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

Table of Contents
- Chapter Overview
- Learning Objectives

Chapter Overview

The sixth edition of Systems Analysis and Design in a Changing World is a completely new and innovative approach to teaching systems analysis and design. The intent of this new approach is 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, you 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 you an introduction to some of the major techniques and models that will be taught later in the text. Please be aware that most projects especially large complex projects take weeks sometimes months even years. The book is just making it simpler by presenting it by days.

Your assignments will be to work on a running case similar to the RMO Tradeshow system case. I assume that the students in this class have various levels of systems development background. So you may not have all of the skills necessary to do an error-free complete development assignment. The objective of the case and its assignments is simply to allow you to practice going through the entire development process so that you obtain a solid overview of systems development. My philosophy on grading is that you must give it your best effort. I am not looking for perfect submissions. I am looking for whether you read the book, understood the concepts, tools, techniques enough to apply them to the case assignment. I will know if you did not give it your best effort. I am not looking for perfect submissions. I am looking for whether you read the book, understood the concepts, tools, techniques enough to apply them to the case assignment. I will know if you did not give it your best effort. Meaning you submitted it last minute, did the minimum just to submit something by the deadline, and/or you did a sloppy job. Trust me I can tell. I suggest that you submit the assignment early with your best effort. If there’s time and I think you can improve on your work, then I will allow for resubmission for more points. It is completely at my discretion whether I will allow you to resubmit the assignment or not. If it is on the due date then most likely I will NOT allow for resubmission. I want to mentor you through the case study. Don’t be shy about sending me questions if you are confused. This is NOT a license for me to review your work before you submit and do pre-grading. Be specific about your question(s) and attach your file(s). I will help you out.
Learning Objectives

After reading this chapter, you should be able to:

- Describe the purpose of systems analysis and design in the development of information systems
- Describe the characteristics of iterative systems development
- Explain the six core processes of the Systems Development Life Cycle
- Identify key documents that are used in planning a project
- Identify key diagrams used in systems analysis and systems design
- Explain the utility of identifying use cases in systems development
- Explain the utility of identifying object classes in systems development

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 activities that enable a person to define and describe in detail the system that solves the need

The objective of this section is to introduce you 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.

Systems Development Life Cycle

Key Terms

- **project** — a planned undertaking that has a beginning and an end, and that produces some definite result
- **Systems Development Life Cycle (SDLC)** — the entire process consisting of all the activities required to build, launch, and maintain an information system
- **information systems development process** — the actual approach used to develop a particular information system
- **Agile Development** — an information systems development process that emphasizes flexibility
to anticipate new requirements during development

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 presents the six core processes required for the development of any new system. In other words, these six core processes are common to all types of SDLCs. By understanding these six core processes, you will not only be able to develop new systems effectively, but you will be able to adapt to any other SDLC that you 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 8, 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.

It is important to understand that all projects are not linear, point A to point B. There are always things that change throughout the systems development life cycle (SDLC). So information technology (IT) professionals, users and management have to be flexible and receptive to changes. That does NOT mean that every project will be chaotic. The project manager must be very skilled in managing the changes and the project team and management must understand the changes. So a good change control process should be implemented.

Please note that the following case (RMO) is an example of software development. The many other types of systems development projects such as upgrading a new network. The majority of projects are software projects or have software components.

**Introduction to Ridgeline Mountain Outfitters**

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.
Iterative Development

Key Terms

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

Finally before the development of the Tradeshow system begins, Iterative Development is introduced. Figure 1-4 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 you the general concepts so that you can apply them in your own activities or within the framework of a corporate environment.

Developing RMO’s Tradeshow System

Key Terms

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

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.

Pre-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. You should be able to understand this document and even produce a simple version of it. Review each section of the document shown in Figure 1-5.

Day 1 Activities

*RMO – Supplier Information System:* The first day of an iteration is usually the initial 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 First Iteration:** 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 1-6 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, the authors 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, the authors show it with boxes and connecting arrows as shown in Figure 1-7.

Note: You most likely do not know how to identify project tasks. The objective for this topic is to show you that a project needs to be organized and the work scheduled. Also, please note that in the beginning this is the initial planning. Planning and project management tasks are done throughout all phases of the SDLC. If you bought the custom book from the Cerritos College bookstore you will notice that Chapter 9 Project Planning and Project Management are NOT included. This topic is an entire class subject matter and I teach the CIS 202A Project Management for Information Technology class in the spring term. I took the chapter out because we already have plenty of material to cover and you can learn more about project management by taking that class.

**Day 2 Activities**

**Fact Finding and User Involvement:** Now that the project and the iterations 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 1-8.

You may not understand the techniques to identify use cases yet. 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 clerk **uses** the system to enter a new sale.” The objective of this step is to teach you that the functions of the new system must be identified and
documented. See Figure 1-8.

**Identifying Object Classes:** The classes are documented either in a table, as shown in Figure 1-9, or a class diagram as shown in Figure 1-10.

You may not understand the techniques of how to identify classes yet. 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 you that information systems always have information that is maintained and this is the method used to determine what is needed. Use cases will be covered in a later chapter. Remember this is the introduction/overview chapter.

Figure 1-10 is included simply to introduce the idea of a model and a class diagram to you. You are not expected to be able to develop or even understand the elements of a class diagram at this point. How to create class diagrams will be presented in later chapters. At most you should see that classes are information components that have a name and detailed elements or attributes.

**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 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)

The chapter begins the process with a **use case diagram**, which is just another way to illustrate the use cases that have been identified. Figure 1-11 is a sample use case diagram.

You may not understand the techniques required to develop a use case diagram yet. At this point in the course, you should just be aware of the fact that a use case diagram is one tool that is used to document use cases. You 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 workflow diagram. A workflow 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 1-12 is a workflow diagram.

Again, you may not understand the techniques of how to develop a workflow diagram at this point. You should be able to read a workflow diagram. You 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 1-13 is a sample screen layout.

You do not have to know all the steps required to layout computer screens. That’s a later chapter. You should be able to read and understand a sample screen layout. You should also be aware that sample screen layouts are an important technique for developing the user interface.
Day 4 Activities

Day 4 begins the design activities. Although analysis (understanding what) and design (structuring the how), often are done concurrently, the authors 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. Figure 1-14 shows a tabular form for the database. Figure 1-16 shows an enhanced class diagram, which provides the detail required to create the database schema.

Approaching High-Level Systems Design: Designing the overall system structure or architecture is done using architecture configuration diagrams. Figure 1-15 shows an architectural diagram for the entire system, and Figure 1-17 shows a subsystem package diagram showing the mid-level design for the Supplier subsystem.

Designing the Overall Architecture: These high-level design decisions will determine the detailed structure 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.

Designing the Subsystem 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 1-17 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. 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. You 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. You are not expected to be able to correctly create these models.

Day 5 Activities

Day 5 activities focus on programming the system. Figure 1-18 illustrates some PHP code for the SupplierView class.

Day 6 Activities

Day 6 is final testing and deployment. Figure 1-19 is a flow chart depicting the testing cycle. Figure 1-20 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. This additional chapter will be posted to the Resources folder in the TalonNet class site. Some questions will be on the test from this chapter. Please read this chapter.

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. This additional chapter will be posted to the Resources folder in the TalonNet class site. Some questions will be on the test from this chapter. Please read this chapter.

Part 3: Essentials of Systems Design

Chapters 6 and 7 provide the foundation principles for systems design. Chapter 6 provides a broad and comprehensive discussion of systems design principles. Chapter 7 teaches how to design the user interfaces and system interfaces.

Part 4: Projects and Project Management

The authors have moved project management principles to the middle of the text so that you 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, you do not have enough experience to grasp the importance and utility of some of the more abstract project management principles.

Chapter 8 introduces various types of SDLC options and different methodologies used to develop
software applications.

Chapter 9 is NOT in the custom book from the Cerritos College bookstore. As stated above we have enough material to cover in this class. If you bought the complete book on the Internet then feel free to read the chapter. Chapter 9 briefly teaches the fundamental principles of project planning and project management. Again this topic is an entire class, the CIS 202A Project Management for Information Technology. Online Chapter C, "Project Management Techniques," teaches the details of calculating net present value and building a project schedule with Microsoft Project. This chapter is also in the Resources folder in TalonNet. It is optional. If you want to learn more about MS Project then take the CIS 202B MS Project class in the fall term and of course CIS 202A in the spring term. NO questions will be on the test from chapter 9 or Chapter C.

Part 5: Advanced Design and Deployment Concepts

Chapters 10 and 11 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 12 explains how to build the database schema from the design class diagram. Other technical topics are included such as security and controls.

The primary focus of this class is on the analysis and design activities. Therefore we do not cover chapters 13 and 14. Again if you bought the complete book on the Internet, then you will have these two chapters and you are free to read them. There will be NO questions on the exams from chapters 13 & 14.

Chapter 13 explains the final steps of acceptance test, data conversion, and deployment of the new system.

Chapter 14 provides a look at the current trends in application software development. New trends include designing for interconnectedness and mobile devices. The Web as an application platform is also a major trend in new application software. This chapter should be an interesting capstone for your course.

Final Thoughts

The danger in chapter 1 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. Analysis activities focus on what the system must do (requirements) while design activities focus on how the system will accomplish the requirements.

For all of the models presented, focus only on the overall purpose of the model. 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. In later chapters will get in to the details of each model.

I hope these lecture notes help you to study and understand the concepts.