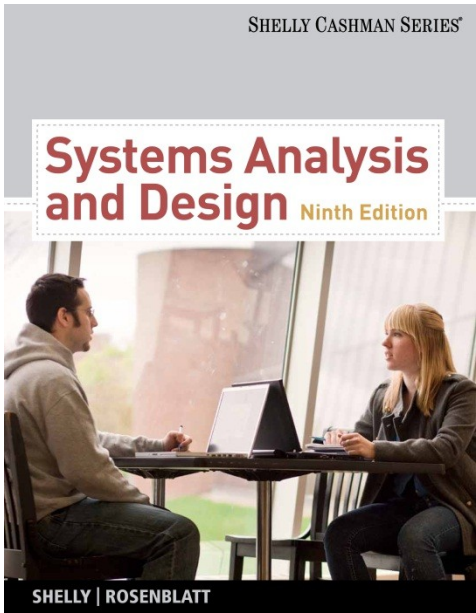
COURSE TECHNOLOGY
CENGAGE Learning

Systems Analysis and Design 9th Edition

Chapter 4

Requirements Modeling



Phase Description

- Systems analysis is the second of five phases in the systems development life cycle (SDLC)
- Will use requirements modeling, data and process modeling, and object modeling techniques to represent the new system
- Will consider various development strategies for the new system, and plan for the transition to systems design tasks

Chapter Objectives

- Describe systems analysis phase activities
- Explain joint application development (JAD), rapid application development (RAD), and agile methods
- Use a functional decomposition diagram (FDD) to model business functions and processes

Chapter Objectives

- Describe the Unified Modeling Language (UML) and examples of UML diagrams
- List and describe system requirements, including outputs, inputs, processes, performance, and controls
- Explain the concept of scalability

Chapter Objectives

- Use fact-finding techniques, including interviews, documentation review, observation, questionnaires, sampling, and research
- Define total cost of ownership (TCO)
- Conduct a successful interview
- Develop effective documentation methods to use during systems development

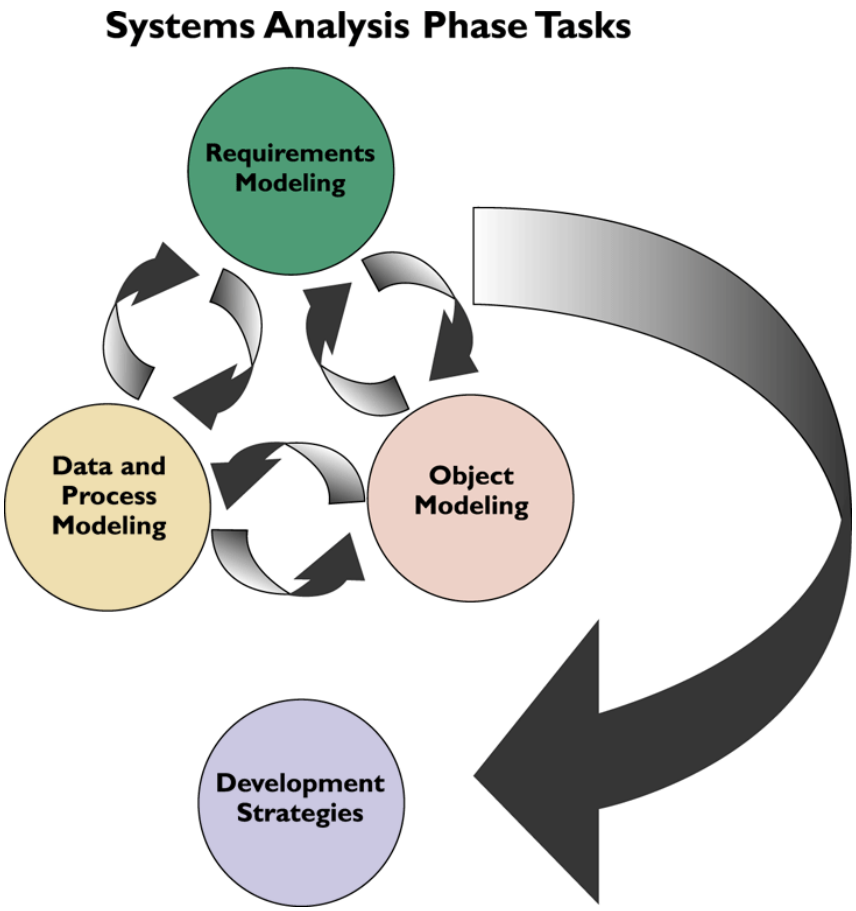
Introduction

- This chapter describes requirements modeling techniques and team-based methods that systems analysts use to visualize and document new systems
- The chapter then discusses system requirements and fact-finding techniques, which include interviewing, documentation review, observation, surveys and questionnaires, sampling, and research

Systems Analysis Phase Overview

- The overall objective of the **systems analysis** phase is to understand the proposed project, ensure that it will support business requirements, and build a solid foundation for system development
- You use models and other documentation tools **to visualize** and describe the proposed system

Systems Analysis Phase Overview



- Systems Analysis Activities
  - Requirements modeling
    - Outputs
    - Inputs
    - Processes
    - Performance
    - Security

Systems Analysis-**Requirements Modeling**

It is the process of defining the **functionalities**, **services**, and **constraints** of a system based on user needs.

- **Key Activities**:
  - **Gathering requirements** from users, stakeholders, and managers.
  - **Analyzing business processes** to understand the **workflow** and how the system will support it.
  - **Defining system functionalities** and expected behaviors.
- **Techniques Used**:
  - Interviews, surveys, observation, and document analysis to collect requirements.
  - **Use case modeling** and **data flow diagrams (DFDs)** to visually represent the system's operations

Systems Analysis-**Inputs**

- Inputs are the data or commands **entered into the system to be processed**. Inputs can come from various sources like users, other systems, sensors, or databases.
- **Examples**:
  - User input in a form (e.g., a customer filling out an online order form).
  - Automated inputs from IoT devices (e.g., temperature sensors sending data to a control system).

Systems Analysis -**Processes**

- Processes refer to the **set of activities or operations** that the system performs on inputs to generate outputs. This can involve calculations, data transformations, filtering, validation, or decision-making algorithms.
- **Key Types**:
  - **Business Processes**: Series of steps to achieve a business goal (e.g., order processing, payroll calculations).
  - **System Processes**: Logical operations or

Systems Analysis- **Performance**

- Performance defines how **well a system operates**, particularly regarding speed, responsiveness, and efficiency.
- **Performance Criteria**:
  - **Response time**: The time taken to respond to user input.
  - **Throughput**: The volume of data processed in a given time.
  - **Scalability**: The system's ability to handle increasing amounts of data or users.
  - **Reliability**: How consistently the system functions without failures

Systems Analysis Phase Overview

- Systems Analysis Activities
  - Data and process modeling
  - Object Modeling
  - Development Strategies
    - System requirements document

Systems Analysis Phase Overview

- Systems Analysis Skills
  - Analytical skills
  - Interpersonal skills
- Team-Oriented Methods and Techniques
  - Joint application development (JAD)
  - Rapid application development (RAD)
  - Agile methods

Joint Application Development

- User Involvement
  - Users have a vital stake in an information system and they should participate fully
  - Successful systems must be user-oriented, and users need to be involved
  - One popular strategy for user involvement is a JAD team approach

Joint Application Development

- JAD Participants and Roles

JAD PARTICIPANT	ROLE
JAD project leader	Develops an agenda, acts as a facilitator, and leads the JAD session
Top management	Provides enterprise-level authorization and support for the project
Managers	Provide department-level support for the project and understanding of how the project must support business functions and requirements
Users	Provide operational-level input on current operations, desired changes, input and output requirements, user interface issues, and how the project will support day-to-day tasks
Systems analysts and other IT staff members	Provide technical assistance and resources for JAD team members on issues such as security, backup, hardware, software, and network capability
Recorder	Documents results of JAD sessions and works with systems analysts to build system models and develop CASE tool documentation

Joint Application Development

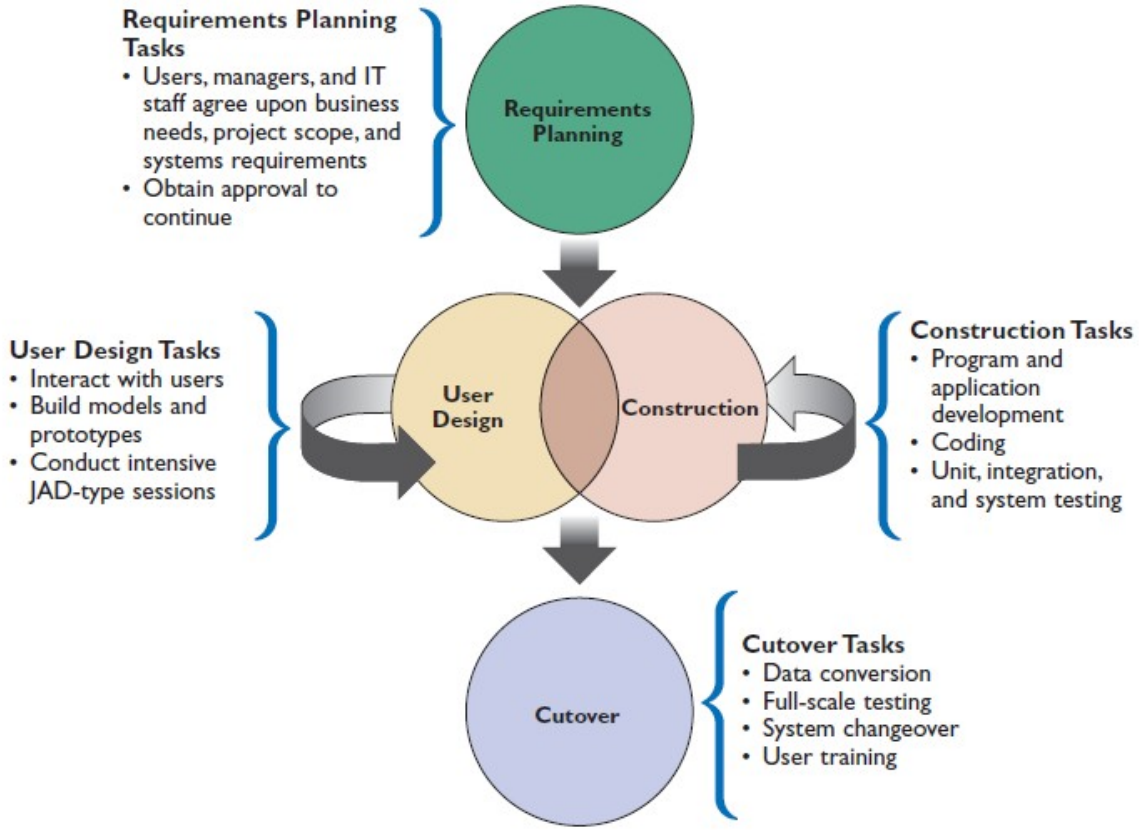
- JAD **Advantages** and **Disadvantages**
  - More **expensive** and can be cumbersome if the group is too large relative to the size of the project
  - Allows **key users to participate** effectively
  - When properly used, JAD can result in a more **accurate statement of system requirements**, a better understanding of common goals, and a stronger commitment to the success of the new system

Rapid Application Development

- Is a team-based technique that speeds up information systems development and produces a functioning information system
- Relies heavily on prototyping and user involvement
- Interactive process continues until the system is completely developed and users are satisfied

Rapid Application Development

- RAD Phases and Activities



Rapid Application Development

- RAD **Objectives**
  - To **cut development time** and expense by involving the users in every phase of systems development
  - Successful RAD team must have IT resources, skills, and **management support**
  - Helps a development team design **a system that requires a highly interactive** or complex user interface

Rapid Application Development

- RAD Advantages and Disadvantages
  - Systems can be developed more quickly with significant cost savings
  - RAD stresses the mechanics of the system itself and **does not emphasize the company's strategic business needs**
  - Might allow less time to develop quality, consistency, and design standards

Agile Methods

- Attempt to develop a system incrementally
- Agilian modeling toolset includes support for many modeling tools
- Some agile developers prefer not to use CASE tools at all, and rely instead on whiteboard displays and arrangements of movable sticky notes

Agile Methods

- Scrum is a rugby term
- Pigs include the product owner, the facilitator, and the development team; while the chickens include users, other stakeholders, and managers
- Scrum sessions have specific guidelines that emphasize time blocks, interaction, and team-based activities that result in deliverable software

Agile Methods

- **Attempt to develop a system incrementally:**

➢ Agile development emphasizes delivering a working system in small, usable **increments** rather than a single, large release at the end. This means the system is built and improved in stages, allowing for regular feedback and adjustments.

➢ **Example:** In a software project, an initial release might include core features. With each increment, additional features or refinements are added, allowing the team to adapt to changes in requirements or user feedback without overhauling the entire system.

Modeling Tools and Techniques

Agilian modeling toolset includes support for many modeling tools:

- The Agilian modeling toolset is an example of CASE (Computer-Aided Software Engineering) tools used to assist in **visualizing and designing software systems**. These tools typically support different types of modeling such as UML (Unified Modeling Language) diagrams, data flow diagrams, and process modeling, all of which are commonly used in Agile development to keep models lightweight and adaptive to changes.
- **Example:** A project team might use a tool like Visual Paradigm (Agilian) to create a UML class diagram that evolves as the system changes incrementally, instead of creating a static and rigid design document upfront.

Use CASE tools for Agile

Some agile developers prefer NOT to use CASE tools at all, and rely instead on whiteboard displays and arrangements of movable sticky notes:

- Agile teams often prefer **lightweight, informal methods for organizing their work and sharing ideas**. Whiteboards and sticky notes provide a simple, flexible way to visualize tasks and progress, enabling constant updates and real-time collaboration.
- **Example:** In a Scrum meeting, a team might use a whiteboard with sticky notes representing user stories. These can be moved across columns like "To Do," "In Progress," and "Completed," reflecting the status of tasks without the need for digital tools.

Agile Method **Advantages**

- **Very flexible and efficient in dealing with change:** Agile methods prioritize adaptability, allowing teams to respond to changes in requirements or user feedback throughout the development process.
  - **Example:** If a client requests a new feature mid-project, Agile teams can reprioritize their work to accommodate the change in the next sprint.
- **Frequent deliverables constantly validate the project and reduce risk:** Delivering small, frequent updates ensures that the project stays aligned with user needs and minimizes the risk of large-scale failures.
  - **Example:** Releasing a new feature every two weeks allows the client **to test and provide feedback**, reducing the chances of major rework later on

Agile Method **Disadvantages**

- **Team members need a high level of technical and interpersonal skills:** Agile teams are expected to work autonomously and collaboratively, requiring not only technical expertise but also strong communication and teamwork skills.
  - **Example:** Team members must communicate openly and adapt quickly to changes, which can be challenging for individuals who are used to more rigid project structures.
- **May be subject to significant change in scope:** Because Agile encourages flexibility and responsiveness to feedback, projects can sometimes experience "scope creep" where additional features or changes are requested, potentially leading to delays or increased costs.
  - **Example:** A project might begin with a defined scope, but over time, new features are added based on user feedback, extending the timeline and increasing the workload

Modeling Tools and Techniques

- Business Process Modeling
  - Business process model (BPM)
  - Business process modeling notation (BPMN)
  - Pool
  - Swim lanes

Business Process Modeling (BPM)

- Business Process Modeling (BPM) is the practice of **representing an organization's processes and workflows in a visual format**. The goal of BPM is to analyze and improve existing processes or design new ones to ensure efficiency and effectiveness in operations. By mapping out the steps of a process, businesses can identify bottlenecks, inefficiencies, and opportunities for improvement.
- **Example:** A company may model its **order** fulfillment process to visualize how orders **are received**, **processed**, and **shipped**, aiming to reduce delays and optimize resource allocation.

Pool

- In BPMN, a **pool** represents a major participant in the process, **usually a company, an organization, business unit, or system**. It is a container for a business process that shows how a specific entity interacts with other entities. Pools can be divided into multiple processes but always represent the boundary of an entity's interaction.
- **Example:** In a business-to-business process, one pool might represent "Company A" and another pool might represent "Company B," each interacting with the other through the exchange of documents or data

Swim Lanes

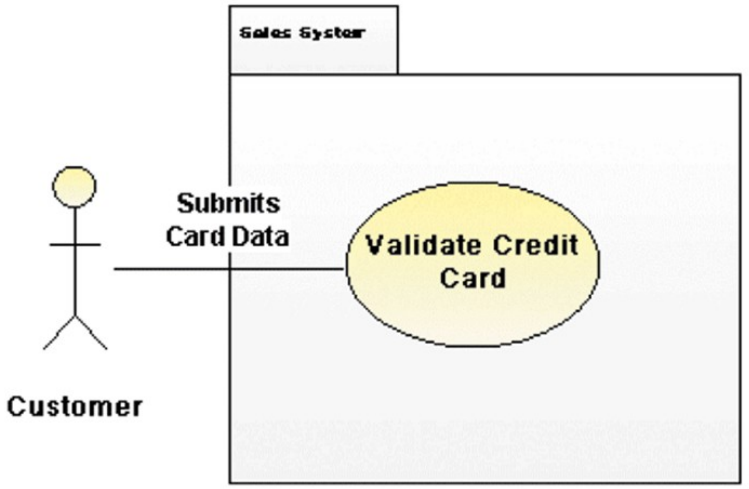
- Swim lanes are **sub-divisions within a pool** in BPMN diagrams. They represent **specific roles**, departments, or systems involved in the process. Each swim lane is responsible for **certain tasks** or activities within the process. Swim lanes help clarify which participant or group is responsible for specific steps within the process.
- **Example:** In a customer service process, a swim lane might be used to show that the "Sales Department" is responsible for customer inquiries, the "IT Department" is responsible for technical issues, and the "Billing Department" handles payment-related tasks

Modeling Tools and Techniques

- Data Flow Diagrams
  - Data flow diagram (DFD)
  - show how the system stores, processes, and transforms data
  - Additional levels of information and detail are depicted in other, related DFDs

Modeling Tools and Techniques

- Unified Modeling Language
  - Widely used method of visualizing and documenting software systems design
  - Use case diagrams
    - Actor
  - Sequence diagrams



System Requirements Checklist

- Outputs
  - The Web site must report online volume statistics every four hours, and hourly during peak periods
  - The inventory system must produce a daily report showing the part number, description, quantity on hand, quantity allocated, quantity available, and unit cost of all sorted by part number

System Requirements Checklist

- Inputs
  - Manufacturing employees must swipe their ID cards into online data collection terminals that record labor costs and calculate production efficiency
  - The department head must enter overtime hours on a separate screen

System Requirements Checklist

- Processes
  - The student records system must calculate the GPA at the end of each semester
  - As the final step in year-end processing, the payroll system must update employee salaries, bonuses, and benefits and produce tax data required by the IRS

System Requirements Checklist

- Performance
  - The system must support 25 users online simultaneously
  - Response time must not exceed four seconds

System Requirements Checklist

- Controls
  - The system must provide logon security at the operating system level and at the application level
  - An employee record must be added, changed, or deleted only by a member of the human resources department

Future Growth, Costs, and Benefits

- Scalability
  - A scalable system offers a better return on the initial investment
  - To evaluate scalability, you need information about projected future volume for all outputs, inputs, and processes

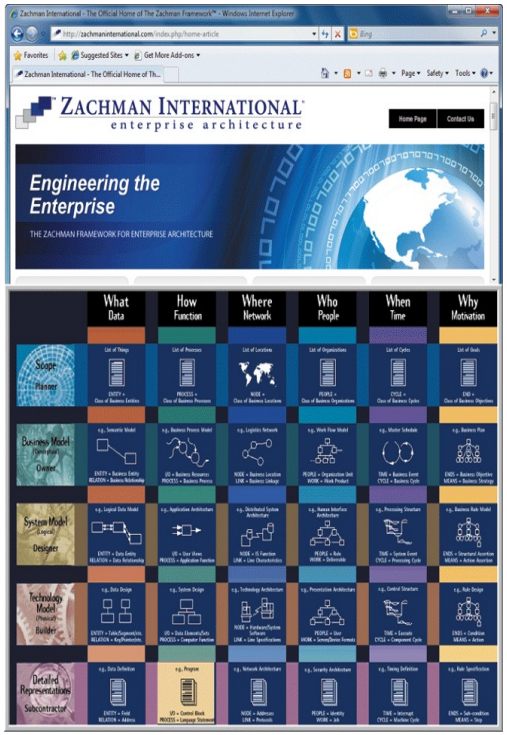
Future Growth, Costs, and Benefits

- Total Cost of Ownership
  - Total cost of ownership (TCO) is especially important if the development team is evaluating several alternatives
  - One problem is that cost estimates tend to understate indirect costs
  - Rapid Economic Justification (REJ)

Fact-Finding

- Fact-Finding Overview
  - First, you must identify the information you need
  - Develop a fact-finding plan
- Who, What, Where, When, How, and Why?
  - Difference between asking what is being done and what could or should be done

Fact-Finding



- The Zachman Framework
- **it helps organizations manage complexity by organizing data, processes, and technology into an understandable framework**
  - Zachman Framework for Enterprise Architecture
  - Helps managers and users understand the model and assures that overall business goals translate into successful IT projects

Example - The Zachman Framework

Perspective	What (Data)	How (Function)	Where (Network)	Who (People)	When (Time)
Planner	Patient, Doctor Info	Appointment Booking	Web and Mobile Access	Patients, Doctors, Admin	Anytime Booking
Owner	Patient Profiles, Doctor Schedules	Search Doctors, Book Appointment	Cloud-based System	Patients, Doctors, IT Support	Notifications 24 Hours Before
Designer	Database Schema (Patients, Doctors, Appointments)	Search, Book, Sync Calendars	Hosted on Cloud, Mobile Access	User Roles: Patient, Doctor, Admin	Automated Reminders
Builder	PostgreSQL Database	Python/ JavaScript for Booking	AWS Cloud, Secure API	OAuth 2.0 for Authentication	Event-Driven Notifications

Interviews

- Step 1: Determine the People to Interview
  - Informal structures
- Step 2: Establish Objectives for the Interview
  - Determine the general areas to be discussed
  - List the facts you want to gather

Interviews

- Step 3: Develop Interview Questions
  - Creating a standard list of interview questions helps to keep you on track and avoid unnecessary tangents
  - Avoid leading questions
  - Open-ended questions
  - Closed-ended questions
  - Range-of-response questions

Interviews

- Step 4: Prepare for the Interview
  - Careful preparation is essential because an interview is an important meeting and not just a casual chat
  - Limit the interview to no more than one hour
  - Send a list of topics
  - Ask the interviewee to have samples available

Interviews

- Step 5: Conduct the Interview
  - Develop a specific plan for the meeting
  - Begin by introducing yourself, describing the project, and explaining your interview objectives
  - Engaged listening
  - Allow the person enough time to think about the question
  - After an interview, you should summarize the session and seek a confirmation

Interviews

- Step 6: Document the Interview
  - Note taking should be kept to a minimum
  - After conducting the interview, you must record the information quickly
  - After the interview, send memo to the interviewee expressing your appreciation
  - Note date, time, location, purpose of the interview, and the main points you discussed so the interviewee has a written summary and can offer additions or corrections

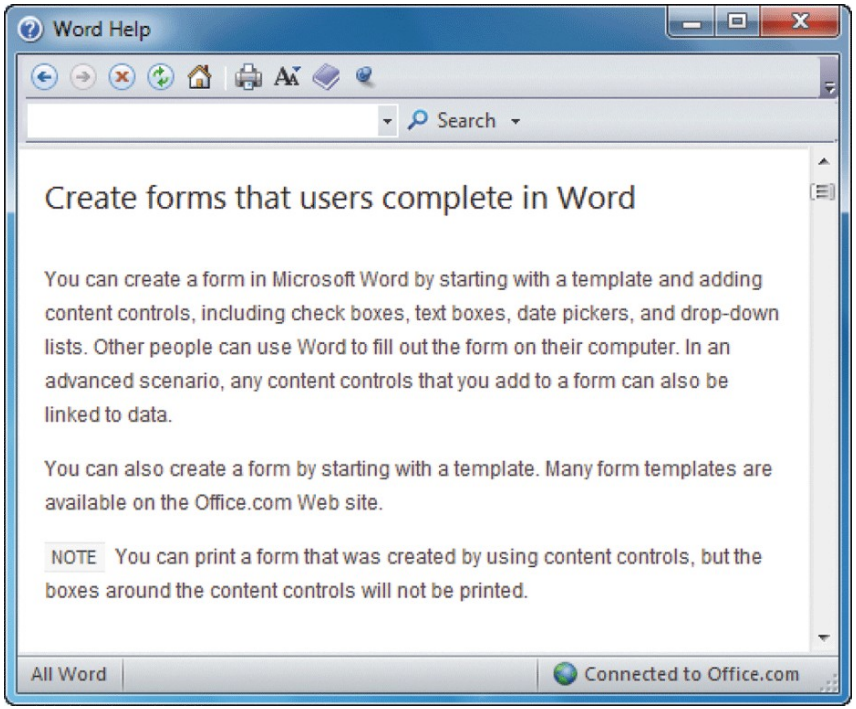
Interviews

- Step 7: Evaluate the Interview
  - In addition to recording the facts obtained in an interview, try to identify any possible biases
- Unsuccessful Interviews
  - No matter how well you prepare for interviews, some are not successful

Other Fact-Finding Techniques

- Document Review
- Observation
  - Seeing the system in action gives you additional perspective and a better understanding of the system procedures
  - Plan your observations in advance
  - Hawthorne Effect

Other Fact-Finding Techniques



- Questionnaires and Surveys
  - When designing a questionnaire, the most important rule of all is to make sure that your questions collect the right data in a form that you can use to further your fact-finding
  - Fill-in form

Other Fact-Finding Techniques

- Sampling
  - Systematic sample
  - Stratified sample
  - Random sample
  - Main objective of a sample is to ensure that it represents the overall population accurately

Other Fact-Finding Techniques

- Research
  - Can include the Internet, IT magazines, and books to obtain background information, technical material, and news about industry trends and developments
  - Site visit

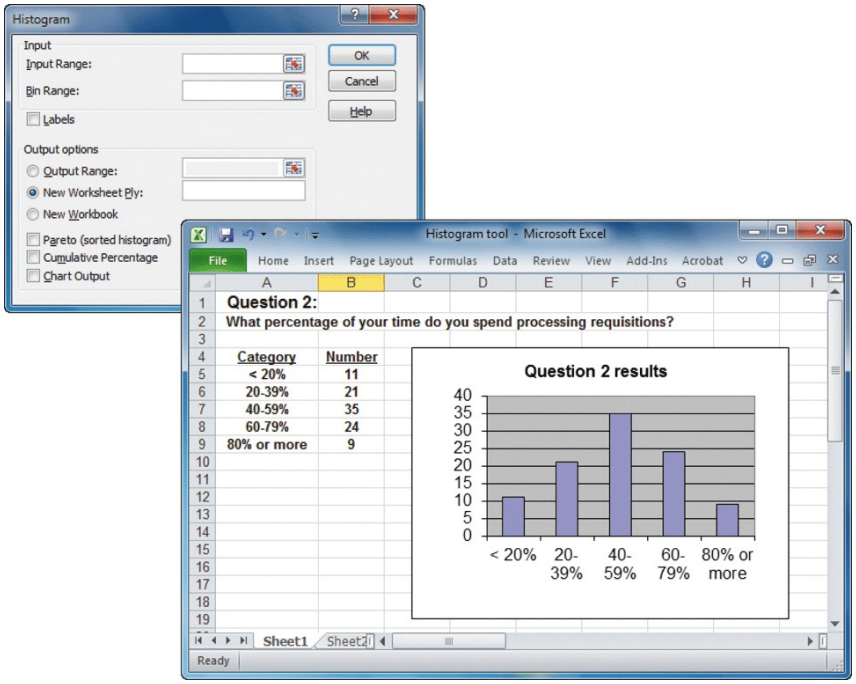
Other Fact-Finding Techniques

- Interviews versus Questionnaires
  - Interview is more familiar and personal
  - Questionnaire gives many people the opportunity to provide input and suggestions
  - Brainstorming
  - Structured brainstorming
  - Unstructured brainstorming

Documentation

- The Need for Recording the Facts
  - Record information as soon as you obtain it
  - Use the simplest recording method
  - Record your findings in such a way that they can be understood by someone else
  - Organize your documentation so related material is located easily

Documentation



- Software Tools
  - CASE Tools
  - Productivity Software
    - Word processing, spreadsheets, database management, presentation graphics, and collaborative software programs
    - Histogram

Documentation

- Software Tools
  - Graphics modeling software
  - Personal information managers
  - Wireless communication devices

Preview of Logical Modeling

- At the conclusion of requirements modeling, systems developers should have a clear understanding of business processes and system requirements
- The next step is to construct a logical model of the system
- IT professionals have differing views about systems development methodologies, and no universally accepted approach exists

Chapter Summary

- The systems analysis phase includes three activities: requirements modeling, data and process modeling, and consideration of development strategies
- The main objective is to understand the proposed project, ensure that it will support business requirements, and build a solid foundation for the systems design phase

Chapter Summary

- The fact-finding process includes interviewing, document review, observation, questionnaires, sampling, and research
- Systems analysts should carefully record and document factual information as it is collected, and various software tools can help an analyst visualize and describe an information system

- Chapter 4 complete