recurring issues, assisting senior officials with difficult strategy formulations, and linking office information and corporate data. In some instances, the factors to be considered in an information systems project, such as the most appropriate aspect of computer or communications technology to be applied, the impact of a new system on the people in a firm, and the specific features the system should have, can be determined in a sequential fashion. In other instances, experience must be gained through experimentation and the staged evolution of a system.

As computers are used more and more by persons who are not computer professionals, the face of systems development is taking on an additional dimension. Users themselves are undertaking development of some of the systems they use, as the executive in the vignette emphasized.

These different situations are represented by three distinct approaches to the development of computer information systems:

1. Systems Development Life Cycle Method
2. Structured Analysis Development Method
3. Systems Prototype Method

This section discusses each approach, focusing on the characteristics of the method and the conditions under which it is likely to have the highest value to the organization. Table 1.2 summarizes the conditions under which each strategy is most useful.

Classical Systems Development Life Cycle

Systems development, a process consisting of the two major steps of systems analysis and design, starts when management or sometimes systems development personnel realize that a particular business system needs improvement.

The *systems development life cycle (SDLC) method* (Figure 1.8) is classically thought of as the set of activities that analysts, designers, and users carry out to develop and implement an information system. This section examines each of the six activities that make up the systems development life cycle. In most business situations the activities are all closely related, usually inseparable, and even the order of the steps in these activities may be difficult to determine. Different parts of a project can be in various phases at the same time, with some components undergoing analysis while others are at advanced design stages.

The systems development life cycle method consists of the following activities:

1. Preliminary investigation
2. Determination of system requirements
3. Design of system

TABLE 1.2 Characteristics of Alternative System Development Strategies

Development Strategy	Description	Application Characteristics
System development life cycle method	Includes the activities of preliminary investigation, requirements determination, system design, software development, systems testing, and implementation.	Predictable information systems requirements. Manageable as a project. Requires entry of data into files and databases. High transaction and processing volume. Requires validation of data input. Spans several departments. Long development timetable. Development by project teams.
Structured analysis method	Focuses on what the system or application does rather than on how it does it (i.e., emphasis is logical, not physical). Uses graphic symbols to describe movement and processing of data. Important components include data flow diagrams and data dictionary.	Suitable for all types of applications. Most useful as a supplement to other development methods.
System prototype method	Iterative or evolutionary development where the user is directly involved in the process.	Unique application settings where developers have little information or experience, or where costs or risks of error may be high. Also useful for testing system feasibility, identifying user requirements, evaluating a system design, or examining application usage.

4. Development of software
5. Systems testing
6. Implementation and evaluation

Preliminary Investigation

A request to receive assistance from information systems can be made for many reasons, but in each case someone—a manager, an employee, or a systems specialist—initiates the request.

When that request is made, the first systems activity, the *preliminary investigation,* begins. This activity has three parts: request clarification, feasibility study, and request approval.

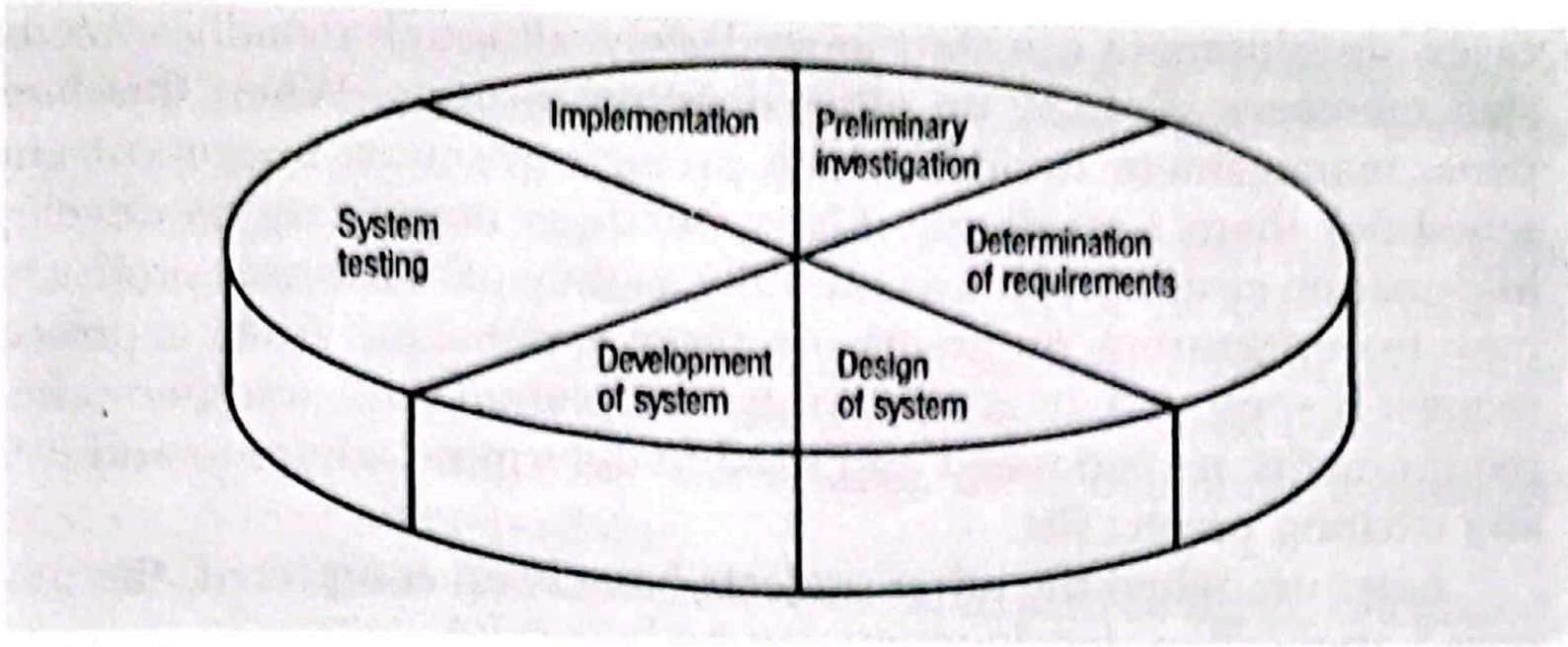

FIGURE 1-8
Activities in classical systems development life cycle.

Request Clarification Many requests from employees and users in organizations are not clearly stated. Therefore, before any systems investigation can be considered, the project request must be examined to determine precisely what the originator wants. A telephone call may suffice if the requester has a clear idea of what is needed but is not sure how to state it. On the other hand, if the requester is asking for help without knowing what is wrong or why there is a problem, problem clarification is much more difficult. In either case, before any further steps can be taken, the project request must be clearly stated.

Feasibility Study An important outcome of the preliminary investigation is the determination that the system requested is feasible. There are three aspects in the *feasibility study* portion of the preliminary investigation:

1. *Technical Feasibility.* Can the work for the project be done with current equipment, existing software technology, and available personnel? If new technology is required, what is the likelihood that it can be developed?
2. *Economic Feasibility.* Are there sufficient benefits in creating the system to make the costs acceptable? Or, are the costs of *not* creating the system so great that the project must be undertaken?
3. *Operational Feasibility.* Will the system be used if it is developed and implemented? Will there be resistance from users that will undermine the possible application benefits?

The feasibility study is carried out by a small group of people (sometimes even one or two) who are familiar with information systems techniques, understand the part of the business or organization that will be involved or affected by the project, and are skilled in the systems analysis and design process. People typically responsible for feasibility assessments are experienced analysts or managers.

Request Approval Not all requested projects are desirable or feasible. Some organizations receive so many project requests from employees that only a few of them can be pursued. However, those projects that are both feasible and desirable should be put into a schedule. In some

cases, development can start immediately, although usually systems staff members are busy on other ongoing projects. When this happens, management decides which projects are most important and schedules them accordingly. Many business organizations develop information systems plans as carefully as they plan for new products, new manufacturing programs, or plant expansion. After a project request is approved, its cost, priority, completion time, and personnel requirements are estimated and used to determine where to add it to any existing project list.

Later on, when the other projects have been completed, the proposed application development can be launched.

Determination of System Requirements

At the heart of systems analysis is a detailed understanding of all important facets of the business area under investigation. (For this reason, the process of acquiring this information is often termed the *detailed investigation*.) Analysts, working closely with employees and managers, must study the business process to answer these key questions:

1. What is being done?
2. How is it being done?
3. How frequently does it occur?
4. How great is the volume of transactions or decisions?
5. How well is the task being performed?
6. Does a problem exist?
7. If a problem exists, how serious is it?
8. If a problem exists, what is the underlying cause?

To answer these questions, systems analysts talk to a variety of persons to gather details about the business process and their opinions of why things happen as they do and their ideas for changing the process. Questionnaires are used to collect this information from large groups of people who cannot be interviewed individually. Detailed investigations also require the study of manuals and reports, actual observation of work activities, and, sometimes, collection of samples of forms and documents to fully understand the process.

As the details are gathered, the analysts study the requirements data to identify features the new system should have, including both the information the system should produce and operational features such as processing controls, response times, and input and output methods.

Design of System

The design of an information system produces the details that state how a system will meet the requirements identified during systems

analysis. Systems specialists often refer to this stage as *logical design*, in contrast to the process of developing program software, which is referred to as *physical design*.

Systems analysts begin the design process by identifying reports and other outputs the system will produce. Then the specific data on each are pinpointed. Usually, designers sketch the form or display as they expect it to appear when the system is complete. This may be done on paper or on a computer display, using one of the automated system design tools available.

The systems design also describes the data to be input, calculated, or stored. Individual data items and calculation procedures are written in detail. Designers select file structures and storage devices, such as magnetic disk, magnetic tape, or even paper files. The procedures they write tell how to process the data and produce the output.

The documents containing the design specifications portray the design in many different ways—charts, tables, and special symbols. The detailed design information is passed on to the programming staff so that software development can begin.

Designers are responsible for providing programmers with complete and clearly outlined software specifications. As programming starts, designers are available to answer questions, clarify fuzzy areas, and handle problems that confront the programmers when using the design specifications.

Development of Software

Software developers may install (or modify and then install) purchased software or they may write new, custom-designed programs. The choice depends on the cost of each option, the time available to write software, and the availability of programmers. Typically, in larger organizations computer programmers (or combination analyst-programmers) are part of the permanent professional staff, as we learned in the vignette at the beginning of the chapter. In smaller firms, without programmers, outside programming services may be retained on a contractual basis.

Programmers are also responsible for documenting the program, providing an explanation of how and why certain procedures are coded in specific ways. Documentation is essential to test the program and carry on maintenance once the application has been installed.

Systems Testing

During systems testing, the system is used experimentally to ensure that the software does not fail, i.e., that it will run according to its specifications and in the way users expect. Special test data are input for processing, and the results examined. A limited number of users may be allowed to use the system so analysts can see whether they try to use it in unforeseen ways. It is preferable to discover any

surprises before the organization implements the system and depends on it.

In many organizations, testing is performed by persons other than those who wrote the original programs to ensure more complete and unbiased testing and more reliable software.

IMPLEMENTATION AND EVALUATION

Implementation is the process of having systems personnel check out and put new equipment into use, train users, install the new application, and construct any files of data needed to use it.

Depending on the size of the organization that will be involved in using the application and the risk associated with its use, systems developers may choose to pilot (test) the operation in only one area of the firm, say in one department or with only one or two persons. Sometimes they will run the old and new systems together to compare the results. In still other situations, developers will stop using the old system one day and begin using the new one the next. As we will see, each implementation strategy has its merits, depending on the business situation in which it is considered. Regardless of the implementation strategy used, developers strive to ensure that the system's initial use is trouble-free.

Once installed, applications are often used for many years. However, both the organization and the users will change, and the environment will be different over weeks and months. Therefore, the application will undoubtedly have to be maintained; modifications and changes will be made to the software, files, or procedures to meet emerging user requirements. Since organization systems and the business environment undergo continual change, the information systems should keep pace. In this sense, implementation is an ongoing process.

Evaluation of the system is performed to identify its strengths and weaknesses. The actual evaluation can occur along any of the following dimensions:

- *Operational Evaluation*
  Assessment of the manner in which the system functions, including ease of use, response time, suitability of information formats, overall reliability, and level of utilization.
- *Organizational Impact*
  Identification and measurement of benefits to the organization in such areas as financial concerns (cost, revenue, and profit), operational efficiency, and competitive impact. Includes impact on internal and external information flows.
- *User Manager Assessment*
  Evaluation of the attitudes of senior and user managers within the organization, as well as end-users.

McDonald's: A System That Spells Success

When we think of the 1950s, we think of rock n' roll, ducktail haircuts, fins on automobiles, Elvis Presley, and lots more. But this colorful era was also a time of postwar economic boom. The country was growing, its population was producing, and its people were busy—so busy they often needed to eat on the run. Against the colorful and dynamic backdrop of the fifties arose the golden arches of McDonald's. Now, more than 30 years later, more people eat at McDonald's every day than live on the entire subcontinent of Australia.

The secret of this extraordinary success story lies in the conception and utilization of a consistent operating system. Ray Kroc, founder of McDonald's, was a systems analyst supreme. He worked out and perfected strategies that worked because they met both the wishes of consumers and the needs of operators to serve the desires of their customers.

Under Kroc's direction, countless consumers were quizzed about what they wanted from a restaurant. Cooking procedures, packaging, menu content, seating, and lighting were all studied and restudied to determine what worked and appealed best. Kroc was always directly involved in these studies as well as in operations. He washed windows, warmed hamburger buns, and greeted customers, knowing that second-hand information was no substitute for first-hand experience.

Using the data gathered through his continuing investigation, Kroc learned precisely what the American hamburger consumer desired and then redesigned the entire hamburger-making industry. Methods were selected to ensure an end product of consistently high quality. Kroc realized the importance of items that other restaurants overlooked. Speed of service became a high priority. Clean rest rooms were made a trademark of the chain.

Capitalizing on his earlier successes, Kroc continued to augment his data by more study and investigation, and designed the company's principle of duplication: When a strategy, procedure, or operating method is successful, duplicate it in other locations.

Today, that operating system offers customers consistency and uniformity on a worldwide basis. The tried and proven methods are instilled in managers and workers at a massive training facility—Hamburger University—in Chicago. Although Ray Kroc died in 1984 at the age of 81, his ideas still live on. He can still be seen and heard at Hamburger University on video, advising his disciples that "Luck is a dividend of sweat. The more you sweat, the luckier you get."

Because of its ubiquitous and successful system, McDonald's puts people at ease in unfamiliar places. Whether on board a military ship in a rolling sea or just off the Champs-Elysées in Paris, the golden arches are a friendly sign, a symbol of a place where one can feel comfortable, almost at home, all because the McDonald's system is much the same around the globe.

- *Development Performance*
  Evaluation of the development process in accordance with such yardsticks as overall development time and effort, conformance to budgets and standards, and other project management criteria. Includes assessment of development methods and tools.

Unfortunately, system evaluation does not always receive the attention it merits. Where properly managed, however, it provides a great deal of information that can improve the effectiveness of subsequent application efforts.

Structured Analysis Development Method

Many information systems specialists acknowledge the difficulty of fully understanding large, complex systems. The structured analysis development method is aimed at overcoming that difficulty through (1) partitioning the system into components and (2) constructing a model of the system. It consists of elements of both analysis and design.

What Is Structured Analysis?

Structured analysis focuses on specifying what the system or application is required to do. It does not state how the requirements should be accomplished or how the application should be implemented. Rather, it allows individuals to see logical elements (what the system should do) apart from the physical components it uses (computers, terminals, storage systems, etc.). Later a physical design can be developed that will be effective for the situation in which it is to be used.

Elements of Structured Analysis

Essential elements of structured analysis include graphic symbols, data flow diagrams, and a centralized data dictionary.

Graphic Description One way to describe a system is to prepare a narrative outlining its features, identifying the function it serves, describing how it interacts with other elements, etc. However, describing a large system in this manner would be most tedious and error prone—it is easy to omit a detail or to describe it in a manner that others may not understand.

Instead of words, structured analysis uses symbols, or icons, to create a graphic model of the system. Graphic models show details of the system without introducing manual or computer processes, tape or disk files, or program and operating procedures. If the right symbols and notation are selected, virtually anyone can follow the way components in the system fit together.

As Figure 1.9 indicates, the icons identify the basic elements of processes, data flows, data stores, and data sources and destinations.