6
System Design and Engineering in Health Care

GIO WIEDERHOLD AND EDWARD H. SHORTLIFFE

After reading this chapter, you should know the answers to these questions:

- What key functions do medical computer systems perform?
- Why is communication between medical personnel and computing personnel crucial to the successful design and implementation of a health information system?
- What are the trade-offs between purchasing a turnkey system and developing a custom-designed system?
- What resources are available remotely for medical computer systems?
- What design features most heavily affect a system's acceptance by health professionals?
- Why do systems in health care, once implemented and installed successfully, have a long lifetime?

6.1 How Can a Computer System Help in Health Care?

In Chapter 5, we introduced basic concepts related to computer and communications hardware and software. In this chapter, we show how information systems created from these components can be used by health professionals to support health care delivery. We describe the basic functions performed by health information systems and discuss important considerations in system design, implementation, and evaluation. You should keep these concepts in mind as you read about the various medical computing applications in the chapters that follow. Think about how each system meets (or fails to meet) the needs of its users and about the practical reasons why certain systems have been accepted for routine use in patient care whereas other systems have failed to make the transition from the research environment to the real world.

At a minimum, a system's success depends on the selection of adequate hardware and sufficient data-storage, and data-transmission capabilities. More crucial is the software, which defines how data are obtained, organized, and processed to yield information. The technical issues related to specific hardware and software choices are beyond the scope of this book. Instead, we provide a general introduction to practical issues in the design and implementation of systems. In particular, we stress the importance of designing systems that not only meet users' requirements for information but also fit smoothly into users' everyday routines. There are many types of users of a health care information system, and often it is necessary to consider each, one at a time.

There are *health care professionals*, for whom the quality of the results is paramount, but who are invariably pressed for time. There are *administrators*, who have to make personnel and financial decisions that are crucial to institutional well-being. There are *clerical personnel*, who enter and retrieve much of the data. Some systems also provide for direct interaction by *patients*. In addition, there are *operational personnel* who maintain the system and ensure its reliability. Initially there are professional system *designers*, *implementers*, and *integrators*, but their numbers and availability decrease as the systems move into routine operation. Before the system is turned over to users, there must be adequate documentation and training. For instance, clerks require clear procedures for their interaction with the system so that errors are minimized. A central theme of this chapter is the importance of communication between health care and computing professionals in defining problems and developing solutions that can be implemented within an institution. With this perspective, we explore the factors that create a need for automation and discuss important considerations in the design, development, and evaluation of health information systems.

A problem in moving systems from their originators, the computing experts, to health care personnel is that their education and experience has too often emphasized different scientific principles. The formality of computing is evidenced in rules that can be applied to many instances, and hence provide consistency and efficiency. Practice in health care focuses on individual instances, one at a time. Some of those will require deep and unique considerations. Much learning in health care stems from examples, and flexibility is expected when a new case differs in some crucial ways. For a computing expert, adapting to a very flexible approach can lead to an explosion of rules, and impossibly complex software. One role of this book is to help in mutual inderstanding of the isues on both sides, leading to an appreciation of the value of obtaining training and expertise at the intersection of biomedicine and computing.

6.1.1 What Is a System?

Until now, we have referred informally to *health information systems* and *computer systems*. What do we mean when we refer to a *system*? In the most general sense, a **system** is an organized set of procedures for accomplishing a task. It is described in terms of (1) the problem to be solved; (2) the data and knowledge required to address the problem; and (3) the internal process for transforming the available **input** into the desired **output** (Figure 6.1). When we talk about systems in this book, we usually mean *computer-based* (or just *computer*) systems. A **computer system** combines both manual and automated processes; people and machines work in concert to manage and use information. A computer system has these components:

- **Hardware**: The physical equipment, including processing units (e.g., the central processing unit (CPU)), data-storage devices, comunication equipment, terminals, and printers
- **Software**: The computer programs that direct the hardware to carry out the automated processes—i.e., to respond to user requests and schedules, to process input data, to store some data for long periods, and to communicate informative results to the users; at times the software will prompt the users to perform manual processes

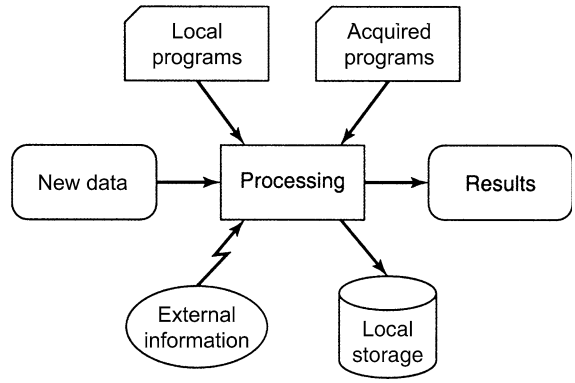


Figure 6.1. A computer system applies locally defined and general procedures to produce results from new input data, from stored data, and from information obtained from remote external sources.

- **Customers**: The users who interact with the software and hardware of the system, issue requests, and use the results or forward them to others; there will be other users who are concerned with providing input, system operations, backup, and maintenance

The role of a computer is, broadly speaking, the conversion of data into information. Every piece of data must be supplied by a person, by another computer system, or by data collection equipment, as seen in patient monitoring (see Chapter 17). Information that is output is delivered to health care professionals or becomes input to another computer system. In other words, a medical computer system is a module within the overall health care delivery system.

The overall health care system not only determines the need for the computer system (e.g., which data must be processed and which reports must be generated) but also the requirements for the system's operation (e.g., the degree of reliability and responsiveness to requests for information). Acquisition and operation of a computer system has implications for the organization of an institution. Who controls the information? Who is responsible for the accuracy of the data? How will the system be financed?

The installation of a computer system has sociological consequences as well. The introduction of a new system alters the work routines of health care workers. Furthermore, it may affect the traditional roles of health care workers and the existing relationships among groups of individuals—e.g., between physicians and nurses, between nurses and patients, and between physicians and patients. Important ethical and legal questions that arise include the confidentiality of patient information, the appropriate role of computers in patient care (especially in medical decision making), and the responsibility of developers and users for ensuring the correct operation of the system (see Chapter 10). Although the technical challenges in system development must be met, organizational factors are crucial determinants of the success of a computer system within the institution. These factors can differ greatly among institutions and can make the transfer of a well-functioning system to another site difficult.

6.1.2 Functions of a Computer System

Computers have been used in every aspect of health care delivery, from the simple processing of business data, to the collection and interpretation of physiological data, to the education of physicians and nurses. Each chapter in Unit II of this book describes an important area for the application of computers in biomedicine. The unique characteristics of each problem area create special requirements for system builders to address. The motivation for investing in these applications, however, is the computer's ability to help health professionals in some aspect of information management. We identify eight topics that define the range of basic functions that may be provided by medical computer systems:

1. Data acquisition and presentation
2. Record keeping and access
3. Communication and integration of information
4. Surveillance
5. Information storage and retrieval
6. Data analysis
7. Decision support
8. Education

These functions are detailed in the discussions of each of the types of health care and biomedical applications addressed in Chapters 12 through 22. Any system will assist its users with several of those functions. In hospital information systems (Chapter 13) most of these functions will be performed, but typically within multiple departments by a variety of computing systems. Those systems must, in turn, communicate with each other (Webster, 1988). Although decision support is a primary function for only two categories of applications, essentially all uses of computers in medicine will support decision making by the variety of staff that uses them.

Data Acquisition

The amount of data needed to describe the state of even a single person is huge. Health professionals require assistance with data acquisition to deal with the data that must be collected and processed.. One of the first uses of computers in a medical setting was the automatic analysis of specimens of blood and other body fluids by instruments that measure chemical concentrations or that count cells and organisms. These systems then present the results of such analyses in a clear form. Signalling when results are outside the expected range alerts the health care staff. Computer-based patient-monitoring systems that collect physiological data directly from patients were another early application of computing technology (see Chapter 17). Such systems ensure that vital signs, electrocardiograms (ECGs), and other indicators of patient status are measured frequently and consistently. More recently, researchers have developed medical imaging applications as described in Chapters 9 and 18, including computed tomography (CT), magnetic resonance imaging (MRI), and digital subtraction angiography. The calculations

for these computationally intensive applications cannot be performed manually; the computers collect and manipulate millions of individual observations.

Early computer-based medical instruments and measurement devices that perform data acquisition provided their results only to human beings. Today, most instruments supply data directly into the patient record, although the interfaces are still awkward and poorly standardized (see Chapter 7). Computer-based systems that acquire information, such as one's health history, from patients are also data-acquisition systems; they free health professionals from the need to collect and enter routine demographic and history information.

Record Keeping

Given the data-intensive nature of health care delivery, it is no surprise that collecting and keeping records is a primary function of many medical computer systems. Computers are well suited to performing tedious and repetitive data-processing tasks, such as collecting and tabulating data, combining related data, and formatting and producing reports. They are particularly useful for processing large volumes of data. Automated billing is a natural application of computers in health care settings, and was typically the first component installed when a hospital, clinic, or private practice decided to use computer technology. Unfortunately, the level of documentation required for performing billing is inadequote for treating patients.

Individual departments within a hospital also have their own computer systems and maintain their own records. For instance, clinical laboratories use computer-based information systems to keep track of orders and specimens and to report test results; most pharmacy and radiology departments use computers to perform analogous functions. Their systems may connect to outside services (e.g., pharmacy systems are typically connected to one or more drug distributors) so that ordering and delivery are rapid and local inventories can be kept small. By automating processing in areas such as these, health care facilities are able to speed up services, reduce direct labor costs, and minimize the number of errors.

Computer systems acquired by hospital departments are often obtained from specialized vendors. Such vendors contribute their experience in serving clinical laboratories, pharmacy operations, or other areas. They will supply their customers with updates when capabilities improve or regulations change. Sometimes the services may actually be operated remotely at a vendor's site. Unfortunately, this diversity makes it difficult to integrate the information from the disparate systems into a coherent whole, a problem addressed in Chapter 13.

Communication and Integration

In hospitals and other large-scale health care institutions, myriad data are collected by multiple health professionals who work in a variety of settings; each patient receives care from a host of providers—nurses, physicians, technicians, pharmacists, and so on. Communication among the members of the team is essential for effective health care delivery. Data must be available to decision makers when and where they are needed,

independent of when and where they were obtained. Computers help by storing, transmitting, sharing, and displaying those data. As we describe in Chapters 2 and 12, the patient record is the primary vehicle for communication of clinical information. The limitation of the traditional paper-based patient record is the concentration of information in a single location, which prohibits simultaneous entry and access by multiple people. Hospital information systems (HISs) (see Chapter 13) and EHR systems (Chapter 12) allow distribution of many activities, such as admission, appointment, and resource scheduling; review of laboratory test results; and inspection of patient records to the appropriate sites.

Information necessary for specific decision-making tasks is rarely available within a single computer system. Clinical systems are installed and updated when needed, available, and affordable. Furthermore, in many institutions, inpatient, outpatient, and financial activities are supported by separate organizational units. Patient treatment decisions require inpatient and outpatient information. Hospital administrators must integrate clinical and financial information to analyze costs and to evaluate the efficiency of health care delivery. Similarly, clinicians may need to review data collected at other health care institutions, or they may wish to consult published biomedical information. Communication networks that permit sharing of information among independent computers and geographically distributed sites are now widely available. Actual integration of the information they contain requires additional software, adherence to standards, and operational staff to keep it all working as technology and systems evolve.

Surveillance

Timely reactions to data are crucial for quality in health care, especially when a patient has unexpected problems. Data overload, created by the ubiquity of information technology, is as detrimental to good decision making as is data insufficiency. Data indicating a need for action may be available but are easily overlooked by overloaded health professionals. Surveillance and monitoring systems can help people cope with all the data relevant to patient management by calling attention to significant events or situations, for example, by reminding doctors of the need to order screening tests and other preventive measures (see Chapters 12 and 20) or by warning them when a dangerous event or constellation of events has occurred.

Laboratory systems routinely identify and flag abnormal test results. Similarly, when patient-monitoring systems in intensive care units detect abnormalities in patient status, they sound alarms to alert nurses and physicians to potentially dangerous changes. A pharmacy system that maintains computer-based drug-profile records for patients can screen incoming drug orders and warn physicians who order a drug that interacts with another drug that the patient is receiving or a drug to which the patient has a known allergy or sensitivity. By correlating data from multiple sources, an integrated clinical information system can monitor for complex events, such as interactions among patient diagnosis, drug regimen, and physiological status (indicated by laboratory test results). For instance, a change in cholesterol level can be due to prednisone given to an arthritic patient and may not indicate a dietary problem.

Surveillance also extends beyond the health care setting. Appearances of new infectious diseases, unexpected reactions to new medications, and environmental effects should be monitored. Thus the issue of data integration has a national or global scope (see the discussion of the National Health Information Infrastructure in Chapter 1 and Chapter 15 that deals with public health informatics).

Information Storage and Retrieval

Storage and retrieval of information is essential to all computer systems. Storage enables sharing of information with people who are not available at the same time. Storage must be well organized and indexed so that information recorded in an EHR system can be easily retrieved. Here the variety of users must be considered. Getting cogent recent information about a patient entering the office differs from the needs that a researcher will have in accessing the same data. The query interfaces provided by EHR and clinical research systems assist researchers in retrieving pertinent records from the huge volume of patient information. As we discuss in Chapter 19, bibliographic retrieval systems are an essential component of health information services.

Data Analysis

Raw data as acquired by computer systems are detailed and voluminous. Data analysis systems must aid decision makers by reducing and presenting the intrinsic information in a clear and understandable form. Presentations should use graphs to facilitate trend analysis and compute secondary parameters (means, standard deviations, rates of change, etc.) to help spot abnormalities. Clinical research systems have modules for performing powerful statistical analyses over large sets of patient data. The researcher, however, should have insight into the methods being used. For clinicians, graphics are essential for interpretation of data and results.

Decision Support

In the end, all the functions described here support decision making by health professionals. The distinction between decision-support systems and systems that monitor events and issue alerts is not clear-cut; the two differ primarily in the degree to which they interpret data and recommend patient-specific action. Perhaps the best-known examples of decision-support systems are the clinical consultation systems or event-monitoring systems that use population statistics or encode expert knowledge to assist physicians in diagnosis and treatment planning (see Chapter 20). Similarly, some nursing information systems help nurses to evaluate the needs of individual patients and thus assist their users in allocating nursing resources. In Chapter 20, we discuss computer-based systems that use algorithmic, statistical, or artificial-intelligence (AI) techniques to provide advice about patient care.

Education

Rapid growth in biomedical knowledge and in the complexity of therapy management has produced an environment in which students cannot learn all they need to know during training—they must learn how to learn and must make a lifelong educational commitment. Today, physicians and nurses have available a broad selection of computer programs designed to help them to acquire and maintain the knowledge and skills they need to care for their patients. The simplest programs are of the drill-and-practice variety; more sophisticated programs can help students to learn complex problem-solving skills, such as diagnosis and therapy management (see Chapter 21). Computer-aided instruction provides a valuable means by which health professionals can gain experience and learn from mistakes without endangering actual patients. Clinical decision-support systems and other systems that can explain their recommendations also perform an educational function. In the context of real patient cases, they can suggest actions and explain the reasons for those actions.

6.1.3 Identifying and Analyzing the Need for a Computer System

The first step in the introduction of computers into health care settings is to identify a clinical, administrative, or research need—an inadequacy or inefficiency in the delivery of health care. The decision to acquire or replace a computer system may be motivated by a desire to improve the *quality* of care, to lower the *cost* of care, to improve *access* to care, or to collect the information needed to document and evaluate the health care delivery process itself. A new computer-based system may correct defects in the old system, for example, by reading bar codes on containers to reduce the level of drug-administration errors. Other computer systems can provide functions not possible with a manual system—e.g., allow integrated access to patient records. In some cases, a computer system simply duplicates the capabilities of the prior system but at lower maintenance cost.

The sophistication of medical computer systems has increased substantially since their inception, when computers were first applied to the problems of health care delivery. The developers of new systems make progress by building on lessons learned from past operations, emulating the successes, and trying to avoid the mistakes of earlier systems. As the discipline has matured, researchers and users have gained a better understanding of the types of problems computer systems can solve and the requirements for system success.

Clearly, computers facilitate many aspects of health care delivery. Operating a computer system, however, is not a panacea; an information system cannot aid in decision making, for example, if critical information is not available or if health professionals do not know how to apply the information once they have it. Similarly, a computer will not transform a poorly organized process into one that operates smoothly—automating a defective approach makes matters worse, not better.

Communication with other system components is today viewed as crucial. Performing local tasks in a unit of the hospital better but not communicating the results with other units just creates more work and delays. A careful workflow analysis before attempting computer-based improvements allows system developers and health care personnel to clarify the requirements for change and may identify correctable deficiencies in current systems (Leymann and Roller, 2000).

Ideally, we first recognize a *need*, and then search for techniques to address it. At times, this logical sequence has been inverted; the development of new hardware or computing methodologies may motivate system developers and marketeers to apply innovative technology in a medical context. Development driven by technology often fails to deal with clinical realities. The adoption of any new system requires users to learn and to adjust to a new routine, and, given the time constraints under which health professionals operate, users may be unwilling to discard a working system unless they perceive a clear reason to change.

Once health professionals have recognized a need for a computer system, the next step is to identify the function or combination of functions that fulfills that need. There usually are many possible solutions to a broadly defined problem. A precise definition of the problem narrows the range of alternative solutions. Is the problem one of access to data? Do health professionals have the data they need to make informed decisions? Is the problem an inability to analyze and interpret data? As we explained in the previous section, computer systems perform a variety of functions, ranging from simply displaying relevant information to aiding actively in complex decision making.

The natural temptation is to minimize this important first step of problem definition and to move directly to the solution phase. This approach is dangerous, however, and it may result in the development of an unacceptable system. Consider, for example, a situation in which physicians desire improved access to patient data. Health care personnel may seek assistance from technologists to implement a specific technical solution to the perceived problem. They may request that each patient's complete medical history be stored in a computer. When that is achieved, however, they may find that the relevant information is hidden among the many irrelevant data and is more tedious to access than before. If the system developers had analyzed the problem carefully, they might have realized that the raw medical data simply were too voluminous to be informative. A more appropriate solution would include integration, filtering and ranking so that only easy-to-read summaries with essential information are displayed.

The development of information systems requires a substantial commitment in terms of labor, money, and time. Once health professionals have clearly defined the need for a system, the question of value inevitably arises. Scarce resources devoted to this project are unavailable for other potential projects. The administrator of a health care institution who works within a fixed budget must decide whether to invest in a computer system or to spend the money in other ways—e.g., an institutional decision maker may prefer to purchase new laboratory equipment or to expand the neonatal intensive care unit.

To assess the value of a health information system relative to competing needs, the administrator must estimate the costs and benefits attributable to the system. Some

benefits are relatively easy to quantify. If admission clerks can process each admission twice as fast using the new system as they could using the old one, an institution needs fewer clerks to perform the same amount of work—a measurable savings in labor costs. Many benefits, however, are less easily quantified. For example, how can we quantify the benefits due to reduced patient mortality and morbidity, increased patient satisfaction, or reduced stress and fatigue among the staff? In Chapter 11 we introduce cost–benefit and cost-effectiveness analyses—two methodologies that can help decision makers to assess the worth of a computer system relative to alternative investments.

6.2 Understanding Health Information Systems

Whether they aim to produce a comprehensive information system for a 500-bed hospital, a patient record system for a small clinic, or a simple billing program for a physician in private practice, system developers should follow the same basic process. In the initial phase of system development, the primary task is to define the problem. The goal is to produce a clear and detailed statement of the system's objectives—i.e., what the system will do and what conditions it must meet if it is to be accepted by its users. The systems analysts also must establish the relative priorities of multiple, sometimes conflicting, goals—e.g., low cost, high efficiency, easy maintenance, and high reliability.

We frame the discussion of this section in terms of institutional system planning and development. Many of the same issues, however, apply to the development of smaller systems as well, albeit on a correspondingly smaller scale of complexity. Since a variety of systems exist in this world, the first step is to assess the use of computers in similar settings. Creating a detailed list of candidate functions is the first task (Webster, 1988). These functions can then by ranked by local importance.

Ideally, a commercial system exists that provides all the essential functions. If there is none, priorities may be reviewed, and systems may be adapted or augmented. Excessive adaptations diminish the benefits of a commercial choice, since now maintenance responsibilities devolve onto the hospital or clinic. Sometimes it is necessary to design a system to handle new requirements and novel functions. After acquisition or development of a system, the next step is to establish the system within the organization. Major activities at this stage include training users, installing and testing the system, and, finally, evaluating and maintaining the operational system on an ongoing basis.

Maintenance is a demanding task. It invoves correcting errors, ongoing adaptions to growth, new hardware, new communication capabilities, new standards, and new professional regulations. Long-lived systems also require ongoing perfection of interfaces, performance, and linkages to other sources (Pigoski, 1997). Over the lifetime of a computer system these tasks exceed by a factor of two to five the original acquisition costs. Many software suppliers will provide most maintenance services for 15 to 30 percent of the purchase price annually. If those services are performed well, they are more than worth the price.

6.2.1 An Illustrative Case Study

In addition to identifying functional requirements, a **requirements analysis** must be sensitive to the varying needs and probable concerns of the system's intended users. These human aspects of computer system design often have been overlooked, and the results can be devastating. Consider, for example, the following hypothetical case, which embodies many of the issues that are the subject of this chapter.[1]

A major teaching hospital purchased and installed a large computer system that assists physicians with ordering drugs and laboratory tests, the clinical laboratories with reporting laboratory test results, head nurses with creating nursing schedules, and the admissions staff with monitoring hospital occupancy. Personnel access the system using workstations located in each nursing unit. There also are printers associated with each unit so that the computer can generate reports for the patient charts (which continue to be paper-based) and worksheets used by the hospital staff. This information system depends on a large, dedicated computer, which is housed in the hospital complex and is supported by several full-time personnel. It has modules to assist hospital staff with both administrative and clinical duties. The following four modules are the primary subsystems used in patient care.

1. *The pharmacy system*: With this component of the information system, physicians order drugs for their patients; the requests are displayed immediately in the hospital pharmacy. Pharmacists then fill the prescriptions and affix computer-printed labels to each bottle. The drugs are delivered to the ward by a pneumatic-tube system. The computer keeps a record of all drugs administered to each patient and warns physicians about possible drug interactions at the time that new prescriptions are ordered.
2. *The laboratory system*: With this component, physicians order laboratory tests for their patients. The requests are displayed in the clinical laboratory, and worksheets are created to assist laboratory personnel in planning blood-drawing schedules and performing tests. As soon as test results are available, health professionals can display them on the screen of any workstation, and paper summaries are printed on the wards for inclusion in the patients' charts.
3. *The bed-control system*: The admissions office of the hospital, in conjunction with the various ward administrators, uses this component to keep track of the location of patients within the hospital. When patients are transferred to another ward, the computer is notified so that physicians, telephone operators, and other personnel can locate them easily. The system also is used to identify patients whose discharge has been ordered; thus, the system aids the admissions office in planning bed assignments for new patients.
4. *The diagnosis system*: To help physicians reach correct diagnoses for their patients, this component provides a clinical consultation program. Physicians enter their patient's signs and symptoms and can combine them with laboratory test results and X-ray examination results. The system then suggests a list of likely diagnoses.

[1] This case study is adapted from Shortliffe E.H. (1984) Coming to terms with the computer. In Reiser S.J., Anbar M. (Eds.), *The Machine at the Bedside: Strategies for Using Technology in Patient Care* (pp. 235–239). Cambridge, UK: Cambridge University Press. It is used here with permission from Cambridge University Press.

Despite the new capabilities provided by the system, after 3 months of use it received mixed reviews about its effectiveness. Most of the people who raised concerns were involved in patient care. A consulting expert was called in to assess the computer system's strengths and weaknesses. She interviewed members of the hospital staff and noted their responses.

One nurse said: I like the system a lot. I found it hard to get used to at first (I never have been a very good typist), but once I got the hang of it, I found that it simplified much of my work. The worst problem has turned out to be dealing with doctors who don't like the system; when they get annoyed, they tend to take it out on us, even though we're using the system exactly as we've been trained to do. For instance, I can't log onto the computer as a physician to log verbal orders in someone else's name, and that makes some of the doctors furious. The only time I personally get annoyed with the computer is when I need to get some work done and the other nurses are using all the ward workstations. They ought to have a few more machines available.

One medical resident was less than enthusiastic about the new clinical system: I wish they'd rip the darn thing out! It is totally unrealistic in terms of the kinds of things it asks us to do or won't allow us to do. Did the guys who built it have any idea what it is like to practice medicine in a hospital like this? For example, the only way we used to be able to keep our morning ward rounds efficient was to bring the chart rack with us and to write orders at the bedside. With the new system, we have to keep sending someone back to the ward workstation to log orders for a patient. What's worse, they won't let the medical student order drugs, so we have to send an intern. Even the nurses aren't allowed to log orders in our name—something to do with the "legality" of having all orders entered by a licensed physician—but that was never a problem with paper order sheets as long as we eventually countersigned the orders. Some of the nursing staff are doing everything by the book now, and sometimes they seem to be obstructing efficiency rather than aiding it. And the designers were so hung up on patient confidentiality that we have a heck of a time cross-covering patients on other services at night. The computer won't let me write orders on any patient who isn't "known" to be mine, so I have to get the other physicians' passwords from them when they sign out to me at night. And things really fall apart when the machine goes down unexpectedly. Everything grinds to a halt, and we have to save our management plans on paper and transcribe them into the system when it finally comes up. I should add that the system always seems to be about three hours late in figuring out about patient transfers. I'm forever finding that the computer still thinks a patient is on the first floor when I know he's been transferred to the intensive care unit.

In addition, the "diagnosis system" is a joke. Sure, it can generate lists of diseases, but it doesn't really understand what the disease processes are, can't explain why it thinks one disease is more likely than another, and is totally unable to handle patients who have more than one simultaneous disease. I suppose the lists are useful as memory joggers, but I no longer even bother to use that part of the system.

And by the way, I still don't really know what all those options on the screen mean. We had a brief training session when they first installed the system, but now we're left to fend for ourselves. Only a couple of the house staff seem to know how to make the system do what they want reliably. What's the best part of the system? I guess it is the decrease in errors in orders for drugs and lab tests and the improved turnaround time on those orders—but I'm not sure the improvement is worth the hassle. How often do I use the system? As rarely as possible!

A hospital pharmacist said: The system has been a real boon to our pharmacy operation. Not only can we fill new orders promptly because of the improved communication but also the system prints labels for the bottles and has saved us the step of typing them ourselves. Our inventory control also is much improved; the system produces several useful reports that help us to anticipate shortages and to keep track of drugs that are about to expire. The worst thing about the system, from my point of view, is the effect it has had on our interaction with the medical staff. We used to spend some of our time consulting with the ward teams about drug interactions, for example. You know, we'd look up the relevant articles and report back at ward rounds the next day. Now our role as members of the ward teams has been reduced by the system's knowledge about drugs. Currently, a house officer finds out about a potential drug interaction at the moment she is ordering a treatment, and the machine even gives references to support the reported incompatibility.

One member of the hospital's computing staff expressed frustration: Frankly, I think the doctors have been too quick to complain about this system. It has been here for only three months, and we're still discovering problems that will take some time to address. What bothers me is the gut reaction many of them seem to have; they don't even want to give the system a chance. Every hospital is a little different, and it is unrealistic to expect any clinical system to be right for a new institution on the first day. There has to be a breaking-in period. We're trying hard to respond to the complaints we've heard through the grapevine. We hope that the doctors will be pleased when they see that their complaints are being attended to and new features are being introduced.

This example, although hypothetical, does not exaggerate the kinds of reactions that computer systems sometimes have evoked in clinical settings. Developers of real-world systems have encountered similar problems after introducing their systems into real clinical settings. This example was inspired in part by the experience of one of the authors (E.H. Shortliffe) with a pharmacy system that was implemented in several teaching wards of the hospital where he served his medical internship. The initial version of the system failed to account for key aspects of the way in which health professionals practiced medicine. All drug orders had to be entered into the computer, and, because the terminals were located at the nursing stations, physicians could no longer complete their orders at the bedside. They either had to return to the nursing station after seeing each patient or had to enter all the orders after completing patient rounds. Furthermore, physicians personally had to enter the orders. The system did not allow them to countersign orders entered by nurses or medical students. The inflexibility of the system forced physicians to alter their practice patterns. The physicians objected loudly, and the system was subsequently removed. Although the system was later redesigned to remedy the earlier problems, it was never fully reimplemented because of persistent negative bias caused by the failure of the earlier version.

In another instance, the introduction of a new blood bank system at a large institution instigated disputes between physicians and nurses over responsibility for the entry of blood orders. In this case, the system was designed to encourage direct order entry but also allowed nurses to enter orders. Given the option, physicians continued to write

paper orders and relied on nurses for order entry. Nurses, however, balked at performing this task, which they perceived as being the physicians' responsibility. Even seemingly trivial matters can cause problems. At the same institution, for example, objection by surgeons was one factor in the decision to use passwords rather than machine-readable identification cards to control system access; surgeons typically do not carry personal belongings when wearing surgical garb and thus would have been unable to log onto the system (Gardner, 1989).

These cases help to emphasize the importance of responsiveness to detailed needs of the intended *users* and of awareness of different users' varying constraints and motivations when a system is intended to meet both clinical and administrative goals. The key considerations suggested by the scenario include (1) analyzing where the system is to fit into the existing workflow; (2) deciding what to purchase from a commercial system vendor and what to develop internally; (3) designing for the actual customers; (4) involving those customers throughout the development; and (5) planning for subsequent changes. We discuss these topics in detail in Section 6.3.

6.2.2 *Involving Customers During Development*

Although the central focus of this chapter is *computer*-based information systems, it is important to realize that people are the critical component of these systems (see Chapter 4). People identify the need for systems; people select, develop, implement, and evaluate the systems; and, eventually, people operate and maintain the systems. A successful system must take into account both the needs of the intended users and the constraints under which these users function.

Even the most perceptive and empathetic developer cannot anticipate all the needs of all types of customers. Thus, the success of a system depends on interaction between health and technical personnel as well as among the heath care professionals. Effective communication among the participants in designing a system, however, is potentially difficult because these people are likely to have widely varying background, education, experience, and styles of interaction. Appointing a wide variety of personnel to a large design committee is likely to be ineffective as well, because committees are best suited for forging compromise solutions, whereas computer systems can and should serve a precise set of objectives.

A major barrier to communication is attributable to a difference between the health care and more formal scientific paradigms. In Chapter 1, we discussed the ways in which clinical information differs from the information used in the basic sciences (recall the difference between low-level and high-level sciences discussed in Section 1.3), and we examined reasons why biomedical informatics differs from basic computer science. In medical practice, as in other human tasks, we expect that a person who can deal with a certain type of problem can, with little incremental effort, extrapolate to handle similar and related problems. In the formal mathematical sciences, however, the ability to solve problems can depend critically on what appear to be small, but fundamental differences in basic assumptions.

The rigor of the mathematical approach also is reflected in computer systems so that computer systems will never be as flexible as people are. Although it is easy to

imagine that a computer program that deals with one class of problems embodies sufficient concepts to deal with other (seemingly) similar problems, the work required to adapt or extend the program often costs as much as the original development and sometimes much more. Occasionally, adaptation is impossible. For health care professionals the difficulty of altering some aspects of software is hard to appreciate, because these people interact mainly with human beings rather than with seemingly smart machines.

Empathy for the differences in the two approaches to problem solving can minimize certain problems. Health information specialists—people trained in both computer science and health science—can facilitate communication and mediate discussions. They can ease the process of specifying accurately and realistically the need for a system and of designing workable solutions to satisfy those needs. One objective of this book is to provide basic material for people who serve in this intermediary role.

6.3 Developing and Implementing Systems in Health Care

The initial task is to circumscribe what a new system in a health care setting should encompass. Will the new system replace all existing computing capabilities, or will it provide new functions, or will it replace some existing systems? If it replaces an existing system, the current functions being provided can be enumerated and the data requirements specified. For new functions additional data will be needed, and if some older functions can be omitted, the input requirements might actually be less. Sometimes a new system may replace parts of multiple existing processes, making the task of defining its functions and data requirements yet harder. Studying related services at other sites will help. Relying only on vendor presentations is risky. In Section 6.3.2 methods to chart those considerations are presented.

6.3.1 System Acquisition Alternatives

There are vendors willing to sell systems for any task that a health care institution may require. For most tasks some vendor will have a system already designed and ready to be adapted to your specifications. The actual functions and data requirements of a system acquired from a vendor may, however, differ from what was envisaged. The manner in which desired functions are provided will also differ among vendors. Demonstrations by vendors to customers can provide insights, and the feedback obtained may well change the expectations that were specified initially. Some desired features may turn out to be costly to implement and maintain, and compromises are likely.

Some required services can be obtained by contracting for them remotely. Searches for published information are best performed over the Internet (Chapter 19). Some services, such as managing supply inventories, may be jointly supported by in-house staff and external contractors. We address remote operations in Section 6.3.4.

For essential operations, a health care institution must control its own systems. We focus on the software requirements because the selection of hardware is primarily determined by the requirements that software imposes.

Commercial Off-the-Shelf Software

Much software is available off the shelf from commercial companies. The selection is widest for the more general needs, such as financial management—general ledger, accounts payable, and general accounts receivable. For health care institutions the selection is more limited or may require adaptation to the institution. For instance, financial receivables in health care—largely based on mixes of insurance, government, and private payments—have more complexity than standard business systems allow. Several software companies that serve the health care industry provide appropriate software. The major concern becomes ensuring smooth interaction among software packages obtained from different vendors. Following the emerging standards for remote services is helpful.

Once an organization has decided to acquire a new computer system, it faces the choice of buying a commercial system or building a system in-house. The primary trade-off between purchasing a **turnkey system**—a vendor-supplied system that requires only installation and "the turn of a key" to make it operational—and developing a **custom-designed system** is one of compatibility with the conventions in the institution versus expense, delay, and ongoing maintenance. Substantial new systems take years to design and develop. A vendor-supplied system usually is less expensive than is a custom-designed system, because the vendor can spread the costs of development and subsequent maintenance over multiple clients. A compromise is to custom-tailor a vendor system to the needs that are particular to one's institution, but the costs for changes not foreseen by the vendor will be disproportionally high. The maintenance costs for computer systems is on the order of 15 to 30 percent per year, and custom-tailoring greatly increases those costs. If an institution can find a commercial system that approximately meets its needs, it should purchase that system, even it has to change its own methods somewhat.

If no available commercial system for some function is adequate, the institution may choose to build its own system or to make do with the current system; the option of keeping the current system should always be considered as one of the possible alternatives. Building an entire system for all functions in a health care institution is not a viable option today. Before embarking on in-house development of any software system, administrators must assess whether the institution possesses the resources and expertise necessary for long-term success. For example, do in-house staff have the knowledge and experience to manage development and implementation of a new system? Can subsystems and components be obtained so that the local effort is minimized? Can outside consultants and technical staff be hired to assist? Who will maintain the system if it is installed successfully? In Section 6.3.3, we describe the development process, mainly to show its complexity.

Turnkey systems include all the hardware, software, and technical support necessary to operate the system. They should become operational rapidly, and most delays will be due to integration of the system with existing services. Unfortunately, the functions supplied by a turnkey system rarely match an institution's information management needs. The system may not perform all the desired functions, may provide superfluous features, or may require some reorganization and modification of responsibilities and established workflows within the institution. It is also important to consider carefully

the reputation of the vendor and the terms of the contract and to answer questions such as "What is the extent of the support and maintenance?" and "To what extent can the system be parameterized to the institution (e.g., selecting interchange standards to other systems in the institution, handling local billing policies, and dealing with multiple pharmacies)?"

Technology Transfer

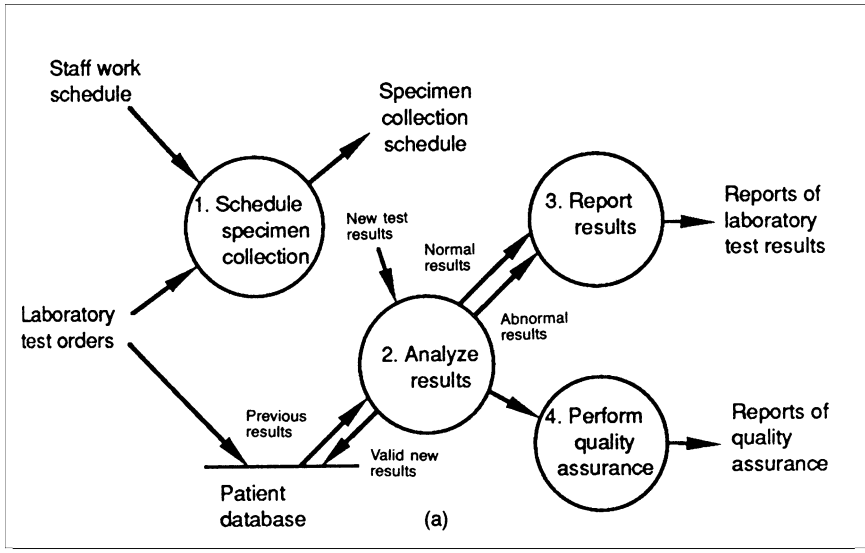
Attractive innovations in health care services are typically demonstrated in a research setting. Subsequent adopters should not underestimate the difficulty of transferring novel technology to a working environment. One rule of thumb we accept is that the work needed to develop an academically successful demonstration of a new computing technology is one-seventh of the work required to transform the demonstration into a practical system. Professional experience is needed to validate that the new system works under all conditions encountered in a clinical practice and that the system can recover from mechanical failures. If new linkages are needed to other systems, yet more work is needed, and if those systems have to be adapted, even slightly, arrangements with their owners and vendors have to be made. Even though changes required for integration may be minor, understanding and validating changes to other systems incur a high cost. During integration, computing personnel must modify the existing system, develop and check interfaces, and retrain the various types of users. The effect is that the time and cost already estimated must be multiplied by a further factor of three or four. The difficulty of software technology transfer has been a significant obstacle to the growth of medical computing.

6.3.2 Specifying Information Processes

In the health care environment, the major portion of computing deals with data rather than complex algorithms. Most data are obtained from patients, laboratories, health care personnel, and insurance providers. The data then are transmitted, stored, transformed, summarized, and analyzed to help health professionals, managers, and patients to plan actions and interventions. Certain data must be archived for legal purposes. To understand the task that a system, or subsystem, performs, it is best to consider the **data flow**.

Data Flow

A graphical representation called the **data-flow diagram** (DFD) provides a succinct way to understand the objectives of a system. It represents the sources of data, the processes for transforming the data, and the points in the system where long-term or short-term data storage is required, and the destinations where reports are generated or where results from queries are presented. The DFD in Figure 6.2a is a model of a simple laboratory information system. It illustrates the flow of laboratory test orders and results, as well as the basic functions of the proposed system: (1) creation of specimen collection schedules; (2) analysis of results; (3) reporting of test results; and (4) performance



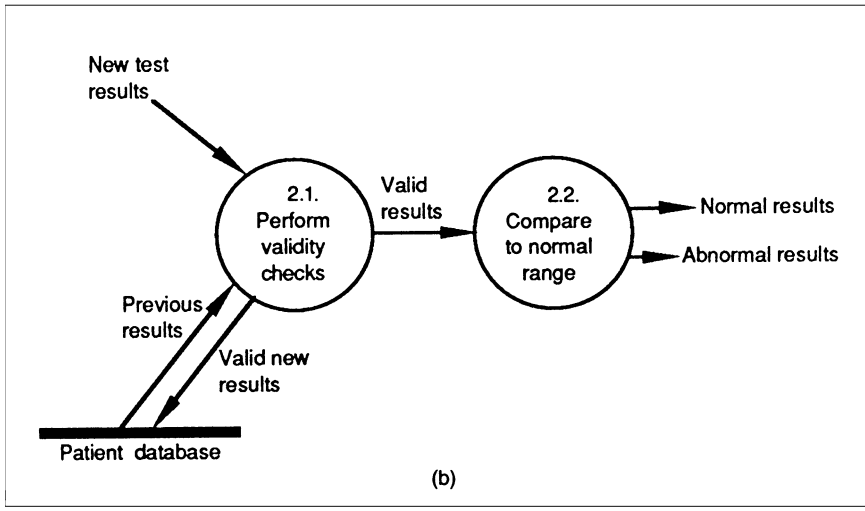


Figure 6.2. A data-flow diagram (DFD) that graphically represents the processes and data flows within a laboratory information system. (a) *Bubbles* depict processes (or functions), *vectors* depict data flows, and *straight lines* depict databases. (b) Often, DFDs are layered to show greater detail within higher-level processes. This second-level DFD decomposes the process of analyzing test results into two lower-level processes. Note that the net inputs and outputs of this DFD match the inputs and outputs of the higher-level process shown in part (a).

of quality-assurance activities. The designer can describe each higher-level process in the DFD by creating a more detailed DFD (Figure 6.2b).

The initial DFD is often based on an analysis of current workflow activities and processes. That approach helps people to identify outputs that are not obvious or that are needed only for rare cases—e.g., for investigation of infrequent infections. It is easy to overlook information needs that are informally obtained. Many informal mechanisms for communication and storage disappear when computers are used to collect and store data. Health professionals trade much valuable information in hallways and at nursing stations; e.g., they will discuss a baffling case to help develop new insights and approaches. Some crucial information never enters the formal record but is noted on scraps of paper or stored in people's memories. System designers must take care to allow support of such mechanisms, perhaps by having unconstrained note fields with the records. Once all needed outputs are indicated on the DFD, analysts can trace back to ensure that all required inputs are available in the flow. The absence of even minor functions can produce errors in health care delivery and will be perceived as a failure of the system, leading to resistance by the customers. Recall from the hypothetical case, for example, the friction between physicians and nurses that arose when the new system failed to allow for verbal orders. Often, technologists discount resistance as an unwillingness to keep up with progress, but they should interpret it as a signal that something is wrong with the new system. When such requirements arise later, they should be accepted as part of routine maintenance, since not all future requirements can be foreseen.

Using the clear graphical representation of the DFD helps the intended users to recognize aberrations and to provide feedback to the system designers. A DFD in which everything seems to connect to everything else is not helpful for analysis or implementation. Complex data flows should first be simplified so that the implementation is straightforward. If the result is still complex, a good layout can help. Major data flows should go left to right, minor flows should be shown in lighter colors, and special cases can use overlays. The final DFD can differ greatly from the initial one.

Data Storage

Most of the data and information that is documented in a DFD is stored in databases. Database software is large and crucial to reliable operations. Alternate database management systems will differ in scale and cost, and many vendors support several choices. An institution may already have made commitments to some database system vendor. It is best to limit the variety of such complex software and to share its maintenance costs. It is hence wise to ensure that the databases that come with commercial software are open—i.e., documented, set up, and maintained in a stable manner so that foreign applications can extract and contribute data to them.

Most business software today depends on relational databases. Such databases provide a means for exchanging information among software components that are otherwise independent. A vendor's schema must be adaptable, so that it can specify all the data that the software will exchange. Once health care databases are widely accessible, serious issues about access to, and release of, private medical data arise, as discussed in Chapter 10.

Further software components may need to be written to interact with an accessible database, but the database schema constrains the design of the programs while simplifying the implementation choices. The standards associated with the relational approach also simplify exchange of data over communication links, so redundant effort in data entry and result reporting can be avoided.

For health records systems, off-the-shelf software has not been easy to adapt. Clinical patient information is too complex to be easily and efficiently mapped into relational databases, so the benefits of open databases and component-based transaction services are just being realized in the core of medical applications. This means that many health care institutions have to deal with databases for medical functions that are not easy to access. For off-the-shelf systems even minor changes may require contractual negotiations with a vendor.

6.3.3 Building a New Software System

If the institution is prepared to build a computer system to meet some specific needs, system developers can employ a variety of software-engineering tools to organize and manage the development process. Although major efforts are rare in today's health care settings, it is useful to understand this process for interacting with vendors or enthusiastic innovators. Software system development tools include formal techniques for system analysis, methodologies of **structured programming**, and testing methods. They are supplemented by methods for managing the project and metrics for assessing the performance of the product and of the software development process itself. The field of **software engineering** had its beginnings in the late 1960s and early 1970s. As the systems being developed grew more complex, teams of programmers, analysts, and managers replaced individual programmers who had worked in isolation. Reuse of existing modules became important (Boehm, 1999). Many software prototypes are still developed in isolation but must be moved to a comprehensive setting before they can be marketed and distributed.

In the 1980s, the cost of hardware (which previously had dominated the total cost of a system) declined rapidly. The largest system cost now is software maintenance, an aspect rarely considered when building prototypes. More recently, standard software components have become available so that new approaches to the creation of systems are opening up. There is no single right way to implement computer systems in health care.

Software Lifecycle

Figure 6.3 depicts the classic **waterfall model** of any engineering development process from requirements analysis through specification, design, implementation, testing, and maintenance. Specialists may be engaged for each of the phases. The tasks of one phase should be completed and documented before the next phase is tackled so that the specialists for the next phase are presented with a complete specification. As Figure 6.3 suggests, problems found in a later phase require feedback to earlier phases and can cause expensive rework. If the work delegated to future phases is well understood, the

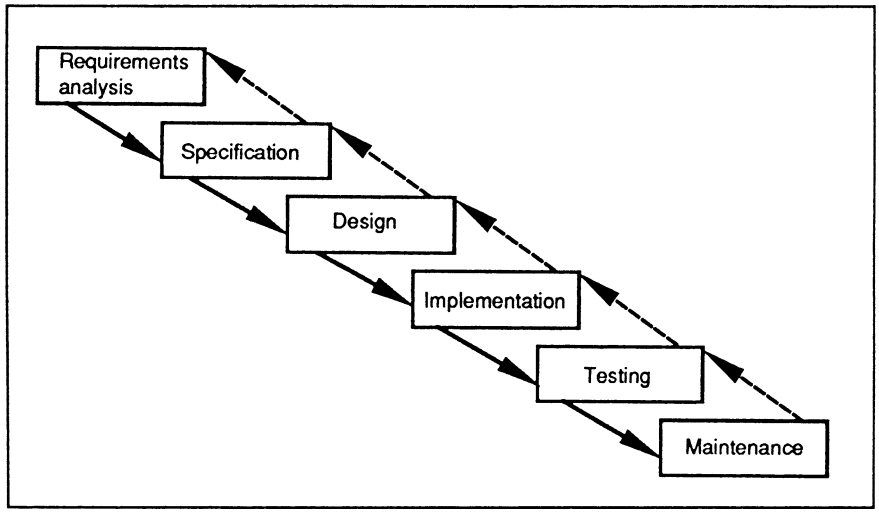


Figure 6.3. The waterfall model represents the traditional system-development process. It recognizes six sequential phases, each with feedback to the earlier phase.

waterfall model can work well. If the specifications challenge later phases, the risk of cost overruns and even failure is great. For instance, specifying X-ray images to be displayed at every terminal within one-tenth of a second requires extensive resources, and, if their cost becomes intolerable, the entire design might have to be revised.

Although the waterfall model conveys a clear view of the software life cycle, the delays inherent in such a phased development process make this methodology inappropriate for innovative applications. It is hard to obtain adequate specifications when you are developing an application for inexperienced customers. They will not see the product until the testing phase and cannot provide feedback until that time. If it takes years to complete the first four phases of the process, even good specifications are likely to have become obsolete—the organizations and their information sources and needs will have changed. It typically takes a second implementation to get a smoothly operating system.

Requirements for some health care systems are presented in Section 6.1.2, but just listing the requirements is insufficient for phase 2. The requirements have to be quantified: where is the information needed, by whom, in what form, within what context, how fast, how often, and how reliable should it be? High reliability can mean that redundant storage and processing has to be included in the design. A single program for a single user can be built without explicit attention to phases 2 and 3, but a large project requires a more careful approach.

In the **specification phase**, the general system requirements are analyzed and formalized. Systems analysts specify the intended behavior precisely and concisely—the specific functions the system must accomplish, the data it will require, the results it will produce, the performance requirements for speed and reliability, and so on. They answer

questions such as "What are the sources of necessary information?" and "What are the viable mechanisms for communication in the current system?" Here the DFD becomes an important tool.

Once the functional specification has been produced and validated against the requirements specified by the prospective users, it is expanded into a design that can be implemented. The initial task is to partition the system into manageable modules. All but the smallest systems are partitioned into subsystems, subsystems into smaller components, and so on, until the whole has been decomposed into manageable components (Figure 6.4). A single module should not require more than a few implementors and not take more than a few months to write and test. Any subsystem should also have few data flows into and out of it, as determined by the DFD. Simplicity allows subsequent testing to cover all cases. Notice in Figure 6.4, for example, that an HIS can be viewed as a hierarchy of nested and interrelated subsystems. In general, a subsystem has strong internal linkages relative to its linkages with the external world.

As part of the design, the hardware needed now and in the future is determined; frequently more hardware can simplify the design and its implementation. For instance, copying data to locations close to a critical customer can improve response time and availability in case of communication problems.

In the **implementation phase**, the subsystems are created in-house or are obtained from public or commercial sources. Even good design specifications cannot cover all the details at the module level. For a good implementation, the module programmers must understand the health care setting. If the system does not function suitably for the end users—namely, if they find the output wrong or not sufficiently helpful—costly rework will be needed after the technical testing phase. Such problems might be caused because

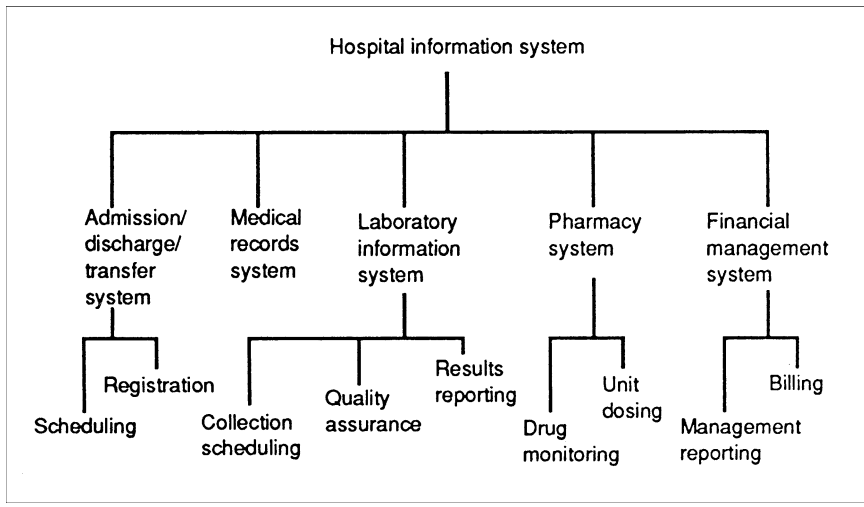


Figure 6.4. A hospital information system comprises interrelated subsystems that serve individual departments. In turn, each subsystem comprises multiple functional components.

the implementation was wrong, the design did not foresee performance problems, the specification was imprecise, or requirements were incomplete or became obsolete. If the errors are due to earlier phases, making the corrections becomes more costly.

In practice, the implementation phase will acquire as many readymade components as can be found and will put them together into a system. The components are selected to match the specifications, and less detailed design takes place. Analysts can visit sites where the components are used, and customers can see sample results early on. Early feedback to the design phase can occur. Because available components will rarely match requirements precisely, compromises in specifications will have to be made, although many vendors will try to create adaptable components that match the needs of many institutions. If some components fail to meet the needs of the customer, the vendor may be induced to improve them, or staff may decide to build some components in-house, raising the subsequent maintenance costs.

When all the components are ready, **system integration** takes place. Mistakes in partitioning, missing data flows, and incomplete interface specification become apparent. For large systems, built by many people, this is the riskiest phase. Even competent people may interpret specifications and design documents differently; some interface or data organization differences make integration impossible without redesign.

Testing of the system has to be planned carefully. Subsystems are tested by their developers, because customers probably will not be able to test incomplete functions. Because testing implies the discovery of failure points, this phase has to start slowly to allow for repair of errors so as to avoid frustration by the testers—especially when real customers begin to do testing. It is in this phase that we find out whether the hypothetical requirements, stated initially, are adequate. If they are not, major rework may be needed. Testing takes as long as implementation. Many new systems are abandoned at this point.

If the system, after testing by builders and customers, is accepted for routine use, the **maintenance phase** starts. Ongoing changes in requirements will necessitate changes in the system. In theory, the same cycle should be followed for any substantial changes, but the original large staff rarely remains available, and thus the adjustments may be made haphazardly. Because software systems often remain in place for 10 to 20 years, the maintenance phase, costing perhaps 25 percent per year, requires a greater financial commitment over time than the original development phases required.

Alternative Methodologies

To obtain early results and reduce the risk of failure, a software development group may follow alternative methodologies. In the **spiral model**, the group generates a simple prototype system by performing the four initial phases rapidly (Boehm, 1998). The result is presented to the customers, who assess it and expand and modify the requirements. Then a second cycle ensues (Figure 6.5). After a few cycles, the prototype is made operational; the operational prototype will require some maintenance, but any significant changes have to wait for yet another iteration of the spiral. Ideally, each cycle takes 3 to 6 months, so the patience of the customers is not exhausted. In the spiral model, the staff is expected to remain intact for many iterations so that essential knowledge is not lost, because little time is spent on documentation.

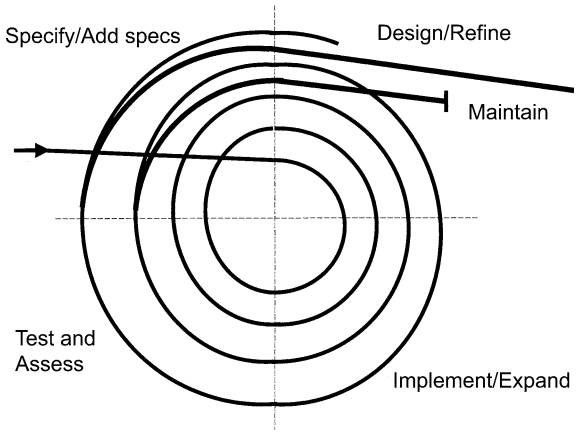


Figure 6.5. The spiral model of implementation. Rapid iterations lead to system cycles that can be assessed incrementally by the customers.

The spiral model methodology also entails substantial risks:

1. The initial prototype is so minimal that the result is not of sufficient interest to the users, and no meaningful feedback ensues; for instance, a system may not permit access to actual patient data.
2. The system does not attain sufficient functionality after several cycles, so it cannot be placed into service; for instance, laboratory results are not yet automatically included in a patient's record.
3. An acceptable small system cannot be scaled to the next level on a standard cycle because of an overly simplified initial design decision, so major rework is needed; for example, no means exist for out-of-queue processing, so stat (emergency) laboratory tests cannot be given priority.

Developers have experimented with other software engineering methodologies, but all require that developers and customers perform similar tasks. For instance, by assessing which components are likely to be hard and risky to build and giving them more attention, the team can reduce implementation risks while accelerating delivery. This *water sluice* methodology avoids a problem of the spiral model in which, to satisfy the cyclic time constraints, the easy processes are done first. Making correct assessments requires considerable experience.[1] "See http://www.db.stanford.edu/~burback/watersluice/node34.html".

Business Objects

Object-oriented programming focuses on the specification of small, reusable data structures and associated methods, called *objects*. For instance, a *visit object* may be defined as a component of a patient record with methods as *schedule*, *check-in*, *record findings*, *issue orders*, etc. The visit object must have well-defined interactions with other objects of concern to the system, such as *patient*, *doctor*, *nurse*, *clinic*, or *pharmacy*. Given a suite of suitable objects, this approach allows rapid composition of software modules.

Libraries of such business objects are becoming available. Programmers must have substantial experience to define objects of sufficient competence and generality while allowing for their many interactions, as required in health care systems.

For complex data, the object-oriented approach may be best. All the data related to major concepts in the information systems is brought together and is accessed via methods that isolate the complexity of the data structures from the programs that use the data. Programming languages that enable object-oriented programming are listed in Table 5.1. The design of object classes to support a specific business domain is a complex task, but there are standard libraries of defined objects. If these libraries are used, the programming task is reduced to prescribing the sequence and content of the interactions among these objects—e.g., patients, their problems, and their visits.

Standard object libraries for some domains are now available—e.g., presentation graphics, telecommunications, and business finance. Many organizations are defining and implementing collections of business objects for a variety of other areas. However, if multiple groups enter a single topic area without common standards (see Chapter 7), their products will not interoperate, and it will take some time before dominant library products emerge. If multiple libraries are needed for a project, an interchange format for data is needed as well. Having a standard database can provide the required storage, but such a database must be capable of holding the data organized in object formats.

Summary

As we indicated in Section 6.3.1, few health care organizations have the resources to see major software system developments through to a successful completion. If a modest computer system development effort can address problems specific to some health care setting, however, it may be worthwhile to undertake a local effort. There are still problems that require novel solutions and technology not available from most vendors of health care systems. Many problems in health care are poorly structured and may require methods based on AI. Such an effort will typically be established as a subsystem in the institutional network and be initially isolated from daily operational responsibilities. The subsystem will depend on available complementary system resources, such as an existing patient-admitting system. Using the spiral approach, experience can be gathered from local customers that outsiders would not be able to supply. It is still essential to ensure reliability, security, and data confidentiality, as discussed in Chapter 10. A new subsystem, no matter how beneficial, is likely to be rejected if it entails risks to the routine services of health care delivery, patients' confidence, or cost accounting.

6.3.4 Incorporating Remote Services

Our health care systems do not operate in isolation. Within major institutions, there will be several distinct computer systems, such as in the laboratory, in physician offices, and in account billing. Outside of the institution, there are other organizations that use computer systems, such as drug and linen suppliers, laboratories that do outsourced work, insurance companies, public health service agencies, and the National Institutes of Health (NIH), including the National Library of Medicine (NLM). The capabilities of modern networks,

as introduced in Chapter 5, enable direct linkages to all these systems, and greatly reduce paper copies and manual transcription of information. The interfaces are diverse, however, unless standards are adopted, as discussed in Chapter 7. Even when standard interfaces are agreed on at one point in time, technology eventually renders them obsolete. Nonetheless, the ability to interact with remote services is essential, and the cost of maintaining interfaces is less than that of transcribing the information by hand or of entering it manually into the home system (see Section 6.3.5). External services require different business arrangements depending on whether they are **informational services** (informal, broad public access) or **business services** (contractual, user access only).

Information on the World Wide Web

For informational systems accessed via the World Wide Web, the Hypertext Markup Language (HTML) dominates Web services. A **markup language** identifies items of the contents. The term *hyper* refers to the capability of references in an HTML document to link to other documents through a Universal Resource Locator (URL), which identifies files at remote locations, such as http://www.dlib.org. The HTML focuses on the presentation of information and allows the definition of text, headings, tables, images, or even program fragments that provide dynamic displays, using the Java language (see Chapter 5). This information is interpreted by a **browser**, which provides the most suitable presentation for display. Browsers deal well with inconsistencies, errors, or novel specifications, providing the most reasonable presentation. The utility of these tools for health care information is presented in Chapter 19.

For business services, the inherent flexibility of the Web and HTML is also a liability. There is no guarantee of completeness of the information or of consistency of content. Many standards have been developed for data interchange; each typically is created for a specific domain, as discussed in Chapter 7. To allow data–quality interchange on the Web, the eXtensible Markup Language (XML) has been widely adopted. Rather than focusing on presentation formats, it specifies contents through extensive use of tags. The meaning of the tags is specified for specific domains (Connolly, 1997). The **XML format** provides a common syntactic technical infrastructure and allows sharing of software tools. A variety of efforts are underway that are specific to health care, as described in Chapter 7, which will enable easier integration of remote services.

6.3.5 Designing for Effectiveness

A system's success depends not only on whether it meets the users' informational needs but also on how it interacts with those users. There are many types of users, ranging from technical specialists to people who only require easy-to-use information services. The resident's complaints in the hypothetical example illustrated an important point: the disruption of effective and established routines and the inconvenience often associated with computer information systems can cause users to work around the system and thus to fail to share its capabilities. Studies of the attitudes of health care personnel regarding computer-based clinical consultation systems have shown that successful programs not only must provide expert-level advice but also must be integrated into the

daily routines of physicians and other users (Friedman and Gustafson, 1977; Teach and Shortliffe, 1981; Detmer and Friedman, 1994). Computer-based information systems should acknowledge the hectic schedules of health professionals and should demystify and simplify the mechanics of the human–computer interface (HCI). By involving users in system design, developers can avoid many of the impediments to widespread success.

The following parameters of computer systems are among the most important to consider during system design:

- *Quality and style of interface*: From a customer's point of view, a system's interface is the system. To be effective, interfaces must have clear presentations, avoid unnecessary detail, and provide a consistent interaction. Since users will have many interaction points with computers, consistency is a global issue. Menus, graphics, and consistent use of color all help to make systems more attractive and simpler to learn and use. Today, clicking on a mouse is ubiquitous, but requires that users are settled in front of the screen. Traditionally physicians were reluctant to type at a keyboard; furthermore, sitting down and starting an interaction is awkward in many health care settings. Thus, system developers have experimented with a number of alternative devices for interaction, including light pens and touch screens. Researchers are investigating speech recognition and natural language understanding. These technologies are now available for limited vocabulary, but not yet as natural and easy as talking with human beings. Mobile devices such as personal digital assistants (PDAs) and laptops have opened up major new opportunities for interaction between clinical systems and their users. In systems that perform decision support and offer advice about patient care, the style of communication is particularly important. Is the system too terse or too verbose? Does it project a helpful attitude or does it seem pedantic or judgmental? Can it justify its recommendations?
- *Convenience*: Customers must have convenient access to the system. The number and placement of workstations and printers are important considerations if the system is to be assimilated into users' routines. For example, if an EHR system is intended to replace the traditional patient record, it must be accessible wherever and whenever health professionals need to look up patient data—in the physician's office, in the nursing station, possibly at the bedside, and so on. Sufficient workstations should be available that users do not have to wait to use the system, even during times of peak use. Logging in must be rapid. Convenience is a major reason for the growth in interest in mobile devices, especially given the highly mobile nature of health workers in clinical environments.
- *Speed and response*: System developers must choose hardware and communication methods that have sufficient capacity to handle customers' demands for information during peak hours, whether through wired or wireless connections. The software must allow users timely access to the data in the form they need. Minor errors in formulating information requests must be easily undone. Health professionals are understandably reluctant to use systems that are hostile, tedious, or unduly time-consuming.
- *Reliability*: In the event of hardware or software failure, health personnel must resort to manual procedures. Redundant hardware and frequent data backup can minimize the loss of time and data.

- *Security*: The confidentiality of medical data is an important issue in the design of health information systems (see Chapter 10). The system should be easily accessible to authorized personnel, yet it should not release information to unauthorized users. These conflicting goals are difficult to achieve. The most common compromise solution is to assign an account with a password to each user, and sometimes a physical token (*cryptocard* with a second password that changes frequently) or **biometric identifier** (such as the user's thumbprint). Variable access to the data then can be controlled for individual users or for classes of users. Certain operations may be restricted to particular workstations. For example, the ability to modify patient charges is best restricted to financial personnel working from workstations located in the accounting office. Medical notes are hard to classify and may require filtering before they can be released to outsiders (Wiederhold et al., 1996).
- *Integration*: The integration of information from independent systems eliminates some of the difficulties and enhances the benefits of using computer systems. If, for example, a laboratory system and a pharmacy system are independent and incompatible, health professionals must access two separate systems (possibly from different workstations) to view a single patient's data. If the two systems need to share data, people will have to collect output from the first and reenter it into the second system. The development of local-area networks (LANs) has provided the infrastructure to allow exchange of information among independent systems and has reduced the need for redundant entry and storage of critical information (a waste of time and a source of errors). Standards, such as those based on XML, reduce the cost of acquiring and maintaining software for integration. The ensuing collaboration will help in developing common terminologies and reduce misunderstandings.

Technological progress is enabling the introduction of effective systems in health care. While the path is difficult and often frustrating, there is no doubt that health care has come to depend on computing and data processing as core capabilities in modern care-delivery environments.

6.3.6 *Planning for Change*

Many health information systems—particularly custom-designed systems—take a long time to develop. This delay is risky; it is hard for the medical establishment to perceive what is being done by the system development staff, and the development staff receives no feedback on the correctness of the assumptions on which they are basing their design. The involvement of users through demonstrations and training sessions and the incremental installation of the system can build the enthusiasm and support of users and can provide computing personnel an invaluable means for evaluating progress.

Visiting similar sites can provide the insights that are needed to develop judgments about new systems or proposed improvements. **Prototypes**—working models that exhibit the essential features of the system under development—facilitate communication between computing personnel and users. Users develop a realistic idea of what the system will look like, how it will work, and what it will do. Developers receive feedback and can modify the system in response to users' comments, thus improving the likelihood

that the final system will be deemed acceptable. A good prototype provides a realistic demonstration of the method of interaction with the system and should be able to deal with most of the common varieties of data input. Much simplification is possible when reliability and data permanence can be ignored, as they can be in a prototype system that is intended only for demonstration and discussion.

Formal training courses also can help to dispel the mystery of a new system. Without adequate reinforcement, however, people are apt to forget what they learn. A training program for a new system should start slowly and extend over the period of its implementation so that surprises are minimized. When some subsystem installation is imminent, health professionals should receive specific and intensive training in its use. Experience with windows and mouse interfaces shows that intuitive metaphors allow users to experiment and explore the system's capabilities and reduce the need for formal training. The feedback provided to the developers during successive phases helps to ensure that problems will not be repeated or multiplied as the system nears completion.

Training is simplified if part of the system is operational. Customers can see real examples of the system in action and thus are less likely to develop overly ambitious expectations. Because those parts of the system that are installed first usually are less problematic, the initial phase is likely to be a success. The attitude generated by initial success can enhance the acceptance of the system in other areas, including those areas that inherently are more difficult to address using computer-based techniques. In hospitals, for instance, computer systems often are used first in the admissions office, where the applications are relatively straightforward, and later are expanded for use on medical wards, where the user community is particularly diverse and demanding.

6.4 Summary

A variety of software and information services are available for health care applications, ranging from turnkey systems, covering many of the needs of an institution, through major subsystems for hospital departments and large Web-based services that require only remote access to databases and to object libraries that are helpful for building local applications. Unfortunately, composing an effective and economical system that is fully sufficient for the financial, clinical, and ancillary needs in health care is nearly impossible. Today, most health care institutions are faced with the choice of either committing to a major vendor, and accepting that vendor's complete system or a subset thereof, or dealing with a variety of purchased, adapted, and in-house developed subsystems that overlap and have gaps. Gaps often require human transcription and replicated data input. The maintenance of the systems requires ongoing efforts to keep them up to date as operational and clinical requirements and processes change.

Information systems development is a political as well as a technical process. Health institutions, just like all other organizations, are composed of different groups of individuals, and often these groups have conflicting priorities, objectives, and values. The technical issues are generally complex and difficult; careful implementation processes can conflict with urgent needs. Risks are rife in internal software development, in

acquiring vendors' software, and in their integration. Health administrators, physicians, nurses, ancillary personnel, and patients have diverse needs that the computer system must accommodate. Information systems can alter relationships among these people—they affect patterns of communication, perceived influence, authority, and control. A strategy for implementation must recognize and deal with these political forces. A new system should disrupt the organizational infrastructure as little as possible. The article by Keen (1981) (listed in the Suggested Readings) discusses the political aspects of system development and implementation in greater detail.

Suggested Readings

Anderson J.G., Jay S.J. (Eds.). (1987). *Use and Impact of Computers in Clinical Medicine*. New York: Springer-Verlag.
This collection of papers presents research on the factors that affect the adoption, diffusion, and utilization of clinical information systems in hospitals. It includes chapters on the attitudes of health professionals toward computers and the probable effects of clinical systems on aspects of medical practice, such as the role of physicians, relations between doctors and patients, and the organization of the health care delivery system.

Blum B. (1992). *Software Engineering: A Holistic Approach*. New York: Oxford University Press.
This philosophical but unbiased textbook on software engineering was written by a practioner who has had substantial experience developing clinical applications. It covers the system-development process, data-flow diagrams, and structured coding techniques.

Boehm B. (1999). Managing Software Productivity and Reuse. *IEEE Computer*, 31(9):111–113.
Making modules reusable, at some additional costs, reduces subsequent development costs.

Boehm B., Egyed A., Kwan J., Port D., Shah A., Madachy, R. (1998). Using the WinWin Spiral Model: A case study. *IEEE Computer*, 31(7):33–44.
This brief note introduces the WinWin Spiral Model, mainly in relation to student projects. The WinWin Spiral Model has two add-ons to the classic spiral model: (1) analysis at the beginning of each spiral cycle and (2) process anchor points. A risk of using the spiral model is that methods used initally may not scale to the eventual requirements.

Booch G. (1994). *Object-Oriented Design with Applications* (2nd ed.). Redwood City, CA: Benjamin-Cummings.
This is an introduction to object-oriented programming, by one of its chief proponents and tool builders.

Keen P.G.W. (1981). Information systems and organizational change. *Communications of the ACM*, 24:24.
This paper emphasizes the pluralistic nature of organizations. It discusses the changes in patterns of communication, influence, and control that often occur when new information systems are implemented, and it suggests strategies for minimizing social inertia and resistance.

Leymann, F. and Roller D. (2000). *Production Workflow: Concepts and Techniques*. Englewood Cliffs, NJ: Prentice-Hall.
This book illustrates the principles of linking computing and manual tasks for smooth interaction.

Monson-Haefel, R. (1999). *Enterprise JavaBeans*. Cambridge, MA: O'Reilly & Associates.
JavaBeans is a technology to define business objects that can be composed to assemble rapidly an analyzable and maintainable data-processing system.

Pigoski, T.M. (1997). *Practical Software Maintenance: Best Practices for Managing Your Software Investment*. Los Alamitos, CA: IEEE Computer Society Press.
A basic reference for computer software maintenance. Corrections are 20 to 25 percent of the effort, adaptations 40 to 60 percent, and perfection consumes the remainder.

Reiser S.J., Anbar M. (Eds.). (1984). *The Machine at the Bedside: Strategies for Using Technology in Patient Care*. Cambridge, UK: Cambridge University Press.
This book discusses the theory and use of health care technologies—such as intensive care, diagnostic imaging, and electronic fetal monitoring—in the context of legal, ethical, economic, and social concerns. It contains 23 case studies that depict the benefits and limitations of using these technologies.

Shlaer S., Mellor S.J. (1992). *Object Life Cycles, Modeling the World in States*. Englewood Cliffs, NJ: Prentice-Hall.
This influential textbook was developed by protagonists of the object-oriented software development approach.

Webster, J. G. (1988) (Ed.). *Encyclopedia of Medical Devices and Instrumentation*. New York: Wiley.
This reference includes many technical definitions, including an extensive description of hospital information systems, listing over 200 functions that such a system might provide.

Questions for Discussion

1. Reread the hypothetical case in Section 6.2.1.
   a. What are three primary benefits of the clinical system? What are three primary disadvantages?
   b. Do you think that the benefits of the system outweigh the disadvantages? Are there adequate noncomputer-based solutions to the problems with which the system was designed to help? If so, what are they?
   c. How would you change the system in your institution or in one you have read about? Among the topics you might address are the effects of the system on hospital routine, computer reliability, and terminal availability and the adequacy of user training programs.
2. Describe an outpatient clinic's billing system in terms of inputs, outputs, and processes. Sketch a simple data-flow diagram that represents your model of the system.
3. Discuss the inherent tension between protecting the confidentiality of patient records and providing health professionals with rapid and convenient access to clinical information. What level of system security do you think provides an appropriate balance between these conflicting goals?
4. Discuss three barriers to technology transfer among health care institutions.
5. Explain the difference between outcome and process measures of system performance. Identify two outcome and two process parameters that you might use to

evaluate the performance of a clinical consultation system that assists physicians in diagnosing disease. Describe an experiment that you could perform to evaluate the effect of the system on one of these parameters. What potential difficulties can you foresee in conducting your experiment? What can you do to compensate for these difficulties?

6. In what three ways is the use of a clinical consultation system similar to the use of human consultants or static sources of health information such as textbooks? In what three ways is it different?