Chapter 1 - From Beginning to End: An Overview of Systems Analysis and Design

IM Table of Contents

Chapter Overview

This edition, the seventh, continues with the approach we began in the last edition to provide a much more example based learning opportunity. As such the entire first chapter is an example of a complete development project from beginning to end. Obviously, in a single chapter example, not every step is elaborated. However, by going through the entire process, students will be able to obtain a broad overview and a clear perspective of the entire development process.

The chapter first introduces several basic concepts that are needed to understand systems development. The first few sections define Systems Analysis, Systems Design, the Systems Development Life Cycle (SDLC), iterative development, Agile development and the six core processes of systems development.

The remainder of the chapter illustrates the first iteration of a typical development project for the Ridgeline Mountain Outfitters (RMO) Tradeshow system. It divides the project into seven steps, including pre-project activities and then six other development steps. For simplicity purposes, each step is assigned a day, so the project has Day 1 activities, Day 2 activities, etc. Throughout this set of daily activities, all six core processes are explained and illustrated. Sample models and diagrams are presented to give the students an introduction to some of the major techniques and models that will be taught later in the text.

At the end of the chapter, there is a chapter case, which has assignments for similar to the RMO Tradeshow system. Obviously the students do not have all the skills necessary to do an error-free complete development. The objective of the case and its assignments, is simply to allow the students to practice going through the entire development process so that they obtain a solid overview of systems development. It is suggested that grading for these assignments be lenient, and credit could even be given simply for completing the assignment, whether it is correct or not.

Learning Objectives

After reading this chapter, the student should be able to:

- Describe the purpose of systems analysis and design when developing information systems
- Explain the purpose of the system development life cycle (SDLC) and identify its six core processes
- Explain how information system methodologies provide guidelines for completing the six core processes of the SDLC
- Describe the characteristics of Agile methodologies and iterative system development
- Based on the Ridgeline Mountain Outfitters Tradeshow System example:
    - Describe how the six core processes of the SDLC are used in each iteration
    - Identify key documents used in planning a project
    - Identify key diagrams used in systems analysis and systems design

Instructor's Notes

Software Development and Systems Analysis and Design

Key Terms

- **computer application (app)**—a computer software program that executes on a computing device to carry out a specific function or set of related functions
- **information system**—a set of interrelated computer components that collects, processes, stores, and provides as output the information needed to complete business tasks
- **systems analysis**—those activities that enable a person to understand and specify what the new system should accomplish
- **systems design**—those system development activities that enable a person to describe in detail how the resulting information system will actually be implemented

Lecture Notes

The objective of this section is to introduce the student to the basic concepts of a **computer application** and an **information system**. The section includes an introduction to and definition of systems analysis and design and explains how it fits into the process of developing a new information system, i.e., systems development. **Systems analysis** means to understand what the system must do. **Systems design** means to specify how the components are configured to provide the solution.

Quick Quiz

Q: What is the basic purpose of a course in systems analysis and design?

A: To provide the student with the necessary tools to (1) understand and document the business need, i.e. requirements, (2) define a solution, (3) work in a team to build the solution and (4) launch the application so that it is in productive use.

Systems Development Life Cycle (SDLC)

Key Terms

- **project**—a planned undertaking that has a beginning and an end, and that produces some definite result
- **Systems Development Life Cycle (SDLC)—** a framework that identifies all the activities required to research, build, deploy, and often maintain an information system
- **system development process** or **methodology**—a set of comprehensive guidelines for carrying out all of the activities of each core process of the SDLC
- **Agile Development**—an information system development process that emphasizes flexibility and rapid response to anticipate new and changing requirements during development

Lecture Notes

The second section in this chapter is an explanation of the **Systems Development Life Cycle (SDLC)**. The SDLC defines all the activities required to develop a new system. There are many different versions of the SDLC. This section distills out six core process required for the development of any new system. In other words, these six core process are common to all types of SDLCs. By understanding these six core processes, students will not only be able to develop new systems effectively, but they will be able to adapt to any other SDLC that they may encounter in industry. The six core processes are: [Note: These six core processes are used throughout the textbook.]

1. Identify the problem or need and obtain approval to proceed.
2. Plan and monitor the project**—**what to do, how to do it, and who does it.
3. Discover and understand the details of the problem or the need.
4. Design the system components that solve the problem or satisfy the need.
5. Build, test, and integrate system components.
6. Complete system tests and then deploy the solution.

This section also introduces **Agile Development** as an iterative approach to systems development. Agile development will be covered in more detail in Chapter 10, but an introduction is appropriate and required in the context of the example in the chapter.

Agile projects must be agile and flexible. They must have procedures in place to allow for, anticipate, and even embrace changes and new requirements during the development process.

Quick Quiz

Q: What is the basic philosophy of Agile development?

A: That the user cannot predict all of the needs of a new system, so the development process must be structured to anticipate the many requirements changes that normally occur. The development process must be flexible and agile.

Q: What are the six core processes? A: See the list above.

Iterative Development

Key Terms

- **iterative development—** an approach to system development in which the system is "grown" piece by piece through multiple mini-projects called iterations

Lecture Notes

Finally before the development of the Tradeshow system begins, Iterative Development is introduced. Figure 1-5 illustrates how the six core processes are utilized across various iterations to develop a new stem. Core processes and iterative development are common themes for many types of SDLCs, including the Unified Process, Extreme Programming, and Scrum. Again, the approach in this textbook is to teach the students the general concepts so that they can apply them in their own activities or within the framework of a corporate environment.

Quick Quiz

Q: What are the key benefits of iterative development?

A: Quicker deployment of important portions of the system, being able to address tough problems early, and having a flexible development process that can respond to changing requirements.

Introduction to Ridgeline Mountain Outfitters (RMO)

Lecture Notes

The next section introduces Ridgeline Mountain Outfitters (RMO). RMO is a sportswear company that sells both its own branded products as well as well as other national brands. There are two systems for RMO that are discussed in the textbook. The Tradeshow system is a small system that serves as the example development project in Chapter 1. The Consolidated Sales and Marketing System (CSMS) is a major system that serves as the running example throughout the rest of the textbook. This chapter only introduces RMO and the Tradeshow system.

Developing RMO's Tradeshow System

Key Terms

- **Subsystem –** An identifiable and partitioned portion of the overall system

Lecture Notes

This development project is designed as a six-day iteration with some pre-project activities to get the project started. The activities in the pre-project and each of the six days provide the supporting detail for the six core processes for systems development.

> Note: There is a case at the end of the chapter with student assignments for pre-project activities and each of the six days of activities. There are more assignments than can reasonably be expected for a single class period, so you may want to judiciously choose those that fit your course objectives, or spread the assignments out over multiple class periods. Obviously students will not have the skills to develop all the required elements of a development project. The assignments are meant to give them a brief introduction to some of the tasks, models, and components that are created during development. At this point the benefit comes from the effort of thinking about those components, rather than producing correct models or solutions. It is suggested that grading for these assignments be on a pass-fail approach, where credit is given for any serious attempt.

Initial Project Activities

Pre-project activities are those activities required to get the project approved and started. The two major objectives are:

- Identify the problem and document the objectives of the solution system.
- Obtain approval in order to begin the project.

Systems Vision Document: In this section the System Vision Document is introduced. It provides a brief description of the problem, the basic functionality of the new system, e.g. system capabilities, and the business benefits to be derived by the new system. Students should be able to understand this document and even produce a simple version of it. Discuss each section of the document shown in Figure 1-8.

Day 1 Activities

RMO – Supplier Information System: The first day of an iteration is usually a planning day. In this project Day 1 will focus on planning the overall project, which is the objective of core process number 2. There are three activities that are completed:

1. Determine the major components (functional areas) that are needed. (Core Process 2)
2. Define the iterations and assigning each functional area to an iteration. (Core Process 2)
3. Determine team members and responsibilities. (Core Process 2)

Planning the Overall Project and the Project Iterations: During the first activity, it was determined that two **subsystems** are needed: The Supplier Information subsystem and the Product Information subsystem. It was decided that the remainder of the iteration would focus on the Supplier Information subsystem.

Planning the Rest of the First Iteration: The Supplier Subsystem: The next planning activities, Activities 2 and 3 above, are to plan the rest of the iteration. Planning each iteration requires three steps:

- Identify the tasks required for the iteration.
- Organize and sequence these tasks into a schedule.
- Identify required resources (especially people) and assign people to tasks.

The first step - identify the tasks required for the iteration - is done through the development of a Work Breakdown Structure. Figure 1-9 is a sample Work Breakdown Structure with specific tasks for the project team and estimates of the time required for each task.

In a full scale project, the Work Breakdown Structure is used to build a schedule, usually with some tool such as Microsoft Project. A simple version of a schedule can be done using a simple PERT/CPM chart, which is called a "work sequence" diagram. In order to keep the example simple in this chapter, we only show a draft copy of a work sequence. Sometimes a work sequence is developed using sticky notes on a white board. In the example in the textbook, we show it with boxes and connecting arrows as shown in Figure 1-10.

> Note: The students will not know how to identify project tasks. The objective for this topic is to show students that a project needs to organized and the work scheduled.

Quick Quiz

Q: What is the purpose of a Work Breakdown Structure?

> A: To identify all of the tasks that must be done to complete a particular iteration and produce the defined deliverable.

Q: What is the purpose of a work sequence diagram?

> A: To define the order of the work, e.g. the tasks in the WBS. It also helps to monitor the progress of the iteration. It also helps to assign resources to the tasks to avoid duplication and plan for tasks that require extensive time or resources.

<u>Day 2 Activities</u>

Fact Finding and User Involvement: Now that the project and the iteration have been planned, Day 2 initiates actual development work. Day 2 focuses on systems analysis activities. In particular three activities are required:

- Do preliminary fact-finding tasks to understand the requirements. (Core Process 3)
- Develop a preliminary list of use cases and a use case diagram. (Core Process 3)
- Develop a preliminary list of classes and a class diagram. (Core Process 3)

Identifying Use Cases: Fact finding is usually done by interviewing the users and other stakeholders. The use cases are documented in a table, as shown in Figure 1-11.

> Note: Students will not understand the techniques to identify use cases. Simply say that a use case is something that the user needs the system to do. Use a phrase such as, "The clerk *uses* the system to...." where the prepositional clause is the use case, such as "...to enter a new sale." The objective of this step is to teach the students that the functions of the new system must be identified.

Identifying Domain Classes: The classes are documented either in a table, as shown in Figure 1-12, or a class diagram as shown in Figure 1-13.

> Note: Students will not understand the techniques of how to identify classes. At this point simply try to identify "informational things" that the system must remember -- such things as a "supplier." The objective of this step is to teach the students that information systems always have information that is maintained and this is the method used to determine what is needed.

> Note: Figure 1-10 is included simply to introduce the idea of a model and a class diagram to the students. They are not expected to be able to develop or even understand the elements of a class diagram. At most they should see that classes are information components that have a name and detailed elements or attributes.

Quick Quiz

Q: What is the purpose of a list of user cases?

> A: To identify and list the functions that must be provided by the new system.

Q: What is the purpose of a table of classes or class diagram?

> A: To identify and list the information components that must be maintained by the new system.

Day 3 Activities

The purpose of Day 3 activities is to flesh out the details of each of the use cases identified earlier (in Day 2). Each use case supports some user work flow, e.g. a sequence of user tasks. It is important to understand the user work flow because in order to define the screens, reports, and system processing steps. The activities for day three are the following:

- Perform in-depth fact finding to understand details. (Core Process 3)
- Understand and document the detailed work flow of each use case. (Core Process 3)
- Define the user experience with screens and reports. (Core Processes 3 and 4)

We begin this process with a use case diagram, which is just another way to illustrate the use cases that have been identified. Figure 1-14 is a sample use case diagram.

> Note: The students will not understand the techniques required to develop a use case diagram. At this point in the course, they should just be aware of the fact that a use case diagram is one tool that is used to document use cases. They should be able to read a use case diagram.

Developing Use Case Descriptions and Workflow Diagrams: One powerful technique to understand the detail work flow of a particular use case is with a work flow diagram. A work flow diagram documents a particular user, his/her actions and what the system must do to respond to the user's actions. It is used to document the internal work flow of one use case. Figure 1-15 is a work flow diagram.

> Note: Student will not understand the techniques of how to develop a work flow diagram. They may be able to read a work flow diagram. Mostly they should understand that it is used to document the internal steps of a particular use case.

Defining Screen Layout: Another technique to define and describe the user actions is a screen layout. The user works with a computer system through the use of computer screens. Sample screens drawings are useful to describe how the user works with the system. Figure 1-16 is a sample screen layout.

> Note: Students will not understand the steps required to layout computer screens. They should be able to read and understand a sample screen layout. They should also be aware that sample screen layouts are an important technique for developing the user interface.

Quick Quiz

Q: What is the purpose of a use case diagram?

> A: To document the use cases and the actors in a system or subsystem.

Q: What is the purpose of a work flow diagram?

> A: To define the detailed processing steps of a use case.

Q: What is the purpose of sample screen layouts?

> A: To design the layout of a computer screen, including all the data elements required on the screen.

Day 4 Activities

Day 4 begins the design activities. Although analysis (understanding what) and design (structuring the how), often are done concurrently, we divide them into separate activities for learning purposes. Day 4 includes two major activities:

- Design the database structure (schema). (Core Process 4)
- Design the system’s high-level structure. (Core Process 4)

Designing the Database: Database design flows directly from the class diagram developed earlier. Figure1-17 shows a tabular form for the database. Figure 1-19 shows an enhanced class diagram, which provides the detail required to create the database schema.

A General Approach to Design: Designing the overall system structure or architecture is done using architecture configuration diagrams. Figure 1-18 shows an architectural diagram for the entire system, and Figure 1-20 shows a subsystem package diagram showing the mid-level design for the Supplier subsystem.

Designing the Software Components: These high-level design decisions will determine the detailed software components of the system. A browser-based system is structured and constructed differently than an application system that runs on a smart phone or a tablet computer.

Defining the Preliminary Design Class Diagrams: A design class diagram (DCD) identifies the OOP classes that will be needed for the system. The set of design classes includes problem domain classes, view layer classes, sometimes separate data access classes, and utility classes. Figure 1-19 is an example.

Designing the Subsystem Software Architecture: Once we have an overall structure and an overall approach for implementing the new system, we begin to drill down to the subsystem design. Figure 1-20 illustrates the architectural design of the Supplier Information subsystem. Notice that this subsystem is further divided into layers: a view layer and a model layer.

Managing the Project: Systems design is a rather complex endeavor. The students should understand that high-level architecture design should be done before low-level design and programming. It is important to have an overview of the entire system before building the detailed components.

> Note: These architecture diagrams can be quite complex and difficult to understand. The students should primarily be aware that architectural design models do exist (these and others that are not shown), and provide an effective tool to think about and document the system in its entirety. They should not be expected to be able to correctly create these models.

Quick Quiz

Q: What is the purpose of a design class diagram?

> A: To define the details of the information requirements in preparation for building the database schema.

Q: What is the purpose of the database schema?

> A: It is the detailed database structure. Used to build the database with MySql, Oracle, or SQLServer.

Q: What is the purpose of the system architectural configuration diagram?

> A: To configure the overall solution system.

Q: What is the purpose of the subsystem architectural design diagram?

> A: To define the individual components of a subsystem and identify how they relate to each other.

Day 5 Activities

Day 5 activities focus on programming the system. Figure 1-21 illustrates some PHP code for the SupplierView class.

Day 6 Activities

Day 6 is final testing and deployment. Figure 1-22 is a flow chart depicting the testing cycle. Figure 1-23 is a sample final screen from a screen capture of the system as it was programmed.

First Iteration Recap

Also during the last day of an iteration it is common to do an "introspection" of the iteration to discuss what went well and what problems were encountered. Because there are multiple iterations in an iterative SDLC, the project team has the opportunity to evaluate and improve their internal work processes.

Where You Are Headed - The Rest of This Book

The following sections in this chapter provide a brief overview of the remainder of the textbook.

Part 1: Introduction to Systems Development

Part 1 consists of Chapter 1 and Online Chapter A.

Online Chapter A, "The Role of the Systems Analyst" describes the skills required of a systems analyst and the various career options available.

Part 2: Systems Analysis Tasks

Chapters 2 through 5 cover systems analysis in detail, including gathering information and building analysis models. The primary focus is on system process, as documented by use cases, and system information, as documented by classes.

Online Chapter B, "The Traditional Approach to Requirements," discusses Data Flow Diagrams and other traditional structured analysis models.

Part 3: Essentials of Systems Design

Chapters 6, 7, 8, and 9 provide the foundation principles for systems design. Chapter 6 provides a broad and comprehensive discussion of systems design principles. Chapter 7 teaches about the overall environment into which the new software must reside. It also covers issues related to the system architecture. Chapter 8 teaches how to design the user interfaces. Chapter 9 explains database design.

Part 4: Projects and Project Management

We have moved project management principles to the middle of the text so that the students will have some experience in project related activities before discussion of how to organize and manage those activities. Sometimes when teaching project management principles at the beginning of the course, the students did not have enough experience to grasp the importance and utility of some of the more abstract project management principles.

Chapter 10 introduces various types of SDLC options and different methodologies used to develop software applications. Chapter 11 extends those concepts and teaches the fundamental principles of project planning and project management. Online Chapter C, "Project Management Techniques," teaches the details of calculating net present value and building a project schedule with Microsoft Project.

Part 5: Advanced Design and Deployment Concepts

Chapters 12 and 13 explain the detail models and techniques to do detailed object-oriented systems design. These two chapters are rather detailed since design can be a complex activity.

Chapter 14 explains the final steps of acceptance test, data conversion, and deployment of the new system.

Classroom Activities

Since the objective of this chapter is to provide an example of a complete systems development project, one interesting activity is to ask the students what they would do, i.e. what are the steps, to build a system. Give an example, such as a patient monitoring system for a dentist, or an inventory tracking system for a small business, and ask them how they would build this system for the owner (who is a good friend). Be careful to make your example large enough that it is not simply a "cottage industry" single person application where the student just builds something that is already in his/her head. Then you can ask pointed question such as:

- How do you find out exactly what the user needs the system to do?
- How do you remember what the user tells you about what it must do?
- How do you know what information is important to keep?
- How do you make sure it does what it is supposed to do?
- How do you divide up the work, if there are two or more of you working together?

The chapter has a rather detailed case at the end of the chapter. You can assign the entire case, or select specific problems out of the case. The case can be assigned to individuals or to small groups to do. On the following day, you could allow a few students or groups to present their models. Remember that the intent of the chapter is to give an example and teach about the entire process of systems development and not about how correct any individual model is. The students should be able to observe that developing software applications for a third party can be a challenging process.

Troubleshooting Tips

The danger in this chapter is to become too focused on the individual models or the specific solutions. Keep the focus on the overall process of planning, analysis, design, programming, testing, deployment with the emphasis on Systems Analysis and Design.

For all of the models presented, focus only on the overall purpose of the model. Simplify it as much as possible. For example work flow diagrams are simply steps of what needs to happen with arrows showing the order of the steps. Class diagrams are simply the "things" that the system needs to know about and remember. The attributes are simply the details. Use cases are simply how the user "uses" the system.

Discussion Questions

1. The need to have a development methodology.

Many students today will have experience building simple applications such as a Facebook app or an iPhone app. These students may not see the need to have a development process with distinct steps, plans, and models. Often these students built an app they envisioned themselves. Hence they were the client, user, stakeholder, analyst, design, and tester all in a single person. It is very different to build an application where there are multiple users, multiple developers, and multiple testers. Function decisions must be discovered and then remembered, ire. documented, designs must be communicated across multiple programmers, acceptance tests must be coordinated, and so forth. Trying to build even a medium sized system without a project plan and a development methodology will spell disaster. Some typical questions to ask in this type of discussion might include the following.

- What purpose does systems analysis serve? Is it really necessary? Why?
- Is it important to build models? What function do they serve? How do the developers ensure that they understand the requirements? How do they remember what decisions were made by the user (and themselves)?
- Evaluate the six core processes? Are there better ways to execute a project? Are the six core processes a minimal set? Are there things missing that should be added?
- How long should we keep documentation and models that were built? Should we have external documentation that must be maintained?
- Is it important to have a formal SDLC? What are the alternatives? How formal should it be?

2. The effectiveness of iteration and the Agile philosophy

When we get to Chapters 8 and 9 we will discuss various forms of the SDLC and adaptive projects versus predictive projects. However, it is would be an effective classroom technique to begin the discussion with this example. Some discussion questions might include the following.

- What does "iteration" mean in software development? What is the alternative? What are some advantages of an iterative process? What are the disadvantages or potential problems with this approach?
- This example (Tradeshow system) appears to have sequential steps for analysis, design, and programming within the iteration. Is this the best way to execute a project? Is there a better way? Why do you suppose the example presents it sequentially (Day 1, Day 2, etc.)?
- What does it mean to use Agile principles for a project? Is Agile an effective way to execute a project? What are some of the disadvantages of an Agile approach?

- Is it important to do project planning? How much time and energy should be spent planning the project? Is planning important to an Agile project? What are some of the dangers of not planning? Of over planning?