chapter 9

Managing Electronic Health Record Systems: Collaboration and Implementation

Mary Alice Hanken and Gretchen F. Murphy

Student Study Guide activities for this chapter are available on the Evolve Learning Resources site for this textbook. Please visit *http://evolve.elsevier.com/Abdelhak.*

When you see the Evolve logo ⓔ, go to the Evolve site and complete the corresponding activity, referenced by the page number in the text where the logo appears.

Key Words

Business data
Collaboration
Downtime
Project
Stakeholder
System life cycle
System testing
Training Manual
User
User manual

This is an adaptation of a chapter written by Ellen Anderson in "Electronic Health Records: Changing the Vision."

Chapter Contents

Abbreviations

EHR—electronic health record
HIM—health information management
I/S—information systems
IT—information technology
MIS—management information systems

Objectives

- Develop a strong working relationship with information systems staff.
- Recognize the strengths of the health information management professional and the information systems professional in developing, selecting, implementing, and modifying health care information systems.
- Understand the difference between project activities and operational activities.
- Identify the necessary skills to bring an information systems project to completion.
- Describe the principles of collaboration between health information management and information systems that help projects be successful.

- Recognize the areas where health information management and information systems do not automatically share the same views or information or approaches to a project.
- Recognize the value of training in project success.
- Differentiate between a training manual and a user manual.
- Evaluate the success of systems projects.
- Identify both measurable metrics and organizational values addressed by a system.
- Recognize that systems projects move into an improvement phase after implementation.

The growth of electronic systems across all business settings redefines the way information is used for the delivery of products and services. Healthcare, while slower to embrace technology in all areas, is now firmly engaged in bringing electronic tools to delivering health services of all kinds. This book presents the evolution of technology from many health care perspectives including legal and regulatory frameworks, business process technological advances, patient care outcomes, more effective use of data and, threaded among all of them, a recognition of the value, roles, and essential nature of information systems users. To make change happen effectively, health care leaders must identify, engage, and appreciate the user community and customers of institutional health information systems.

VALUE OF USER PERSPECTIVE

The business community, or **"user,"** perspective of systems projects can help business communities identify and initiate high level system project sponsors and stakeholders as well as address fundamental roles for users at multiple levels. Appreciating user roles enables institutional sponsors to: (1) understand the dependencies and the work effort involved in systems implementations, (2) hold realistic expectations of process and outcome, and (3) become a much more sophisticated partner in systems development and implementation.

A Natural Dependence

COLLABORATION PRINCIPLE

Collaboration Principle: There is a natural dependence between users and information systems in the pursuit of definition, design, development, purchase, implementation, and usage of any EHR system.

Computer systems in the workplace are tools applied to business functions. They are the workhorses of business. On an equal plane, computer systems fail to perform if data structure and databases, operating systems, interfaces, and networks are not well planned, robust, and technically compatible. This business community–computer division dependence should be more than just recognized; it should be embraced.

Environmental basics is a term the original author, Ellen Anderson, has created to define the "human elements" that exist as a natural part of humans composing work teams intended to accomplish specific tasks. The eight environmental basics covered in this chapter include personal commitment, trust, respect, policy setting to manage human behavior, expectations, vocabulary, understanding, and knowledge.

A **stakeholder** is any individual who has a stake or real interest in the computerization project and implemented electronic health record (EHR), regardless of department, position, or functional role in the company (Table 9-1).

The information technology (IT) division of most organizations has a title that includes the word "information." This is true in health care organizations as well. A few examples are management of information systems (MIS), information technology (IT), and information services or systems (I/S). The abbreviation I/S is used in this chapter when referring to either the department itself or the staff of individuals in this unit of work activity in any organization. It is not surprising to note that more than one work unit of an organization may use the word "information" in its title. Both the computer department and the health information department are in the business of managing information for the health care industry.

Webster's New Twentieth Century Dictionary of the English Language[1] defines **collaboration** as "to labor, especially in literary or scientific pursuits, as the associate of another or others." In this chapter, the term is used to refer to the combined labor of users and I/S staff in the pursuit of EHRs. The important word here is, of course, together.

Project, in the context of this chapter, means a unit of work to acquire or build and implement a computer system

Table 9-1 USERS OF THE ELECTRONIC HEALTH RECORD

Physicians
Nurses
Nurse's aides
Physical therapists
Occupational therapists
Speech therapists
Health information staff
Medical records staff
Care/case coordinators
Dieticians
Decision support analysts
Pharmacists
Pharmacy technicians
Laboratory technicians
Radiologists
Radiology technicians
Data analysts
Quality improvement analysts
Utilization review coordinators
Business office personnel
Social workers
Mental health specialists

AND THE LIST GOES ON AND ON . . .

(software, hardware, and all infrastructure) with a definitive start date, targeted stop date, steps, milestones, and outcome.

Project work is quite different from regular operations in any organization. Table 9-2 illustrates some comparisons. The nuances of projects make collaboration that much more important and that much more challenging.

Table 9-2 PROJECTS VERSUS OPERATIONS

Projects	Operations
Peaks and valleys work flow	Steady stream of work
No direct authority, full responsibility	Full authority, full responsibility
Ad hoc teams	Business-aligned teams
Done once	Done repeatedly
Highly time dependent	Function dependent
Defined start and end dates	Not "start date" and "end date" dependent
Product/service driven	Business function/process driven

ENVIRONMENTAL BASICS (REQUIRED FOR TRUE COLLABORATION)

The **system "life cycle"** has been used in describing computer system development for many years. Simply put, the stages of a system can be expressed in simple verbs: define, design, develop or buy, do (or implement), and support. That system life cycle then recycles to redefine, redesign, and so forth, in a continuous fashion until the system is no longer used and another takes its place. (See Chapter 8 for more information on the system life cycle.)

Define, design, develop, implement, redesign, redevelop, and do are all action verbs that involve groups, and lots of them. Because EHR systems happen through groups, environmental basics that are group oriented must be in place before an EHR project is started. All these environmental basics must be intact for project success. These same principles apply to a working relationship to improve and expand systems within an organization or in a community setting.

Four Shared Environmental Basics

COLLABORATION PRINCIPLE

Collaboration Principle: Personal commitment, trust, respect, and policy setting are four mandatory environmental basics shared between users and I/S for true collaboration.

Personal Commitment

COLLABORATION PRINCIPLE

Collaboration Principle: Personal commitment builds computer systems champions.

Commitment is not easy when resources are tight, especially if the capital costs are high, the risks are great, and the resources are limited. Yet personal commitment to a project produces the political will to make the system happen. Nothing goes farther than a user who truly believes. Commitment needs to come, however, from all sides of the collaboration equation—from organizational leaders, system users, and I/S. Personal commitment is more than verbalization. It is quite easy to verbalize commitment; it is quite another matter to stand behind that commitment.

If you are starting a major systems project, use focus groups, surveys, and interviews to assess the level of commitment. Study the positive and negative feedback from all focus groups, surveys, or interviews. Is there a strong personal commitment to this project? Is the positive or negative perception warranted? Is it based on correct or incorrect facts? If incorrect, how can you correct them? Will correcting the facts change the degree of personal commitment? If correct facts have created a negative perception, can conditions be altered to raise the level of commitment? If users show strong personal commitment to the project, are they grounded in reality?

Grounded Champions

User leaders and managers make the best system and project champions. Sometimes, however, they need accurate resource information to be truly grounded in their commitment and genuine excitement. User managers can be very serious about the project but also blind to the resource commitment of their operational people and budget to ensure success. Test their commitment before beginning the project. They will usually underestimate resources in terms of both people and money—it is human nature to wish it done quickly and at a cost less than realistic.

I/S managers are another important group who must be personally committed to the project. They are faced with an overwhelming number of requests to automate or update existing automation from all business units within the organization. Setting a high I/S priority to a project is usually propelled by a combination of (1) a compelling business reason and (2) the right technology fit and direction. For example, it makes no sense to automate a business function when the desired technology is not ready for general usage.

If I/S leaders declare a project a high priority, you are halfway there. Are there true capital dollars behind it? Is it assigned a budget? Is it a shared budget among a number of departments, including I/S? Is the budget "assigned" realistic? Does I/S appear ready with pen in hand to sign initial project documents if you moved forward immediately?

Trust

COLLABORATION PRINCIPLE

Collaboration Principle: The worst of projects can succeed within a basis of trust, and the best of projects will likely fail without it.

It has been said that the worst of projects were successful because, even during the down-and-out periods, project members and business communities trusted each other. Together they pulled a project through the battles to victory. It has also been said that the best of projects and systems failed because no trust existed.[2] When no trust exists, each source of irritation or project component failure brings the project closer to its knees. A baseline level of trust must be in place as a project begins. Trust develops more completely as a project's work groups or teams grow comfortable in the natural dependencies between them. There are various levels of trust in any collective group of people; EHR systems are no exception. All levels within an organization must trust, and be trusted, for maximum use of group synergy.

Those who hold the corporate purse strings, or control the organizational finances, must trust the rest of the organization to be good stewards of the millions of dollars often required to get a system into production. They must also trust that the proposed budget reflects the true costs or costs as commonly computed within the organization. If they take the fiscal stewardship of the organization seriously, they will ask for a cost/benefit analysis or strong reasons about value to help in their prioritization of this capital expenditure relative to other capital expenditure requests.

Project steering committee members need to trust that their fight for the capital dollars will result in tangible computer systems. They also must trust that the people who will be empowered to select or create and implement the system have the talents and the will to do so. They may also trust that the potential system users have or will acquire the computer skills and aptitudes needed to use the system. Do they have these trusts?

User managers are the most trusting of all, especially if they are true champions of a potential system. They trust that the people above them will act on their commitment of resources. They trust that the users they supervise will embrace the system, or at least accept it and make it work. They trust that I/S staff know how to implement and support the desired system, have budgeted appropriately, and will supply resources when needed.

If a system will be purchased, the organization must trust vendor representatives who provide sales information about the product, help to configure the system for use, and support implementation at time of going live. Do they trust that the vendor will deliver the promised system? Will the vendor work to fit their product to the client's environment or build only what sells in the marketplace?

The I/S staff must trust the user community to know what it wants and needs in system functionality. They must also believe that the users will deliver the human resources necessary to participate actively in the project that the user community will ultimately own. They must also trust that the business community stands ready to basically own and functionally support the system after implementation.

Users, then, must trust all of the above. Do users see the above groups as trustworthy? Do they perceive that the above groups will deliver? Is this particular project a "wise" use of the organization's dollars in their minds? Will the various departments really support users through initial training, implementation, institutionalized training and retraining, and system support?

Through the focus groups, interviews, and surveys, distrust may appear. Has I/S "let down" the user community in the past on the basis of technical promises not kept or a cost that is three times the estimate? Has the business community ill defined a system and "wasted lots of I/S dollars" trying to define it while in the middle of system development? These are two very common distrust statements. Neither contributes to future project success.

The potential list of double-sided "done me wrongs" could go on and on. This is truly only project background garbage and needs to be cleaned up before a good working relationship can develop in the dynamics of working groups and team efforts. Distrust does not contribute to positive group synergy.

Respect

COLLABORATION PRINCIPLE

Collaboration Principle: Mutual respect breeds collaboration. Disrespect destroys it.

In information systems projects, the caste system that exists in most health care organizations can quickly disappear. The appointment clerk can be as important or more important than the physician in an appointment scheduling system implementation. This creates some interesting team membership. A baseline level of respect between project players can hasten the comfort level of multidiscipline, multibusiness unit project teams who have seldom worked side by side. Individuals need to feel respect, regardless of what role they play in the organization or in an automation project.

Equally critical factors in system implementation are two common problems with systems work: (1) resource tug-of-war and (2) timeline tag. Because there are so many components included with systems, planned people resources will likely get "shuffled" a lot, and unplanned resources will continually knock on managers' doors for staffing. For example, a user manager may have planned to have two of the staff "free" in May for 2 weeks for user testing, but the user testing will be delayed 3 weeks because of a delay in receipt of equipment. The staff is now needed in June, possibly not a good time from the business unit's perspective.

Using the delayed equipment example above, the project timeline has now leaped forward by at least 3 weeks. Project teams will be trying to play timeline tag to catch up or move the "go-live" date forward. If the "go-live" date is a "hard date" (which means it cannot be changed—perhaps it is a regulatory compliance date, for example), the pressure to play the tag intensifies.

Project teams and managers must be sensitive to user frustration resulting from this resource shuffling and the timeline movement, and the user community must be sensitive to what components of the project work are truly under the control of the project manager and teams.

It is also important to understand that the user community can cause the delays as well. Turning the tide, the user community can cause a delay because an operations activity suddenly "takes priority" or a consensus activity within the user community cannot be scheduled for when it was originally planned.

Gauge what mutual respect exists between the business units to be involved with the automation project and between the collective business units and the I/S staff. Does each appreciate the role and work effort of the other? Does each appreciate the time it takes to accomplish the designated work effort that contributes to the whole of the organization?

Policy Setting

COLLABORATION PRINCIPLE

Collaboration Principle: In the absence of clear policies and principles of information management, there will be internal conflict and civil war.[3]

It has been appropriately said that policies and principles manage human behavior. These directives for human behavior guide both system development and the project teams developing them. In large measure, data access, data use, data content, data standards, authentication, electronic signature, data exchange standards, data release, and data security are focused subjects for policies and principles directly guiding EHR systems. For example, patient confidentiality will remain a major concern until appropriate policies are in place that address patient identifying data running through FAX lines, internal messaging software, the Intranet, the Internet, and today's rapidly developing wireless technology, and any other technology that develops in the future. The same concern appears as a data warehouse or clinical data repository project kicks into gear because data are suddenly pulled from a transaction processing system where individuals have an easily established "need to know" and moved into the organizational hands of an entire enterprise. The warehouse or repository will create a need for new policies and procedures related to data access, use, and retention.

As the rate of technology advancement intensifies, policies and principles quickly become outdated and ineffective guides to human behavior amid exciting new automation. Project teams can become foes as one pushes ahead with no clear organizational directive while the other refuses to move without it. The users and I/S staff can quickly become antagonistic. Before this happens, the organization must create the appropriate behavioral framework for use in that organizational culture. Collaboration, then, is not hindered by lack of organizational directives for human behavior.

It takes real courage to develop such policies. Those who are pushing hard for the system can label policy makers a hindrance to automation because they anticipate that policy setting will result in a more complex system or a time delay. They may be right. Those who have this perception do not understand that this policy-setting work is as normal to system development as programming screens are. It is best to establish this policy through an enterprise-wide group that (1) takes the stewardship of data, especially patient identifiable data, to heart and (2) takes on the scope of "setting confidentiality criteria that would provide a basis for new applications and for new delivery system models."[4]

Four Unshared Environmental Basics

So far, we have talked about four shared environmental basics. There are four other environmental basics; however, that will not be initially nor automatically shared between I/S and the user community.

COLLABORATION PRINCIPLE

Collaboration Principle: Expectations, vocabulary, understanding, and knowledge are four environmental basics not common between the user community and I/S.

Work on these environmental basics because they will make or break a project.

Expectations

COLLABORATION PRINCIPLE

Collaboration Principle: Perception is an individual's reality built on expectations. Individuals in the user community and individuals in I/S will not share initial project expectations.

Everyone has expectations. It's human nature. Users often "expect that the system would do" this or that. I/S developers often "expect that the user would use the system" this way or that way. Positive expectations are important because they have tremendous motivating power. User perception has a powerful influence on the success of EHR and other systems. In one very controlled man–machine study, "the most striking finding was that perceived usefulness was 50% more influential than ease of use in determining usage"[5] of new information systems. When the user community and I/S have shared expectations, projects can embrace common goals and objectives built from those expectations. This builds the ultimate group synergy. If they do not share expectations (and it is human nature that they will not), work needs to be done to bring them together.

All expectations, good ones as well as bad ones, must be "held in common." It is imperative that gathered expectations be brought to all stakeholders as a group and discussed. Shared expectations lead to project success, so work on group consensus. This defines project purpose. Powerful project goals and objectives depend on it.

Vocabulary

COLLABORATION PRINCIPLE

Collaboration Principle: It is likely that no common language exists between the user community and the I/S staff.

We don't speak the same language. Business buzzwords and I/S buzzwords abound. It's human nature to create them and then institutionalize them. Abbreviations are the mental shorthand for repeated communications of often-used words. Particularly since the age of the computer and data automation, codes now often replace words and abbreviations because codes are numerical and computer storage "friendly." They also consume less space on printed reports.

I/S staff especially needs to be consciously aware of the vocabulary blockade. Technology does not have to be discussed in terms unnecessarily technical. Highly technical language often results in the user member of the group sitting quietly, quite lost in the discussion. Technical-oriented staff members are often not comfortable communicators. It can, however, have little to due with personality. They may likely lack the communication skills to know how to simplify their terminology. They often face several big stumbling blocks. The first is their technical terminology anchorage. Once a new and important term is learned, it can be difficult for them to find another, more relevant term from a business function perspective that would be more successful in discussion. The second is detail. Once buried in detail, it can be tough to bring yourself to a higher, conceptual explanation of that detail. Yet the project team members often need and want only the concept. A good conceptual explanation is sufficient. Ask the I/S member to explain and use analogies until the language is clear.

The user member, on the other hand, may use business jargon or abbreviations. The I/S individuals lose concentration and understanding in the midst of this jargon. If the terms are misunderstood, the I/S members can assume that they know the definition of business buzzwords, abbreviations, or codes and make faulty programming errors as a result. This can be very costly to the project and a real waste of organizational resources. Again, it is okay to ask questions and get clarification.

Vocabulary Blockade

Be proactive about vocabulary blockade. Before a group convenes, grasp the level of jargon a technical person intends to use in group settings. If you know that he or she will "lose the users" and will have difficulty in simplifying the terminology, suggest less technical terms that will get the message across without frustrating the group. After all, the listener usually will need only a conceptual understanding of the term used.

Encourage clarification through establishing and monitoring ground rules about vocabulary. "Don't interpret. Ask." should be a ground rule for all project groups. The project leaders should encourage a "no stupid questions" policy and then create a discreet procedure for stakeholders and project team members to submit questions if they are uncomfortable asking questions in a group setting. This gives team members a second, but indirect, way of obtaining clarification.

Keep It Simple

If you are in the middle of a group setting and the technical team member uses very technical words, again suggest simple terms to clear the communication early in the discussion. If that task seems too difficult for the technical team member, and no data definition role has been assigned to a team member, take a stab at your own nontechnical explanation through a simplified feedback statement. It can start as simply as, "If I understand correctly," or "I'd like to see if I could restate that in my own words." and end with a simplified explanation of the subject technology. This process clears the vocabulary blockade and keeps communications flowing. This is usually deeply appreciated by the user members and gives the technical person a clear message to simplify his or her vocabulary. Even the most complicated computer hardware configuration or network can be simplified dramatically so that users understand the basic concept.

The same holds true if you have a user member who uses lots of business buzzwords, abbreviations, or business codes. This often happens with front-line day-to-day operations personnel. Their world is very focused, and those shortened communication aids are second nature to them in their daily work. Ask what an abbreviation stands for or for a descriptive phrase or a word to help others in work groups understand. Sometimes even the manager of the department doesn't know the newest buzzwords or abbreviations. Clarification helps everyone on the team, not just the I/S members. The day-to-day operations personnel may not even know what an abbreviation or code means because they use it only in its shortened form. If so, record the term in question and get clarification after the session; then share it with the team.

Never lose the "business balance" in dealing with vocabulary. Ultimately, the system exists for business reasons, will use **business data**, and will be documented primarily in business terms. In a health care organization, this includes all aspects of the business, clinical as well as financial.

Understanding

COLLABORATION PRINCIPLE

Collaboration Principle: Much of understanding revolves around how people think. I/S people think in systems; users think in business tasks.

Have you ever heard the expression, "I never thought of it that way"? Understanding is dramatically affected by how we think. I/S people tend to think in systems. The business community tends to think in business tasks. For example, a physician may be describing the step-by-step process she uses to review critical laboratory test results and x-ray results, order additional laboratory work and pharmaceuticals, and finally refer the patient to home health care. The I/S person is interpreting her words in terms of systems—that laboratory system that feeds batch results to the clinical results reporting system twice a day that provides physicians with data to do orders that are entered at the time the laboratory or pharmacy order is placed, and so on. This different way of processing information allows varying views of the same material. Both brains need to work in their own thought process to maximize their skill sets in a successful implementation. The two styles should be encouraged. Problems can occur, however, if either type of thinking misinterprets the facts as presented by the other.

The team may be communicating through a mutually understood vocabulary but in very vague terms. Vagueness blocks understanding. Each team member can then interpret vague language to fit his or her own expectations and knowledge base. This hinders collaboration.

This is good time to discuss the differences between vagueness and concept and appropriate detail. We looked at concept versus detail in the previous discussion of vocabulary. Concept is most appropriate in "mixed" groups—those encounters where both user team members and various I/S technical team members are present. An example would be identifying the specific physical "closet" for server placement in the network architecture that will support the new system. (Because the user community may not recognize the term "closet" and "server," the technical team needs to make sure that the user community understands these terms as well.) Specific detail, rather than concept, is most appropriate in sessions where everyone in the room not only wants to but needs to understand even the smallest detail. Using the previous example, the network team and the hardware team must know which wires in which specific closet will connect which specific server to which specific workstations and which specific printers, and so forth. Vagueness is the lack of enough fact in either of the above examples to communicate your message clearly. Suppose the project manager states that the server "will be placed in a closet in the facility" in a meeting with business unit leaders. Without understanding what is entailed in a closet from a computer network perspective, the user community may mentally and immediately place the server in the basement utility closet.

A clarification feedback exercise is also a helpful activity in team meetings when not all user community representatives are always present at every meeting. For example, one user member describes the desired process one way at a meeting on Monday and is absent at Friday's meeting. Her counterpart describes it another way on Friday. By listening carefully on Friday, you catch the different description. If you have a user facilitator in the meeting, he or she should catch this. Use the notes from the meeting on Monday, maybe even a model or flowchart or diagram or some visual way to show Monday's description, to verify the discrepancy. Ask that the group stop the discussion and clarify the discrepancy. Validate that clarification with the user member who was absent from the meeting. If there is a true inconsistency of belief between users, allow the whole team to understand that the inconsistency exists and that it needs resolution before moving on with project tasks. This can also work with user understanding of computer hardware layout for the office space or navigational demands on the user through network connectivity. Get the user members to explain their understanding of the configuration or navigational steps in simple language to the I/S people responsible for the placement. An amazing amount of understanding can result through this simple feedback and clarification technique.

Understanding Process

COLLABORATION PRINCIPLE

Collaboration Principle: Team members must understand not only the content of the project and product but also, equally important, the process by which the team is obtaining the outcome.

How many times in a work effort have you heard the phrase, "I don't have a clue what I'm doing"? If that statement is indeed true, that individual does not understand the process. Often project team members clearly understand what they are working toward. They "don't have a clue," however, about how they are getting there. If the process is clearly known; clearly communicate it for understanding. The effect on team members enables them to truly perform at their very best. If the process is not known, say so. At least give guidance about continuous work effort without a well-defined process. Do not leave project teams in process limbo. As process limbo increases, project team(s) morale decreases.

A word about the unknown is critical here. Many aspects of a project can be "unknown": unknown from an operations side (for example, a new procedure around a system that has never been in place for some new business functionality) and unknown from a technical perspective (for example, new sites are involved in the project or the volume estimated in the operational work expected on the system is much greater than volumes in other systems). The unknowns are constant and predictably there. To have true understanding from all team members, including the user members, project leaders must give members of the team the courtesy of knowing what aspects of the project are "unknowns." A project phenomenon then occurs. It can build incredible team unity, a "we are all on the same, uncharted waters together" camaraderie. It also identifies vulnerabilities that the team as a whole must embrace and work to minimize or mitigate. For example, if there is real concern on the part of I/S staff related to the volume of traffic over their lines with this "additional system," team members will want to take that into consideration in planning business procedures. Understanding the vulnerability allows the team members to do this planning and act. Lacking this knowledge and understanding leaves them in the dark, possibly unknowingly planning a use or activity that will have a negative impact on the system. This sets the project teams up for unnecessary conflict and possible collaboration failure.

Knowledge

COLLABORATION PRINCIPLE

Collaboration Principle: I/S and the user community each initially have a knowledge base NOT well known to the other.

Another ground rule for projects should be "Never assume. Know." When a project starts, the user community has a knowledge base not well known to the I/S people. The I/S people have a knowledge base not well known to the user community. The knowledge that each member needs to support the project "rubs off" from other members as a natural part of communicating and understanding throughout the project's life. Each team member evolves into a teacher to the others. Each member also learns to absorb or filter out knowledge on the basis of their perceived need to know that information to function in their role.

Users should be aware of the knowledge base of today's I/S staff. It is never as good as it could be or should be because the speed and volume of today's knowledge in information technology is beyond the mental capacity of any I/S department or individual staff member. I/S staff members tend to specialize to address the amount of information and its ever-changing complexity. The I/S staffers will never know all that they could know about the desired database, operating system, network, workstations, and other components of the desired system's infrastructure. The question is, do they know enough to make solid technical decisions? Probe the rationale for their potential choices to get clues about the supporting knowledge base.

It is frustrating for user communities when the I/S staff and users disagree over the "right" component to use in the new system. It may be based on different I/S staffers gaining different bases of knowledge that lead to differing conclusions. It might be based on the same knowledge base interpreted differently. Try to understand how much knowledge plays into the I/S decision making rather than, for example, a general preference toward a vendor or operating system, a likable sales representative, or price, or some other criteria in question. One of the legitimate computer-age questions that project teams often tackle is whether to go with an "older" but well-known system or component or strike out for the new, better, but untested technology or perhaps an untested company. This dilemma will only get more complex as information technology leaps forward, rapidly outdating the technology known and used in today's health care world.

This issue comes into play when decisions are being made about what electronic health record system vendor is the safest choice.

With all that said, users must still rely on I/S's knowledge for the physical infrastructure of the system through the design, development, selection, implementation, and support stages of the project. That does not preclude, however, users self-educating about these components through their own professional journals or other educational endeavors. As users read and explore this knowledge base, it is helpful for them to use I/S people as a sounding board to discuss their learning outside of the group sessions to clarify their understanding.

I/S units have another kind of knowledge that will be used during the development of a built system or the configuration of a purchased product and as problems are found in testing the system before implementation. This same knowledge spills over to support usage after the product is put into production. The I/S staff has a ready body of knowledge to draw on to "problem solve." When it could be "one of 10 possibilities," they will need to work their way through their problem-solving triage and knowledge bank to find the problem and then resolve it. There is no question that problem solving today is far more complicated than it was in the mainframe computer days when limited and isolated applications that rarely talked to other systems ran on a "tried and true" operating system.

The user community has no need to share this knowledge bank in detail. However, if a user is involved in experiencing the problem, he or she should provide specific information to the I/S staff. Users will want to understand the general cause of the problem so that they can learn how to avoid the problem in the future, particularly if is it a "user error" problem. Users will also want to know the solution and have clear directions for what they should do if it occurs again. If triaging the problem and reaching solution involves "**downtime**" (time when the computer system is not available to the user community), that community also has a need and a right to know whether the resolution process could take an hour or several days. "Stretching" users from a promise of an hour that becomes several days of "downtime" destroys trust, although initially it may keep them quiet. The short-term peace is not worth the long-term price.

THE TIDE TURNS

What becomes an interesting twist is the knowledge advantage the user community gains over the I/S people as the system is implemented and then used over time. No one will know that system better than the user. Although I/S people will initially know the system better than the users do, they will become hard pressed to keep up with the learning curve of users who have incorporated the EHR system and other systems into their daily lives.

Users can get frustrated by the tendency of the I/S staff members to initially label reported bugs or problems found by the users as "user error" before true knowledge of the problem is unraveled. This is a dangerous practice because it can often be wrong—perhaps it is a programming error, network connectivity error, software incapability error, a client–server communications error, and so on. It also destroys mutual respect because the users feel they are branded as incompetent in the eyes of I/S.

If you have a shared and well-defined proposal, built on relevant policies and principles that has been validated and driven by users and I/S staff who have a personal commitment to the **project** and its people on the basis of solid trust and respect that has solidified expectations through the beginnings of a common vocabulary and understanding of departmental knowledge bases, jump on this EHR project. We now have the environmental basics for moving ahead with collaborating EHR projects successfully.

PROCESS MODELING

Remember that the four components we must integrate in system design are technology, people, data, and business process. Events in a business environment trigger various kinds of transaction processing. Process has hierarchy too, much like data, to systems-thinking people. Process hierarchy has been identified as function, process, subprocess, and procedure. In an EHR system and in other systems, several challenges will need attention in modeling current process. First, workflow inconsistencies are possible if two or more persons perform the same subprocess. These inconsistencies may be unknown to the user manager. Second, there may be multiple versions of the same subprocess within the business unit or whole organization, all sanctioned by multiple user managers. Documentation of "variance," sanctioned or unsanctioned, is important for understanding the old way of doing things and in planning and implementing new systems.

Although we are documenting the current work activity, this is a critical step in new system design. Some argue that focus on the "old" fosters a "stuck in a box" approach when they want a "new" visionary path. This visualization of current activity is detailed "baseline data," which becomes invaluable for (1) doing "what if" exercises to analyze the business impact of the new system in system design and implementation planning, (2) a powerful aid in discussions with vendors when buying a new EHR system that will be "retrofitted" to the business environment, and (3) a true measuring stick for "benefits" of new system once implemented. These models should be sanctioned by the business community before becoming working tools for new system design, development, and implementation. User team members are responsible for and may require the user member or manager to muster resources quickly and without fanfare.

Users may need to get creative now. Sometimes solutions lay in choosing among a technical fix, a software fix, or a people fix. For every system created or purchased, there are likely multiple "constraints" involved in implementation that will have a business impact. If environmental basics are solid, project teams and departments can work through constraints to a solution.

User Processes and Procedures

Users have plans for what operational workflow will likely change with the introduction of the new system. It is now time to review these plans and the system development and identify any changes to the plans. I/S staff will have little to do with this task. A user or group of users will take on the sole responsibility of "process owner." It should be done in concert with the systems team members configuring the system and defining the database load. Include here direct business activities and also any system support activities that have been delegated to user's responsibility. This is a huge job, but it is dramatically foreshortened with the planning aid. Poor process planning is generally one weak area for the user community, who do not comprehend the impact on implementation without carefully defined and orchestrated new workflow. User members should understand that the first go-round will not necessarily be totally functional. Thus, part of implementation is the iterative refinement of the workflow. It is also a helpful message to the user community who may be nervous about the impending implementation.

Discussion Scenario: An outpatient behavioral health clinic requires clients/patients to sign a consent for medication for each new prescription medication. Signing the consent form demonstrates the client's participation in the decision and is a reminder trigger to share current information about the medication, its positive effects and potential side effects. Signing the consent is a paper and pen process. Medical team members who prescribe medications (physicians, advanced registered nurse practitioners, and physician's assistants) keep paper consent forms in their offices and use them as needed. Typically, the client is able to sign the consent form during the clinic visit. Clients who are children or who have an appointed guardian typically must have a parent or guardian sign the consent form. The consent form is scanned into the clinical record. Because the organization has an electronic health record and wants to reduce the number of scanned documents, the HIM staff has suggested that the organization use digital signature pads. The digital signature pad can be used for several applications but beginning with the medication consent form allows a small group to try this technology before it is used for a variety of other applications in the organization.

Discuss the current and future work flows involved. How do before and after flow charts differ? What are the potential benefits and potential costs?

Plan Transition

Teams should know what the business unit is doing now and what is expected when the new system is in place, but the spatial move between systems can be wrought with bumps. The conversion from one system to another should be planned for a time least disruptive to the business operations. It may be a time, however, when other unautomated but important system rollouts are occurring. It may involve temporary staffing and their logistics or automated work activity to greatly foreshorten the task. Seasoned veterans of implementations are great resources for planning this work. User members need to work closely with the planning staff and the user managers to smooth the conversion path.

Users not only define the implementation process, they completely drive it. Do you want to implement the system throughout the organization, department, or unit, or "phase it in"? If it is phase-in, who goes first and why? Do others agree with that rationale? If the system is to be used by many disciplines, do you start with one, or a handful of many, and then spread out the implementation on the basis of your ability to train the users? When is the best time to implement? What other activities are occurring that might interfere with the implementation? Will you implement one module or the whole system at once? Is there space for more hardware in work areas? When can training be scheduled? How will you get appropriate security access for the system to the users? Users must drive these decisions to ensure workflow continuity.

Users will get lots of feedback from existing clients interviews and site visits about these decisions, often conflicting styles of implementation and conflicting opinions about their success or failure. For example, one client will have used only "champions" to begin implementation. Another may have used a mix of those computer proficient and those computer illiterate or those initially trained. The user member must be comfortable with decision making amid conflicting information from the past implementation experiences of others.

Training

The critical task of training is usually underfunded, understaffed, underscheduled, and often only concentrates on system functionality, forgetting process. Users are not interested, nor should they be asked, to separate learning the new system from learning the operational processes in place to support the new system. This combination approach (new system supported by new processes) to training takes more energy if you buy a system because vendor manuals will be geared only to the system's features and functionality. When flexibility is part of the system's design, the flexibility will be discussed at length in the vendor's training materials. The chosen way the organization specifically configured the system to best meet the business needs of that particular organization will not be included. This leads to unnecessary system functionality information directed at the trainees, and the trainee may be lost or confused about how he or she is expected to take the new system and apply it to daily work activities. In other words, the training materials sold by the vendor must be applied to each client's environment.

Often user members on design and development teams make good trainers. The training plan may call for "train the trainer," in which case system developers or vendor trainers may train the user trainer who will go forth and train the rest of the user community. This is tiring, repetitive work that can be frustrating if users are not grasping the important concepts or seem resistant to change. Sometimes trainers train the user community directly.

Lots of questions will need to be answered in planning training. How much is enough? Who needs it when? How will you backfill your staff with other employees who can do the work during the staff's training time? What prerequisites does the user community need? Do you need to survey their skills and knowledge? If the system is purchased, did you "buy" training? If so, what training did you buy—train the trainer, train the users, and so forth? Does the vendor have a trainer? Do vendor's training materials—training agenda, training manual, goals, objectives, and evaluations—exist? Are they appropriate for users' needs? Who is going to put documented processes and procedures that support the new system into the training materials and curriculum? If vendor training materials do not exist (often the vendor offers only one "manual," which may not be suitable for training), then how do you get these materials? If you built the system, who will produce the training manual and training curriculum? Where can you do training and when? Is a training room temporary, or can it serve continuing training needs? How can you "institutionalize" training and system trainers after implementation? How can you get a practice time and place for users between training and implementation and long after?

A word here about a **training manual** versus a **user's manual**. They are different, and they have different goals, as depicted in Table 9-3.

A training manual is intended to train: to teach concepts that serve as background to hands-on exercises that teach system functionality and outcomes. A user's manual should be a complete textural documentation of the system that includes all system features and functionality, complete rationale behind use of particular features, complete picture of system usage within organization or user environment, solutions to potential problems with usage, sources of help for users, and so forth. One does not replace the other. Part of the plans for training entail securing (or creating) a useful training manual.

Try orienting a few users to the new system and then let them determine what would most help others in learning the system. Test any newly created materials on a few other users to gain valuable feedback. "Cheat sheets" are helpful in the initial training and in the first month or two of system implementation. They come in many flavors and forms.

Prepare User Manual(s)

If you build the system, you are automatically forced to develop a user's manual that serves as a reference to system functionality, navigation, and work flow to support the system. The user member may be asked to document the designed front-end user features and functionality of the system, whereas other members of the team write other appropriate chapters of the documentation. This will become the bible for the system, so this work must be done to reflect the user's view accurately. It then needs to be updated when system changes occur.

If you buy the system, you will need to review carefully the user's manual that is sold with the system for accuracy, completeness, and ease of use. It will only describe the features and functions of the system. It will not describe any of the processes the user community has put in place to support the new system. This will need to be added before training so that workflow can be learned along with the new system. Include new work-flow documentation in the user's manual, which will serve as a reference guide to the system after it is implemented.

An important consideration for training is to tailor it to the specific user community and the criticality of the application.

Table 9-3 TRAINING VERSUS USER (REFERENCE) MANUAL

Training Manual	User (Reference) Manual
Training is objective	System knowledge reference is objective
Focuses on select, critical system functions	Encompasses documentation of entire system
Exposes trainees to select system features critical to usage of this unique application	Defines and describes all system features, which are available in the system design
Limits explanation of system logic to subject area being taught	Explanation of full system logic incorporated into format and text
Incorporates the adult learning principles of discuss, demonstrate, do, and evaluate	Lacks "exercises" and evaluation components in manual format
Outcome is satisfactory work performance	Outcome is self help in problem solving while accomplishing work tasks

Major EHR implementations may deploy trainers as "coaches" who are readily available at the clinic sites so that the care process documentation can occur safely. A key example of this is were computerized provider order entry (CPOE) is implemented and the entire medical staff must "go live" at the same time or on an aggressive schedule. Offering coaches for a 24×7 training support program facilitates the application's implementation and supports the clinical users in the change-over experience.

Data Migration

If the decision has been made that data from an old automated system will migrate to the new system, programs must be written to accommodate this. Business units decide the need for such action and the structure of the data through migration, and they influence the timing of the migration. They then need to test for data availability and integrity. This data mapping activity forces users to assess the value of data in legacy systems.

System Testing

System testing is a BIG DEAL. It is often called the "user acceptance test," and it means just that. If this is an EHR application, I/S staff will set up a "test database" that consists of the full system features and functions complete with dummy patients supplied by users. This test database is isolated from the "production" database and made available to selected users who add fake data and then test the system's features and functionality. Performance is measured against test plan criteria established earlier. Does it work like you expected? Are you getting "strange messages"? What does it not do that you thought it would? System testing will also occur when a system is being upgraded. Users should actively participate in the testing process for initial and ongoing implementations. Just as business process scenarios add strength to vendor evaluation earlier in the process, the careful provision of "test' scenarios strengthen the testing experience.

"Go Live"

Implementation is a time to be highly organized because things happen quickly. A response plan aids in focusing on the immediate tasks at hand. Identify what roles team members will play at "go live." Clearly and completely communicate to the user community the series of implementation events that are about to occur. Implementation teams can expect to do lots of hand holding on the initial days of "go live." There is a natural learning curve to every system, every user, and every I/S staff member involved in implementation. Everyone is learning: the user community, all of the I/S staff, the vendor representatives (if the system was purchased), everyone.

This is the time when the "unknowns" strike as problems and test the teams that designed, developed, and implemented the system for their patience and endurance. If you are supporting the system in any way during this time, you will be "going the extra mile"—probably putting in long hours in problem solving, helping users navigate through difficulties, supporting the user community through technical "snags," and so forth. It is an exciting and tiring time, and tempers can fly. Draw heavily on the environmental basics to sustain the teams and add humor whenever possible. Keep track of problems and issues that arise by using a formal log; these could need immediate attention or could be fodder for system improvement or phase II work.

The implementation will not happen perfectly. The odds are against perfection with so many new variables in the pot. Be prepared to document and problem solve for at least the first several months of system usage. Keep a log that serves to track the problem, triage, and solution status because the volume identified soon overwhelms the human brain. Problem solving after "go live" is a continuation of major testing of the environmental basics in systems work.

SYSTEM AND PROJECT EVALUATION

In all the efforts that led up to system implementation, you should have accumulated some real figures to measure change or new business impact. Use these as a baseline to determine system implementation success or failure. Ask whether the system implementation is the only factor that has caused the change. Users will likely be asked to measure against these baselines because much of the data will be coming directly from their units. Once the changes to baselines are measured, it is time to review the implementation. Was it a success or failure? What made it so? If problems exist, can and should they be corrected now? Can some things wait until an identified phase II or upgrade?

For example, the EHR system evaluation should evaluate how the system performs against both strategic values of the organization and value benchmarks.[6] Once electronic systems are in place, the sharing of episodes of care enhance the quality and continuity of care. The benefit may be stronger in ambulatory settings because of the longitudinal nature of care; however, all care settings should benefit. A number of opportunities to measure improvements are listed here:

- Improved access to medical history, allergies, and medications should reduce the number of adverse reactions from treatment.

this. The finalized model should be documented in writing, possibly by using project method tools including visuals.

After the current system is documented, the workflow for the new system is designed. User members should be visionary people, and the team must have the new data defined for the new system to guide new process definition. This will be a conceptual model of process that will change as constraints and opportunities present themselves down the development path as teams take the system design documents and breathe life into them. In fact, using scenarios to describe the basic business processes brings concrete analysis to evaluating vendor products. Health care enterprise decision makers need to provide enough clear descriptions for vendors to fully explain just how their product will be able to perform the processes in question. Investing in the "what if" approach through scenarios is worth the time and effort required.

New, unique system-oriented processes may be handled by the user community as the system is implemented and used. These might include system backup, system table maintenance, data purges and archiving, and so on. Today as computer systems become "friendlier," more and more system maintenance/support is being designed to be handled by the users—those who have the most stake in system reliability and integrity. Traditionally, these processes have been completed by computer operations/support staff.

Again, user members need to obtain user community acceptance of the conceptual workflow for the new system. The approval should be documented in writing, possibly through some project method document including visuals. Expect multiple queries from front-end users at this time. Suddenly, they begin to understand the system plan and start assessing the personal impact because the proposed workflow directly includes them. They start seeing a living system rather than just plans. It is important for user members to work with user managers to address and honestly answer these queries now. Be honest about the unknowns: there are many of them in this design or conceptual stage of projects.

SELF-ASSESSMENT

1. What are the elements of process modeling?
2. Can you describe an example that uses all of the elements of the process model?

DATABASE DESIGN

We now move from design activities to development activities. The first of these is often the database design. Database designers and administrators take the conceptual model of data turned into a logical model for the new system and plan the physical database. The database administrator will likely plan the physical database somehow isolated from the user members, appearing infrequently to clarify or possibly to determine whether he or she might structure the database one way or the other, depending on the user's response to the database administrator's question about operations impact. This "plan" is usually a textual listing of database tables with "keys" (unique identifiers for the each of the subject tables) and component data and identified physical storage mode and location. This is not for user community usage. However, the user members of the teams that identified the desired data should understand and indeed formally accept the proposed physical structure of the new database. This requires a clear understanding of the proposed database presented by I/S staff.

After the database plan is finalized, the database administrator builds it as a foundation for system feature and functionality. The programmers can dig in with actual screen planning and design, menu development, report planning and creation, and so forth.

The user team members usually enjoy this task. This is where all their prior work starts to congeal. They will have defined their data and business rules, anticipated, and flowcharted proposed new processes. If the "mock screen" technique was used in any of these earlier tasks, the users and the designers probably have a "gut feeling" for how many screens may work, how the screens would flow through the system, how much of the data would be chosen from "pop-up tables" versus entered as "free-form text," and so on. Find creative but patient people for the job—often a dichotomy. This task can be time-consuming.

Reach consensus on "seed" screens to be used in screen design sessions; there is no limit on number or versions. Screen ideas can be dropped quickly in these sessions, so multiple versions are actually a good idea. If the automated EHR has a data entry staff, plan some design sessions for data entry personnel only. If the system has a large number of data viewers from many departments or units, include them in design sessions. If the system has a lot of report functionality, do report design sessions that include supervisors and managers who request and read them.

Iterative design is the name of the game here. Avoid long time spans between sessions that allow (1) the programmers to get too far into a full-screen design and (2) the users too long to forget their prior creativity.

IMPLEMENTATION READINESS

System development will never go quite as project managers and teams expect. "Challenges" will surface as the teams attempt to "put it all together." A detail-oriented problem solver from the user community is the best choice to work through "challenges." This task can also be time-consuming

- Improved access should reduce the amount of time spent in obtaining and reviewing patient history before ordering diagnostics or treatment, which should shorten the time it takes to provide care and should reduce the cost of care.
- Improved access should allow caregivers to place orders that are legible, timely, and screened for possible errors, which should result in fewer errors in the treatment process.
- Improved access to patient information should reduce the need to delay seeing drop-in or emergency care patients until a record is obtained from current or remote storage, which should reduce the amount of time patients spend waiting for this information to be available to the treating clinician.
- System prompts and reminders should reduce the possibility of adverse reactions to care.
- Use of electronic and embedded disease protocols should result in consistent comparison to an agreed on standard of care.
- Use of online formularies should save time for the ordering caregiver and result in lower medication costs.
- Use of electronic clinical decision support systems should improve the efficiency and effectiveness of care.
- Use of electronic systems should result in standardization of terms and data, which should allow comparison of both the quality and cost of care.
- Legibility is improved, which should reduce medication errors.
- Laboratory technicians can check and see when a medication was ordered/administered, which should allow better timing of specimen collections.
- Caregivers can complete their documentation without obtaining access to a paper record, which should allow more timely completion of documentation.
- Administrative access to clinical records for billing, coding, and release of information can be completed without access to a paper record, which should allow more timely completion of these functions.

Use of a value-based approach will help an organization assess both the financial and more intangible values associated with a major system, such as the EHR system (Table 9-4).

Use of value benchmarks will give an organization a broader view of measuring the potential value of an EHR system (Table 9-5). If the system is realizing a value benefit that translates into cost reduction, the most operationally valid benchmark is the "average cost per case."[4] Value benchmarks can help measure the achievement of objectives and expectations for an EHR system.

It is safe to say the quest for new and better ways to conduct business is an eternal one that continually requires collaboration between the user community and the I/S staff. Out of implementation and usage will come new desires for improving the system. This starts the project cycle all over again in search of new and better information systems.

Go to the Evolve site and complete the Chapter Review questions for this chapter.

Table 9-4 ELECTRONIC HEALTH RECORD: STRATEGIC AND OPERATIONAL VALUES

Strategic Value	Operational Value
Improved patient satisfaction	Faster registration
Improved clinician satisfaction	Improved work flow
Improved data resources	Elimination of data entry
Better and safer patient care	Elimination of duplicate data storage
Enhanced decision support capabilities	Reduction of identification and billing errors
Improved ability to integrate into community care systems	Greater access to complete and accurate information
Improved standardization of information resource	Direct entry of orders by responsible clinician
	Clinical reminders and alerts
	Access to new sources of medical information
	Enhanced development and use of multidisciplinary clinical pathways/ protocols
	Activity-based costing
	Standardization of quality of care measures
	Outcomes measurement

Integral components: Master Patient Index (MPI), Electronic Health/Patient Record (EHR), Clinical Data Repository (CDR).

Table 9-5 SAMPLE VALUE BENCHMARKS

Opportunity	Basis	Benchmark
	Pharmacy	
FTE reduction	Improved efficiency	FTE/thousand pharmacy orders
Error reduction	Bar-coded inventory	Dispensing errors/per thousand orders
Improved drug utilization	Limited formulary	% Reduction in formulary
	Decrease in drug costs	Drug cost/patient day
		Drug cost/thousand covered lives
	Enterprise Master Patient Index	
Reduction in denied claims	Improved eligibility verification	% Claims denied
		FTE/claim
Dollars saved	Improved cash flow	Days in AR
Increased revenue	Income growth from improved service	% Increase in net income
	Enterprise Scheduling	
FTE reduction	Improved efficiency	Transactions/FTE
Increased revenue	Reduced no-shows and cancellations, improved service	% Increase in net income
	Clinical documentation	
Overtime reduction	Decreased shift overlap	% Nursing overtime hours
Full-time equivalent reduction	Improved efficiency	Nursing FTEs/AOB
		Nursing FTEs/thousand covered lives
	Health information management	
FTE reduction	Reduced manual chart processing	HIM FTEs/AOB
	Reduced storage costs	Total paper charts stored
		Health information management FTEs/thousand covered lives
	Order/Results Management With Decision Support	
Reduced test volume	Guided ordering/protocols	Tests/unit of service
Reduced morbidity	Clinical rules	Adverse medical events
Reduction in length of stay	Results flags/alerts	Average length of stay
	Enterprise-wide benchmarks	
Improved efficiency	Integrated systems	Cost/unit of service
		Cost/diagnosis-related group
		Cost/pathway

Adapted from Murphy G, Hanken MA, Waters K: *The electronic health record: changing the vision,* Philadelphia, 1999, Saunders/Elsevier.
FTE, full-time equivalent; AR, accounts receivable; AOB, adjusted occupied bed.

REFERENCES

1. *Webster's new twentieth century dictionary of the English language,* ed 2, unabridged, New York, 1979, William Collins.
2. Goverman I: Planning for success. Presented at the Health Information Management Systems Society Northwest winter conference, January 23, 1997, Seattle, WA.
3. Kloss L: The politics of information system integration, *JAHIMA* 68:18, 1997.
4. Murphy G, Anderson E: An organizational model for data access and management—work in progress, *JAHIMA* 65:52, 1994.
5. Davis F: User acceptance of information technology: system characteristics, user perception and behavioral impacts, *Int J Man Machine Stud* 38:475–487, 1993.
6. Murphy G, Hanken MA, Waters K: *The electronic health record: changing the vision,* Philadelphia, 1999, Saunders/Elsevier.
7. Ciotti V: Applying TQM/CQI principles to information system selection, *Healthcare Fin Man* 49:56, 1995.

RESOURCES

Whitten J, Bentley L: *Systems analysis and design methods,* New York, 2007, McGraw-Hill.