer-based information system application.

Our goal is to help you understand and follow the software engineering process that leads to the creation of information systems. As shown in Figure 1-3, proven methodologies, techniques, and tools are central to software engineering processes.

Methodologies are a sequence of step-by-step approaches that help develop your final product: the information system. Most methodologies incorporate several development techniques, such as direct observations and interviews with users of the current system.

Techniques are 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 with current and future users of the information system to determine what your system should do, planning and managing the activities in a systems development project, diagramming how the system will function, and designing the reports, such as invoices, your system will generate for its users to perform their jobs.

Tools are computer programs, such as computer-aided software engineering (CASE) tools, that make it easy to use specific techniques. These three elements—methodologies, techniques, and tools—work together to form an organizational approach to systems analysis and design.

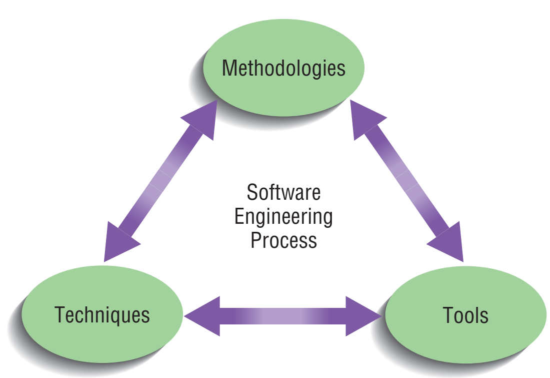


FIGURE 1-3
The software engineering process uses proven methodologies, techniques, and tools.

In the rest of this chapter, you will learn about approaches to systems development—the data- and process-oriented approaches. You will also identify the various people who develop systems and the different types of systems they develop. The chapter ends with a discussion of some of the methodologies, techniques, and tools created to support the systems development process. Before we talk more about computer-based information systems, let's briefly discuss what we mean by the word *system*.

Systems

The key term used most frequently in this book is *system*. Understanding systems and how they work is critical to understanding systems analysis and design.

Definition of a System and Its Parts

System
A group of interrelated procedures used for a business function, with an identifiable boundary, working together for some purpose.

A **system** is an interrelated set of business procedures (or components) used within one business unit, working together for some purpose. For example, a system in the payroll department keeps track of checks, whereas an inventory system keeps track of supplies. The two systems are separate. A system has nine characteristics, seven of which are shown in Figure 1-4. A detailed explanation of each characteristic follows, but from the figure you can see that a system exists within a larger world, an environment. A boundary separates the system from its environment. The system takes input from outside, processes it, and sends the resulting output back to its environment. The arrows in the figure show this interaction between the system and the world outside of it.

1. Components
2. Interrelated components
3. Boundary
4. Purpose
5. Environment
6. Interfaces
7. Constraints
8. Input
9. Output

Component
An irreducible part or aggregation of parts that makes up a system; also called a *subsystem*.

Interrelated
Dependence of one part of the system on one or more other system parts.

Boundary
The line that marks the inside and outside of a system and that sets off the system from its environment.

Purpose
The overall goal or function of a system.

Environment
Everything external to a system that interacts with the system.

A system is made up of components. A **component** is either an irreducible part or an aggregate of parts, also called a *subsystem*. The simple concept of a component is very powerful. For example, as with an automobile or a stereo system, with proper design, we can repair or upgrade the system by changing individual components without having to make changes throughout the entire system. The components are **interrelated**; that is, the function of one is somehow tied to the functions of the others. For example, the work of one component, such as producing a daily report of customer orders received, may not progress successfully until the work of another component is finished, such as sorting customer orders by date of receipt. A system has a **boundary**, within which all of its components are contained and that establishes the limits of a system, separating it from other systems. Components within the boundary can be changed, whereas systems outside the boundary cannot be changed. All of the components work together to achieve some overall **purpose** for the larger system: the system's reason for existing.

A system exists within an **environment**—everything outside the system's boundary that influences the system. For example, the environment of a state university includes prospective students, foundations and funding agencies, and

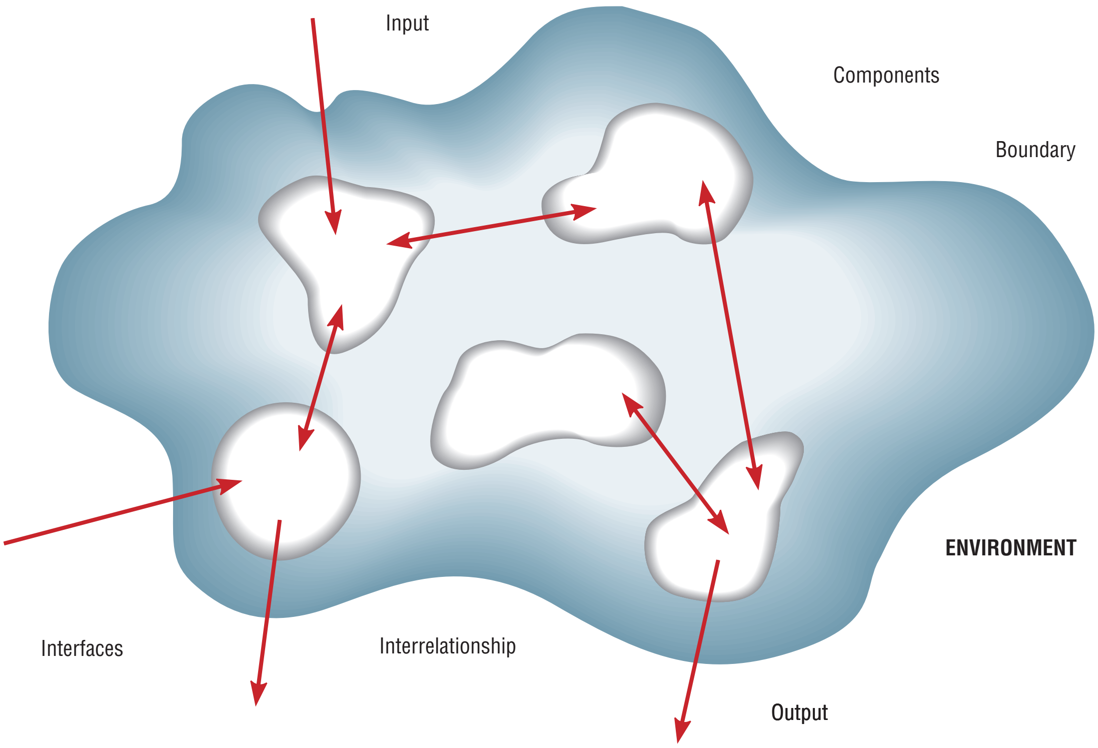


FIGURE 1-4
Seven characteristics of a system.

the news media. Usually the system interacts with its environment. A university interacts with prospective students by having open houses and recruiting from local high schools. An information system interacts with its environment by receiving data (raw facts) and information (data processed in a useful format). Figure 1-5 shows how a university can be seen as a system. The points at which the system meets its environment are called **interfaces**; an interface also occurs between subsystems.

Interface
Point of contact where a system meets its environment or where subsystems meet each other.

In its functioning, a system must face **constraints**—the limits (in terms of capacity, speed, or capabilities) to what it can do and how it can achieve its purpose within its environment. Some of these constraints are imposed inside the system (e.g., a limited number of staff available), and others are imposed by the environment (e.g., due dates or regulations). A system takes input from its environment in order to function. People, for example, take in food, oxygen, and water from the environment as input. You are constrained from breathing fresh air if you're in an elevator with someone who is smoking. Finally, a system returns output to its environment as a result of its functioning and thus achieves its purpose. The system is constrained if electrical power is cut.

Constraint
A limit to what a system can accomplish.

Important System Concepts

Systems analysts need to know several other important systems concepts:

- Decomposition
- Modularity
- Coupling
- Cohesion

Decomposition is the process of breaking down a system into its smaller components. These components may themselves be systems (subsystems) and can be broken down into their components as well. How does decomposition

Decomposition
The process of breaking the description of a system down into small components; also known as *functional decomposition*.

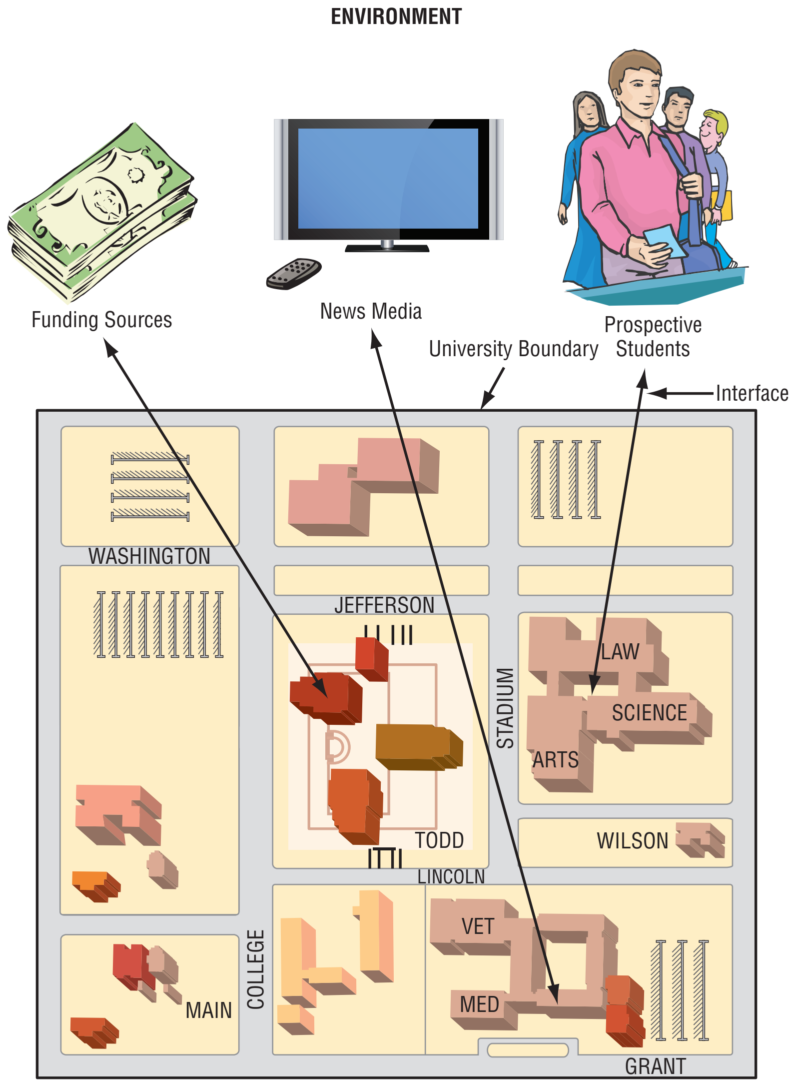


FIGURE 1-5
A university as a system.

aid understanding of a system? It results in smaller and less complex pieces that are easier to understand than larger, complicated pieces. Decomposing a system also allows us to focus on one particular part of a system, making it easier to think of how to modify that one part independently of the entire system. Decomposition is a technique that allows the systems analyst to:

- Break a system into small, manageable, and understandable subsystems
- Focus attention on one area (subsystem) at a time, without interference from other areas
- Concentrate on the part of the system pertinent to a particular group of users, without confusing users with unnecessary details
- Build different parts of the system at independent times and have the help of different analysts

Figure 1-6 shows the decomposition of a portable MP3 player. Decomposing the system into subsystems reveals the system's inner workings. You can decompose an MP3 player into at least three separate physical subsystems. (Note that decomposing the same MP3 player into *logical* subsystems would

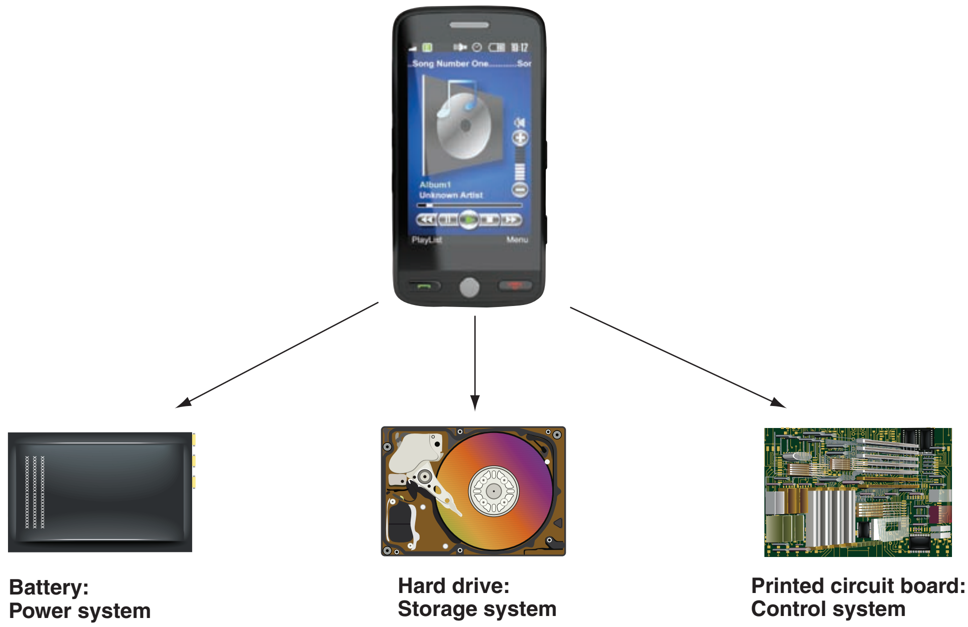


FIGURE 1-6
An MP3 player is a system with power supply, storage, and control subsystems.
Source: Shutterstock.

result in a different set of subsystems.) One subsystem, the battery, supplies the power for the entire system to operate. A second physical subsystem, the storage system, is made up of a hard drive that stores thousands of MP3 recordings. The third subsystem, the control subsystem, consists of a printed circuit board (PCB), with various chips attached, that controls all of the recording, playback, and access functions. Breaking the subsystems down into their components reveals even more about the inner workings of the system and greatly enhances our understanding of how the overall system works.

Modularity is a direct result of decomposition. It refers to dividing a system into chunks or modules of a relatively uniform size. Modules can represent a system simply, making it easier to understand and easier to redesign and rebuild. For example, each of the separate subsystem modules for the MP3 player in Figure 1-6 shows how decomposition makes it easier to understand the overall system.

Modularity
Dividing a system up into chunks or modules of a relatively uniform size.

Coupling means that subsystems are dependent on each other. Subsystems should be as independent as possible. If one subsystem fails and other subsystems are highly dependent on it, the others will either fail themselves or have problems functioning. Looking at Figure 1-6, we would say the components of a portable MP3 player are tightly coupled. The best example is the control system, made up of the printed circuit board and its chips. Every function the MP3 player can perform is enabled by the board and the chips. A failure in one part of the circuit board would typically lead to replacing the entire board rather than attempting to isolate the problem on the board and fix it. Even though repairing a circuit board in an MP3 player is certainly possible, it is typically not cost effective; the cost of the labor expended to diagnose and fix the problem may be worth more than the value of the circuit board itself. In a home stereo system, the components are loosely coupled because the subsystems, such as the speakers, the amplifier, the receiver, and the CD player, are all physically separate and function independently. If the amplifier in a home stereo system fails, only the amplifier needs to be repaired.

Coupling
The extent to which subsystems depend on each other.

Cohesion is the extent to which a subsystem performs a single function. In the MP3 player example, supplying power is a single function.

Cohesion
The extent to which a system or subsystem performs a single function.