DENNIS • WIXOM • ROTH

SYSTEMS ANALYSIS & DESIGN

6TH EDITION

WILEY

Visible®
SYSTEMS CORPORATION

Visible Analyst Student Edition

Educating tomorrow's developers today

Visible Analyst is a "hands-on" tool for teaching students all aspects of analysis and design including dynamic rules, consistency checking, managing change, and understanding the integration issues across an IT project. Visible Analyst prepares students to enter the IT world as business or data architects, analysts, designers, and modelers.

Visit us at **www.visible.com** to learn more.

YOU CAN Start Today with the Visible Analyst!

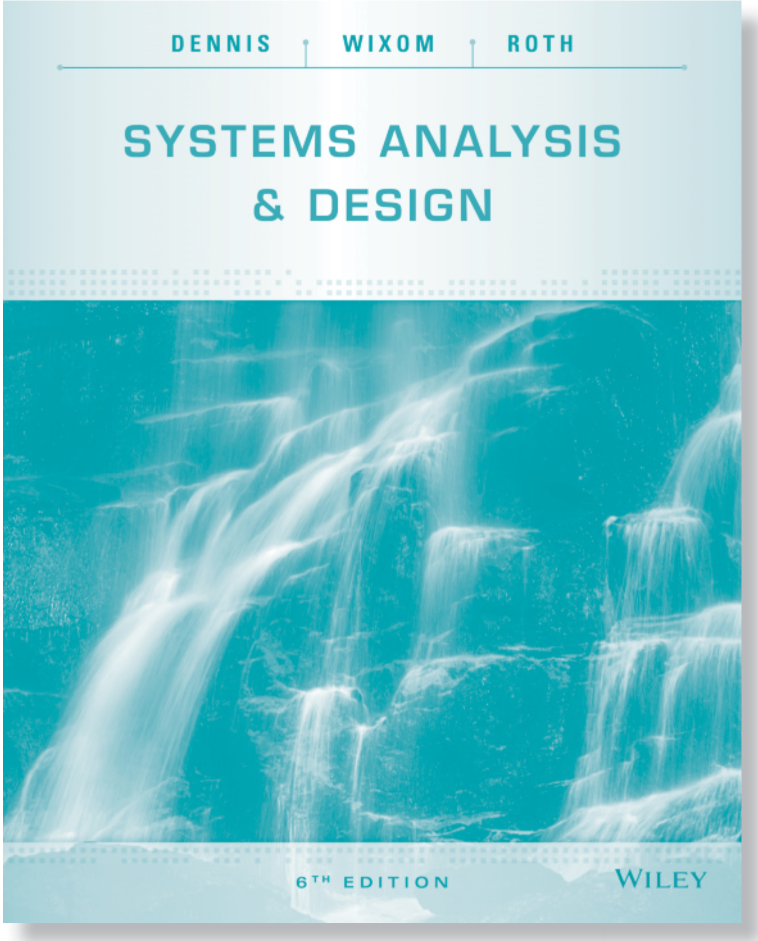
Only takes 2 minutes to install!
Save... 33% discount!

Please visit
http://store.visible.com/Wiley.aspx
to purchase and register with your information (see below) and obtain a valid license for your student edition of the software. To purchase the discounted software you will need to enter the following code:

978112014

Email support is provided to all registered students at support@visible.com. Your registration includes

- the latest release of the Visible Analyst Student Edition (software)
- the Visible Analyst eTutorial
- a preloaded Sample Instructional Project
- access to Webcast "How to" and "Get Started" Instructional Videos.



DENNIS WIXOM ROTH

SYSTEMS ANALYSIS & DESIGN

6TH EDITION

WILEY

Disclaimer: The publisher of the textbook does not sponsor, review, or make decisions about Visible Analyst software, and will not be responsible for, or involved in, any changes to the software.

SYSTEMS ANALYSIS AND DESIGN

Sixth Edition

Alan Dennis
Indiana University

Barbara Haley Wixom
Massachusetts Institute of Technology

Roberta M. Roth
University of Northern Iowa

WILEY

VP & EXECUTIVE PUBLISHER	Don Fowley
EXECUTIVE EDITOR	Beth Lang Golub
CONTENT EDITOR	Mary O’Sullivan
ASSOCIATE EDITOR	Ellen Keohane
MARKETING MANAGER	Christopher Ruel
OPERATIONS MANAGER	Yana Mermel
ASSOCIATE PRODUCTION MANAGER	Joyce Poh
DESIGNER	Wendy Lai

This book was set by Aptara.

Library of Congress Cataloging-in-Publication Data

Dennis, Alan.
Systems analysis and design / Alan Dennis, Barbara Haley Wixom, Roberta M. Roth.—Sixth edition.
pages cm
Includes bibliographical references and index.

ISBN 978-1-118-89784-3 (paperback)
1. System design. 2. System analysis. 3. Computer architecture. I. Wixom, Barbara Haley, 1969– II. Roth, Roberta M. (Roberta Marie), 1955– III. Wixom, Barbara Haley. IV. Roth, Roberta M. V. Title.
QA76.9.S88D464 2014
004.2′2—dc23

2014034532

Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

To Kelly
To Chris, Haley, and Hannah
To Rich and the boys, always.

PREFACE

PURPOSE OF THIS BOOK

Systems Analysis and Design (SAD) is an exciting, active field in which analysts continually learn new techniques and approaches to develop systems more effectively and efficiently. However, there is a core set of skills that all analysts need to know no matter what approach or methodology is used. All information systems projects move through the four phases of planning, analysis, design, and implementation; all projects require analysts to gather requirements, model the business needs, and create blueprints for how the system should be built; and all projects require an understanding of organizational behavior concepts like change management and team building.

This book captures the dynamic aspects of the field by keeping students focused on doing SAD while presenting the core set of skills that we feel every systems analyst needs to know today and in the future. This book builds on our professional experience as systems analysts and on our experience in teaching SAD in the classroom.

This book will be of particular interest to instructors who have students do a major project as part of their course. Each chapter describes one part of the process, provides clear explanations on how to do it, gives a detailed example, and then has exercises for the students to practice. In this way, students can leave the course with experience that will form a rich foundation for further work as a systems analyst.

OUTSTANDING FEATURES

A Focus on Doing SAD

The goal of this book is to enable students to do SAD—not just read about it, but understand the issues so that they can actually analyze and design systems. The book introduces each major technique, explains what it is, explains how to do it, presents an example, and provides opportunities for students to practice before they do it in a real-world project. After reading each chapter, the student will be able to perform that step in the system development life cycle (SDLC) process.

Rich Examples of Success and Failure

The book includes a running case about a fictitious company called Tune Source. Each chapter shows how the concepts are applied in situations at Tune Source. Unlike running cases in other books, this text focuses examples on planning, managing, and executing the activities described in the chapter, rather than on detailed dialogue between fictitious actors. In this way, the running case serves as a template that students can apply to their own work. Each chapter also includes numerous Concepts in Action boxes that describe how real companies succeeded—and failed—in performing the activities in the chapter. Many of these examples are drawn from our own experiences as systems analysts.

Incorporation of Object-Oriented Concepts and Techniques

The field is moving toward object-oriented concepts and techniques, both through UML 2.0, the new standard for object-oriented analysis and design, as well as by gradually incorporating object-oriented concepts into traditional techniques. We have taken two approaches to incorporating object-oriented analysis and design into the book. First, we have integrated several object-oriented concepts into our discussion of traditional techniques, although this may not be noticed by the students because few concepts are explicitly labeled as object-oriented concepts. For example, we include the development of use cases as the first step in process modeling (i.e., data flow diagramming) in Chapter 4, and the use (and reuse) of standard interface templates and use scenarios for interface design in Chapter 9.

Second, and more obvious to students, we include a final chapter on the major elements of UML 2.0 that can be used as an introduction to object-oriented analysis and design. This chapter can be used at the end of a course—while students are busy working on projects—or can be introduced after or instead of Chapters 5 and 6.

Real-World Focus

The skills that students learn in a systems analysis and design course should mirror the work that they ultimately will do in real organizations. We have tried to make this book as "real" as possible by building extensively on our experience as professional systems analysts for organizations such as IBM, the U.S. Department of Defense, and the Australian Army. We have also worked with diverse industry advisory boards of IS professionals and consultants in developing the book and have incorporated their stories, feedback, and advice throughout. Many students who use this book will eventually apply the skills on the job in a business environment, and we believe that they will have a competitive edge by understanding what successful practitioners feel is relevant in the real world.

Project Approach

We have presented the topics in this book in the SDLC order in which an analyst encounters them in a typical project. Although the presentation necessarily is linear (because students have to learn concepts in the way in which they build on each other), we emphasize the iterative, complex nature of SAD as the book unfolds. The presentation of the material should align well with courses that encourage students to work on projects, because it presents topics as students need to apply them.

Graphic Organization

The underlying metaphor for the book is doing SAD through a project. We have tried to emphasize this graphically throughout the book so that students can better understand how the major elements in the SDLC are related to each other. First, at the start of every major phase of the system development life cycle, we present a graphic illustration showing the major deliverables that will be developed and added to the electronic "project binder" during that phase. Second, at the start of each chapter, we present a checklist of key tasks or activities that will be performed to produce the deliverables associated with this chapter. These graphic elements—the deliverables tied to each phase and the task checklist tied to each chapter—can help students better understand how the tasks, deliverables, and phases are related to and flow from one to another.

Finally, we have highlighted important practical aspects throughout the book by marking boxes and illustrations with a “push pin.” These topics are particularly important in the practical day-to-day life of systems analysts and are the kind of topics that junior analysts should pull out of the book and post on the bulletin board in their office to help them avoid costly mistakes.

WHAT’S NEW IN THE SIXTH EDITION

The sixth edition contains several significant enhancements, including new and updated content, new Concepts in Action features, and updated exercises. The most significant content changes are:

- Expanded discussion of Agile methodologies, particularly Scrum, in chapter 2.
- Considerable reorganization of chapter 4 (Use Case Analysis) so as to clarify concepts and add more illustrations.
- New content on mobile application architectures in chapter 8.
- Extensive revision of chapter 9, User Interface Design, in order to streamline the presentation for added clarity. New content has been added to reflect some important current user interface concepts, including usability, user experience (UX), issues of designing for touch screen interfaces, and several additional user interface design tools, including site maps, wireframe diagrams, and wireflow diagrams.
- New content in the chapter 11, Database Design, on the newer NoSQL databases and the BigData concept.

Throughout the book, the chapter objectives have been revised to reflect active learning objectives. Chapter references to outside sources have been updated to current resources wherever possible. New Concepts in Action features appear throughout the book to provide updated, real-world illustrations of the textbook content.

For this edition, a series of tutorial lessons have been created that will teach students how to use and apply the Visible Analyst™ computer-assisted software engineering (CASE) software to a simple systems development project scenario. (Please see the **Supplements** section of this Preface for more information on purchasing Visible Analyst software at a reduced price for use in your course.)

We know that most professors and students find the Systems Analysis and Design class to have a lot of demanding content, particularly in those classes that include a significant project. Many professors would like their students to be able to experience first-hand how useful a CASE tool is to a systems analyst, but find it difficult to include instruction on a CASE tool in an already full course. The goal of these lessons is to enable students to learn the basics of the Visible Analyst CASE software with little involvement on the part of the professor. The lesson material is found on the student web site for this textbook (at www.wiley.com/college/dennis), with references to these lessons found in new Your-Turn boxes throughout the book. Professors have the flexibility to assign these tutorial lessons if they want to include the Visible Analyst software in their courses, but are also free to exclude this material if they prefer. The tutorial lessons have been written to provide students with a sufficient foundation to apply Visible Analyst to a more significant systems development project, should that be a part of their course.

ORGANIZATION OF THIS BOOK

This book is organized by the phases of the systems development life cycle (SDLC). Each chapter has been written to teach students specific tasks that analysts need to accomplish over the course of a project, and the deliverables that will be produced from the tasks. As students complete the book, tasks will be "checked off" and deliverables will be completed and stored in project folders.

Part 1 covers the first phase of the SDLC, the Planning Phase. Chapter 1 introduces the SDLC, the roles and skills needed for a project team, project initiation, the systems request, and feasibility analysis. Chapter 2 discusses project selection, the selection of an SDLC methodology for the project, and project management, with emphasis on the work plan, staffing plan, project charter, risk assessment, and tools used to help manage and control the project.

Part 2 presents techniques needed during the analysis phase. In Chapter 3, students are introduced to requirements determination and are taught a variety of analysis techniques to help with business process automation, business process improvement, and business process reengineering. Chapter 4 focuses on use cases, Chapter 5 covers process models, and Chapter 6 explains data models and normalization.

The Design Phase is covered in Part 3 of the textbook. In Chapter 7, students create an alternative matrix that compares custom, packaged, and outsourcing alternatives. Chapter 8 focuses on designing the system architecture, which includes the architecture design, hardware/software specification, and security plan. Chapter 9 focuses on the user interface and presents interface design; in this chapter, students learn how to create use scenarios, the interface structure diagram, interface standards, and interface prototypes. Finally, data storage design and program design are discussed in Chapters 10 and 11, which contain information regarding the data storage design, the program structure chart, and program specifications.

The Implementation Phase is presented in Chapters 12 and 13. Chapter 12 focuses on system construction, and students learn how to build and test the system. It includes information about the test plan and user documentation. Conversion is covered in Chapter 13, where students learn about the conversion plan, the change management plan, the support plan, and the project assessment.

Chapter 14 (online) provides a background of object orientation and explains several key object concepts supported by the standard set of object-modeling techniques used by systems analysts and developers. Then, we explain how to draw four of the most effective models in UML: the use case diagram, the sequence diagram, the class diagram, and the behavioral state machine diagram.

SUPPLEMENTS (www.wiley.com/college/dennis)

Online Instructors Manual

The instructors manual provides resources to support the instructor both in and out of the classroom:

- Short experiential exercises can be used to help students experience and understand key topics in each chapter.
- Short stories have been provided by people working in both corporate and consulting environments for instructors to insert into lectures to make concepts more colorful and real.
- Additional mini-cases for every chapter allow students to perform some of the key concepts that were learned in the chapter.
- Answers to end-of-chapter questions and exercises are provided.

Online Instructor's Resources

- PowerPoint slides, prepared by author Roberta Roth, are provided that instructors can tailor to their classroom needs and that students can use to guide their reading and studying activities.
- Test Bank includes a variety of questions ranging from multiple choice to essay-style questions. A computerized version of the Test Bank is also available.

Student Web Site

- Web Quizzes help students prepare for class tests. See www.wiley.com/college/dennis.

Wiley E-Text: Powered by VitalSource

This Wiley e-text offers students continuing access to materials for their course. Your students can access content on a mobile device, online from any Internet-connected computer, or by a computer via download. With dynamic features built into this e-text, students can search across content, highlight, and take notes that they can share with teachers and classmates. Readers will also have access to interactive images and embedded podcasts.

Visible Analyst

Wiley has partnered with Visible Analyst to give students a discounted price for Visible Analyst software, an intuitive modeling tool for all aspects of traditional or object-oriented systems analysis and design. All new copies of the text will have a Key Code (printed on a page near the front of this text) that will provide a discount on Visible Analyst software. To obtain the software, students should visit http://store.visible.com/Wiley.aspx and enter their Key Code. Students who buy a new print text or digital e-book will receive one-third off the price of a downloadable edition of the software with a 6-month license. With the software, they will also receive tutorials, how-to videos, and a sample project. Students who buy used copies of this text may buy Visible Analyst at full price using the URL provided.

Project Management Software

You can download a 60-day trial of Microsoft Project Professional 2013 from the following Web site: http://technet.microsoft.com/en-us/evalcenter/hh973401. Note that Microsoft has changed its policy and no longer offers the 120-day trial previously available.

Another option now available to education institutions adopting this Wiley title is a free introductory 3-year membership for DreamSpark Premium. DreamSpark Premium is designed to provide the easiest and most inexpensive way for academic departments to make the latest Microsoft software available in labs, classrooms, and on student and instructor PCs. Microsoft Project software is available through this Wiley and Microsoft publishing partnership, free of charge with the adoption of any qualified Wiley title. Each copy of Microsoft Project is the full version of the software, with no time limitation, and can be used indefinitely for educational purposes. Contact your Wiley sales representative for details. For more information about the DreamSpark Premium program, contact drmspkna@Microsoft.com.

ACKNOWLEDGMENTS

We extend our thanks to the many people who contributed to the preparation of this sixth and past editions. We are indebted to the staff at John Wiley & Sons for their support, including Beth Lang Golub, Executive Editor; Christopher Ruel, Marketing Manager; Joyce Poh, Associate Production Manager; and Wendy Lai, Senior Designer.

We would like to thank the following reviewers of the sixth edition for their helpful and insightful comments:

Name	**School**
Las Adams	*Bethune-Cookman University*
Jon W. Beard	*George Mason University*
Claudia Howery	*Delta College*
Glynn Johnson	*University of Houston*
Kay Lee	*University of Kansas*
Long Li	*Grambling State University*
Suzanne Nordhaus	*Lee College*

We would like to thank the many practitioners from private practice, public organizations, and consulting firms for helping us add a real-world component to this project. A special remembrance goes to Matt Anderson from Accenture, who was a role model for all who knew him—who demonstrated excellence in systems analysis and design and in life in general.

Thanks also to our families and friends for their patience and support along the way, especially to Christopher, Haley, and Hannah Wixom; Alec Dennis; and Richard Jones.

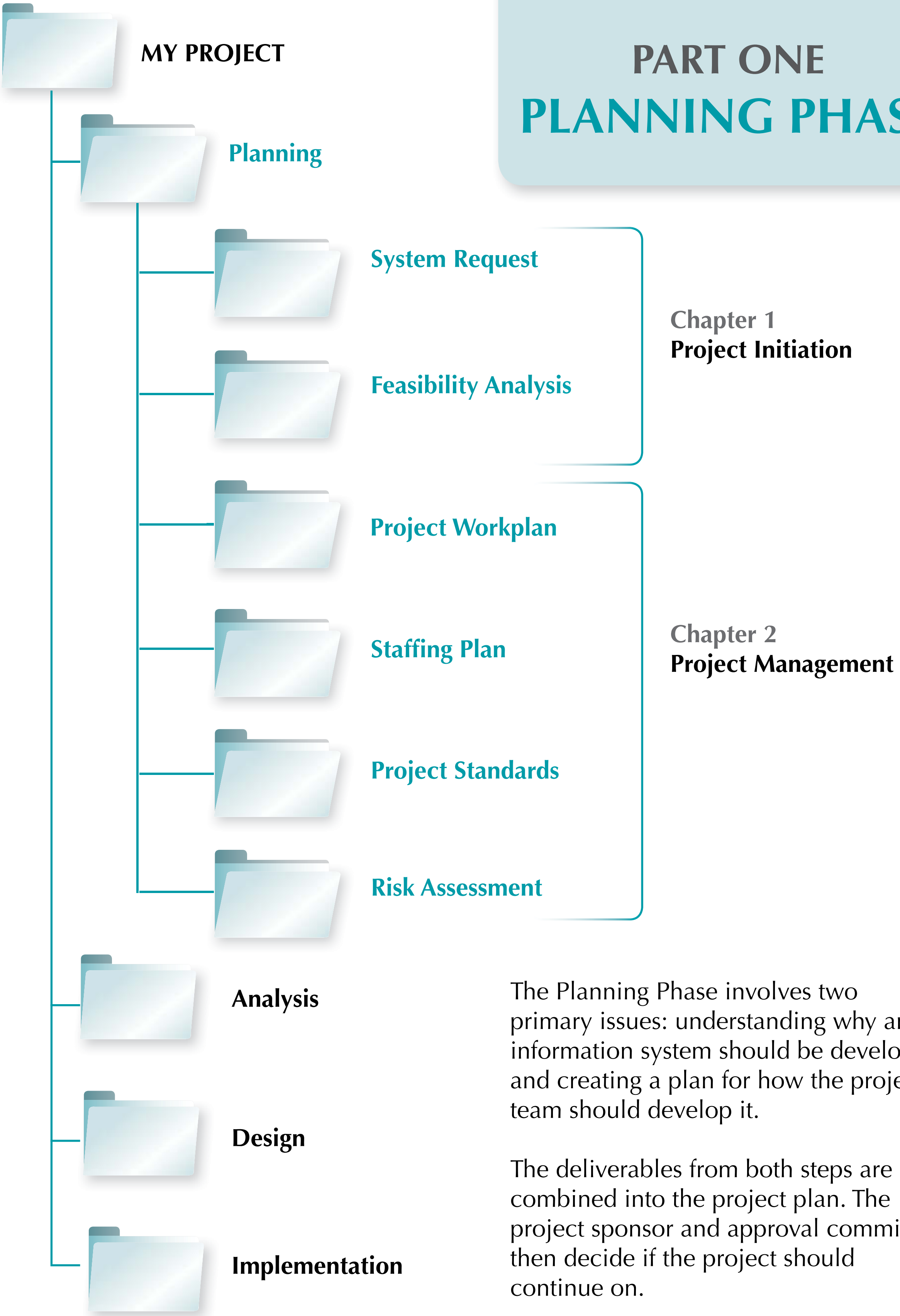
Alan Dennis
ardennis@indiana.edu

Barb Wixom
bwixom@mindspring.com

Robby Roth
Roberta.Roth@uni.edu

CONTENTS

PART ONE
PLANNING PHASE



MY PROJECT

Planning

System Request

Feasibility Analysis

Chapter 1
Project Initiation

Project Workplan

Staffing Plan

Project Standards

Risk Assessment

Chapter 2
Project Management

Analysis

Design

Implementation

The Planning Phase involves two primary issues: understanding why an information system should be developed and creating a plan for how the project team should develop it.

The deliverables from both steps are combined into the project plan. The project sponsor and approval committee then decide if the project should continue on.

CHAPTER 1

THE SYSTEMS ANALYST AND INFORMATION SYSTEMS DEVELOPMENT

PLANNING

TASK CHECKLIST

- □ Identify project.
- □ Develop systems request.
- □ Analyze technical feasibility.
- □ Analyze economic feasibility.
- □ Analyze organizational feasibility.
- □ Perform project selection review.
- □ Estimate project time.
- □ Identify project tasks.
- □ Create work breakdown structure.
- □ Create PERT Charts.
- □ Create Gantt Charts.
- □ Manage scope.
- □ Staff project.
- □ Create project charter.
- □ Set up CASE repository.
- □ Develop standards.
- □ Begin documentation.
- □ Assess and manage risk.

This chapter introduces the role of the systems analyst in information systems development projects. First, the fundamental four-stage systems development life cycle (planning, analysis, design, and implementation) is established as the basic framework for the information system (IS) development process. Next, ways in which organizations identify and initiate potential projects are discussed. The first steps in the process are to identify a project that will deliver value to the business and to create a system request that provides the basic information about the proposed system. Next, the analysts perform a feasibility analysis to determine the technical, economic, and organizational feasibility of the system.

OBJECTIVES

- Explain the role played in information systems development by the systems analyst.
- Describe the fundamental systems development life cycle and its four phases.
- Explain how organizations identify IS development projects.
- Explain the importance of linking the information system to business needs.
- Be able to create a system request.
- Describe technical, economic, and organizational feasibility assessment.
- Be able to perform a feasibility analysis.

INTRODUCTION

The *systems development life cycle* (SDLC) is the process of determining how an information system (IS) can support business needs, designing the system, building it, and delivering it to users.

If you have taken a programming class or have programmed on your own, you have probably experienced some success in developing small software applications. Creating high-quality information systems that meet expectations and provide meaningful value to organizations is a much more complex endeavor, however.

Numerous studies over the years report that projects involving information technology experience failure rates from 30 to 70%.[1] The definition of failure in these studies is often quite different, so the meaning of these statistics is hard to pin down. It is clear, though, that bringing an information system development project to a successful conclusion is difficult and many things need to be done right if we hope to achieve a positive outcome.

Although we would like to promote this book as a "silver bullet" that will keep you from experiencing failed IS projects, we must admit that such a silver bullet guaranteeing IS development success does not exist.[2] Instead, this book will provide you with many fundamental concepts and practical techniques that you can use to improve the likelihood of success.

The systems analyst plays a key role in the SDLC, analyzing the business situation, identifying opportunities for improvements, and designing an information system to implement the improvements. Many systems analysts view their profession as one of the most interesting, exciting, and challenging jobs around. As a systems analyst, you will work as a team with a variety of people, including business and technical experts. You will feel the satisfaction of seeing systems that you designed and developed make a significant positive business impact, while knowing that your unique skills helped make that happen.

It is important to remember that the primary objective of the systems analyst is not to create a wonderful system. The primary goal is to create value for the organization, which for most companies means increasing profits. (Government agencies and not-for-profit organizations measure value differently.) Many failed projects were abandoned because the analysts tried to build a wonderful system without clearly understanding how the system would support the organization's goals, improve business processes, and integrate with other information systems to provide value. An investment in an information system is like any other investment, such as a new machine tool. The goal is not to acquire the tool, because the tool is simply a means to an end; the goal is to enable the organization to perform work better so that it can earn greater profits or serve its constituents more effectively.

This book introduces you to the fundamental skills needed by a systems analyst. This is a pragmatic book that discusses best practices in systems development; it does not present a general survey of systems development that exposes you to everything about the topic. By definition, systems analysts *do things* and challenge the current way that an organization works. To get the most out of this book, you will need to actively apply the ideas and concepts in the examples and in the "Your Turn" exercises that are presented throughout to your own systems development project. This book will guide you through all the steps for delivering a successful information system. In the text, we illustrate how one organization, called Tune Source, applies the steps in one project, developing a Web-based digital music sales system. (Other illustrations of successful IS projects are provided on the course web site.) By the time

[1] Michael Krigsman, "CIO Analysis: Why 37 Percent of Project Fail", *zdnet.com*, accessed February 2014.

[2] The idea of using the silver bullet metaphor was first described in a paper by Frederick Brooks. See Frederick P. Brooks, Jr., "No Silver Bullet—Essence and Accident in Software Engineering," *Information Processing 1986, the Proceedings of the IFIP Tenth World Computing Conference,* H.-J. Kugler (ed.), 1986: 1069–76.

CONCEPTS IN ACTION

1-A Managerial Causes of IT Failures

A significant proportion of IT projects fail to fulfill their original objectives, resulting in wasted resources and a damaged reputation for the responsible IT department. In many cases, the causes of the failure are organizational issues, not technical issues.

Qantas, the Australian national airline, has endured two high-profile IT failures in recent years. In 1995, Project eQ, a 10-year technology services contract with IBM, was cancelled after four years, at a cost of $200 million. Poor planning contributed to the failure to upgrade a complex and unwieldy IT infrastructure saddled with 700-odd applications written in older programming languages.

In 2008, Qantas canceled Jetsmart, a $40 million parts-management system implementation, due in part to a dispute with the unionized users (aircraft mechanics) of the system. The union advised its members not to assist with the implementation, claiming the software unnecessarily increased the members' workload.

An analysis of these IT failures reveals several contributing factors. First, Qantas faced the challenges of a complicated technical infrastructure and outdated legacy applications. More significantly, however, was the failure of company leadership to understand basic IT issues. In public statements, the company CFO seemed not to care about the user perspectives on new software, preferring instead to put in what management thought was appropriate. This attitude, in part, led to union problems and claims of poorly designed, hard-to-use software and inadequate training.

Aging applications and an unwieldy technical infrastructure are challenges faced by many organizations today. But the senior-management attitude that seemingly disregards the views of software users casts serious questions about Qantas' prospects for IT project success in the future.

Adapted from: Michael Krigsman, "Quantas Airways: A Perfect Storm for IT Failures?", 2/29/08, *zdnet.com*, accessed March 2014.

you finish the book, you will not be an expert analyst, but you will be ready to start building systems for real.

In this chapter, we first describe the role of the systems analyst in information systems development projects. We discuss the wide range of skills needed to be successful in this role, and we explain various specialties that systems analysts may develop. We then introduce the basic SDLC that information systems projects follow. This life cycle is common to all projects and serves as a framework for understanding how information systems projects are accomplished. We discuss how projects are identified and initiated within an organization and how they are initially described in a system request. Finally, we describe the feasibility analysis that is performed, which drives the decision whether to proceed with the project.

THE SYSTEMS ANALYST

The systems analyst plays a key role in information systems development projects. The systems analyst works closely with all project team members so that the team develops the right system in an effective way. Systems analysts must understand how to apply technology to solve business problems. In addition, systems analysts may serve as *change agents* who identify the organizational improvements needed, design systems to implement those changes, and train and motivate others to use the systems.

Systems Analyst Skills

New information systems introduce change to the organization and its people. Leading a successful organizational change effort is one of the most difficult jobs that someone can do. Understanding what to change, knowing how to change it, and convincing others of the need for change require a wide range of skills. These skills can be broken down into six major categories: technical, business, analytical, interpersonal, management, and ethical.

Spotlight on Ethics-1

James is a systems analyst on a new account management system for Hometown National Bank. At a recent meeting with the project sponsor, James learned about some new ideas for the system that were not a part of the original project scope. Specifically, the bank's marketing director has asked that some of the data that will be collected by the new system from customers who open new checking and savings accounts also be used as the basis of a marketing campaign for various loan products the bank offers.

James is uncomfortable with the request. He is not sure the bank has the right to use a person's data for purposes other than the original intent. Who "owns" this data, the bank that collected it as a part of a customer opening an account, or the customer who the data describes? Should James insist that the customers give authorization to use "their" data in this way? Or should he say nothing and ignore the issue? Is it necessary (or appropriate) for a systems analyst to be an ethical watchdog in a systems development project? Why or why not?

Analysts must have the technical skills to understand the organization's existing technical environment, the new system's technology foundation, and the way in which both can be fit into an integrated technical solution. Business skills are required to understand how IT can be applied to business situations and to ensure that IT delivers real business value. Analysts are continuous problem solvers at both the project and the organizational level, and they put their analytical skills to the test regularly.

Often, analysts need to communicate effectively, one-on-one with users and business managers (who often have little experience with technology) and with programmers (who often have more technical expertise than the analyst does). They must be able to give presentations to large and small groups and to write reports. Not only do they need to have strong interpersonal abilities, but they also need to manage people with whom they work, and they must manage the pressure and risks associated with unclear situations.

Finally, analysts must deal fairly, honestly, and ethically with other project team members, managers, and system users. Analysts often deal with confidential information or information that, if shared with others, could cause harm (e.g., dissent among employees); it is important for analysts to maintain confidence and trust with all people.

Systems Analyst Roles

As organizations and technology have become more complex, most large organizations now build project teams that incorporate several analysts with different, but complementary roles. In smaller organizations, one person may play several of these roles. Here we briefly describe these roles and how they contribute to a systems development project.

YOUR TURN

1-1 Being an Analyst

Suppose you set a goal to become an analyst after you graduate. What type of analyst would you most prefer to be? Why does this particular analyst role appeal to you? What type of courses should you take before you graduate? What type of summer job or internship should you seek?

Question

Develop a short plan that describes how you will prepare for your career as an analyst.

The *systems analyst* role focuses on the IS issues surrounding the system. This person develops ideas and suggestions for ways that IT can support and improve business processes, helps design new business processes supported by IT, designs the new information system, and ensures that all IS standards are maintained. The systems analyst will have significant training and experience in analysis and design and in programming.

The *business analyst* role focuses on the business issues surrounding the system. This person helps to identify the business value that the system will create, develops ideas for improving the business processes, and helps design new business processes and policies. The business analyst will have business training and experience, plus knowledge of analysis and design.

The *requirements analyst* role focuses on eliciting the requirements from the stakeholders associated with the new system. As more organizations recognize the critical role that complete and accurate requirements play in the ultimate success of the system, this specialty has gradually evolved. Requirements analysts understand the business well, are excellent communicators, and are highly skilled in an array of requirements elicitation techniques (discussed in Chapter 3).

The *infrastructure analyst* role focuses on technical issues surrounding the ways the system will interact with the organization's technical infrastructure (hardware, software, networks, and databases). This person ensures that the new information system conforms to organizational standards and helps to identify infrastructure changes that will be needed to support the system. The infrastructure analyst will have significant training and experience in networking, database administration, and various hardware and software products. Over time, an experienced infrastructure analyst may assume the role of *software architect*, who takes a holistic view of the organization's entire IT environment and guides application design decisions within that context.

The *change management analyst* role focuses on the people and management issues surrounding the system installation. This person ensures that adequate documentation and support are available to users, provides user training on the new system, and develops strategies to overcome resistance to change. The change management analyst will have significant training and experience in organizational behavior and specific expertise in change management.

The *project manager* role ensures that the project is completed on time and within budget and that the system delivers the expected value to the organization. The project manager is often a seasoned systems analyst who, through training and experience, has acquired specialized project management knowledge and skills. More will be said about the project manager in the next chapter.

The roles and the names used to describe them may vary from organization to organization. In addition, there is no single typical career path through these professional roles. Some people may enter the field as a more technically oriented programmer/analyst. Others may enter as a business-oriented functional specialist with an interest in applying IT to solve business problems. As shown in Figure 1-1, those who are interested in the broad field of information systems development may follow a variety of paths during their career.

THE SYSTEMS DEVELOPMENT LIFE CYCLE

In many ways, building an information system is similar to building a house. First, the owner describes the vision for the house to the developer. Second, this idea is transformed into sketches and drawings that are shown to the owner and refined (often, through several drawings, each improving on the other) until the owner agrees that the pictures depict what he or she wants. Third, a set of detailed blueprints is developed that presents much more specific information about the house (e.g., the layout of rooms, placement of plumbing fixtures and

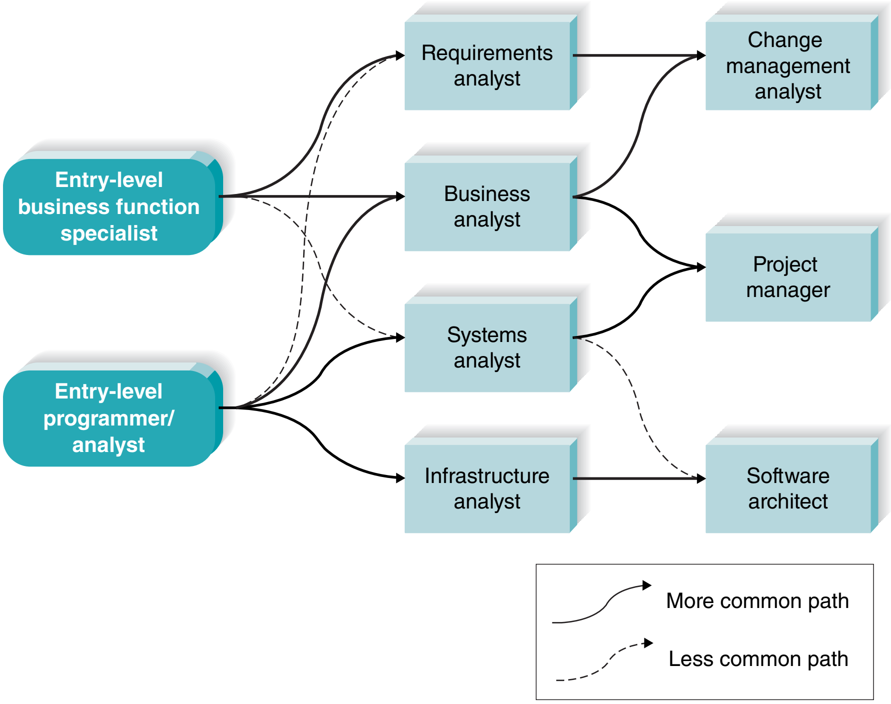


FIGURE 1-1
Career Paths for System Developers

electrical outlets, and so on). Finally, the house is built following the blueprints—and often with some changes and decisions made by the owner as the house is erected.

Building an information system using the SDLC follows a similar set of four fundamental *phases:* planning, analysis, design, and implementation (Figure 1-2). Each phase is itself composed of a series of *steps*, which rely on *techniques* that produce *deliverables* (specific documents and files that explain various elements of the system). Figure 1-3 provides more detail on the steps, techniques, and deliverables that are included in each phase of the SDLC and outlines how these topics are covered in this textbook.

Figures 1-2 and 1-3 suggest that the SDLC phases proceed in a logical path from start to finish. In some projects, this is true. In many projects, however, the project team moves through the steps consecutively, incrementally, iteratively, or in other patterns. Different projects may emphasize different parts of the SDLC or approach the SDLC phases in different ways, but all projects have elements of these four phases.

For now, there are two important points to understand about the SDLC. First, you should get a general sense of the phases and steps that IS projects move through and some of the techniques that produce certain deliverables. In this section, we provide an overview

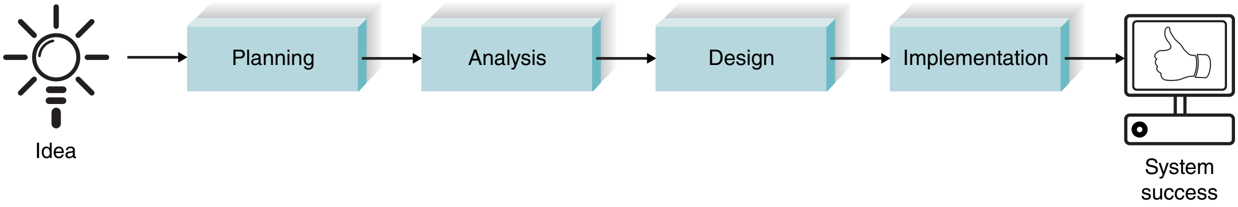


FIGURE 1-2 The Systems Development Life Cycle

Phase	Chapter	Step	Technique	Deliverable
Planning Focus: Why build this system? How to structure the project? Primary outputs: —System request with feasibility study —Project plan	1	Identify opportunity	Project identification	System request
	1	Analyze feasibility	Technical feasibility Economic feasibility Organizational feasibility	Feasibility study
	2	Develop workplan	Time estimation Task identification Work breakdown structure PERT chart Gantt chart Scope management	Project plan —Workplan
	2	Staff project	Project staffing Project charter	—Staffing plan
	2	Control and direct project	CASE repository Standards Documentation Timeboxing Risk management	—Standards list —Risk assessment
Analysis Focus: Who, what, where, and when for this system? Primary output —System proposal	3	Develop analysis strategy	Business process automation Business process improvement Business process reengineering	System proposal
	3	Determine business requirements	Interview JAD session Questionnaire Document analysis Observation	—Requirements definition
	4	Create use cases	Use case analysis	—Use cases
	5	Model processes	Data flow diagramming	—Process models
	6	Model data	Entity relationship modeling Normalization	—Data model
Design Focus: How will this system work? Primary output: —System specification	7	Design physical system	Design strategy	Alternative matrix System specification
	8	Design architecture	Architecture design Hardware and software selection	—Architecture report —Hardware and software specification
	9	Design interface	Use scenario Interface structure Interface standards Interface prototype Interface evaluation	—Interface design
	10	Design programs	Data flow diagramming Program structure chart Program specification	—Physical process model —Program design
	11	Design databases and files	Data format selection Entity relationship modeling Denormalization Performance tuning Size estimation	—Database and file specification —Physical data model
Implementation Focus: Delivery and support of completed system Primary output: —Installed system	12	Construct system	Programming Software testing Performance testing	Test plan Programs Documentation Migration plan
	13	Install system	Conversion strategy selection Training	—Conversion plan —Business contingency plan —Training plan
	13	Maintain system	Support selection System maintenance Project assessment	Support plan Problem report Change request
	13	Post-implementation	Post-implementation audit	Post-implementation audit report

FIGURE 1-3 Systems Development Life Cycle Phases

of the phases, steps, and some of the techniques that are used to accomplish the steps. Second, it is important to understand that the SDLC is a process of *gradual refinement*. The deliverables produced in the analysis phase provide a general idea what the new system will do. These deliverables are used as input to the design phase, which then refines them to produce a set of deliverables that describes in much more detailed terms exactly how the system should be built. These deliverables in turn are used in the implementation phase to guide the creation of the actual system. Each phase refines and elaborates on the work done previously.

Planning

The *planning phase* is the fundamental process of understanding *why* an information system should be built and determining how the project team will go about building it. It has two steps:

1. During *project initiation*, the system's business value to the organization is identified—how will it contribute to the organization's future success? Most ideas for new systems come from outside the IS area and are recorded on a system request. A *system request* presents a brief summary of a business need and explains how a system that addresses the need will create business value. The IS department works together with the person or department generating the request (called the *project sponsor*) to conduct a feasibility analysis. The *feasibility analysis* examines key aspects of the proposed project:

   - The technical feasibility (Can we build it?)
   - The economic feasibility (Will it provide business value?)
   - The organizational feasibility (If we build it, will it be used?)

   The system request and feasibility analysis are presented to an information systems *approval committee* (sometimes called a *steering committee*), which decides whether the project should be undertaken.

2. Once the project is approved, it enters *project management.* During project management, the *project manager* creates a *workplan*, staffs the project, and puts techniques in place to help control and direct the project through the entire SDLC. The deliverable for project management is a *project plan* that describes how the project team will go about developing the system.

Analysis

The *analysis phase* answers the questions of *who* will use the system, *what* the system will do, and *where* and *when* it will be used (Figure 1-3). During this phase, the project team investigates any current system(s), identifies improvement opportunities, and develops a concept for the new system. This phase has three steps:

1. An *analysis strategy* is developed to guide the project team's efforts. Such a strategy usually includes studying the current system (called the *as-is system*) and its problems, and envisioning ways to design a new system (called the *to-be system*).
2. The next step is *requirements gathering* (e.g., through techniques such as interviews, group workshops, or questionnaires). The analysis of this information—in conjunction with input from the project sponsor and many other people—leads to the development of a concept for a new system. The system concept is explained through a set of *requirement statements* and a set of business *analysis models* that describe how the business will operate if the new system is developed. The analysis models represent user/system interactions and the data and processes needed to support the underlying business process.