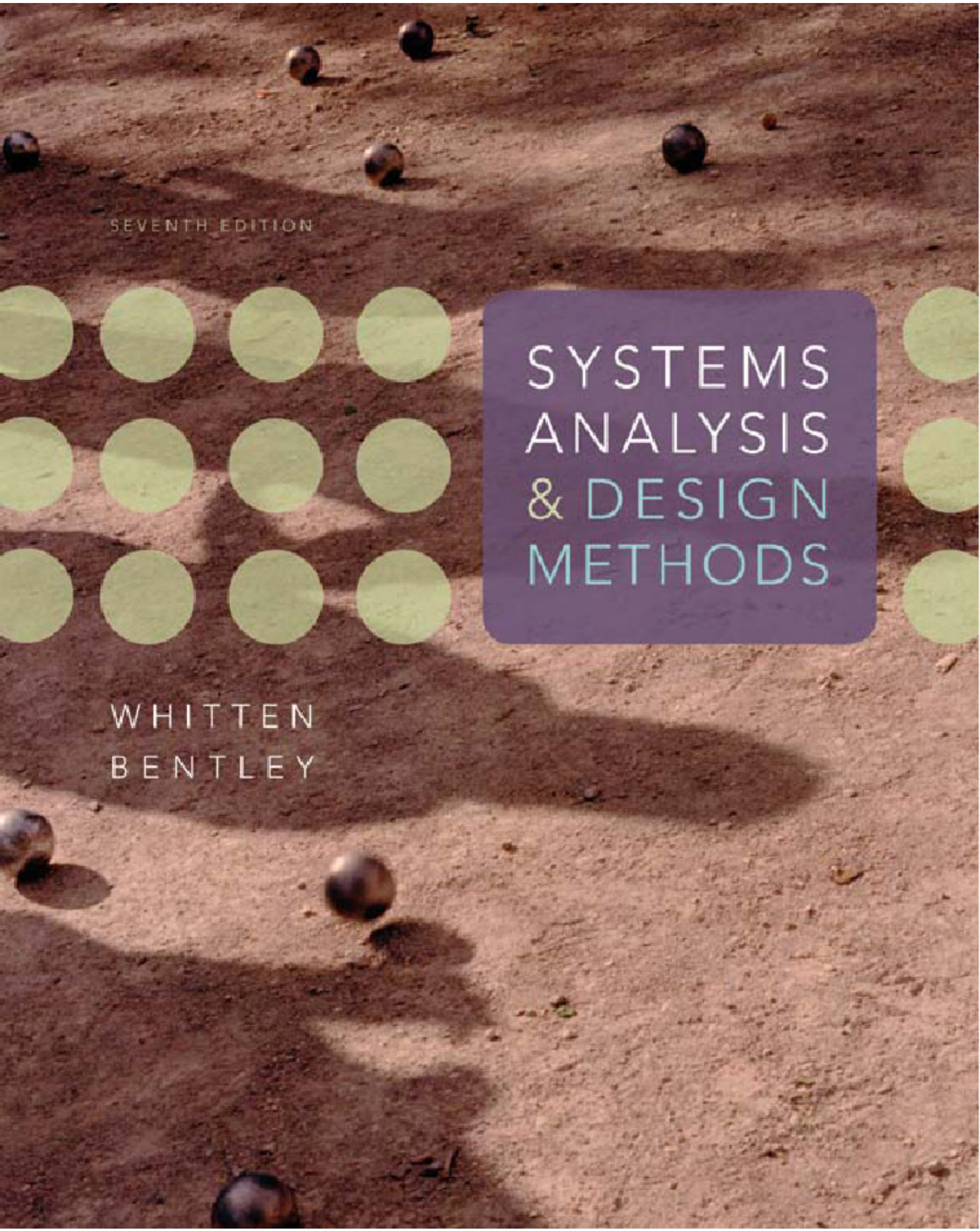

SEVENTH EDITION
SYSTEMS ANALYSIS & DESIGN METHODS
WHITTEN
BENTLEY

6 Fact-Finding Techniques for Requirements Discovery

Chapter Preview and Objectives

Effective fact-finding techniques are crucial to the development of systems projects. In this chapter you will learn about techniques to discover and analyze information system requirements. You will learn how to use various fact-finding techniques to gather information about the system's problems, opportunities, and directives. You will know that you understand fact-finding techniques and requirements discovery when you can:

- Define system requirements and differentiate between functional and nonfunctional requirements.
- Understand the activity of problem analysis and be able to create an Ishikawa (fishbone) diagram to aid in problem solving.
- Understand the concept of requirements management.
- Identify seven fact-finding techniques and characterize the advantages and disadvantages of each.
- Understand six guidelines for doing effective listening.
- Understand what body language and proxemics are and why a systems analyst should care.
- Characterize the typical participants in a JRP session and describe their roles.
- Complete the planning process for a JRP session, including selecting and equipping the location, selecting the participants, and preparing an agenda to guide the JRP session.
- Describe several benefits of using JRP as a fact-finding technique.
- Describe a fact-finding strategy that will make the most of your time with end users.

Introduction

Bob Martinez has spent most of the week reading. He started with memos related to the proposed Member Services System to better understand the problem. He then reviewed SoundStage's procedures manual for any policies related to member services and promotions. He studied nearly 100 member order forms selected at random, noting the kinds of data recorded in each blank and which blanks were always, sometimes, and never used. He read the documentation for the present member services system. He reviewed data and process diagrams from the prior member service systems development project, noting things that would probably need to be changed in the new system. It was grueling work. But in the end he really felt like he was beginning to understand the system. He produced a report for Sandra, his boss, of the key issues and questions that would need to be answered at the upcoming joint requirements planning meeting.

An Introduction to Requirements Discovery

In Chapter 3 we discussed several phases of systems development. Each phase is important and necessary in order to effectively design, construct, and ultimately implement a system to meet the users' (stakeholders') needs. But to develop such a system, we must first be able to correctly identify, analyze, and understand what the users' requirements are or what the users want the system to do. The process and techniques that a systems analyst uses to identify, analyze, and understand system requirements are referred to as **requirements discovery.** As suggested by the chapter's home page, requirements discovery primarily involves systems analysts working with system users and owners during the earlier system development phases to obtain a detailed understanding of the business requirements of an information system.

requirements discovery the process and techniques used by systems analysts to identify or extract system problems and solution requirements from the user community.

What are system requirements? **System requirements** specify what the information system must do or what property or quality the system must have. System requirements that specify what the information system must do are frequently referred to as **functional requirements.** System requirements that specify a property or quality the system must have are frequently referred to as **nonfunctional requirements.**

system requirement something that the information system must do or a property that it must have. Also called a *business requirement*.

The PIECES framework (Table 6-1), introduced in Chapter 3, provides an excellent tool for classifying system requirements. The benefit of classifying the various types of requirements is the ability to group requirements for reporting, tracking, and validation purposes. Plus doing so aids in identifying possible overlooked requirements.

functional requirement something the information system must do.

Essentially, the purpose of requirements discovery and management is to correctly identify the KNOWLEDGE, PROCESS, and COMMUNICATION requirements for the users of a new system. Failure to correctly identify system requirements may result in one or more of the following:

nonfunctional requirement a property or quality the system must have. Examples include security, ease-of-use, performance, etc.

- The system may cost more than projected.
- The system may be delivered later than promised.
- The system may not meet the users expectations, and that dissatisfaction may cause them not to use it.
- Once in production, the costs of maintaining and enhancing the system may be excessively high.
- The system may be unreliable and prone to errors and downtime.
- The reputation of the IT staff on the team is tarnished because any failure, regardless of who is at fault, will be perceived as a mistake by the team.

The impact in terms of cost can be staggering. Take, for example, Table 6-2, by Barry W. Boehm, a noted expert in information technology economics.[1] He studied several

[1]Donald C. Gause and Gerald M. Weinberg, *Exploring Requirements: Quality before Design* (New York: Dorset House Publishing, 1989), pp. 17–18.

TABLE 6-1 PIECES Classification of System Requirements

Nonfunctional Requirement Type	Explanation
Performance	Performance requirements represent the performance the system is required to exhibit to meet the needs of users. • What is the acceptable throughput rate? • What is the acceptable response time?
Information	Information requirements represent the information that is pertinent to the users in terms of content, timeliness, accuracy, and format. • What are the necessary inputs and outputs? When must they happen? • What is the required data to be stored? • How current must the information be? • What are the interfaces to external systems?
Economy	Economy requirements represent the need for the system to reduce costs or increase profits. • What are the areas of the system where costs must be reduced? • How much should costs be reduced or profits be increased? • What are the budgetary limits? • What is the timetable for development?
Control (and security)	Control requirements represent the environment in which the system must operate, as well as the type and degree of security that must be provided. • Must access to the system or information be controlled? • What are the privacy requirements? • Does the criticality of the data necessitate the need for special handling (backups, off-site storage, etc.) of the data?
Efficiency	Efficiency requirements represent the system's ability to produce outputs with minimal waste. • Are there duplicate steps in the process that must be eliminated? • Are there ways to reduce waste in the way the system uses it resources?
Service	Service requirements represent needs in order for the system to be reliable, flexible, and expandable. • Who will use the system, and where are they located? • Will there be different types of users? • What are the appropriate human factors? • What training devices and training materials are to be included in the system? • What training devices and training materials are to be developed and maintained separately from the system, such as stand-alone computer-based training (CBT) programs or databases? • What are the reliability/availability requirements? • How should the system be packaged and distributed? • What documentation is required?

TABLE 6-2 Relative Coasts of Fixing an Error

Phase in which Error Discovered	Cost Ratio
Requirements	1
Design	3–6
Coding	10
Development Testing	15–40
Acceptance Testing	30–70
Operation	40–1,000

software development projects to determine the costs of errors in requirements that weren't discovered until later in the development process.

Based on Boehm's findings, an erroneous requirement that goes undetected and unfixed until the operation phase may cost 1,000 times more than it would if it were detected and fixed in the requirements phase. Therefore, in defining system requirements, it is critical that they meet the following criteria:

- *Consistent*—The requirements are not conflicting or ambiguous.
- *Complete*—The requirements describe all possible system inputs and responses.
- *Feasible*—The requirements can be satisfied based on the available resources and constraints (feasibility analysis is covered in Chapter 11).
- *Required*—The requirements are truly needed and fulfill the purpose of the system.
- *Accurate*—The requirements are stated correctly.
- *Traceable*—The requirements directly map to the functions and features of the system.
- *Verifiable*—The requirements are defined so that they can be demonstrated during testing.

This can be a time-consuming, difficult, and frustrating process that often leads organizations and individuals to take shortcuts to save time and money. But this shortsightedness often leads to the problems mentioned before. Now that we understand our goal, lets look at the process.

The Process of Requirements Discovery

The process of requirements discovery consists of the following activities:

- Problem discovery and analysis.
- Requirements discovery.
- Documenting and analyzing requirements.
- Requirements management.

Let's now examine each one of these activities in detail.

> Problem Discovery and Analysis

As previously stated, requirements solve problems. For systems analysts to be successful, they must be skilled in the activity of problem analysis. To fully understand problem analysis, let's use the following example: A mother takes her young daughter to the doctor because the child is ill. The first thing the doctor tries to do is identify

the problem. The child has an earache, a fever, and a runny nose. Are these the problems? The mother has been giving the child pain medicine to ease the pain, but the child has not gotten better. It turns out the mother is treating the symptoms and not the real problem. Fortunately, the doctor is trained to analyze further. After examining the child, the doctor has concluded that the child has an ear infection, which is the root cause of the child's symptoms. Now that the problem has been identified and analyzed, it is time for the doctor to recommend a cure (solution). Normally, an antibiotic is prescribed to cure an ear infection, but in order to do that, the doctor first needs to determine if there are any constraints on the medicine that he can prescribe. How old is the child, and how much does she weigh? Is the child allergic to any medications? Can she swallow pills? Once these constraints are known, a prescription can be generated. Systems analysts use the same problem-solving process as a doctor uses, but instead of diagnosing medical problems they diagnose system problems.

One of the most common mistakes inexperienced systems analysts make when trying to analyze problems is identifying a symptom as a problem. As a result, they may design and implement a solution that more than likely doesn't solve the real problem or that may cause new problems. A popular tool used by development teams to identify, analyze, and solve problems is an **Ishikawa diagram.** The fishbone-shaped diagram is the brainchild of Kaoru Ishikawa, who pioneered quality management processes in the Kawasaki shipyards of Japan and, in the process, became one of the founding fathers of modern management.

Ishikawa diagram a graphical tool used to identify, explore, and depict problems and the causes and effects of those problems. It is often referred to as a *cause-and-effect diagram* or a *fishbone diagram* (because it resembles the skeleton of a fish).

Drawing the fishbone diagram begins with the name of the problem of interest entered at the right of the diagram (or the *fish's head*). The possible causes of the problem are then drawn as *bones* off the *main backbone,* each on an arrow pointing to the backbone. Typically, these "bones" are labeled as four basic categories: materials, machines, manpower, and methods (the four Ms). Other names can be used to suit the problem at hand. Alternative or additional categories include places, procedures, policies, and people (the four Ps) or surroundings, suppliers, systems, and skills (the four Ss).

The key is to have three to six main categories that encompass all possible areas of causes. Brainstorming techniques (defined later in the chapter) are commonly performed to add causes to the main bones. When the brainstorming is complete, the fishbone depicts a complete picture of all the possibilities about what could be the root cause for the designated problem. The development team can then use the diagram to decide and agree on what the most likely causes of the problem are and how they should be acted on. Figure 6-1 is an example of a fishbone diagram

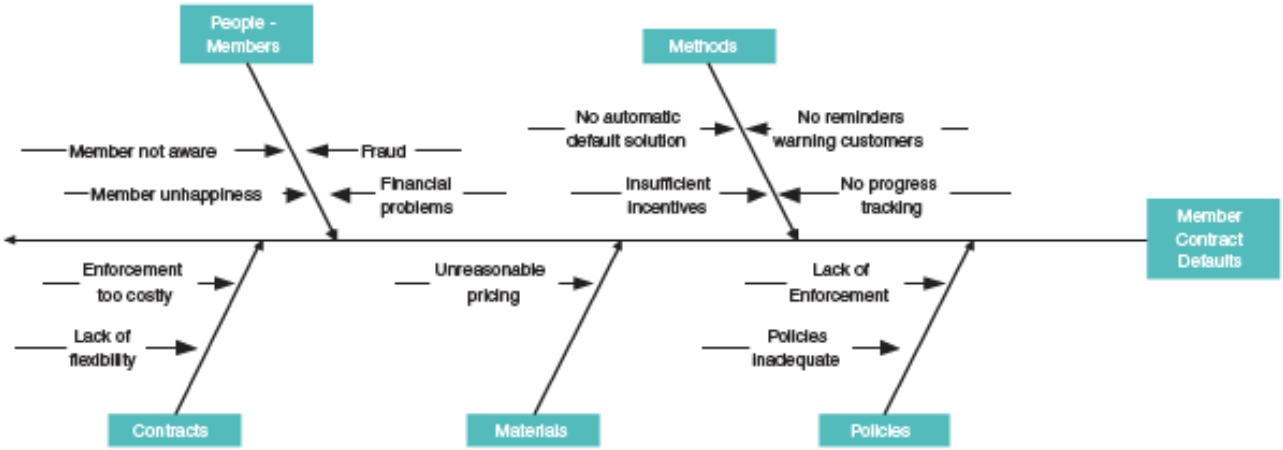


FIGURE 6-1 Sample Fishbone Diagram

depicting the SoundStage problem of members defaulting on contracts. In the diagram, notice that the problem to be solved is in the box at the far right. The five areas that have been identified as categories of causes (People-Members, Methods, Contracts, Materials, and Policies) are listed in boxes above and below the *fish's skeleton* and connected by arrows (bones) pointing to the fish's backbone. The actual causes of the problem for each category are depicted as arrows pointing to the category arrow (bone).

> Requirements Discovery

Given an understanding of problems, the systems analyst can start to define requirements. For today's systems analysts to be successful in defining system requirements, they must be skilled in effective methods for gathering information—fact-finding. **Fact-finding** is a technique that is used across the entire development cycle, but it is extremely critical in the requirements analysis phase. Once fact-finding has been completed, tools such as use cases, data models, process models, and object models will be used to document facts, and conclusions will be drawn from the facts. You will learn about these tools and how to document requirements derived from fact-finding in subsequent chapters of this textbook.

fact-finding the formal process of using research, meetings, interviews, questionnaires, sampling, and other techniques to collect information about system problems, requirements, and preferences. It is also called *information gathering* or *data collection*.

Facts are in the domain of the business application and its end users. Therefore, the analyst must collect those facts in order to effectively apply the documentation tools and techniques. During systems analysis phases, the analyst learns about the vocabulary, problems, opportunities, constraints, requirements, and priorities of a business and a system.

What types of facts must be collected? It would certainly be beneficial if we had a framework to help us determine what facts need to be collected, no matter what project we are working on. Fortunately, we have such a framework. As it turns out, the facts that describe any information system also correspond nicely with the building blocks highlighted on the chapter home page. Notice that fact-finding techniques are used in the early systems development phases to identify information, functional, and communication scopes and visions, as well as to identify business knowledge process, and communication requirements for the system.

> Documenting and Analyzing Requirements

When the systems analyst is performing fact-finding activities, it is important that the analyst assemble or document the gathered information (or *draft requirements*) in an organized, understandable, and meaningful way. These initial documents will provide direction for the modeling techniques the systems analyst will use to analyze the requirements and determine the correct requirements for the project. Once those have been identified, the systems analyst formalizes the requirements by presenting them in a document that will be reviewed and approved by the users.

Documenting the Draft Requirements Systems analysts use various tools to document their initial findings in draft form. They write *use cases* to describe the system functions from the external users' perspective and in a manner and terminology the users understand. *Decision tables* are used to document an organization's complex business policies and decision-making rules, and *requirements tables* are used to document each specific requirement. Each of these tools is examined in more detail later in the chapter.

Analyzing Requirements More often than not, fact-finding activities produce requirements that are in conflict with one another. This is because requirements are solicited from many different sources and each person has his or her own opinions

and desires for the functionality and features of the new system. The goal of the requirements analysis activity is to discover and resolve the problems with the requirements and reach agreement on any modifications to satisfy the stakeholders. The process is concerned with the "initial" requirements gathered from the stakeholders. These requirements are usually incomplete and documented in an informal way in instruments such as use cases, tables, and reports. The focus at this stage is on reaching agreement on the stakeholder's needs; in other words, the analysis should answer the question, "Do we have the right system requirements for the project?" Inevitably, the draft requirements contain many problems, such as:

- Missing requirements
- Conflicting requirements
- Infeasible requirements
- Overlapping requirements
- Ambiguous requirements

These types of requirements problems are very common in many of the requirement documents written today. If left unresolved, they can be extremely costly to fix later in the development cycle.

It was previously mentioned that stakeholders should agree on the resulting system requirements—thus there is an inevitable negotiation process that exists among stakeholders during analysis. If multiple stakeholders submit requirements that are in conflict with each other or if the proposed requirements are too ambitious, the stakeholders must negotiate, often under the guidance of the systems analyst, to agree on any modifications or simplifications to the system requirements. They also must agree on the criticality and priority of the requirements. This is crucial to ensure the success of the development effort.

The fact-finding and requirements analysis activities are very closely associated with each other and in fact are often interwoven. If requirements discovered during the fact-finding process are found to be problematic, the analyst may go ahead and perform analysis activities on the select items in order to resolve the problems before continuing to elicit additional system needs and desires.

This chapter focuses primarily on the business side of requirements, or, in other words, the logical requirements, but it is important to note that additional technical requirements exist that are physical in nature. Examples of technical requirements include specifying a required software package or hardware platform. These types of requirements will be discussed in depth in Chapter 11.

Formalizing Requirements System requirements are usually documented in a formal way to communicate the requirements to the key stakeholders of the system. This document serves as the contract between the system owners and the development team on what is going to be provided in terms of a new system. Thus, it may go through many revisions and reviews before everyone agrees and authorizes its contents. There is no standard name or format for this document. In fact, many organizations use different names such as requirements statement, requirements specification, requirements definition, functional specification, and the like, and the format is usually tailored to the organization's needs. Companies that provide information systems and software to the U.S. government must use the format and naming conventions specified in the government's published standards document MIL-STD-498.[2] Many organizations have created their own standards adapted from MIL-STD-498 because of its thoroughness and because many people are already familiar with it. In this book we will use the term **requirements definition document,** and Figure 6-2 provides a sample outline of one. Please note that this document will be consolidated with

requirements definition document a formal document that communicates the requirements of a proposed system to key stakeholders and serves as a contract for the systems project. Synonyms include requirements statement, requirements specification, and functional specification.

[2]MIL-STD-498 is a standard that merges DOD-STD-2167A and DOD-STD-7935A to define a set of activities and documentation suitable for the development of both weapon systems and automated information systems.

FIGURE 6-2
Sample Requirements Definition Outline

REQUIREMENTS DEFINITION DOCUMENT

1. Introduction
   - 1.1. Purpose
   - 1.2. Background
   - 1.3. Scope
   - 1.4. Definitions, Acronyms, and Abbreviations
   - 1.5. References
2. General Project Description
   - 2.1. Functional Requirements
3. Requirements and Constraints
   - 3.1. Functional Requirements
   - 3.2. Nonfunctional Requirements
4. Conclusion
   - 4.1. Outstanding Issues

Appendix (optional)

other project information to form the requirements statement, which is the final deliverable of the requirements analysis phase. A requirements definition document should consist of the following:

- The functions and services the system should provide.
- Nonfunctional requirements, including the system's features, characteristics, and attributes.
- The constraints, which restrict the development of the system or under which the system must operate.
- Information about other systems with which the system must interface.

Who will read the requirements definition document? This document is probably the most widely read and referenced document of all the project documentation. System owners and users use it to specify their requirements and any changes that may arise. Managers use it to prepare project plans and estimates, and developers use it to understand what is required and to develop tests to validate the system. With this in mind, it is important to note that requirements are read more often than they are written. Therefore, taking the time to write them correctly, concisely, and clearly not only will save time in terms of the schedule but is also more cost-efficient and reduces the risk of costly requirements errors. Performing requirements validation, therefore, is a necessary step toward achieving that goal. Requirements validation is performed on a final draft of the requirements definition document after all input has been solicited from the system owners and users. The purpose of this activity is for the systems analyst to ensure the requirements are written correctly. Examples of errors the systems analyst might find are:

- System models that contain errors.
- Typographical or grammatical errors.
- Conflicting requirements.
- Ambiguous or poorly worded requirements.
- Lack of conformance to quality standards required for the document.

> Requirements Management

Over the lifetime of the project it is very common for new requirements to emerge and existing requirements to change after a requirements definition document has been approved. Some studies have shown that over the life of a project as much as

50 percent or more of the requirements will change before the system is put into production. Obviously, this can be a major headache for the development team. To help alleviate the many problems associated with changing requirements, it is necessary to perform **requirements management.** Requirements management encompasses the policies, procedures, and processes that govern how a change to a requirement is handled. In other words, it specifies how a change request should be submitted, how it is analyzed for impact to scope, schedule, and cost, how it is approved or rejected, and how the change is implemented if approved.

requirements management the process of managing change to the requirements.

Fact-Finding Techniques

In this section we present seven common fact-finding techniques:

- Sampling of existing documentation, forms, and databases.
- Research and site visits.
- Observation of the work environment.
- Questionnaires.
- Interviews.
- Prototyping.
- Joint requirements planning.

An analyst usually applies several of these techniques during a single systems project. To be able to select the most suitable technique for use in any given situation, systems analysts need to learn the advantages and disadvantages of each of the fact-finding techniques.

Fact-Finding Ethics During fact-finding, systems analysts often come across or analyze information that is sensitive in nature. It could be a file of an aerospace company's pricing structure for a contract bid or even employee profiles, including salaries, performance history, medical history, and career plans. Analysts must take great care to protect the security and privacy of any facts or data with which they have been entrusted. Many people and organizations in this highly competitive atmosphere are looking for an "edge" to get ahead. Careless system analysts who leave sensitive documents in plain view on their desks, or publicly discuss sensitive data could cause great harm to the organization or to individuals. If such data should fall into the wrong people's hands, the systems analyst may lose the respect, credibility, or confidence of users and management. In some cases, the analyst would be responsible for the invasion of a person's privacy and could be liable.

Most corporations make every effort to ensure they conduct business in an ethical manner because the laws may require them to. There have been many cases where corporations have incurred heavy fines for not conducting business properly. To this end, many corporations require that their employees attend annual training seminars on company ethics, and they reinforce the learning by displaying banners or signs that contain the company's code of conduct and ethics statements throughout the workplace in highly visible locations. The company's ethics policies may be in a hard-copy format that is distributed to all employees, or they may be on the company's Web pages, making them easily accessible to employees no matter where they are currently located. Ethics policies document expected and required behavior. Violations of these policies could lead to disciplinary action or even termination. Ethics play a crucial role in fact-finding.

Sampling of Existing Documentation, Forms, and Files

When studying an existing system, systems analysts develop a pretty good feel for the system by studying existing documentation, forms, and files. A good analyst always knows to get facts first from existing documentation rather than from people.

Collecting Facts from Existing Documentation What kind of documents can teach you about a system? The first document the analyst may wish to seek out is the organization chart. An organization chart serves to identify key individual owners and users for a project and their reporting relationships. The analyst may also want to trace the history that led to the project. To accomplish this, the analyst should collect and review documents that describe the problem. These include:

- Interoffice memoranda, studies, minutes, suggestion box notes, customer complaints, and reports that document the problem area.
- Accounting records, performance reviews, work measurement reviews, and other scheduled operating reports.
- Information systems project requests—past and present.

In addition to documents that describe the problem, there are usually documents that describe the business function being studied or designed. These documents may include:

- The company's mission statement and strategic plan.
- Formal objectives for the organization subunits being studied.
- Policy manuals that may place constraints on any proposed system.
- Standard operating procedures (SOPs), job outlines, or task instructions for specific day-to-day operations.
- Completed forms that represent actual transactions at various points in the processing cycle.
- Samples of manual and computerized databases.
- Samples of manual and computerized screens and reports.

Also, analysts often check for documentation of previous system studies and designs performed by former systems analysts and consultants. This documentation may include:

- Various types of flowcharts and diagrams.
- Project dictionaries or repositories
- Design documentation, such as inputs, outputs, and databases.
- Program documentation.
- Computer operations manuals and training manuals.

All documentation collected should be analyzed to determine whether or not the information is current. Outdated documentation should not be discarded; however, analysts should keep in mind that additional fact-finding will be needed to verify or update the facts collected. What is the analyst looking for in all this material? Things that can be gleaned from these documents include:

- The symptoms and (possibly) causes of the problem.
- What persons in the organization have an understanding of the problem.
- The business functions that support the present system.
- The type of data that needs to be collected and reported by the system.
- Things in the documentation that the analyst does not understand and so need to be covered in interviews.

Document and File Sampling Techniques Because it would be impractical to study every occurrence of every form or record in a file or database, system analysts normally use **sampling** techniques to get a large-enough cross section to determine what can happen in the system. The systems analyst should seek to sample enough forms to represent the full nature and complexity of the data. Experienced analysts avoid the pitfalls of sampling blank forms—blank forms tell little about how the form is actually used, when it is not used, or how it is often misused. When studying documents or records from a database table, analysts should study enough samples to identify all the possible processing conditions and exceptions. Statistical sampling techniques can be used to determine if the sample size is large enough to be representative of the total population of records or documents.

sampling the process of collecting a representative sample of documents, forms, and records.

TABLE 6-3 Partial Table of Certainty Factors

Desired Certainty	Certainty Factor
95%	1.960
90	1.645
80	1.281

There are many sampling issues and factors, and this is a good reason for taking an introductory statistics course. One simple and reliable formula for determining sample size is

$$\text{Sample size} = 0.25 \times (\text{Certainty factor}/\text{Acceptable error})^2$$

The certainty factor is a value that can simply be looked up in statistical tables based on the desired certainty that the sample selected will be representative of the total population. See Table 6-3 for a partial example.

Suppose an analyst wants to be 90-percent certain that a sample of invoices will contain no unsampled variations. The sample size, *SS*, is calculated as follows:

$$SS = 0.25(1.645/0.10)^2 = 68$$

The analyst will need to sample 68 invoices to get the desired accuracy. If a higher level of certainty is desired, a larger number of invoices are needed.

If the analyst knows from experience that 1 in every 10 invoices varies from the norm, then he or she can replace the heuristic 0.25 with $p(1 - p)$ where p is the proportion of invoices with variances:

$$SS = p(1 - p)(1.645/0.10)^2$$

By using this formula, the analyst can reduce the number of samples required to get the desired accuracy:

$$SS = 0.10(1 - 0.10)(1.645/0.10)^2 = 25$$

How are the 25 invoices chosen? Two commonly used sampling techniques are randomization and stratification. **Randomization** involves randomly, or without concern, selecting sample data. Therefore, we just randomly choose 25 invoices based on the sample size calculated above. **Stratification** is a thoughtful and systematic approach aimed at reducing the variance of the sample data. For computerized files, stratification sampling can be executed by writing a simple program. For instance, suppose our invoices are in a database that has a volume of approximately 250,000 invoices. Recall that our sample size needs to include 25 invoices. We will simply write a program that prints every 10,000th record (= 250,000/25). For manual files and documents, we could execute a similar scheme.

randomization a sampling technique characterized by having no predetermined pattern or plan for selecting sample data.

stratification a systematic sampling technique that attempts to reduce the variance of estimates by spreading out the sampling—for example, choosing documents or records by formula—and by avoiding very high or very low estimates.

> Research and Site Visits

A second fact-finding technique is thoroughly researching the problem domain. Most problems are not completely unique. Other people have solved them before us. Many times organizations contact or perform site visits with companies they know have previously experienced similar problems. If these companies are "willing to share," valuable information can be obtained that may save tremendous time and cost in the development process.

Computer trade journals and reference books are a good source of information. They can provide information on how others have solved similar problems. With recent advances in cyberspace, analysts rarely have to leave their desks to do research.

Exploring the Internet and intranet via personal computer can provide immeasurable amounts of information.

A similar type of research involves visiting other companies or departments that have addressed similar problems. Memberships in professional societies such as the Association for Information Technology Professionals (AITP) or Association for Information Systems (AIS), among others, can provide a network of useful contacts.

> Observation of the Work Environment

observation a fact-finding technique wherein the systems analyst either participates in or watches a person perform activities to learn about the system.

Observation is one of the most effective data-collection techniques for learning about a system. **Observation** involves the systems analyst becoming an observer of people and activities in order to learn about the system. This technique is often used when the validity of data collected through other methods is in question or when the complexity of certain aspects of the system prevents a clear explanation by the end users.

Collecting Facts by Observing People at Work Even with a well-conceived observation plan, the systems analyst is not assured that fact-finding will be successful. The following story, which appears in a book by Gerald M. Weinberg called *Rethinking Systems Analysis and Design*, gives us an entertaining yet excellent example of some of the pitfalls of observation.[3]

The Railroad Paradox

About thirty miles from Gotham City lay the commuter community of Suburbantown. Each morning, thousands of Surburbanites took the Central Railroad to work in Gotham City. Each evening, Central Railroad returned them to their waiting spouses, children, and dogs.

Suburbantown was a wealthy suburb, and many of the spouses liked to leave the children and dogs and spend an evening in Gotham City with their mates. They preferred to precede their evening of dinner and theater with browsing among Gotham City's lush markets. But there was a problem. To allow time for proper shopping, a Suburbanite would have to depart for Gotham City at 2:30 or 3:00 in the afternoon. At that hour, no Central Railroad train stopped in Suburbantown.

Some Suburbanites noted that a Central train did pass through their station at 2:30, but did not stop. They decided to petition the railroad, asking that the train be scheduled to stop at Suburbantown. They readily found supporters in their door-to-door canvass. When the petition was mailed, it contained 253 signatures. About three weeks later, the petition committee received the following letter from the Central Railroad:

Dear Committee

Thank you for your continuing interest in Central Railroad operations. We take seriously our commitment to providing responsive service to all the people living among our routes, and greatly appreciate feedback on all aspects of our business. In response to your petition, our customer service representative visited the Suburbantown station on three separate days, each time at 2:30 in the afternoon. Although he observed with great care, *on none of the three occasions were there any passengers waiting for a southbound train.*

We can only conclude that there is no real demand for a southbound stop at 2:30, and must therefore regretfully decline your petition.

Yours sincerely,

Customer Service Agent

Central Railroad

[3]Gerald M. Weinberg, *Rethinking Systems Analysis and Design*, pp. 23–24. Copyright © 1988, 1982 by Gerald M. Weinberg. Reprinted by permission of Dorset House Publishing, 353 W. 12th St., New York, NY 10014 (212-620-4053/800-DH-BOOKS/www.dorsethouse.com). All rights reserved.

What are the lessons learned form the story above? For one, it is necessary to use the appropriate fact-finding technique for the problem at hand. Observation, in this case, was an incorrect choice. Why would anyone be waiting for a 2:30 train when all the town's people knew the train doesn't stop? A second lesson to be learned is to verify fact-finding results with users. Based on the user feedback, you may discover that you need to try other fact-finding techniques to gather additional information. Never jump to conclusions.

Observation Advantages and Disadvantages Observation can be a very useful and beneficial fact-finding technique provided that you have the ability to observe all aspects of the work being performed by the users and that the work is being performed in the usual manner. You should become aware of the pros and cons of the technique of observation. Advantages and disadvantages include:

Advantages

- Data gathered based on observation can be very reliable. Sometimes, observations are conducted to check the validity of data obtained directly from individuals.
- The systems analyst is able to see exactly what is being done. Complex tasks are sometimes difficult to clearly explain in words. Through observation, the systems analyst can identify tasks that have been missed or inaccurately described by other fact-finding techniques. Also, the analyst can obtain data describing the physical environment of the task (e.g., physical layout, traffic, lighting, noise level).
- Observation is relatively inexpensive compared with other fact-finding techniques. Other techniques usually require substantially more employee release time and copying expenses.
- Observation allows the systems analyst to do work measurements.

Disadvantages

- Because people usually feel uncomfortable when being watched, they may unwittingly perform differently when being observed.
- The work being observed may not involve the level of difficulty or volume normally experienced during that time period.
- Some systems activities may take place at odd times, causing a scheduling inconvenience for the systems analyst.
- The tasks being observed are subject to various types of interruptions.
- Some tasks may not always be performed in the manner in which they are observed by the systems analyst. For example, the systems analyst might have observed how a company filled several customer orders. However, the procedures the systems analyst observed may have been the steps used to fill a number of regular customer orders. If any of those orders had been special orders (e.g., an order for goods not normally kept in stock), the systems analyst would have observed a different set of procedures being executed.
- If people have been performing tasks in a manner that violates standard operating procedures, they may temporarily perform their jobs correctly while you are observing them. In other words, people may let you see what they want you to see.

Guidelines for Observation How does the systems analyst obtain facts through observation? Does one simply arrive at the observation site and begin recording

everything that's viewed? No. Much preparation should take place in advance. The analyst must determine how data will actually be captured. Will it be necessary to have special forms on which to quickly record data? Will the individuals being observed be bothered by having someone watch and record their actions? When are the low, normal, and peak periods of operations for the task to be observed? The systems analyst must identify the ideal time to observe a particular aspect of the system.

An analyst should plan to observe a site when there is a typical workload. Once a typical workload has been observed, observations can be made during peak periods to gather information for measuring the effects caused by the increased volume. As part of the analyst's observation, he or she should obtain samples of documents or forms used by those being observed.

work sampling a fact-finding technique that involves a large number of observations taken at random intervals.

The sampling techniques discussed earlier are also useful for observation. In this case, the technique is called **work sampling,** wherein a large number of observations may be conducted at random intervals. This technique is less threatening to the people being observed because the observation period is not continuous. When using work sampling, an analyst needs to predefine the operations of the job to be observed. A sample size then needs to be calculated as was done for document and file sampling. The analyst should make many random observations, being careful to observe activities at different times of the day. By counting the number of occurrences of each operation during the observations, the analyst will get a feel for how employees spend their days.

The following guidelines are key to honing observation skills:

- Determine the who, what, where, when, why, and how of the observation.
- Obtain permission to observe from appropriate supervisors or managers.
- Inform those who will be observed of the purpose of the observation.
- Keep a low profile.
- Take notes during or immediately following the observation.
- Review observation notes with appropriate individuals.
- Don't interrupt individuals at work.
- Don't focus heavily on trivial activities.
- Don't make assumptions.

Living the System In this type of observation the systems analyst actively performs the role of the user for a short period of time. This is one of the most effective ways to learn about problems and requirements of the system. By filling the user's "shoes," a systems analyst quickly gains an appreciation for what the user experiences and what she or he has to do to perform the job. This type of role playing gives the systems analyst a firsthand education in the business processes and functions, as well as the problems and challenges associated with them.

> Questionnaires

questionnaire a document that allows the analyst to collect information and opinions from respondents.

Another fact-finding technique is conducting surveys through **questionnaires.** The document can be mass-produced and distributed to respondents, who can then complete the questionnaire on their own time. Questionnaires allow the analyst to collect facts from a large number of people while maintaining uniform responses. When dealing with a large audience, no other fact-finding technique can tabulate the same facts as efficiently.

Collecting Facts by Using Questionnaires Systems analysts have often criticized the use of questionnaires. Many systems analysts claim that the responses lack reliable and useful information. Nevertheless, questionnaires can be an effective means of fact gathering, and many of these criticisms can be attributed to the inappropriate or

ineffective use of the questionnaires. Before using questionnaires, an analyst should understand the pros and cons associated with their use:

Advantages

- Most questionnaires can be answered quickly. People can complete and return questionnaires at their convenience.
- Questionnaires are a relatively inexpensive means of gathering data from a large number of individuals.
- Questionnaires allow individuals to maintain anonymity. Therefore, individuals are more likely to provide the real facts, rather than telling you what they think their boss would want them to.
- Responses can be tabulated and analyzed quickly.

Disadvantages

- The number of respondents is often low.
- There's no guarantee that an individual will answer or expand on all of the questions.
- Questionnaires tend to be inflexible. There's no opportunity for the systems analyst to obtain voluntary information from individuals or to reword questions that may have been misinterpreted.
- It's not possible for the systems analyst to observe and analyze the respondent's body language.
- There is no immediate opportunity to clarify a vague or incomplete answer to any question.
- Good questionnaires are difficult to prepare.

Types of Questionnaires There are two formats for questionnaires: free format and fixed format. **Free-format questionnaires** are designed to allow the users to exercise more freedom or latitude in their answers to each question.

free-format questionnaire a questionnaire designed to offer the respondent greater latitude in the answer. A question is asked, and the respondent records the answer in the space provided after the question.

Here are two examples of free-format questions:

- What reports do you currently receive and how are they used?
- Are there any problems with these reports (e.g., are they inaccurate, is there insufficient information, or are they difficult to read and/or use)? If so, please explain.

Obviously, responses to such questions may be difficult to tabulate. It is also possible that the respondents' answers may not match the questions asked. In order to ensure useful responses in free-format questionnaires, the analyst should phrase the questions in simple sentences and not use words—such as *good*—that can be interpreted differently by different respondents. The analyst should also ask questions that can be answered with three or fewer sentences. Otherwise, the questionnaire may take up more time than the respondent is willing to sacrifice.

The second type of questionnaire is fixed-format. **Fixed-format questionnaires** are more rigid, requiring that the user select an answer from a predefined set of possible answers. Given any question, the respondent must choose from the available answers. This makes the results much easier to tabulate. On the other hand, the respondent cannot provide additional information that might prove valuable.

fixed-format questionnaire a questionnaire containing questions that require selecting an answer from predefined available responses.

There are three types of fixed-format questions:

1. For *multiple-choice questions*, the respondent is given several answers from which to choose. The respondent should be told if more than one answer can be selected. Some multiple-choice questions allow for very brief free-format responses when none of the standard answers apply. Examples of multiple-choice fixed-format questions are:

   Do you feel that backorders occur too frequently?
   ❑ YES ❑ NO

   Is the current accounts receivable report that you receive useful?
   ❑ YES ❑ NO

   If no, please explain.

2. For *rating questions,* the respondent is given a statement and asked to use supplied responses to state an opinion. To prevent built-in bias, there should be an equal number of positive and negative ratings. The following is an example of a rating fixed-format question:

   The implementation of quantity discounts would cause an increase in customer orders.
   ❑ Strongly agree
   ❑ Agree
   ❑ No opinion
   ❑ Disagree
   ❑ Strongly disagree

3. For *ranking questions,* the respondent is given several possible answers, which are to be ranked in order of preference or experience. An example of a ranking fixed-format question is:

   Rank the following transactions according to the amount of time you spend processing them:
   _____ new customer orders
   _____ order cancellations
   _____ order modifications
   _____ payments

Developing a Questionnaire Good questionnaires can be difficult to develop. The following procedure can prove helpful in developing an effective questionnaire:

1. Determine what facts and opinions must be collected and from whom you should get them. If the number of people is large, consider using a smaller, randomly selected group of respondents.
2. Based on the facts and opinions sought, determine whether free- or fixed-format questions will produce the best answers. A combination format that permits optional free-format clarification of fixed-format responses is often used.
3. Write the questions. Examine them for construction errors and possible misinterpretations. Make sure that the questions don't reveal your personal bias or opinions. Edit the questions.
4. Test the questions on a small sample of respondents. If your respondents had problems with them or if the answers were not useful, edit the questions.
5. Duplicate and distribute the questionnaire.

> Interviews

interview a fact-finding technique whereby the systems analyst collects information from individuals through face-to-face interaction.

The personal interview is generally recognized as the most important and most often used fact-finding technique. Personal **interviews** involve soliciting requirements through direct, face-to-face interaction. Interviewing can be used to achieve any or all of the following goals: find facts, verify facts, clarify facts, generate enthusiasm, get the end user involved, identify requirements, and solicit ideas and opinions. There are two roles assumed in an interview. The systems analyst is the *interviewer,* responsible for organizing and conducting the interview. The system user or system owner is the *interviewee,* who is asked to respond to a series of questions.

There may be one or more interviewers and/or interviewees. In other words, interviews may be conducted one-on-one or many-to-many. Unfortunately, many systems analysts are poor interviewers. In this section you will learn how to conduct proper interviews.

Collecting Facts by Interviewing Users People are the most important element of an information system. More than anything else, people want to be in on things. No other fact-finding technique places as much emphasis on people as interviews. But people have different values, priorities, opinions, motivations, and personalities. Therefore, to use the interviewing technique, a systems analyst must possess good human relations skills for dealing effectively with different types of people. As with other fact-finding techniques, interviewing isn't the best method for all situations. Interviewing has its advantages and disadvantages, which should be weighed against those of other fact-finding techniques for every fact-finding situation:

Advantages

- Interviews give the analyst an opportunity to motivate the interviewee to respond freely and openly to questions. By establishing rapport, the systems analyst is able to give the interviewee a feeling of actively contributing to the systems project.
- Interviews allow the systems analyst to probe for more feedback from the interviewee.
- Interviews permit the systems analyst to adapt or reword questions for each individual.
- Interviews give the analyst an opportunity to observe the interviewee's nonverbal communication. A good systems analyst may be able to obtain information by observing the interviewee's body movements and facial expressions as well as by listening to verbal replies to questions.

Disadvantages

- Interviewing is a very time-consuming, and therefore a costly, fact-finding approach.
- Success of interviews is highly dependent on the systems analyst's human relations skills.
- Interviewing may be impractical due to the location of interviewees.

Interview Types and Techniques There are two types of interviews, unstructured and structured. **Unstructured interviews** are characterized as involving general questions that allow the interviewee to direct the conversation. This type of interview frequently gets off track, and the analyst must be prepared to redirect the interview back to the main goal or subject. For this reason, unstructured interviews don't usually work well for systems analysis and design. **Structured interviews** involve the interviewer asking specific questions designed to elicit specific information from the interviewee. Depending on the interviewee's responses, the interviewer will direct additional questions to obtain clarification or amplification. Some of these questions may be planned and others spontaneous.

Unstructured interviews tend to involve asking **open-ended questions.** Such questions give the interviewees significant latitude in their answers. An example of an open-ended question is "Why are you dissatisfied with the report of uncollectable accounts?" Structured interviews tend to involve asking more **closed-ended questions** that are designed to elicit short, direct responses from the interviewee. Examples of such questions are "Are you receiving the report of uncollectable accounts on time?" and "Does the report of uncollectable accounts contain accurate information?" Realistically, most questions fall between the two extremes.

unstructured interview an interview that is conducted with only a general goal or subject in mind and with few, if any, specific questions. The interviewer counts on the interviewee to provide a framework and direct the conversation.

structured interview an interview in which the interviewer has a specific set of questions to ask of the interviewee.

open-ended question a question that allows the interviewee to respond in any way that seems appropriate.

closed-ended question a question that restricts answers to either specific choices or short, direct responses.

> How to Conduct an Interview

A systems analyst's success is at least partially dependent on the ability to interview. A successful interview will involve selecting appropriate individuals to interview, preparing extensively for the interview, conducting the interview properly, and following up on the interview. Here we examine each of these steps in more detail. Let's assume that the analyst has identified the need for an interview and has determined exactly what kinds of facts and opinions are needed.

Select Interviewees The systems analyst should interview the end users of the information system being studied. A formal organization chart will help identify these individuals and their responsibilities. The analyst should attempt to learn as much as possible about each individual prior to the interview, such as the person's strengths, fears, biases, and motivations. The interview can then be geared to take the characteristics of the individual into account.

The analyst should make an appointment with the interviewee and never just drop in. The appointment should be limited to somewhere between a half hour and an hour. The higher the management level of the interviewee, the less time should be scheduled. If the interviewee is a clerical, service, or blue-collar worker, the analyst must get the supervisor's permission before scheduling the interview. It is also important to ensure that the location for the interview will be available during the time it is scheduled. Interviews should never be conducted in the presence of the analyst's officemates or the interviewee's peers.

Prepare for the Interview Preparation is the key to a successful interview. An interviewee can easily detect when an interviewer is unprepared and may resent the lack of preparation because it wastes valuable time. When the appointment is made, the interviewee should be notified about the subject of the interview. To ensure that all pertinent aspects of the subject are covered, the analyst should prepare an *interview guide.* The interview guide is a checklist of specific questions the interviewer will ask the interviewee. The interview guide may also contain follow-up questions that will be asked only if the answers to other questions warrant the additional answers. A sample interview guide is presented in Figure 6-3. Notice that the agenda is carefully laid out with the specific time allocated to each question. Time should also be reserved for asking follow-up questions and redirecting the interview. Questions should be carefully chosen and phrased. Most questions begin with the standard who, what, when, where, why, and how much type of wording. The following types of questions should be avoided:

- *Loaded questions,* such as "Do we have to have both of these columns on the report?" The question conveys the interviewee's personal opinion on the issue.
- *Leading questions,* such as "You're not going to use this OPERATOR CODE, are you?" The question leads the interviewee to respond, "No, of course not," regardless of actual opinion.
- *Biased questions,* such as "How many codes do we need for FOOD CLASSIFICATION in the INVENTORY FILE? I think 20 ought to cover it." These types of biased questions will influence an interviewee.

Interviewers should always avoid threatening or critical questions. The purpose of the interview is to investigate, not to evaluate or criticize. Additional guidelines for questions include:

- Use clear and concise language.
- Don't include your opinion as part of the question.
- Avoid long or complex questions.
- Avoid threatening questions.
- Don't use "you" when you mean a group of people.

Interviewee:	Jeff Bentley, Accounts Receivable Manager
Date:	January 19, 2003
Time:	1:30 P.M.
Place:	Room 223, Admin. Bldg.
Subject:	Current Credit-Checking Policy

Time Allocated	Interviewer Question or Objective	Interviewee Response
1 to 2 min.	**Objective** Open the interview: • Introduce ourselves. • Thank Mr. Bentley for his valuable time. • State the purpose of the interview — to obtain an understanding of the existing credit-checking policies.	
5 min.	**Question 1** What conditions determine whether a customer's order is approved for credit? **Follow-up**	
5 min.	**Question 2** What are the possible decisions or actions that might be taken once these conditions have been evaluated? **Follow-up**	
3 min.	**Question 3** How are customers notified when credit is not approved for their order? **Follow-up**	
1 min.	**Question 4** After a new order is approved for credit and placed in the file containing orders that can be filled, a customer might request that a modification be made to the order. Would the order have to go through credit approval again if the new total order cost exceeds the original cost? **Follow-up**	
1 min.	**Question 5** Who are the individuals who perform the credit checks? **Follow-up**	
1 to 3 min.	**Question 6** May I have permission to talk to those individuals to learn specifically how they carry out the credit-checking process? **Follow-up** If so: When would be an appropriate time to meet with each of them?	
1 min.	**Objective** Conclude the interview: • Thank Mr. Bentley for his cooperation and assure him that he will be receiving a copy of what transpired during the interview.	
21 minutes	Time allotted for questions and objectives	
9 minutes	Time allotted for follow-up questions and redirection	
30 minutes	Time allotted for interview (1:30 p.m. - 2:00 p.m.)	
General Comments and Notes:		

FIGURE 6-3 Sample Interview Guide

Conduct the Interview Respect your interviewee and his or her time. Dress to match the interviewee. That generally means that you will dress differently to interview managers than you will to interview workers on the loading dock. If the interview will be held in a meeting room other than the interviewee's office, arrive early to make sure it is set up appropriately.

Open the interview by thanking the interviewee in advance. State the purpose and length of the interview and how the gathered data will be used. Then monitor the time so you will keep your promise.

Ask follow-up questions. Probe until you understand the system requirements. Ask especially about exception conditions. As what-if questions, such as "What if the check doesn't clear?" or "What happens if a product is not in stock?"

Listen closely, observe the interviewee, and take notes concerning both verbal and nonverbal responses from the interviewee. It's very important to keep the interview on track; this means anticipating the need to adapt the interview, if necessary. Questions can often be bypassed if they have been answered earlier or they can be deleted if determined to be irrelevant, based on previous answers.

Here is a set of rules that an interviewer should follow:

Do

- Dress appropriately.
- Be courteous.
- Listen carefully.
- Maintain control of the interview.
- Probe.
- Observe mannerisms and nonverbal communication.
- Be patient.
- Keep the interviewee at ease.
- Maintain self-control.
- Finish on time.

Avoid

- Assuming an answer is finished or leading nowhere.
- Revealing verbal and nonverbal clues.
- Using jargon.
- Revealing your personal biases.
- Talking instead of listening.
- Assuming anything about the topic or the interviewee.
- Tape recording—a sign of poor listening skills.

Conclude the interview by expressing appreciation and providing answers to any questions posed by the interviewee. The conclusion is very important for maintaining rapport and trust with the interviewee.

Follow Up on the Interview To help maintain good rapport and trust with interviewees, the interviewer should send them a memo that summarizes the interview. This memo should remind the interviewees of their contributions to the systems project and allow them the opportunity to clarify any misinterpretations that the interviewer may have derived during the interview. In addition, the interviewees should be given the opportunity to offer additional information they may have failed to bring out during the interview.

Listening When most people talk about communication skills, they think of speaking and writing. The skill of listening is rarely mentioned, but it may be the most important skill during the interviewing process. In order to conduct a successful interview, the interviewer must make a distinction between hearing and listening: "To hear is to recognize that someone is speaking, to listen is to understand what the speaker wants to communicate."[4]

We have actually been conditioned most of our lives not to listen. Take, for example, how we can ignore our quarreling brothers and sisters while we enjoy our favorite CD or, as students, how we learn to study by blocking out distractions such as noisy roommates. We have learned not to listen, but we can also learn how to listen effectively and productively.

---

[4]Thomas R. Gildersleeve, *Successful Data Processing Systems Analysis* (Englewood Cliffs, NJ: Prentice Hall, 1978), p. 93.

When working with users trying to solve their problems, analysts may find that getting the users to communicate can be difficult. The following guidelines can help open the lines of communication:

- *Approach the session with a positive attitude.* The interviewer should make the best of any situation, and look at it as a fun, pleasurable experience.
- *Set the other person at ease.* Presenting a nice, cheerful attitude can help the person relax. The interviewer should start by talking about the person's interests or hobbies. Showing an interest in the interviewee's personal life sometimes can serve as an icebreaker and put the person more at ease.
- *Let the other person know you are listening.* The interviewer should always maintain eye contact when listening and use a response such as a head nod or an "uh-huh" to acknowledge what the person is saying. Good posture and leaning forward will tell the speaker that the interviewer is really interested in what the person is saying.
- *Ask questions.* The interviewer should ask questions to make sure he or she clearly understands what the person is saying or to clarify a point. This will show that the interviewer is listening; it will also give the other person the opportunity to expand on the answer.
- *Don't assume anything.* One of the worst things an interviewer can do is to act as if he or she is in a hurry. For example, if an interviewer assumes what the other person is going to say and cuts in and finishes the sentence, he or she will possibly miss what the person intended to say and irritate the speaker. Or if the speaker is interrupted because the interviewer has already heard the information and believes it is not applicable to the topic of the interview, a valuable piece of information may be missed. Don't assume anything! TV host Art Linkletter learned this lesson on his popular television show, *Kids Say the Darnedest Things,* when he asked a child a philosophical question:

  On my show I once had a child tell me he wanted to be an airline pilot. I asked him what he'd do if all the engines stopped out over the Pacific Ocean. He said "First I would tell everyone to fasten their seatbelts, and then I'd find my parachute and jump out."

  While the audience rocked with laughter, I kept my attention on the young man to see if he was being a smart alec. The tears that sprang into his eyes alerted me to his chagrin more than anything he could have said, so I asked him why he'd do such a thing. His answer revealed the sound logic of a child: "I'm going for gas . . . I'm coming back!"[5]

- *Take notes.* The process of taking notes serves two purposes. First, by jotting down brief notes while the other person is speaking, you give the person the impression that what he or she has to say is important enough to be written down. Second, the notes help the interviewer remember the major points of the meeting later.

Body Language and Proxemics What is body language, and why should a systems analyst care about it during the interviewing process? **Body language** is all the non-verbal information being communicated by an individual. Body language is a form of nonverbal communication that we all use and of which we are usually unaware.

body language the nonverbal information we communicate.

Studies have determined a startling fact: Of a person's total feelings, only 7 percent are communicated verbally (in words), whereas 38 percent are communicated by the tone of voice used and 55 percent are communicated by facial and body expressions. If you only listen to someone's words, you are missing most of what the person has to say.

[5]Donald Walton, *Are You Communicating? You Can't Manage without It.* (New York: McGraw-Hill, 1989), p. 31.

For this discussion, we will focus on just three aspects of body language: facial disclosure, eye contact, and posture. *Facial disclosure* means you can sometimes understand how a person feels by watching the expressions on his or her face. Many common emotions have easily recognizable facial expressions associated with them. However, the face is one of the most controlled parts of the body. Some people who are aware that their expressions often reveal what they are thinking are very good at disguising these expressions.

Another form of nonverbal communication is *eye contact.* Eye contact is the least controlled aspect of facial expression. Have you ever spoken to someone who will not look directly at you? How did it make you feel? A continual lack of eye contact may indicate uncertainty. A normal glance is usually from three to five seconds in length; however, direct-eye-contact time should increase with distance. Analysts need to be careful not to use excessive eye contact with users who seem threatened so that they won't further intimidate them. Direct eye contact can cause strong feelings, either positive or negative, in other people.

Posture is the least controlled aspect of the body. As such, body posture holds a wealth of information for the astute analyst. Members of a group who are in agreement tend to display the same posture. A good analyst will watch the audience for changes in posture that could indicate anxiety, disagreement, or boredom. An analyst should normally maintain an "open" body position, signaling approachability, acceptance, and receptiveness. In special circumstances, the analyst may choose to use a confrontation angle of head-on or at a 90-degree angle to another person in order to establish control and dominance.

proxemics the relationship between people and the space around them.

In addition to the information communicated by body language, individuals also communicate via proxemics. **Proxemics,** the relationship between people and the space around them, is a factor in communications that can be controlled by the knowledgeable analyst.

People still tend to be very territorial about their space. Observe where your classmates sit in one of your courses that does not have assigned seats. Or the next time you are involved in a conversation with someone, deliberately move much closer or farther away from the person and see what happens. A good analyst is aware of four spatial zones:

- *Intimate zone*—closer than 1.5 feet.
- *Personal zone*—from 1.5 feet to 4 feet.
- *Social zone*—from 4 feet to 12 feet.
- *Public zone*—beyond 12 feet.

Certain types of communications take place only in some of these zones. For example, an analyst conducts most interviews with system users in the personal zone. But the analyst may need to move back to the social zone if the user displays any signs (body language) of being uncomfortable. Sometimes increasing eye contact can make up for a long distance that can't be changed. Many people use the fringes of the social zone as a "respect" distance.

We have examined some of the informal ways that people communicate their feelings and reactions. A good analyst will use all the information available, not just the written or verbal communications of others.

> Discovery Prototyping

Another type of fact-finding technique is prototyping. Prototyping was introduced in Chapter 3 for use in rapid application development (RAD). As you should recall, the concept behind prototyping is building a small working model of the users' requirements or a proposed design for an information system. This type of prototyping is usually a design technique, but the approach can be applied earlier in the system

development life cycle to perform fact-finding and requirements analysis. The process of building a prototype for the purpose of identifying requirements is referred to as **discovery prototyping.**

discovery prototyping the act of building a small-scale representative or working model of the users' requirements in order to discover or verify those requirements.

Discovery prototyping is frequently applied to systems development projects, especially in cases where the development team is having problems defining the system requirements. The philosophy is that the users will recognize their requirements when they see them. It is important that the prototype be developed quickly so that it can be used during the development process. Usually, only the areas where the requirements are not clearly understood are prototyped. This means that a lot of desired functionality may be left out and quality assurance may be ignored. Also, nonfunctional requirements such as performance and reliability may be less stringent than they would be for the final product. Technologies other than the ones used for the final software are frequently used to build the discovery prototypes. In these cases, the prototypes are most likely discarded when the system is finished. This "throw-away" approach is primarily used to gather information and develop ideas for the system concept. Many areas of a proposed system may not be clearly understood, or some features may be a technical challenge for the developers. Creating discovery prototypes enables the developers as well as the users to better understand and refine the issues involved with developing the system. This technique helps minimize the risk of delivering a system that doesn't meet the user's needs or that can't fulfill the technical requirements.

Discovery prototyping has its advantages and disadvantages, which should be weighed against those of other fact-finding techniques for every fact-finding situation:

Advantages

- Allows users and developers to experiment with the software and develop an understanding of how the system might work.
- Aids in determining the feasibility and usefulness of the system before high development costs are incurred.
- Serves as a training mechanism for users.
- Aids in building system test plans and scenarios to be used last in the system testing process.
- May minimize the time spent on fact-finding and help define more stable and reliable requirements.

Disadvantages

- Developers may need to be trained in the prototyping approach.
- Users may develop unrealistic expectations based on the performance, reliability, and features of the prototype. Prototypes can only simulate system functionality and are incomplete in nature. Care must be taken to educate the users about this fact and not to mislead them.
- Doing prototyping may extend the development schedule and increase the development costs.

> Joint Requirements Planning

Many organizations are using the group work session as a substitute for numerous and separate interviews. One example of the group work session approach is **joint requirements planning (JRP)**, wherein highly structured group meetings are conducted for the purpose of identifying and analyzing problems and defining system requirements. This and similar techniques generally require extensive training to work as intended. However, they can significantly decrease the time spent on fact-finding in one or more phases of the life cycle. JRP is becoming increasingly common in systems planning and systems analysis to obtain group consensus on problems, objectives, and requirements. In this section, you will learn about the participants of a JRP session

joint requirements planning (JRP) a process whereby highly structured group meetings are conducted for the purpose of analyzing problems and defining requirements.

and their roles. We will also discuss how to go about planning and conducting a JRP session, the tools and techniques that are used during a JRP session, and the benefits to be achieved through JRP.

JRP Participants Joint requirements planning sessions include a wide variety of participants and roles. Each participant is expected to attend and actively participate for the entire JRP session. Let's examine the different types of individuals involved in a typical JRP session and their roles:

- *Sponsor*—Any successful JRP session requires a single person, called the *sponsor,* to serve as its champion. This person is normally an individual who is in top management (*not* IT or IS management) and who has authority that spans the different departments and users who are to be involved in the systems project. The sponsor gives full support to the systems project by encouraging designated users to willingly and actively participate in the JRP session. Recalling the "creeping commitment" approach to systems development, it is the sponsor (system owner) who usually makes final decisions regarding the go or no-go direction of the project.

  The sponsor plays a visible role during a JRP session by kicking off the meeting by introducing the participants. Often, the sponsor will also make closing remarks for the session. The sponsor also works closely with the JRP leader to plan the session by helping identify individuals from the user community who should attend and determining the time and location for the JRP session.
- *Facilitator*—JRP sessions involve a single individual who plays the role of the leader or facilitator. The *JRP facilitator* is usually responsible for leading all sessions that are held for a systems project. This individual is someone who has excellent communication skills, possesses the ability to negotiate and resolve group conflicts, has a good knowledge of the business, has strong organizational skills, is impartial to decisions that will be addressed, and does not report to any of the JRP session participants.

  It is sometimes difficult to find an individual within the company who possesses all these traits. Thus, companies often must provide extensive JRP training or hire an expert from outside the organization to fill this role. Many systems analysts are trained to become JRP facilitators.

  The role of the JRP facilitator is to plan the JRP session, conduct the session, and follow through on the results. During the session, the facilitator is responsible for leading the discussion, encouraging the attendees to actively participate, resolving issue conflicts that may arise, and ensuring that the goals and objectives of the meeting are fulfilled. It is the JRP facilitator's responsibility to establish the ground rules that will be followed during the meeting and ensure that the participants abide by these rules.
- *Users and managers*—Joint requirements planning includes a number of participants from the user and management sectors of an organization who are given release time from their day-to-day jobs to devote themselves to active involvement in the JRP sessions. These participants are normally chosen by the project sponsor, who must be careful to ensure that each person has the business knowledge to contribute during the fact-finding sessions. The project sponsor must exercise authority and encouragement to ensure that these individuals will be committed to actively participating.

  A typical JRP session may involve anywhere from a relatively small number of user/management people to a dozen or more. The role of the users during a JRP session is to effectively communicate business rules and requirements, review design prototypes, and make acceptance decisions. The role of the managers during a JRP session is to approve project objectives, establish project priorities, approve schedules and costs, and approve

identified training needs and implementation plans. Overall, both users and managers are relied on to ensure that their critical success factors are being addressed.

- *Scribe(s)*—A JRP session also includes one or more *scribes,* who are responsible for keeping records pertaining to everything discussed in the meeting. These records are published and disseminated to the attendees immediately following the meeting in order to maintain the momentum that has been established by the JRP session and its members. The need to quickly publish the records is reflected by the fact that scribes are increasingly using CASE tools to capture many facts (documented using data and process models) that are communicated during a JRP session. Thus, it is advantageous for scribes to possess strong knowledge of systems analysis and design and be skilled with using CASE tools. Systems analysts frequently play this role.
- *IT staff*—A JRP session may also include a number of IT personnel who primarily listen and take notes regarding issues and requirements voiced by the users and managers. Normally, IT personnel do not speak up unless invited to do so. Rather, any questions or concerns they have are usually directed to the JRP facilitator immediately after or before the JRP session. It is the JRP facilitator who initiates and facilitates discussion of issues by users and managers.

  The IT staff in the JRP session usually consists of members of the project team. These members may work closely with the scribe to develop models and other documentation related to facts communicated during the meeting. Specialists may also be called on to gain information regarding special technical issues and concerns that may arise. When the situation warrants, the JRP facilitator may prompt an IT professional to address the technical issue.

How to Plan JRP Sessions Most JRP sessions span three to five days and occasionally last up to two weeks. The success of any JRP session depends on properly planning and effectively carrying out the plan. Some preparation is necessary well before the JRP session can be performed. Before planning a JRP session, the analyst must work closely with the executive sponsor to determine the scope of the project that is to be addressed through JRP sessions. It is also important that the high-level requirements and expectations of each JRP session be determined. This normally involves interviewing selected individuals who are responsible for departments or functions that are to be addressed by the systems project. Finally, before planning the JRP session, the analyst must ensure that the executive sponsor is willing to commit people, time, and other resources to the session.

Planning for a JRP session involves three steps: selecting a location for the JRP session, selecting JRP participants, and preparing an agenda to be followed during the JRP session. Let's examine each of these planning steps in detail:

1. *Selecting a location for JRP sessions*—When possible, JRP sessions should be conducted away from the company workplace. Most local hotels or universities have facilities designed to host group meetings. By holding the JRP session at an off-site location, the attendees can concentrate on the issues and activities related to the JRP session and avoid interruptions and distractions that would occur at their regular workplace. Regardless of the location of the JRP session, all attendees should be required to attend and be prohibited from returning to their regular workplaces.

   A JRP session typically requires several rooms. A conference room is required in which the entire group can meet to address JRP issues. Also, if the JRP session includes many people, several small breakout rooms may be needed for separate groups of people to meet and focus discussion on specific issues.

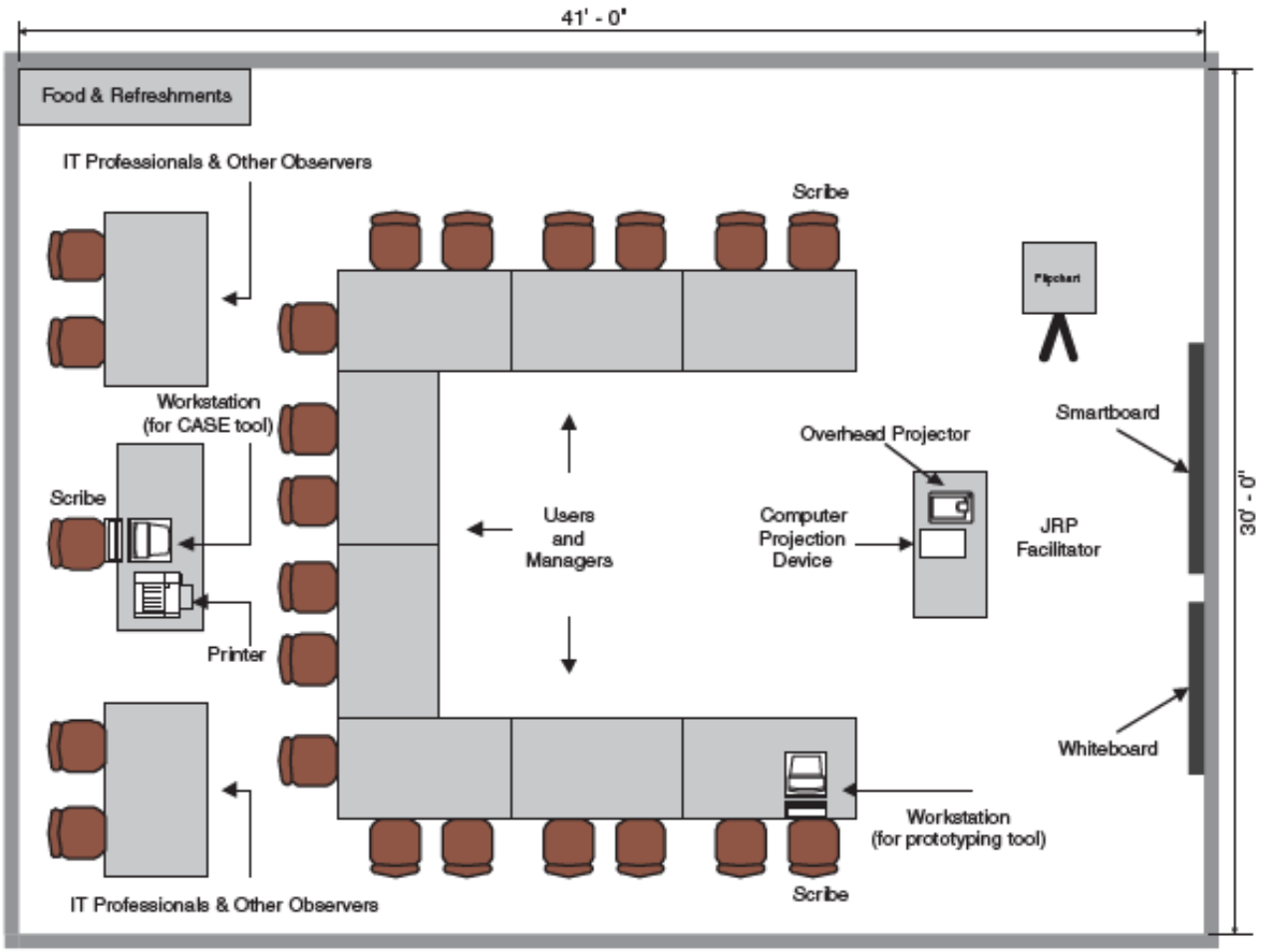


FIGURE 6-4 Typical Room Layout for JRP Session

The conference or main meeting room should comfortably hold all the attendees. The room should be fully equipped with tables, chairs, and other items that meet the needs of all attendees. Figure 6-4 depicts a typical room layout for a JRP session. Typical visual aids for a JRP room should include a whiteboard, smartboard, or blackboard; one or more flipcharts; and one or more projectors.

The room should also include computer equipment needed by scribes to record facts and issues communicated during the session. The computer itself should include software packages to support the various types of records or documentation to be captured and later published by the scribes. Such software may include CASE tool, word processing, spreadsheet, presentation package, prototyping software, printer, copier (or quick access), and computer projection capability. As a guideline, computer equipment (except that used for prototyping) should be located at the rear of the room so that it doesn't interfere or become a distraction for the JRP participants. Personal interaction of the participants, not technology, should be the focus of the session.

Finally, the room should be equipped for teleconferencing so that users at distant locations can participate. The room should include notepads and pencils for users, managers, and other attendees. Attendees should also be provided with nametags, place cards, snacks, and drinks so that they will be as

comfortable as possible. Creature comforts are very important since JRP sessions are very intensive and often run the entire day.

2. *Selecting JRP participants*—As mentioned earlier, participants selected include the JRP facilitator, scribe(s), and representatives from the user community. The users should be key individuals who are knowledgeable about their business area. Unfortunately, managers are often very dependent on these individuals to run their business areas and are often hesitant to release them from their duties. Thus, the analyst must ensure that management is committed to the JRP project and willing to not only permit but also require these key individuals to participate.

   Various IT professionals may also be selected to be involved in the JRP session. Usually all IT individuals assigned to the project team are involved in the JRP session. Other IT specialists may also be assigned to address specific technical issues pertaining to the project.

3. *Preparing a JRP session agenda*—Preparation is the key to a successful JRP session. The JRP facilitator must prepare documentation to brief the participants about the scope and objectives of the sessions. In addition, an agenda for each JRP session should be prepared and distributed before each session. The agenda dictates issues to be discussed during the session and the amount of time allotted to each item.

   The agenda should contain three parts: the opening, body, and conclusion. The opening is intended to communicate the expectations of the session, to communicate the ground rules, and to influence or motivate the attendees to participate. The body is intended to detail the topics or issues to be addressed in the JRP session. Finally, the conclusion represents the time set aside to summarize the day's session and to remind the attendees of unresolved issues (to be carried forward).

How to Conduct a JRP Session The JRP session begins with opening remarks, introductions, and a brief overview of the agenda and objectives for the session. The JRP facilitator will direct the session by following the prepared script. To successfully conduct the session, the facilitator should follow these guidelines:

- Do not unreasonably deviate from the agenda.
- Stay on schedule (agenda topics are allotted specific times).
- Ensure that the scribe is able to take notes (this may mean having the users and managers restate their points more slowly or clearly).
- Avoid the use of technical jargon.
- Apply conflict resolution skills.
- Allow for ample breaks.
- Encourage group consensus.
- Encourage user and management participation without allowing individuals to dominate the session.
- Make sure that attendees abide by the established ground rules for the session.

One of the goals of a JRP session is to generate possible ideas to solve a problem. One approach is brainstorming. **Brainstorming** involves encouraging participants to generate as many ideas as possible, without analyzing the validity of the ideas.

brainstorming a technique for generating ideas by encouraging participants to offer as many ideas as possible in a short period of time without any analysis until all the ideas have been exhausted.

Brainstorming is a formal technique that requires discipline. The following guidelines should be used to ensure effective brainstorming:

1. Isolate the appropriate people in a place that will be free from distractions and interruptions.
2. Make sure that everyone understands the purpose of the meeting (to generate ideas to solve the problem) and focuses on the problem(s).

3. Appoint one person to record ideas. This person should use a flipchart, chalkboard, or overhead projector that can be viewed by the entire group.
4. Remind everyone of the brainstorming rules:
   a. Be spontaneous. Call out ideas as fast as they occur.
   b. Absolutely no criticism, analysis, or evaluation of any kind is permitted while the ideas are being generated. Any idea may be useful, if only to spark another idea.
   c. The goal is quantity of ideas, not necessarily quality.
5. Within a specified time period, team members call out their ideas as quickly as they can think of them.
6. After the group has run out of ideas and all ideas have been recorded, then and only then should the ideas be analyzed and evaluated.
7. Refine, combine, and enhance the ideas that were generated earlier.

With a little practice and attention to these rules, brainstorming can be a very effective technique for generating ideas to solve problems.

As mentioned earlier, the success of a JRP session is highly dependent on planning and the skills of the JRP facilitator and scribes. These skills improve through proper training and experience. Therefore, JRP sessions are usually concluded with an evaluation questionnaire for the participants to complete. The responses will help ensure the likelihood of future JRP successes.

The end product of a JRP session is typically a formal written document. This document is usually created by the JRP facilitator and scribes. It is essential for confirming the specifications agreed on during the session by all participants. The content and organization of the specifications are obviously dependent on the objectives of the JRP session. The analyst may provide a different set of specifications to different participants based on their role—for example, a manager may receive more of a summary version of the document provided to the user participants (especially in cases in which the system owners had minimal actual involvement in the JRP session).

Benefits of JRP Joint requirements planning offers many benefits as an alternative fact-finding and development approach. More and more companies are beginning to realize its advantages and are incorporating JRP into their existing methodologies. An effectively conducted JRP session offers the following benefits:

- JRP actively involves users and management in the development project (encouraging them to take "ownership" in the project).
- JRP reduces the amount of time required to develop systems. This is achieved by replacing traditional, time-consuming one-on-one interviewing of each user and manager with group meetings. The group meetings allow for more easily obtaining consensus among the users and managers, as well as resolving conflicting information and requirements.
- When JRP incorporates prototyping as a means for confirming requirements and obtaining design approvals, the benefits of prototyping are realized.

Achieving a successful JRP session depends on the JRP facilitator and his or her ability to plan and facilitate the JRP session.

A Fact-Finding Strategy

An analyst needs an organized method for collecting facts. Inexperienced analysts will frequently jump right into interviews. They believe, "Go to the people. That's where the real facts are!" Wrong! This approach fails to recognize an important fact of life: People must complete their day-to-day jobs. You may be thinking, "But

I thought you've been saying that the system is for people and that direct end-user involvement in systems development is essential. Aren't you contradicting yourselves?"

Not at all. Time is money. To waste end users' time is to waste their company's money. To make the most of the time spent with end users, analysts should not jump right into interviews. Analysts should first collect all the facts they can by using other methods. Consider the following step-by-step strategy:

1. Learn from existing documents, forms, reports, and files. Analysts can learn a lot without any people contact.
2. If appropriate, observe the system in action.
3. Given all the facts already collected, design and distribute questionnaires to clear up things that aren't fully understood.
4. Conduct interviews (or group work sessions). Because most of the pertinent facts have already been collected by low-user-contact methods, interviews can be used to verify and clarify the most difficult issues and problems. (Alternatively, consider using JRP techniques to replace or complement interviews.)
5. (Optional). Build discovery prototypes for any functional requirements that are not understood or for requirements that need to be validated.
6. Follow up. Use appropriate fact-finding techniques to verify facts (usually interviews or observation).

The strategy is not sacred. Although a fact-finding strategy should be developed for every pertinent phase of systems development, every project is unique. Sometimes observation and questionnaires may be inappropriate. But the idea should always be to collect as many facts as possible before using interviews.

Learning Roadmap

This chapter introduced you to a wide range of techniques for discovering information system requirements. Most systems development methodologies require some level of documentation and analysis of system requirements. Accordingly, the remaining chapters in this part present a number of systems documentation tools and techniques that can be used during the analysis phase of systems development. Most of you will proceed directly to Chapter 7, "Modeling System Requirements with Use Cases." Use-case models serve as a foundation for the development of subsequent models for modeling additional systems requirements and are presented in Chapters 8 through 11.

Summary

1. The process and techniques that a systems analyst uses to identify, analyze, and understand system requirements are referred to as requirements discovery.
2. System requirements specify what the information system must do, or what property or quality the system must have.
3. The process of requirements discovery consists of the following activities:
   a. Problem discovery and analysis.
   b. Requirements discovery.
   c. Documenting and analyzing requirements.
   d. Requirements management.
4. Fact-finding is a technique that is used across the entire development cycle, but it is extremely critical in the requirements analysis phase.
5. A popular tool used by development teams to identify, analyze, and solve problems is an Ishikawa diagram.
6. Conducting business in an ethical manner is a required practice, and analysts need to be more aware of the implications of not being ethical.
7. There are seven common fact-finding techniques:
   a. The sampling of existing documents and files can provide many facts and details with little or no direct personal communication being necessary. The analyst should collect historical documents, business operations manuals and forms, and information systems documents.
   b. Research is an often-overlooked technique based on the study of other similar applications. It now has become more convenient with the Internet and World Wide Web (WWW). Site visits are a special form of research.
   c. Observation is a fact-finding technique in which the analyst studies people doing their jobs.
   d. Questionnaires are used to collect similar facts from a large number of individuals.
   e. Interviews are the most popular but the most time-consuming fact-finding technique. When interviewing, the analyst meets individually with people to gather information.
      i) When most people talk about communication skills, they think of speaking and writing. The skill of listening hardly gets mentioned at all, but it may be the most important, especially during the interviewing process.
      ii) Research studies have determined a startling fact: Of a person's total feelings, only 7 percent are communicated verbally (in words), whereas 38 percent are communicated by the tone of voice used and 55 percent are communicated by facial and body expressions. If you only listen to someone's words, you are missing most of what the person has to say. Experienced systems analysts pay close attention to body language and proxemics.
   f. Discovery prototyping is frequently applied to systems development projects, especially in cases where the development team is having problems defining the system requirements. The philosophy is that the users will recognize their requirements when they see them. It is important that the prototype be developed quickly so that it can be used during the development process.
   g. Many analysts find flaws with interviewing—separate interviews often lead to conflicting facts, opinions, and priorities. The end result is numerous follow-up interviews and/or group meetings. For this reason, many organizations are using a group work session known as the joint requirements planning session as a substitute for interviews.
      i) Joint requirements planning sessions include a wide variety of participants and roles. Each participant is expected to attend and actively participate for the entire duration of the JRP session.
      ii) An effective JRP session involves extensive planning. Planning for a JRP session involves three steps: selecting a location for the JRP session, selecting JRP participants, and preparing an agenda to be followed during the JRP session.
8. To help alleviate the many problems associated with changing requirements, it is necessary to perform requirements management. Requirements management encompasses the policies, procedures, and processes that govern how a change to a requirement is handled.
9. Because "time is money," it is wise and practical for the systems analyst to use a fact-finding strategy to maximize the value of time spent with the end users.
   a. Learn from existing documents, forms, reports, and files. Analysts can learn a lot without any people contact.

b. If appropriate, observe the system in action.
c. Given all the facts already collected, design and distribute questionnaires to clear up things that aren't fully understood.
d. Conduct interviews (or group work sessions). Because most of the pertinent facts have already been collected by low-user-contact methods, interviews can be used to verify and clarify the most difficult issues and problems. (Alternatively, consider using JRP techniques to replace or complement interviews.)
e. (Optional.) Build discovery prototypes for any functional requirements that are not understood or for requirements that need to be validated.
f. Follow up. Use appropriate fact-finding techniques to verify facts (usually interviews or observation).

Review Questions

1. What is the importance of conducting the requirements discovery process?
2. What are the possible consequences if you fail to identify system requirements correctly and completely?
3. What are some of the criteria deemed to be critical in defining system requirements?
4. The requirements discovery process consists of what activities?
5. Briefly describe the purpose and component parts of an Ishikawa diagram.
6. What technique is commonly used in the requirements discovery phase? Why is it important?
7. Why is analyzing requirements essential?
8. When collecting facts from existing documentation, what kind of documents should system analysts review?
9. What are some of the drawbacks of collecting facts by observing employees in their work environment? How can systems analysts deal with these drawbacks?
10. What are the types of survey questionnaires that systems analysts can use to collect information and opinions?
11. What are some of the ways that you can use to help open the lines of communication in an interview?
12. What is joint requirements planning (JRP)?
13. Why has JRP become popular?
14. Why is the facilitator in JRP so important?
15. What is the main concern in selecting a location for JRP sessions?

Problems and Exercises

1. You are managing a project that was postponed twice because its funding was diverted to higher-priority projects. The system owners do not want that to happen again, so they are very anxious to get the new system started and built as quickly as possible. They are putting a great deal of pressure on you to spend no more than a couple of days on requirements discovery. If anything is missed, they tell you, it can be fixed later on. You really want to make them happy, but a little voice of caution is going off. What are the potential consequences and costs of rushing through the requirements discovery process?
2. You have learned the importance of making sure that requirements are correctly identified. But how do you know when you have a correct requirement—that is, what criteria must each requirement meet in order to be considered correct?
3. What common error does a new systems analyst often make when analyzing a problem? What are the potential consequences of this error? What tool can be used to help avoid this problem?
4. System developers use fact-finding techniques in every project phase. Is fact-finding more important during the requirements analysis phase than for other phases? Why or why not?
5. What ethical issues might arise during the fact-finding process, and how should they be handled?
6. What are some of the common tools and techniques a systems analyst can use to document the initial findings? Should the systems analyst expect the requirements to be complete and correct at this point? If not, what are the common problems? What should be the focus of the project team at this point?

7. What is the deliverable that is created once requirements analysis is completed? Why is this deliverable needed, and what does it include? Who are the audience and/or users of this deliverable, and for what reasons?
8. You are a systems analyst in a software development company that has been hired to do the requirements analysis phase for a large organization. What are three categories of existing documentation that you should collect during requirement discovery? What are some examples of each of these three types of documentation? What should the systems analyst watch out for in collecting documentation?
9. Suppose that you are a systems analyst on a project that involves modifying the sales order process. Since your company receives in the neighborhood of 2,500 sales orders per day, how many do you need to sample if you want 95 percent certainty that you have covered all variations? What if the number of sales orders per day was 25,000 orders?
10. Surveys and questionnaires are frequently used to gather facts. What are some of the advantages and disadvantages of questionnaires? When might you choose free-format questionnaires over fixed-format questionnaires? What is one method of determining the effectiveness of a questionnaire?
11. What are some of the reasons to use joint requirements planning (JRP) as a fact-finding technique? What should be the basis for selecting which users and managers will participate in the JRP session, and who generally selects them? What skills should the facilitator and scribe possess? What is the role of IT staff during JRP sessions? What is the typical duration of the JRP sessions?
12. Provide at least five of the critical success factors for JRP sessions.
13. What one thing should an analyst *not* do when beginning the fact-finding portion of requirements discovery, no matter how tempting?

Projects and Research

1. Systems analysts must have expertise in problem analysis. When systems analysts are starting out, they often find it difficult to differentiate symptoms from problems, and to identify the actual causes of the problem. One tool that can help analysts learn to do this is the Ishikawa, or fishbone, diagram.
   a. Find and select a problem that your organization, school, or other organization is currently attempting to resolve. Describe this problem.
   b. Follow the process described in this chapter and create an Ishikawa diagram.
   c. Which categories did you start with in the diagram, and which categories did you add during the process?
   d. Did this diagram help in finding the actual cause(s) of the problem? Did the cause(s) turn out to be what you originally thought, or something different?
2. Observing the work environment is a technique that predates the information age, but that can still be highly effective. Although not applicable for every situation, observing what people actually do and how they do it can be in some cases much more accurate than asking them! Select a system—whether hypothetical or real—and do the following:
   a. Provide an overview of the system and what you are trying to learn about the system for a project.
   b. Develop a work observation plan using the guidelines in this chapter. The format is up to you, but it generally should not need more than 1–2 pages.
   c. Develop a work-sampling plan, and describe the sampling procedures you will use.
   d. What are your thoughts about this method compared to other fact-finding methods?
3. You are a systems analyst working on a project to develop an intranet for a large organization with several thousand employees working in offices throughout the United States. This will be the organization's first intranet, and executive management wants it to help increase employee efficiency and commitment to the organization. As part of fact-finding, information needs to be gathered from employees of the organization regarding intranet content and functionality. Due to the size and geographic distribution of the organization, as well as project time constraints, there is insufficient time

and resources for personal interviews, so you have decided that a questionnaire is needed.

a. What facts and opinions do you need to collect?
b. Should all employees in the organization be surveyed? Why or why not? If not all employees should be surveyed, how would you select the employees to be surveyed?
c. What format do you think would work best for this survey questionnaire? If fixed format, what type(s) of fixed-format questions should be used?
d. How long should the survey questionnaire be in order to get the necessary information without discouraging employees from filling it out?
e. Create the survey questionnaire, using the question-writing guidelines given in this chapter.

4. Based upon the responses to your intranet survey, you feel that it would be helpful to interview someone in another organization who has had experience in developing and/or maintaining company intranets.

a. What type of interview do you think would be most appropriate in this situation—unstructured or structured? Why?
b. Make an appointment with the intranet administrator in your organization or another organization or school to discuss her or his experiences in developing and/or maintaining an intranet. Describe the organization and its intranet.
c. Prepare an interview guide using the format of Figure 6-3 as an example, ensuring that questions are free of the problems discussed in this chapter.
d. Conduct the interview, and record the responses.
e. What do you feel worked well in the interview, and what would you do differently next time?

5. Body language is an extremely important part of communication, as described in the textbook. Analysts need to be aware of not only what is being communicated through the body language of the interviewee but also the impact that their own body language may have upon the interview process. Make an appointment with several co-workers or fellow students to interview regarding the features they would like to see in an intranet; if possible, select interviewees you know well and those that you don't. Prepare for the interviews following the same steps as in the prior question.

a. Describe the interviewees you selected and the questions you asked.
b. During each interview, observe the facial expressions of the interviewee. What did you observe? Were the facial expressions always consistent with the responses?
c. During each interview, observe the eye contact. How long did it last? Observe and describe what happened when you made eye contact for more than three to five seconds with the interviewee.
d. Try changing your spatial zone during the interview. Did the interviewee show any signs of being uncomfortable? At what point did that occur?
e. Did you note any differences in body language between those you knew well and those you didn't?
f. What did you do that was the most successful and the least successful in eliciting information?

6. Analysts typically have access to confidential or sensitive data during the requirements discovery phase of a project, particularly during fact-finding. Analysts need to be aware of situations where there may be a breach of professional ethics, whether by acts of commission or omission, and the possible consequences. Search on the Web and/or in business periodicals in your school library for articles on incidents involving breaches of professional ethics.

a. What articles did you find?
b. What was the nature of each of these incidents?
c. What were the consequences?
d. What was the analyst's personal responsibility in each incident?
e. What could have been done at the organizational and/or individual level to prevent the incident or to reduce its severity?

Minicases

1. In Chapter 5, you developed feasibility studies for a project. Economic feasibility assessments are impacted significantly by intangibles, whose value is obtained in part by interviews and questionnaires. Develop interview questions to determine the value to employees of telecommuting.
   a. Begin with unstructured questions posed to one group of employees to determine what matters to the employees and how they view telecommuting.
   b. Once you know what issues surround employee perceptions of telecommuting and why they might like/dislike it, create open-ended, but structured, questions on those issues, and interview a second set of employees. Why are we using two different groups of employees for this process?
2. Develop a questionnaire for mass employee distribution based on your findings from the previous interviews. Why are we completing the analysis with an anonymous survey?
3. You are in charge of developing a new online class registration system for your school. Develop a set of interview questions to determine issues and needs of students, registration staff, and faculty for an online registration system.
4. Discuss the impact that biased or leading questions may have on an analysis. Create one non-biased interview question and one biased or leading question. Pose each of those questions to five people. What kind of responses did you get? Were they what you expected?

Team and Individual Exercises

1. Create a biased, leading, or loaded set of interview questions. Pose them to another student in the class. The other student, instead of answering the questions, should tell you how you are biased and what response you are looking for.
2. Class exercise: Create as unbiased a set of interview questions as you can on a particular topic. Pose the questions to the class. However, wear a shirt, pins, and so on, that lead the class to respond in a particular way. Have fun, and experiment with visual aids, props, and the like.
3. It has been found in past research studies that employees who are allowed to telecommute actually work approximately three extra unpaid hours a week. But telecommuting is often used as a negotiating tool by an employer—in order to telecommute, employees must accept a lower salary, typically 10 percent. What do you think about this?

Suggested Readings

Andrews, D. C., and N. S. Leventhal. *Fusion Integrating IE, CASE and JAD: A Handbook for Reengineering the Systems Organization.* Englewood Cliffs, NJ: Prentice Hall, 1993.

Berdie, Douglas R., and John F Anderson. *Questionnaires: Design and Use.* Metuchen, NJ: Scarecrow Press, 1974. A practical guide to the construction of questionnaires. Particularly useful because of its short length and illustrative examples.

Davis, William S. *Systems Analysis and Design.* Reading, MA: Addison-Wesley, 1983. Provides useful pointers for preparing and conducting interviews.

Dejoie, Roy; George Fowler; and David Paradice. *Ethical Issues in Information Systems.* Boston, MA: Boyd and Fraser, 1991. Focuses on the impact of computer technology on ethical decision making in today's business organizations.

Fitzgerald, Jerry; Ardra F Fitzgerald; and Warren D. Stallings, Jr. *Fundamentals of Systems Analysis,* 2nd ed. New York: John Wiley & Sons, 1981. A useful survey text for the systems analyst. Chapter 6, "Understanding the Existing System," does a good job of presenting fact-finding techniques in the study phase.

Gane, C. *Rapid Systems Development.* New York: Rapid Systems Development, Inc., 1987. This book provides a good discussion on how to lead a group meeting/interview.

Gause, Donald C., and Gerald M. Weinberg. *Exploring Requirements: Quality before Design.* New York: Dorset House Publishing, 1989. An excellent book describing the techniques of requirements discovery.

Gildersleeve, Thomas R. *Successful Data Processing System Analysis.* Englewood Cliffs, NJ: Prentice Hall, 1978. Chapter 4, "Interviewing in Systems Work," provides a comprehensive look at interviewing specifically for the systems analyst. A thorough sample interview is scripted and analyzed in this chapter.

London, Keith R. *The People Side of Systems.* New York: McGraw-Hill, 1976. Chapter 5, "Investigation versus Inquisition," provides a very good people-oriented look at fact-finding, with considerable emphasis on interviewing.

Lord, Kenniston W., Jr., and James B. Steiner. *CDP Review Manual: A Data Processing Handbook,* 2nd ed. New York: Van Nostrand Reinhold, 1978. Chapter 8, "Systems Analysis and Design," provides a comprehensive comparison of the merits and demerits of each fact-finding technique. This material is intended to prepare data processors for the Certificate in Data Processing examinations, one of which covers systems analysis and design.

Miller, Irwin, and John E Freund. *Probability and Statistics for Engineers.* Englewood Cliffs, NJ: Prentice Hall, 1965. Introductory college textbook on probability and statistics.

Mitchell, Ian; Norman Partington; Peter Dunne; and John Moses. "Practical Prototyping, Part One," *Object Currents,* May 1996. First of a three-part series of articles that explores prototyping and how you can benefit from it. Prototyping is an integral part of JRP.

Robertson, Suzanne, and James Robertson. *Mastering the Requirements Process.* Reading, MA: ACM Press/Addison-Wesley, 1999. This book contains an in-depth coverage of step-by-step procedures for requirements discovery.

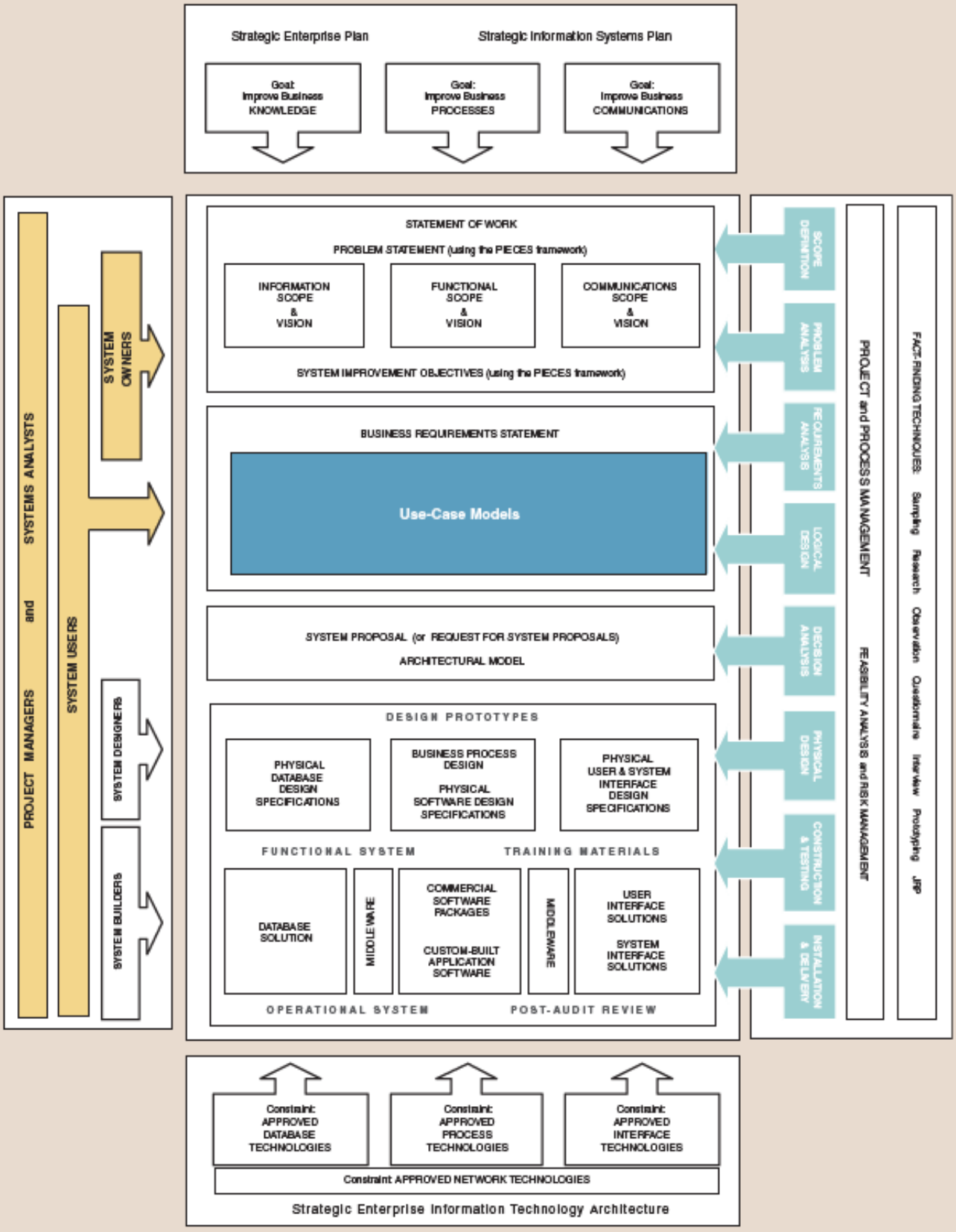
Salvendy, G., ed. *Handbook of Industrial Engineering.* New York: John Wiley & Sons, 1974. A comprehensive handbook for industrial engineers; systems analysts are, in a way, a type of industrial engineer. Excellent coverage on sampling and work measurement.

Stewart, Charles J., and William B. Cash, Jr. *Interviewing: Principles and Practices,* 2nd ed. Dubuque, IA: Brown, 1978. Popular college textbook that provides broad exposure to interviewing techniques, many of which are applicable to systems analysis and design.

Walton, Donald. *Are You Communicating? You Can't Manage without It.* New York: McGraw-Hill, 1989. This book is an easy-to-use guidebook on the process of communications and a must for anyone who must work with people and influence them.

Weinberg, Gerald M. *Rethinking Systems Analysis and Design.* Boston: Little, Brown and Company, 1982. A book created to stimulate a new way of thinking.

Wood, Jane, and Denise Silver. *Joint Application Design.* New York: John Wiley & Sons, 1989. This book provides a comprehensive overview of IBM's joint application design technique.


Strategic Enterprise Plan
Strategic Information Systems Plan
Goal: Improve Business KNOWLEDGE
Goal: Improve Business PROCESSES
Goal: Improve Business COMMUNICATIONS
PROJECT MANAGERS and SYSTEMS ANALYSTS
SYSTEM USERS
SYSTEM OWNERS
SYSTEM DESIGNERS
SYSTEM BUILDERS
STATEMENT OF WORK
PROBLEM STATEMENT (using the PIECES framework)
INFORMATION SCOPE & VISION
FUNCTIONAL SCOPE & VISION
COMMUNICATIONS SCOPE & VISION
SYSTEM IMPROVEMENT OBJECTIVES (using the PIECES framework)
BUSINESS REQUIREMENTS STATEMENT
Use-Case Models
SYSTEM PROPOSAL (or REQUEST FOR SYSTEM PROPOSALS)
ARCHITECTURAL MODEL
DESIGN PROTOTYPES
PHYSICAL DATABASE DESIGN SPECIFICATIONS
BUSINESS PROCESS DESIGN
PHYSICAL SOFTWARE DESIGN SPECIFICATIONS
PHYSICAL USER & SYSTEM INTERFACE DESIGN SPECIFICATIONS
FUNCTIONAL SYSTEM
TRAINING MATERIALS
DATABASE SOLUTION
MIDDLEWARE
COMMERCIAL SOFTWARE PACKAGES
CUSTOM-BUILT APPLICATION SOFTWARE
MIDDLEWARE
USER INTERFACE SOLUTIONS
SYSTEM INTERFACE SOLUTIONS
OPERATIONAL SYSTEM
POST-AUDIT REVIEW
SCOPE DEFINITION
PROBLEM ANALYSIS
REQUIREMENTS ANALYSIS
LOGICAL DESIGN
DECISION ANALYSIS
PHYSICAL DESIGN
CONSTRUCTION & TESTING
INSTALLATION & DELIVERY
PROJECT and PROCESS MANAGEMENT
FEASIBILITY ANALYSIS and RISK MANAGEMENT
FACT-FINDING TECHNIQUES: Sampling Research Observation Questionnaire Interview Prototyping JRP
Constraint: APPROVED DATABASE TECHNOLOGIES
Constraint: APPROVED PROCESS TECHNOLOGIES
Constraint: APPROVED INTERFACE TECHNOLOGIES
Constraint: APPROVED NETWORK TECHNOLOGIES
Strategic Enterprise Information Technology Architecture