

CHAPTER 9

Developing Information Systems

LEARNING OBJECTIVES

After reading this chapter, you will be able to answer the following questions:

1. How does developing new information systems produce organizational change?
2. What are the core activities in the information systems development process?
3. What are the principal methodologies for modeling and designing information systems?
4. What are the alternative methods for building information systems?
5. What are new approaches for systems development in the digital firm era?

OPENING CASE: PROTECTING PATIENTS BY TRACKING INSTRUMENTS

The opening case describes how and why the Ottawa Hospital (TOH) moved to a computer-based tracking system for medical instruments. This ensures that instruments are properly sterilized, stored, assembled, maintained and are available when needed for medical procedures.

The system also allows the hospital to budget and track the costs to maintain the instruments. The case illustrates the improvements that such systems can provide both in terms of function (better health care) and efficiency (better accounting). It also provides an example of how the system was acquired and the need for the vendor and the hospital staff to work together.

9.1

SYSTEMS AS A PLANNED ORGANIZATIONAL CHANGE

Change is difficult for people and organizations. But it is necessary to keep companies in the lead. In this chapter, we're going to focus on using information systems as a way to

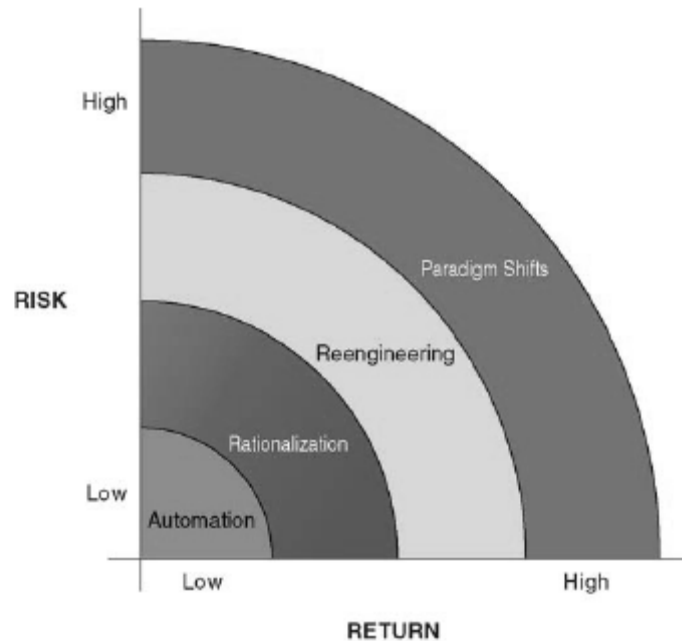
successfully help redesign organizations so they can improve their current processes or establish new ones.

SYSTEMS DEVELOPMENT AND ORGANIZATIONAL CHANGE

This section provides students with an introduction to the methodology of redesigning the organization. Although some of the techniques and methods are considered older or less satisfactory, students have a good chance of seeing these methods in the workplace. It is not a good idea to engage in a lengthy, philosophical discussion of the superiority of methods. However, it is a good idea to help students understand that there is no one best method, that each method is appropriate depending upon the situation and the requirements. You should remind students of the largest problems: It takes too long to develop systems, and they often do not work as intended. Of course, this is not always true, but developing systems is difficult and labour intensive. The point is to familiarize students with methods, which after all, are certainly better than no formal methods at all.

Too many companies buy the hardware they think is necessary for a new or improved information system. Then they purchase some software to go along with the new hardware. Now they realize their hardware is inadequate for the new software, so they buy more powerful hardware. And the vicious circle continues. Pretty soon they have a whole bunch of hardware and a lot of expensive software, but do they have an information system? Only if they have made sure all the hardware and software purchases fit in with their organizational *information systems plan* and their people know how to use them.

Of course, the information plan should support the overall business plan and not conflict with it. The plan must include all levels of the organization, including the strategic and executive levels. These two levels contain the people who often say they are exempt from having to determine information system needs.



The most common forms of organizational change are automation and rationalization. These relatively slow-moving and slow-changing strategies present modest returns but little risk. Faster and more comprehensive change—such as reengineering and paradigm shifts—carries high rewards but offers substantial chances of failure.

This figure shows the four degrees of organizational change. **Automation** is the easiest for the organization and the most common form of change. But that doesn't mean you don't have to plan for the change first.

Rationalization of procedures causes the organization to examine its standard operating procedures, eliminate those no longer needed, and make the organization more efficient.

Both types of change cause some disruption, but it's usually manageable and relatively accepted by the people.

Business process reengineering, on the other hand, can cause radical disruption. The mere mention of the term nowadays strikes fear in the hearts of workers and managers at all levels. Why? Because many companies use it as a guise for downsizing the organization and laying off workers. Business process reengineering causes planners to completely *rethink* the flow of work, how the work will be accomplished, and how costs can be reduced by eliminating unnecessary work and workers.

But if you want to talk radical change, take a look at **paradigm shifts**. This changes the very nature of the business and the structure of the organization itself.

An example of a paradigm shift is higher education, which is undergoing a major paradigm shift in the online delivery of education. Classes are now offered through the Internet so that students don't even go to classrooms. Many tried-and-true teaching

methodologies are being radically altered to accommodate this shift in how education is offered.

The Internet is causing all kinds of industries and businesses to alter their products, their services, and their processes in radical ways. Entire organizations are being created to handle the paradigm shifts involved in e-commerce. Look at the automobile industry as an example of this type of change: Traditional dealerships are being disrupted by automalls and online buying opportunities. How can a local dealer compete on price with these two environmental challenges? What is the dealers' role in the revolutionary changes taking place all around them?

If business process reengineering and paradigm shifting are so disruptive and so dangerous, why even try to do them? Because companies realize they have to take on the challenges in order to stay competitive. They have had to cut costs and streamline their operations because of global economic pressures, in addition to meeting the demands of their shareholders. And done well, the rewards can be tremendous.

BUSINESS PROCESS REENGINEERING

Change certainly should be a theme throughout the entire course. It takes a change agent to carry out major changes, such as business process reengineering. Business process reengineering is a good example of the problems that information systems can face. First, people react negatively to what they may view as a buzzword. Secondly, business process reengineering implies, probably correctly, that the organization is either not doing things correctly or that the environment has changed and old ways of working do not work anymore. It takes some crises or changes in the environment to induce or engender the call for reengineering. ERP is often the computer enterprise incarnation of business process engineering.

In order to make BPR successful, you must first redesign the process, and then apply computing power to the new processes. If problems existed in the process before the new system was installed and those problems aren't resolved, the new system could actually make them worse.

Very few processes in business are as efficient as they can possibly be. It's a fact of life. The idea behind successful BPR is to find improvements or even new opportunities. For instance, ING Direct and Scotiabank had to reengineer their business processes to incorporate this new paradigm shift brought about by the use of the Internet.

New information system software is giving businesses the methodology to redesign their processes. **Work flow management** offers the opportunity to streamline procedures for companies whose primary business is oriented toward paperwork. Instead of 10 people handling a single bank loan application, you can install software that will speed up the process, allow several people to work on the document at the same time, and decrease the total number of people who handle it. Or, you can migrate the application process to the Web and make it even more efficient and customer-friendly.

Bottom Line: Continual change is a necessary part of corporate life. Managing organizational information requirements through planned analyses and structured system development rather than a haphazard approach will help you succeed.

Steps in Effective Reengineering

BPR attempts fail 70 percent of the time. That's an astonishing figure when you think about it. Some of the reasons for the high failure rate are lack of planning, management's inability to fully comprehend the enormity and complexity of the effort, and the fact that BPR usually takes much longer than expected.

What can organizations, their managers, and workers do to help make BPR a success? It may be helpful to make a diagram of how your processes work now and then envision how they will work after they are redesigned. The textbook provides a flow of steps that can be taken to improve the chances of success. You can use these steps to analyze a situation with which students are familiar, registration or applying for on-campus work, for example.

PROCESS IMPROVEMENT: BUSINESS PROCESS MANAGEMENT, TOTAL QUALITY MANAGEMENT, AND SIX SIGMA

Business Process Management

Take a seemingly simple task such as sending out customer invoices and really analyze how many steps are involved in the process. Even in a small business, you may be surprised how many steps there are. **Business process management (BPM)** is the art and science of analyzing every task in a business and helping firms continually optimize them. BPM includes work flow management, business process modelling, quality management, change management, and standardizing processes throughout the organization. Every business, from the smallest to the largest should continually analyze how they accomplish every task and look for ways to improve everything. The business doesn't have to accomplish this with the idea that every process should be automated even though many can. The business simply has to continually look for better methods of performing the work.

Total Quality Management and Six Sigma

Total quality management (TQM), making quality control everyone's responsibility, relies on an excellent information system to supply workers and management with the data necessary to improve products and drive down costs. **Six Sigma** is another initiative companies use to spot problems and correct them before they are too deeply embedded in the company's processes. It provides organizations of a goal of no more than 3.4 defects per million opportunities.

The lack of good, useful information may not be apparent until the organization can't figure out what it's doing wrong, or doing right. Data from all the types of information systems we discussed can be fed into quality management and make it easier to develop and improve products that blow away the competition.

How Information Systems Support Quality Improvements

Here are some ways companies can use information systems to achieve total quality management:

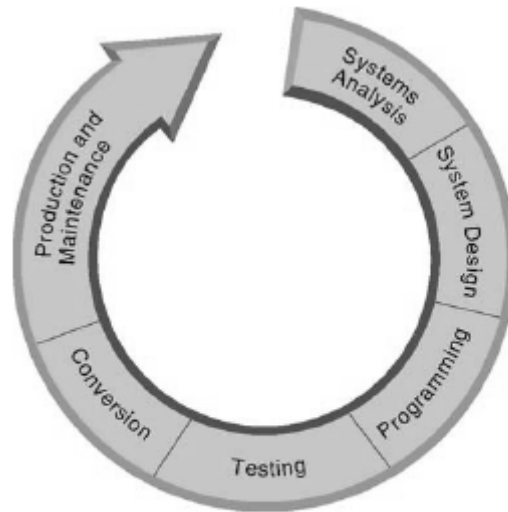
- Simplify processes by using information to determine what the processes are in the first place
- Identify **benchmarking** targets
- Gather, process, and store customer feedback in information systems that are available company-wide
- Reduce cycle time by providing information earlier in the process
- Redesign the process or redesign the product by using information about the process
- Improve production processes by using available information from internal and external sources

Bottom Line: The quality of a company and a product can be improved through the reliable, useful information produced by a well-developed, well-managed and integrated information system.

9.2

OVERVIEW OF SYSTEMS DEVELOPMENT

Systems development includes every resource and every step that goes into producing an information system that solves a problem or helps the organization take advantage of new opportunities.



Building a system can be broken down into six core activities.

The arrow in the figure goes in only one direction. Remember that just because you apparently completed one step doesn't mean you can never go back to it at some point in the development process. In fact, many of the steps should be revisited several times, especially if you are using the prototype method of development.

SYSTEMS ANALYSIS

You have to analyze the current situation to determine the real cause of the problem. Make sure you're addressing the real problem and not just the symptoms. Effective **systems analysis**, adequately determining the real problem, is the key.

Write down everything you do in this stage, especially when it comes to what the real problem or opportunity is. Constantly review it throughout the rest of the system development process to remind you and others of what you're trying to do and where you're trying to go. It's natural to stray from the path! Most of all, determine how your objective fits in with the rest of the current information systems and the business plan itself.

Is your idea even feasible? You might be surprised how often organizations fail to ask this question. A **feasibility study** helps you determine if your proposed solution is achievable *before* you spend thousands of dollars. The study will review the technical feasibility of hardware, software, and people when you're deciding whether your proposed answer is the right one. Too often organizations underestimate the cost of a new system, especially in the human resource area: training, downtime, lost productivity, and employee disruption. The feasibility study will help determine the economic feasibility of the idea.

Something's got to give with the new system. What are the changes, who will manage them, and how will they be incorporated into the existing organizational structure? How much change can the organization handle, monetarily and emotionally? Those are the questions the operational feasibility study will help you answer.

Establishing Information Requirements

Figuring out who needs what information, where, when, and how will challenge the political dynamics of any organization. No system can answer every need, so you're going to have to decide who gets what. That's why you must write down the problem and then keep referring to your notes throughout the development process. It is too easy to get sidetracked by politics.

You must *think* and then *rethink* the proposed solution. Make sure you thoroughly investigate the **information requirements** — you're going to live or die by the outcome. Whatever happens at this stage will carry through to all the other stages.

The final dilemma is whether a new information system is really the answer. Would it be better to address the problem through management changes, more training, or changing existing organizational processes?

SYSTEMS DESIGN

In the **systems design** stage, you can get down to figuring out how the system will actually solve the problem or help you take advantage of new opportunities. Remember, your goal is to fit the system into the organization and not make the organization fit the new system. Or at least you want to keep them in tandem; that is, the organization should decide what technology is necessary, while the system capabilities can help reshape the organization.

When we discussed database management systems, we distinguished between two methods of viewing data: the physical design (how the data would actually be stored in the system) and the logical design (how the data would look to the user). Use the same definitions when you are designing your system, and concentrate on the logical design. In addition to elements that the authors point out in the text, the physical design should determine how the new system will support the current organizational structure, or spell out the changes in that structure that will successfully integrate the new system.

The Role of End Users

Unfortunately, the physical design sometimes overrides the logical design. This is a reminder that both sides have to work together, keeping the goals of the system as the number one priority and remembering that the best system is one that meets the user's needs.

Don't forget that people are the most important component of any system. As soon as users begin to feel they have little input into the development process, you are courting disaster. Keeping the end user involved will produce a better system.

COMPLETING THE SYSTEMS DEVELOPMENT PROCESS

Now that you're through the analysis and design phases, you can move on to the remaining steps in the process. You can always go back to those two steps and probably should at some point.

Programming

The actual **programming** phase will in all likelihood be carried out by the IT department. If you're using a fourth-generation language, the programming could very well be done by the end user. Either way, make sure that the programming supports the analysis and design phases. If not, go back and work through them again. It could very well be that what was designed simply can't be programmed. The usual impulse is to program around the design flaws. Don't! Redesign instead.

Testing

Exhaustive and thorough **testing** must be conducted to ascertain whether the system produces the right results. Several things that go wrong in this phase of the development process can severely hamper the project's success. .

For one thing, this step is glossed over by both techies and non-techies. People assume that because something was designed and programmed according to the specifications in the analysis stage, it is right. So they just fly right through the testing process. Or they run one or two tests, usually by the very people who designed and programmed the system. You should never have the people who were involved with the design and programming stages do all the testing. Get a fresh pair of eyes to look at the system according to the **test plan** that was developed by the programmers and the users.

Most of all, if you do find a flaw in the testing, do not give into the temptation to ignore it or explain it away. Go back to the analysis, design, and programming stages. Get rid of the flaw the right way.

Of the three types explained in the book, **unit testing**, **system testing**, and **acceptance testing**, the last is the one that is most important and yet the most underrated. Managers and users must be adamant about testing the system, measuring it against the analysis and design requirements, and then accepting the system only when it does in fact measure up.

Conversion

Conversion is the process of changing from the old system to the new system. There is no right way or wrong way to implement the system; you have to look at it in the context of your particular organization.

- You can use the **parallel strategy**, but it's expensive to run two separate systems at one time. If you don't have a lot of confidence in your new system, you might want to go with this one.

- If you're really confident in your development process or if the old system simply doesn't work any more, you can use the **direct cutover** strategy. For instance, Friday you're using the old system; come Monday you're using the new one.
- If neither of the above describes your organization or your new information system, you might want to consider the **pilot study** strategy. You can introduce the new system into a single area of the organization. If all goes well there, you can install the new system in other areas. You're still going to have to figure out how to run two systems at once and also figure out how to integrate the new system with the old system.
- The **phased approach** is similar to the pilot strategy, but now you install parts of the new system slowly into specific areas of the organization. Again, two systems, two methods, integration problems, support problems, etc.

However you convert, make sure everyone knows what's going on. Tell them through **documentation** of a formal conversion plan and not the grapevine. Use the information you gathered in the earlier stages of the development process to help guide the implementation plan. Make sure you figure out how to convert the data and train the users. User resistance through fear of the unknown can destroy all your hard work and planning.

Production and Maintenance

After you install the new system and it's in **production**, you want to go back one more time and make sure it's meeting your needs through a **post implementation audit**. Eventually you're going to have to perform **maintenance** on the system no matter how well you designed and built it. And someday you will have to make major changes or replace it altogether.

WINDOW ON ORGANIZATIONS: DORFMAN PACIFIC ROLLS OUT A NEW WIRELESS WAREHOUSE

TO THINK ABOUT QUESTIONS

1. Compare Dorfman Pacific's old and new order-picking processes. Diagram the processes.

The old system: The old process relied on paper-based processes and tacit knowledge of the facility and the company's customers. The company's IT systems were spread out over various functional areas and did little to support a transparent inventory. The warehouse worker, called a picker, received a paper pick ticket from a supervisor. The picker drove a forklift to the area of the warehouse where he or she expected to find the bin that stored the product on the ticket. The worker manually picked boxes off of the shelf and then brought them to a packing area to be boxed, labeled, and loaded onto a truck. Bins were labeled manually, making them difficult to read. Sometimes the boxes held more than one product. Each picker had his or her preferred path to performing picking duties. The company's ERP system did not integrate well with other systems.

The new system: The new system banished paper. The new ERP system and the warehouse management system used software to manage the picking, packing, and shipping processes. Pickers carrying mobile devices receive data telling them where to go, what to pick, and where to bring the merchandise using the most efficient route. Pickers armed with wireless scanning devices were assured that the bar-code-labeled bins contained only one product type each.

2. What role did end users play in developing Dorfman's wireless warehouse system? What would have happened to the project if users hadn't been so involved? Explain your answer.

Probably the most important characteristic of this project is that Dorfman approached the change as a business project rather than an IT project. A cross-functional team consisting of an outside consultant as project manager and managers from distribution, purchasing, customer services, and sales worked on the transformation. The IT department took responsibility for choosing hardware, installing the hardware and software for the wireless warehouse, and appointed an administrator for the new warehouse management system. The employees had to change the way they worked. Dorfman took the job of selling the new systems to its workers very seriously, convincing them that the wireless warehouse would improve their lives and their job performance.

The new system could have easily failed if the employees had felt threatened by the new system. They could have sabotaged the implementation and cause work delays. Because the users were heavily involved in the system's development, they had a feeling of ownership and responsibility to ensure the success of the implementation. They were able to mold the system according to their priorities and business requirements. They were given more opportunities to control the outcome of the project.

3. What types of system-building methods and tools did Dorfman use for building its wireless warehouse system?

It appears Dorfman used a system development life cycle approach in the new project. It first defined the problem, identified the causes, identified the solution objectives, and identified its information requirements. Then it identified alternative solutions. Next it evaluated the alternatives and chose the best solution. In the implementation phase it created detailed design specifications, acquired the hardware, acquired the software, tested the system, trained employees, converted the system, and the evaluated the solution.

4. How did the new system change the way Dorfman ran its business?

The old paper-based system was completely eliminated. Tracking inventory became seamless. All of the company's systems were integrated. The company was able to handle twice the number of orders during peak seasons and reduced its labour costs

almost 30 percent. It eliminated the need for temporary workers and overtime, thus saving the company upwards of \$250,000.

5. What problems did the new system solve? Was it successful?

CEO Douglass Highsmith's goals were to reduce labour costs and create the most efficient way for a streamlined warehouse staff to pick products with the smallest error rate. Apparently the problems were solved because the company eliminated temporary workers and overtime and saved the company at least \$250,000. The new system's success can also be measured in employee satisfaction and acceptance of the new system.

MIS IN ACTION QUESTIONS

Use your Web-searching capabilities to answer the following questions.

1. What are some of the components of a wireless warehouse system?

Data Capture: handheld devices, barcode readers, RFID tags, hardware including central data repository, and software including a database management system.

Output: label printing—both stationary and mobile wireless, reports.

Access points: radio frequency access points, antennas, network routers.

2. What companies manufacture these components?

Many different companies manufacture the necessary components for wireless warehouse systems. Dee Electronics, Micromaster and Apprise are just a few. The link below provides access to the Apprise Web site for further information.

<http://www.apprise.com/products/distribution/>

3. What other businesses or organizations have implemented wireless warehouses?

This link provides access to an IBM Case Study about how Southern Wine & Spirits implemented a wireless warehouse system in four months.

<http://www-304.ibm.com/jct03001c/press/us/en/pressrelease/19865.wss>

4. If you were implementing a wireless warehouse, what potential problems would worry you most?

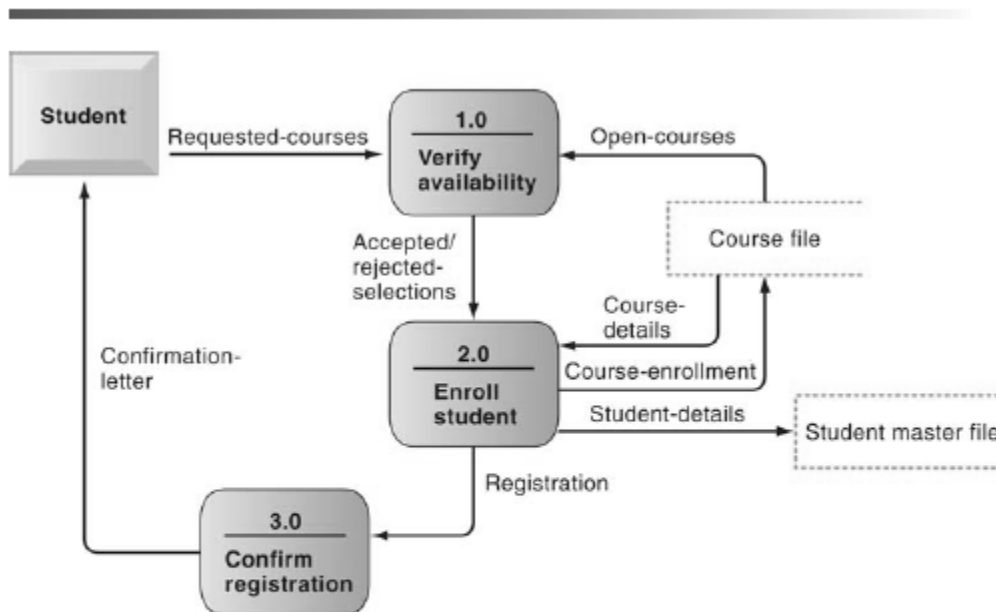
Student answers will vary based on their experiences and depth of knowledge about wireless warehouses.

MODELLING AND DESIGNING SYSTEMS: STRUCTURED AND OBJECT-ORIENTED METHODOLOGIES

Structured Methodologies

Structured refers to the fact that the techniques are step by step, with each step building on the previous one. Traditionally, systems have been structured in a very orderly manner. The methods used to build the systems began at the top and progressed to the lowest detail always with an eye towards keeping the processes separated from the data. The designers sketch out how the data moves through the system by using data flow diagrams. The designers could easily track all the processes and their interrelationships by using the **data flow diagrams (DFD)**. The DFDs can be used to help spot trouble areas before the system is actually built. Figure 9-4 shows a typical data flow diagram.

Two other tools used for structured analysis include process specifications and structure charts. The aim of **process specifications** is to describe the transformation occurring within the lowest level of the data flow diagrams. They express the logic for each process. The **structure chart** is a top-down chart, showing each level of design, its relationship to other levels, and its place in the overall design structure. The design first considers the main function of a program or system, then breaks this function into subfunctions, and decomposes each subfunction until the lowest level of detail has been reached.



The system has three processes: Verify availability (1.0), Enroll student (2.0), and Confirm registration (3.0). The name and content of each of the data flows appear adjacent to each arrow. There is one external entity in this system: the student. There are two data stores: the student master file and the course file.

The advantage of using data flow diagrams is that they can be used to show a very general, high-level process or very minute detail using the same tools. Anyone can view the overall system and then easily drill down through the diagrams to lower levels of the system. Couple the DFDs with a data dictionary and you can develop process specifications that describe how the data is transformed into useable information. Hierarchical structure charts complete the structured methodology by providing top-down charts that show each level and how they interrelate.

Object-Oriented Development

Most software development methods keep data and processes separate. **Object-oriented software development** combines the two and treats them as one object. More importantly, the objects are created once and, if they are done right, can be used many times over. That reduces the cost of creating new objects once you have built up a library of objects. It also makes it easier to create new software, because you aren't continually starting from scratch.

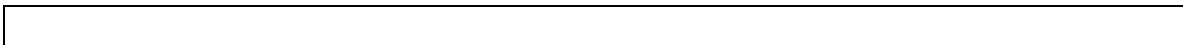
One tool businesses can use to create new software is *object-oriented programming*. This programming language has become much simpler for nontechnical people to use. This type of programming treats text, tables, pictures, or groups of data, as an **object** that can be manipulated and programmed to do special functions. It requires some in-depth training and time to learn, but this kind of programming is very functional for small businesses or a particular department. It can save time and money by allowing the end user to develop a whole program without the help of computer professionals.

The terms used in this section, classes and inheritance, describe how object-oriented programming uses objects to develop a program.

Two important tools used in object-oriented programming are structural diagrams and behavioural diagrams. These tools are part of the unified modelling language (UML) that has become the industry standard for use in this type of programming. The structural diagrams help you understand class relationship while behavioural diagrams help you understand what the system does versus how it accomplishes the process. Both diagrams are equally important in understanding class and inheritance characteristics of object-oriented programming.

Computer-Aided Software Engineering

Computer aided software engineering (CASE) provides developers with an easy to use method of developing software code that is well-documented, well-designed, and easily reused. Because software programs are becoming so complex it's not unusual to have teams of programmers develop the code. CASE tools provide an element of coordination and management for the programming teams. CASE tools also help keep programming elements such as data flow diagrams and data dictionaries synchronized. CASE tools bring discipline and structure to the program design and development process that may otherwise be lacking.



Bottom Line: You should not look at the system development process as a straight line but rather as a series of steps to be revisited and repeated. The most important piece of the puzzle in systems development is the user.

9.3 ALTERNATIVE SYSTEM-BUILDING APPROACHES

The traditional systems life cycle methodology is usually only used for very large, complex systems. As the text makes clear, it is inflexible and does not allow easy changes at any step along the way. However, it can be effective for highly-structured systems such as accounting, payroll, or complex manufacturing systems. Government defence or space systems often are mandated to use the system life cycle methodology because of the rigorous milestones generated by the method.

However, there are alternative methods that are used depending on factors such as the type of system being developed, the complexity of the system, the experience of the end-users, and the impact of the system on the core functions of the organization.

TRADITIONAL SYSTEMS LIFE CYCLE

The **systems life cycle** describes large- and medium-size systems projects. It has existed for years and uses tried-and-true methods that help ensure the success of the system from its humble beginnings as an idea to an old relic that eventually needs replacing.

The traditional life cycle may seem rigid today: There are very definite roles for techies and non-techies. Even though end users or managers have to sign off and accept the system at the end of each stage, they are not as involved in this method as some of the others we'll look at later.

The life cycle approach works well for major systems but doesn't fit the bill for smaller ones. It's expensive, time-consuming, and sometimes doesn't allow techies and non-techies to work together as they should.

PROTOTYPING

Prototyping can be the best way to develop a new system if the end users don't fully understand what they really want the end product to look like. Even if they have a few clues, this approach works well because the user can guide the process based on what they see as the system is built.

Have you ever watched a television show where the police artist draws a picture of the crook as the victim looks on? The artist draws the eyes and gets approval from the onlooker. Then the mouth is sketched in and approved. Pretty soon a composite drawing

is completed, and the cops are off and running. That pretty much describes the **iterative** process used in building a prototype system.

You would generally use prototyping for very small systems or small parts of a larger system. You wouldn't want to use this method to build a company-wide information system. It can be too unstructured, making it harder to manage in large projects. Prototyping works well when you're developing user interfaces and output reports — areas the users will see the most.

Steps in Prototyping

The text outlines the four steps you use when developing a prototype. The important thing to keep in mind is that these steps should be repeated many times over.

Advantages and Disadvantages of Prototyping

Prototyping can be less costly than the traditional systems approach, but if you fail to follow some of the basic principles of systems development, it can be more costly. For instance, if you ignore the basic principle of how the prototype fits into the other information systems in the organization, or how it supports the business plan in general, you may be costing the organization more money than you realize. Did you just create an island of information that is incompatible with other systems, or is it fully supportive and easy utilized in other areas?

The greatest advantage of the prototyping method of developing **end-user interfaces** is that users see the product, or at least a pretty good replica of it, right away. If they like it, you press on. If they don't like it, changes can be made immediately. There's less red tape and bureaucracy (perceived or otherwise) to work through in this method.

END-USER DEVELOPMENT

This method of system development is a bit like prototyping, but the end user designs and develops the new system using the fourth-generation language tools we previously reviewed. It's convenient for small applications, and the user can have complete ownership of the system.

The tools available to the end user are getting easier to use all the time and increase the likelihood that the system will meet the user's specifications, since the user is building it. There's no one else to blame if the system doesn't do what the user wants it to do. But don't attempt to build larger and more complex systems using this method: The capabilities of the tools are limited.

Managers should be aware of some inherent dangers when allowing users to develop their own systems. Standardization can be a tough issue when you use this method of system development. You're almost begging for conflicts in data processing and storage, since each user will have his or her own method of creating, defining, and developing data.

The biggest danger is creating those islands of information. The chance of redundant end products just went up, since each user will have his or her own system with slight differences that may prohibit cross-utilization of the information.

One way to reduce some of the risks associated with this development process is to establish information centres in the organization. These are like technical support units that offer help and guidance to users. They can be the focal point for training, reviews, and advice. They can also ensure that systems meet certain organizational standards. Just don't make them too bureaucratic or difficult to access, because then no one will use them.

Some types of information systems can be developed by end users with little or no formal assistance from technical specialists. This phenomenon is called **end-user development**. **Fourth-generation languages** are software tools that enable end users to create reports or develop software applications with minimal or no technical assistance. In developing their own software, end-users are most likely to work with PC software tools and query languages. **Query languages** are software tools that provide immediate online answers to requests for information that are not predefined. Many students are familiar with Microsoft's Access relational database software in which they have constructed and executed database queries.

APPLICATION SOFTWARE PACKAGES AND OUTSOURCING

We mentioned earlier that software programs are becoming extremely complex to design, develop, and build. Many companies either don't have the in-house staff to accomplish the task or decide to focus on their core competencies and have someone else develop the software they need.

Fast, easy, convenient, user-driven. Many software packages are extremely convenient for non-techies to use to develop their information products. Commonly called "off-the-shelf" software, these packages can be the best method of creating an information system if that system is fairly standard across different types of businesses.

You don't have to worry about system documentation, since that usually comes with the software. You still have to write local procedures for using the program, but you don't have to start from scratch. Training is easier because once you learn how to use the menus and toolbars in one program, the same skills can be carried over to other programs. Training manuals often come with the program or are available through online help functions.

Application Software Packages

Application software packages also provide an easier method of obtaining code corrections, updates, and enhancements: simply go to the Web site of the company that wrote the software and download the latest version.

Most of the common programs still need to have standards for use within the organization. For instance, if you use an accounts receivable application software program, you should still set standards for how you will adapt that program to meet the unique requirements of your company.

Most off-the-shelf software can't be changed, so you have to take what comes. The unique requirements of your organization probably won't be met. You'll end up having to change your methods to match the software, instead of the other way around. Some software packages do allow some **customization**, but not as much as a program developed solely for your organization.

Application software packages still need lots of planning, especially when it comes to integrating them with the other information systems throughout the organization. Compatibility is key.

You should determine the total cost of ownership with these programs beforehand. What are the training costs, implementation costs, and integration costs? They all add up.

Selecting software packages can be just as demanding as developing a system on your own. You have to evaluate:

- the program's functions to make sure they fit your needs
- flexibility to adapt to your business
- user-friendliness
- hardware and software resources
- database requirements
- installation and maintenance efforts
- documentation
- vendor quality (including follow-up)
- cost

Just as you would for any piece of equipment, you would seek **Requests for Proposal (RFP)** from several vendors to fully evaluate the software package according to your needs. Remember, you give up a lot of control when you choose to go with a prewritten software package.

Outsourcing

What happens if an organization decides it doesn't have the in-house expertise to support the system development process or any of the system maintenance required? No problem: *outsource*. There are hundreds of outside companies that will do the job. These companies offer expertise and experience, often at a lower cost than in-house staff. They can also offer smaller organizations economies of scale that make overall information processing cheaper.

The total cost of ownership of a system can be cheaper because of outsourcing. Perhaps the outsourcing company can keep up with changing technologies better than the

organization. It may simply be that the organization decides to spend in-house information resource dollars in other ways.

Should you decide to use an outsourcing company to develop an information system, you must be more careful than ever to ensure that everything, right down to the smallest detail, is in writing and agreed upon by both sides. You are signing a contract with the outsourcer that carries the full force of law. You must agree on how changes will be made to the current system. How responsive will the outsourcer be to changing requirements? You still have some responsibilities for the system; what will they be? Get it in writing!

You must continually analyze the outsourcing company's performance and cost and make sure it remains the cheaper, better way to handle the organization's needs. At some point in time, you may find a different method is in fact cheaper.

Bottom Line: There are different ways to develop information systems: prototyping, application software packages, end-user development tools, outsourcing. Analyze each and then pick the right tool for the right job.

WINDOW ON MANAGEMENT: HOW TO GET OUTSOURCING RIGHT: AVOID GETTING IT WRONG

A firm is most likely to benefit from outsourcing if it understands its own requirements, knows exactly how the outsourcing vendor will provide value, and identifies providers with capabilities and objectives that are best aligned with its specific needs. The firm must also be able to design an outsourcing contract that allows it to manage the vendor relationship.

TO THINK ABOUT QUESTIONS

- 1. What is the basis for vendor firms claiming they can provide IT services more economically than a firm's own IT staff?**

Outsourcing vendor firms claim that they can provide IT services more economically than a firm's own IT staff by being able to do the job better, faster, and cheaper than a company's internal staff, and still make a profit. They claim to have the economies of scale to do the basic computing cheaper, and to have the complimentary assets in the form of people and expertise that your firm needs.

- 2. Why is it difficult to write iron-clad legal contracts specifying in detail strategic alliance outsourcing relationships?**

The deals are so big and so long term, it's impossible to write iron-clad contracts.

Although many sources could be cited, the main reason is that the majority of partners do not know how to adequately value their internal staffs, and they fail to understand their own cost structure. Outsourcing companies also have no data to evaluate the claims that an outsourcer makes in stating that they can do it cheaper, faster, and better than the companies own internal IT staff could do the work in-house.

3. Why do joint ventures and co-sourcing outsourcing relationships have a better chance of success?

Joint ventures:

- Both parties share the same interests and both parties want to benefit from the relationship.
- Both parties perform the work and share the risks and rewards. In this relationship, both parties have a vested interest in saving money, do the work properly, and ensure that the venture is successful for each other.
- Companies develop long-term and lasting strategic alliances with their outsourcing vendors. By doing so, both parties are in a better position to take each other interests into account as the relationship moves forward.
- Flexibility, trust, performance, and self-sacrifice seem to be some of the ingredients that have proven essential in making these types of relationships stronger and lasting.

Co-sourcing alliances:

- Both the vendor and the firm co-manage and co-own the outsourcing project.
- More precise understanding of costs and benefits, and a greater transparency because both partners “own” the outcome.

MIS IN ACTION QUESTIONS

1. Capgemini is one of the largest domestic IT service firms. Visit their Web site (<http://www.capgemini.com>). Identify and describe their outsourcing services. What are the benefits they promise to deliver?
2. At the Capgemini Web site, read one of the company's success stories of collaboration with a firm for providing IT services. Describe what is being outsourced and identify what type of outsourcing relationship is being used (strategic, transaction, or co-sourcing). Do you think this is the right model for the work being performed?
3. Review the article “In Depth: Customers Analyze Outsourcing Vendors and Strategies,” by Paul McDougall, *Information Week*, June 19, 2006. This article is available at <http://www.informationweek.com>. What are the motivations for using outsourcing? Is outsourcing expected to grow, and why?

9.4**APPLICATION DEVELOPMENT FOR THE DIGITAL FIRM**

Digital firms can't hold onto their old systems that fit requirements developed ten or even five years ago. Continual change, growth, and migration are required to remain competitive in today's business environment. New development processes that allow a firm to build systems to meet today's and tomorrow's requirements must be faster and easier than ever before.

RAPID APPLICATION DEVELOPMENT (RAD), JOINT APPLICATION DESIGN (JAD), AND AGILE DEVELOPMENT

The supply of technical specialists is not enough to support the demand for new systems, or maintenance of the old ones. Something has to fill the gap — that's why you see so many new methods already on the market and more advanced, easier-to-use tools coming down the road. The shortage of skilled technicians is also why you see more and more companies moving away from the structured methods we've reviewed. There just isn't enough time.

Rapid Application Development (RAD) reduces the time it takes to build systems by using many of the tools that we've discussed. You can choose from prototyping or fourth-generation tools to develop systems much more quickly. End users and techies can work hand-in-hand with **joint application design (JAD)** tools to reduce the development time for new applications.

COMPONENT-BASED DEVELOPMENT AND WEB SERVICES

Component-based development is simply the practice of developing reusable components that are commonly found in many software programs. Create a "Save File" function for one application and use it in all applications. Not only does it save development time but creates functions that users have to learn only once and use multiple times. In short, why keep re-inventing the wheel when it works just fine across a multitude of vehicles.

Web Services and Service-Oriented Computing

The Internet and the Web provide the open standards that so many businesses and users now demand and were lacking from closed or proprietary systems. Advancing upon that openness is a new idea called Web services.

The potential reward is untold savings in time and money and a major boost in productivity. Web services are based on industry standards so that all the services can speak to one another. That keeps companies from having to cope with pricey, proprietary software that can cost 10 times as much as Web service software. At the same time, companies are able to

automate mundane tasks. “This is about automation — replacing manual processes with machine processes,” says analyst Ted Schadler of Forrester Research Inc. “It’s poised to revolutionize the way people solve business problems.” (*BusinessWeek*, e.biz, March 18, 2002)

Bottom Line: New methods of developing systems are continually being introduced. These new technologies, rapid application development, joint application development, component-based development, and Web services are reducing the time, effort and cost for businesses and organizations of supplying employees, customers and suppliers with the information they need.

SUMMARY

1. How does developing new systems produce organizational change?

Building a new information system is a form of planned organizational change that involves many different people in the organization. Because information systems are sociotechnical entities, a change in information systems involves changes in work, management, and the organization. Four kinds of technology-enabled change are: (a) automation, (b) rationalization of procedures, (c) business reengineering, and (d) paradigm shift, with far-reaching changes, carrying the greatest risks and rewards. Many organizations are attempting business reengineering to redesign work flows and business processes in the hope of achieving dramatic productivity breakthroughs. Information systems can also be used to support business process management, total quality management (TQM), six sigma, and other initiatives for incremental process improvement.

2. What are the core activities in the information systems development process?

The core activities in systems development are systems analysis, systems design, programming, testing, conversion, production, and maintenance. Systems analysis is the study and analysis of problems of existing systems and the identification of requirements for their solutions. Systems design provides the specifications for an information system solution, showing how its technical and organizational components fit together.

3. What are the principal methodologies for modelling and designing information systems?

The two principal methodologies for modelling and designing information systems are structured methodologies and object-oriented development. Structured methodologies focus on modelling processes and data separately. The data flow diagram is the principal tool for structured analysis and the structure chart is the

principal tool for representing structured software design. Object-oriented development models a system as a collection of objects that combine processes and data. Object-oriented modelling is based on the concepts of class and inheritance.

4. *What are the alternative methods for developing information systems?*

There are a number of alternative methods for building information systems, each suited to different types of problems. The oldest method for building systems is the systems life cycle, which requires that information systems be developed in formal stages. The stages must proceed sequentially and have defined outputs; each requires formal approval before the next stage can commence. The system life cycle is useful for large projects that need formal specifications and tight management control over each stage of systems building. However, this approach is very rigid and costly and is not well suited for unstructured, decision-oriented applications for which requirements cannot be immediately visualized.

Prototyping consists of building an experimental system rapidly and inexpensively for end users to interact with and evaluate. The prototype is refined and enhanced until users are satisfied that it includes all of their requirements and can be used as a template to create the final system. Prototyping encourages end-user involvement in systems development and iteration of design until specifications are captured accurately. The rapid creation of prototypes can result in systems that have not been completely tested or documented or that are technically inadequate for a production environment.

Developing an information system using an application software package eliminates the need for writing software programs when developing an information system. Using a software package reduces the amount of design, testing, installation, and maintenance work required to build a system. Application software packages are helpful if a firm does not have the internal information systems staff or financial resources to custom develop a system. To meet an organization's unique requirements, packages may require extensive modifications that can substantially raise development costs.

End-user development is the development of information systems by end users, either alone or with minimal assistance from information systems specialists. End-user-developed systems can be created rapidly and informally using fourth-generation software tools. The primary benefits of end-user development are improved requirements determination; reduced application backlog; and increased end-user participation in, and control of, the systems development process. However, end-user development, in conjunction with distributed computing, has introduced new organizational risks by propagating information systems and data resources that do not necessarily meet quality standards and that are not easily controlled by traditional means.

Outsourcing consists of using an external vendor to build (or operate) a firm's information systems. The work is done by the vendor rather than by the organization's internal information systems staff. Outsourcing can save application development costs or enable firms to develop applications without an internal information systems staff. However, firms risk losing control over their information systems and becoming too dependent on external vendors.

Selection of a systems-building approach can have a big impact on the time, cost, and end product of systems development. Managers should be aware of the strengths and weaknesses of each systems-building approach and the types of problems for which each is best suited. The impact of application software packages and of outsourcing should be carefully evaluated before they are selected because these approaches give organizations less control over the systems-building process.

5. *What are the new approaches for systems development in the digital firm era?*

Businesses today are often required to build e-commerce and e-business applications very rapidly to remain competitive. New systems are likely to have more interorganizational requirements and processes than in the past. Companies are turning to rapid application design, joint application design (JAD), and reusable software components to improve the systems development process. Rapid application development (RAD) uses object-oriented software, visual programming, prototyping, and fourth-generation tools for very rapid creation of systems. Component-based development expedites application development by grouping objects into suites of software components that can be combined to create large-scale business applications.

Web services enable firms to obtain software application components delivered over the Internet for building new systems or integrating existing systems. Web services provide a common set of standards that enable organizations to link their systems regardless of their technology platform through standard plug and play architecture.

KEY TERMS

The following alphabetical list identifies the key terms discussed in this chapter.

Acceptance testing — provides the final certification that the system is ready to be used in a production setting.

Automation — using the computer to speed up the performance of existing tasks.

Benchmarking — setting strict standards for products, services, or activities and measuring organizational performance against those standards.

Business process management (BPM) — methodology for revising the organizations business processes to use business processes as fundamental building blocks of corporate information systems.

Business process reengineering — the radical redesign of business processes, combining steps to cut waste and eliminating repetitive, paper-intensive tasks to improve cost, quality, and service and to maximize the benefits of information technology.

Component-based development — building large software systems by combining preexisting software components.

Computer-aided software engineering (CASE) — automation of step-by-step methodologies for software and systems development to reduce the amounts of repetitive work the developer must do.

Conversion — the process of changing from the old system to the new system.

Customization — the modification of a software package to meet an organization's unique requirements without destroying the packaged software's integrity.

Data flow diagram — primary tool for structured analysis that graphically illustrates a system's component process and the flow of data between them.

Direct cutover — a risky conversion approach in which the new system completely replaces the old one on an appointed day.

Documentation — descriptions of how an information system works from either a technical or end-user standpoint.

End-user development — the development of information systems by end users with little or no formal assistance from technical specialists.

End-user interface — the part of an information system through which the end user interacts with the system, such as online screens and commands.

Feasibility study — as part of the systems analysis process, the way to determine whether the solution is achievable, given the organization's resources and constraints.

Fourth-generation languages — a programming language that can be employed directly by end users or less-skilled programmers to develop computer applications more rapidly than conventional programming languages.

Information requirements — a detailed statement of the information needs that a new system must satisfy; identifies who needs what information, and when, where, and how the information is needed.

Iterative — a process of repeating over and over again the steps to build a system.

Joint application design (JAD) — process to accelerate the generation of information requirements by having end users and information systems specialists work together in intensive interactive design sessions.

Maintenance — changes in hardware, software, documentation, or procedures to a production system to correct errors, meet new requirements, or improve processing efficiency.

Object — software building block that combines data and the procedures acting on the data.

Object-oriented development — approach to systems development that uses the object as the basic unit of systems analysis and design. The system is modeled as a collection of objects and the relationship between them.

Paradigm shift — radical re-conceptualization of the nature of the business and the nature of the organization.

Parallel strategy — a safe and conservative conversion approach where both the old system and its potential replacement are run together for a time until everyone is assured that the new one functions correctly.

Phased approach — introduction of the new system in stages either by functions or by organizational units.

Pilot study — a strategy to introduce the new system to a limited area of the organization until it is proved to be fully functional; only then can the conversion to the new system take place across the entire organization.

Postimplementation audit — formal review process conducted after a system has been placed in production to determine how well the system has met its original objectives.

Process specifications — describe the logic of the processes occurring within the lowest levels of a data flow diagram.

Production — the stage after the new system is installed and the conversion is complete; during this time the system is reviewed by users and technical specialists to determine how well it has met its original goals.

Programming — the process of translating the system specifications prepared during the design stage into program code.

Prototype — the preliminary working version of an information system for demonstration and evaluation purposes.

Prototyping — the process of building an experimental system quickly and inexpensively for demonstration and evaluation so that users can better determine information requirements.

Query languages — software tool that provides immediate online answers to requests for information that are not predefined.

Rapid application development (RAD) — process for developing systems in a very short time period by using prototyping fourth-generation tools, and close teamwork among users and systems specialists.

Rationalization of procedures — the streamlining of standard operating procedures, eliminating obvious bottlenecks, so that automation makes operating procedures more efficient.

Request for Proposal (RFP) — a detailed list of questions submitted to vendors of software or other services to determine how well the vendor's product can meet the organization's specific requirements.

Six sigma — a specific measure of quality, representing 3.4 defects per million opportunities; used to designate a set of methodologies and techniques for improving quality and reducing costs.

Structure chart — system document showing each level of design, the relationship among the levels, and the overall place in the design structure; can document one program, one system, or part of one program.

Structured — refers to the fact that techniques are carefully drawn up, step by step, with each step building on a previous one.

Systems analysis — specialists who translate business problems and requirements into information requirements and systems and who act as liaisons between the information systems department and the rest of the organization.

Systems design — details how a system will meet the information requirements as determined by the systems analysis.

Systems development — the activities that go into producing an information systems solution for an organizational problem or opportunity.

Systems life cycle — a traditional methodology for developing an information system that partitions the systems development process into formal stages that must be completed sequentially with a very formal division of labour between end users and information systems specialists.

System testing — tests the functioning of the information system as a whole to determine whether discrete modules will function together as planned.

Test plan — a plan prepared by the development team in conjunction with the users; it includes all of the preparations for the series of tests to be performed on the system.

Testing — the exhaustive and thorough process that determines whether the system produces the desired results under known conditions.

Total quality management (TQM) — a concept that makes quality control a responsibility to be shared by all people in an organization.

Unit testing — the process of testing each program separately in the system. Sometimes called program testing.

Work flow management — the process of streamlining business procedures so that documents can be moved easily and efficiently from one location to another.

REVIEW QUESTIONS

1. How does building new systems produce organizational change?

Describe each of the four kinds of organizational change that can be promoted with information technology.

Figure 9-1 shows four kinds of organizational change and the risks and rewards of each:

- **Automation:** low risk, low reward. Employees perform tasks more efficiently and effectively.
- **Rationalization:** medium risk, medium reward. Involves streamlining standard operating procedures, redesigning business processes, work flows, and user interfaces.
- **Business process reengineering:** higher risk, higher reward. Organizations rethink and streamline business processes to improve speed, service, and quality. BPR reorganizes work flows, combining steps to cut waste and eliminate repetitive, paper-intensive tasks. May eliminate jobs also.
- **Paradigm shift:** highest risk, highest reward. Transforms how an organization carries out its business or even the nature of the business.

Define business process reengineering and explain how it differs from business process management. Describe the steps required for effective reengineering.

Business process reengineering: business processes are analyzed, simplified, and redesigned. Information technology helps companies rethink and streamline their business processes to improve speed, service and quality. It requires a new vision of how the process is to be organized. BPR is normally a one-time effort.

Business process management: companies manage incremental process changes that are required simultaneously in many areas. Organizations need to revise and optimize numerous internal business processes and BPM provides the methodologies and tools necessary to be successful. BPM is more about continual improvements to business processes and using processes as building blocks in information systems.

Steps required for effective reengineering: A business first needs to understand what business processes need improvement. Improving the wrong processes simply allows a business to continue doing what it shouldn't do in the first place. Next, an organization must understand and measure the performance of existing processes as a baseline. Otherwise, the effectiveness of the changes can't be determined. The majority of BPR projects don't provide the breakthroughs expected because organizations changes are very difficult to manage.

Explain how information systems support process changes that promote quality in an organization.

Total quality management (TQM) and Six Sigma are two quality improvement programs that differ from business process reengineering which is based on radical one-time redesigns of business processes. They are more incremental in their approach to change than BPR. TQM requires a series of continuous improvements. Six Sigma requires statistical analysis to detect flaws and make minor adjustments.

Information systems help achieve organizational goals in these two programs by

- helping companies simplify products or processes
- making improvements based on customer demands
- reducing cycle times
- improving the quality and precision of design and production
- meeting benchmarking standards

2. What are the core activities in the systems development process?

Distinguish between systems analysis and systems design. Describe the activities for each.

System analysis: requires an organization to analyze problems it will try to solve with information systems. It includes defining a problem, identifying its causes, specifying the solution, and identifying information requirements that must be met by the system solution. System analysis requires determining if the solution is feasible from a financial, technical, and organizational standpoint.

System design: shows how the system will fulfill the objective of the system analysis. It's the overall plan or model for the system that serves as a blueprint and consists of all the specifications that give the system its form and structure. The

specifications outline managerial, organizational, and technological components of the system solution.

Define information requirements and explain why they are difficult to determine correctly.

Information requirements involve identifying who needs what information, where, when, and how. They define the objectives of the new or modified system and contain a detailed description of the functions the new system must perform. Gathering information requirements is perhaps the most difficult task of the systems analyst, and faulty requirements analysis is a leading cause of systems failure and high systems development costs. Some problems require adjustments in management, additional training, or refinement of existing organizational procedures rather than an information system solution.

Explain why the testing stage of systems development is so important. Name and describe the three stages of testing for an information system.

Testing is critical to the success of a system because it is the only way to ascertain whether the system will produce the right results. Three stages of information system testing are:

- **Unit testing:** refers to separately testing or checking the individual programs.
- **System testing:** the entire system is tested to determine whether program modules are interacting as planned.
- **Acceptance testing:** the system undergoes final certification by end users to ensure it meets established requirements and that it's ready for installation.

Describe the role of programming, conversion, production, and maintenance in systems development.

- **Programming:** specifications that were prepared during the design stage are translated into software program code.
- **Conversion:** the process of changing from the old system to the new system.
- **Production:** the operation of the system once it has been installed and conversion is complete. The system will be reviewed during production by both users and technical specialists to determine how well it has met its original objectives and to decide whether any revisions or modifications are needed.
- **Maintenance:** modifications to hardware, software, documentation, or procedures to a production system to correct errors, meet new requirements, and improve processing efficiency.

3. What are the principal methodologies for modeling and designing systems?

Compare object-oriented and traditional structured approaches for modeling and designing systems.

Structured methods for designing systems separate processes from data in the modeling process. System analysis is separated from system design. Object-oriented development (OOD) combines processes and data into an object that becomes the basic unit of system analysis and design. Data encapsulated in an object is accessed and modified only by the operations or methods associated with that object. Processing logic resides within the objects. Object-oriented development is more iterative and incremental than traditional structured development. The analysis phase of OOD requires system builders to document the functional requirements of a system and specify its most important properties. During the design phases, system builders describe how the objects will behave and how they will interact with each other. Similar objects are grouped together to form a class. Classes are further grouped into hierarchies in which a subclass inherits the attributes and methods from its superclass.

4. What are the alternative methods for developing information systems?

Define the traditional systems lifecycle. Describe each of its steps and its advantages and disadvantages for systems building.

The traditional systems lifecycle is a formal methodology for managing the development of systems and is still the principal methodology for large projects. The overall development process is partitioned into distinct stages, each of which consists of activities that must be performed to fashion and implement an information system. The stages are usually sequential with formal “sign-off” agreements among end users and data processing specialists to validate that each stage has been completed. Users, managers, and data processing staff have specific responsibilities in each stage. The approach is slow, expensive, inflexible, and is not appropriate for many small desktop systems.

The systems lifecycle consists of systems analysis, systems design, programming, testing, conversion, and production and maintenance. Systems analysis requires an organization to define a problem that needs solving. Technical specialists identify the problem, gather information requirements, develop alternative solutions, and establish a project management plan. Business users provide information requirements, establish financial or operational constraints, and select the solution. During systems design, technical specialists model and document design specifications and select the hardware and software technologies for the solution. Business users approve the specifications.

Technical specialists translate the design specifications into software during the programming phase. After that, technical specialists develop test plans and conduct unit, system, and acceptance tests. Business users provide test data and scenarios and validate test results.

During the conversion phase, technical specialists prepare a conversion plan and supervise conversion. Business users evaluate the new system and decide when the

new system can be put into production. During the production and maintenance phase, technical specialists evaluate the technical performance and perform maintenance. Business users use the system and evaluate its functional performance.

The main advantages of using this method for building information systems are:

- it is highly structured;
- it has a rigorous and formal approach to requirements and specifications and tight controls over the system building process;
- it is appropriate for building large transaction processing and management information systems and for building complex technical systems.

Disadvantages are:

- it is very costly and time-consuming;
- it is inflexible and discourages change even though requirements will change during the project due to the long time this method requires;
- it is ill-suited to decision-oriented applications that can be rather unstructured and for which requirements are difficult to define.

Define information system prototyping. Describe its benefits and limitations. List and describe the steps in the prototyping process.

Information system prototyping is an explicitly interactive system design methodology that builds an experimental model of a system as a means of determining information requirements. Prototyping builds an experimental system quickly and inexpensively for demonstration and evaluation so that users can better determine information requirements. A preliminary model of a system or important parts of the system is built rapidly for users to experiment with. The prototype is modified and refined until it conforms precisely to what users want. Information requirements and design are determined dynamically as users interact with and evaluate the prototype.

Prototyping is most valuable when requirements are uncertain and cannot be entirely pre-specified or when the appropriate design solution is unclear. Prototyping is especially helpful for designing end-user interfaces (screens and reports) and for determining elusive requirements of decision-support type applications. Prototyping can help reduce implementation costs by capturing requirements more accurately at an earlier point in the implementation process. It is not so useful for a very structured, well-understood, or routine problem.

It is best suited for smaller applications oriented toward simple data manipulation. Large systems with complex processing may only be able to have limited features prototyped. The prototype may be built so rapidly that design is not well thought out or must be reworked for a production environment. The problem arises when the prototype is adopted as the production version of the system without careful analysis and validation. Prototypes are built so rapidly that documentation and testing are glossed over. The system is so easily changed that documentation may not be kept

up-to-date.

The steps in prototyping include identifying the users basic requirements; developing a working prototype of the system outlined in the basic requirements, using the prototype, and revising and enhancing the prototype based on the users reaction. The third and fourth steps are repeated until users are satisfied with the prototype.

Define an application software package. Explain the advantages and disadvantages of developing information systems based on software packages.

Application software packages are common to all business organizations and are built around universal functions with standard processes that don't change a great deal over time. Examples include payroll, accounts receivable, general ledger, or inventory control.

Software packages provide several advantages:

- the vendor has already established most of the design that may easily consume up to 50 percent of development time
- programs are pretested, reducing testing time and technical problems
- the vendor often installs or assists in the installation of the package
- periodic enhancement or updates are supplied by the vendor
- vendors also maintain a permanent support staff well versed in the package, reducing the need for individual organizations to maintain such expertise in-house
- the vendor supplies documentation.

The disadvantages of application software packages are:

- there are high conversion costs for systems that are sophisticated and already automated
- they may require extensive customization or reprogramming if they cannot easily meet unique requirements
- a system may not be able to perform many functions well in one package alone

Define end-user development and describe its advantages and disadvantages. Name some policies and procedures for managing end-user development.

End-user development refers to the development of information systems by end users with minimal or no assistance from professional systems analysts or programmers. This is accomplished through sophisticated user-friendly software tools and gives end users direct control over their own computing.

Advantages include improved requirements determination, realizing large productivity gains when developing certain types of applications, enabling end users to take a more active role in the systems development process, many can be used for prototyping, and some have new functions such as graphics, modeling, and ad-hoc

information retrieval.

Disadvantages include not being suited for large transaction-oriented applications or applications with complex updating requirements, standards for testing and quality assurance may not be applied, and proliferation of uncontrolled data and private information systems.

End-user development is suited to solving some of the backlog problem because the end users can develop applications themselves. It is suited to developing low-transaction systems. End-user development is valuable for creating systems that access data for such purposes as analysis (including the use of graphics in that analysis) and reporting. It can also be used for developing simple data-entry applications.

Policies and procedures to manage end-user development include the following:

- The organization must establish sufficient support facilities for end-user computing: information centers or distributed end-user computing centers.
- Training and support should be targeted to the specific needs of those being trained.
- End-user application development should be incorporated into an organization's strategic plan.

Management should develop controls over end-user computing in the following areas:

- Cost justification of end-user information system projects
- Hardware and software standards for user-developed applications
- Company-wide standards for microcomputers, word processing software, database management systems, graphics software, and query and reporting tools
- Quality assurance reviews that specify whether the end-user systems must be reviewed by information systems and internal audit specialists
- Control for end-user developed applications covering testing, documentation, accuracy, and completeness of input and update, backup, recovery, and supervision
- Critical applications that supply data to other important systems should be flagged and subjected to more rigorous standards

Describe the advantages and disadvantages of using outsourcing for building information systems.

Outsourcing is the process of turning over an organization's computer center operations, telecommunications networks, or applications development to external vendors who provide these services. Outsourcing is seen as a way to control costs or to develop applications rather than try to use in-house staff.

Advantages of outsourcing are:

- Outsourcing firms possess skills, resources, and assets that clients don't

- It may be less costly than hiring, training, and maintaining in-house staff
- The Internet and networking technologies have drastically reduced costs associated with using offshore outsourcing firms

The main disadvantage is that firms hiring outsourcers tend to underestimate costs of

- Identifying and evaluating vendors
- Transitioning to a new vendor
- Improving internal software development methods to match those of vendors
- Monitoring vendors to make sure they are meeting contract requirements
- Coping with cultural differences

5. What are new approaches for information systems development in the digital firm era?

Define rapid application development (RAD) and agile development and explain how they can speed up system-building?

RAD is a process for developing systems in a very short time period by using prototyping, fourth-generation tools, and close teamwork among users and systems specialists. RAD allows the creation of working software in a very short time by using objects and by automating much of the code generation.

Agile development rapidly creates working software by breaking a large project into smaller sub-projects. Each sub-project requires a team to plan the project development, analyze requirements, design, code, test, and document it. The overall risk is minimized and projects can adapt to changes more quickly. Agile development emphasizes face-to-face communications rather than written documents. People collaborate more and make decisions more quickly and effectively using this method of software development.

Explain how component-based development and Web services help firms build and enhance their information systems.

Component-based development expedites application development by grouping objects into suites of software components that can be combined to create large-scale business applications. Systems are built by assembling and integrating existing software components like shopping carts, user authentication, search engines and online catalogs.

Web services enable firms to obtain software application components delivered over the Internet for building new systems or integrating existing systems. Web services provide a common set of standards that enable organizations to link their systems regardless of their technology platform through standard plug-and-play architecture. Web services offer significant cost savings and open up new opportunities for collaborating with other companies.

DISCUSSION QUESTIONS

1. Why is selecting a systems development approach an important business decision? Who should participate in the selection process?

The development of new systems or major enhancements to existing systems is often the result of significant changes made to the business processes supported by the systems. Organizations are being faced with the requirement to meet higher quality standards, but often with fewer resources. Basically, organizations need to do more with less and do it better. This trend forces organizations to find simpler and more efficient methods for performing their work. Usually the effort to simplify the business processes themselves precedes any major systems development effort. It is appropriate that the business processes be reviewed before systems work begins, to avoid the unfortunate mistake of simply automating existing cumbersome processes. Ideally, the efforts to simplify business processes will be done by the functional office in conjunction with technical personnel, so that current technology can be considered as the business processes are reviewed. In some cases, particularly when a vendor package is selected for implementation, the simplification of business processes may occur during the systems development or installation process. One other issue is critical. A decision on the selection of a development approach often is dependent upon the system itself. A large, complex system simply cannot be done using a quick prototype method, for example.

2. Some have said that the best way to reduce system development costs is to use application software packages or fourth-generation tools. Do you agree? Why or why not?

Student views will likely vary because no simple, straightforward answer exists. The answer to this question is dependent upon the type of problem to be solved. For example, large transaction-heavy applications are not suited to end-user software tools. Also, remember that purchasing and installing an application software package or fourth-generation tool rather than developing a system from scratch might save significant time, and often significant resources, in terms of cost. However, there are many issues to consider when deciding whether to purchase a product or develop a custom system. The following guidelines help determine if a vendor package is the right way to go:

- Are packages available on the market that can satisfy at least 80 percent of the functional requirements of the system? Although it may not be possible to find a package that exactly matches the specific functional requirements, if less than 80 percent of the requirements are satisfied, then the purchase of a package is likely to be more costly than custom development.
- Are available packages compatible with the organizations current technical environment (or planned for installation in the immediate future)?
- Is the overall cost of purchasing, installing, and maintaining a package no more

than (and preferably less than) the cost of custom development and maintenance?

- Do the vendors have a proven track record of installation and support and can they show evidence of financial stability (this is particularly critical for large systems)?

COLLABORATION AND TEAMWORK: PREPARING WEB SITE DESIGN SPECIFICATIONS

With three or four of your classmates, select a system described in this text that uses the Web. Review the Web site for the system you select. Use what you have learned from the Web site and the description in this book to prepare a report describing some of the design specifications for the system you select. If possible use Google Sites to post links to Web pages, team communication announcements, and work assignments; to brainstorm; and to work collaboratively on project documents. Try to use Google Docs to develop a presentation of your findings for the class.

Because Web systems play such a central role in today's information systems world, the purpose of this project is to give the students experience in evaluating a Web system and think through how well it meets IS requirements. After selecting the Web system, the groups should begin this project by developing a set of requirements they conclude the system will have met (strengths and weaknesses must be measured against some standards).

Students also need to ask hardware and operating system requirements questions, such as: What are the operating systems under which it will run? What are the minimum and optimal amounts of RAM needed and the minimum and optimal amounts of disk storage space required (including holding data)?

Finally, students need to examine the user interface with such questions as: Does it use or require a mouse? How many keystrokes are necessary to access a given function? Is it easy to learn? Is there an expert operating mode so that the expert will not need to go through a long series of menus to access a function?

LEARNING TRACK MODULES

- 1. Unified Modelling Language (UML)***
- 2. Primer on Business Process Design and Documentation***

Students will find Learning Track Modules on this topic at the MyMISLab for this chapter.

HANDS-ON MIS: PROJECTS

Management Decision Problems

1. Sears Repair Service: customers purchasing appliances can purchase a three-year service contract that provides free repair service and parts. When a customer needs Sears' appliance repair service they call for an appointment that make take up to two weeks. The repairman arrives and diagnoses the problem. If the repair requires a new part, the repairman will replace it if he is carrying it on his truck. If he doesn't have the part on hand, he must order it. If the part is not in stock at Sears, it is ordered and sent to the customer. After the part arrives, the customer must make another appointment to have the repair technician replace the part. It make take 2 weeks to schedule the first repair visit, another 2 weeks to order and receive the part, and another week to schedule a second repair visit after the part has been received.

- **Diagram the existing process:** Students should use Figure 9-2 as an example of how to diagram this process. It's important to remember the "if-then" rules that apply when the technician diagnoses the problem. If he has the faulty part on his truck then he fixes it immediately. If he doesn't then he must order it.
- **What is the impact of the existing process on Sears' operational efficiency and customer relationships?** Customers can be inconvenienced and without the use of their appliances for up to five weeks. Customers may have to take off work twice or make other arrangements to accommodate the repair technician instead of just once. Sears has to schedule a repair technician's visit twice which is an expensive process.
- **What changes could be made to make this process more efficient? How could information systems support these changes? Diagram the new improved process.** Sears should analyze the most frequent reasons why appliances break and which parts fail most often. They can gather the information from their transaction processing system that manages parts ordering and gather information from repair technicians. Those parts that fail most often should be stocked on the repair trucks or in the Sears stores. Then only one repair visit is required to fix the most common problems.

2. Agricultural chemicals corporation: Management at your agricultural chemicals corporation has been dissatisfied with production planning. Production plans are created using best guesses of demand for each product, which are based on how much of each product has been ordered in the past. If a customer places an unexpected order or requests a change to an existing order after it has been placed, there is no way to adjust production plans. The company may have to tell customers it can't fill their orders, or it may run up extra costs maintaining additional inventory to prevent stock-outs.

At the end of each month, orders are totaled and manually keyed into the company's production planning system. Data from the past month's production and inventory systems are manually entered into the firm's order management system. Analysts from

the sales department and from the production department analyze the data from their respective systems to determine what the sales targets and production targets should be for the next month. These estimates are usually different. The analysts then get together at a high-level planning meeting to revise the production and sales targets to take into account senior management's goals for market share, revenues, and profits. The outcome of the meeting is a finalized production master schedule.

The entire production planning process takes 17 business days to complete. Nine of these days are required to enter and validate the data. The remaining days are spent developing and reconciling the production and sales targets and finalizing the production master schedule.

- **Draw a diagram of the existing production planning process.** Students should use Figure 9-2 as a guide for diagramming the existing production planning process.

- **Analyze the problems this process creates for the company.** The main problem this process creates for the company are the potentially lost sales or extra costs in maintaining additional inventory to prevent stock-outs if customers place unexpected orders or requests to change existing orders. It's also very costly to spend 17 business days to complete the entire production planning process. Manually keying in data to two different systems (production planning system and the order management system) creates potential problems if data are entered incorrectly. The employee overhead for this process is also extra costs the company can easily eliminate. Using 'best guesses' to create production plans is never a good idea.

- **How could an enterprise system solve these problems? In what ways could it lower costs? Diagram what the production planning process might look like if the company implemented enterprise software.** The first step is to eliminate having to manually key in any data into any system. Production planning should be based on data derived from both systems (production planning system and the order management system) and not on best guesses. That eliminates having different estimates from the sales department and the production department. The company can use business process management (BPM) to revise and optimize work flow management, business process modeling, and change management. BPM uses process-mapping tools to identify and document existing processes and to create models of improved processes. BPM software tools automatically manage processes across the business, extract data from various sources and databases, and generate transactions in multiple related systems. Middleware can connect the production and inventory system to the order management system. A single database should serve both systems. Web services can connect the two systems if the company decides to go that way. One user interface can be developed jointly between sales and production using prototyping techniques. The company would lower its costs by eliminating data entry, creating production plans based on real data, and allow more flexibility in its production planning process that could easily adapt to customer demands.

IMPROVING DECISION MAKING: USING DATABASE SOFTWARE TO DESIGN A CUSTOMER SYSTEM FOR AUTO SALES

This is a challenging assignment and will take longer to solve than the other Application Software exercises. Students will have to perform a systems analysis and then design a system solution using database software. They will need to identify information requirements and then map out entities, attributes, and relationships to guide the design of database tables. They will need to populate the database and generate queries and reports that satisfy management information requirements. This project should not be assigned unless students have the requisite database skills.

The data for students for this exercise is found in the file named Ch09_Auto_Sales.mdb in the Chapter 9 folder.

- 1. Description of the problem and its organizational and business impact.**
- 2. Proposed solution, solution objectives, and solution feasibility.**
- 3. Costs and benefits of the solution you have selected. The company has a PC with Internet access and the full suite of Microsoft Office desktop productivity tools.**
- 4. Information requirements to be addressed by the solution.**
- 5. Management, organization, and technology issues to be addressed by the solution, including changes in business processes.**

Ace's customer information is primarily paper based. There is valuable customer information in those paper records but it is of little use to the company because it cannot be easily organized or analyzed. The company cannot easily find out which customer touchpoints are most effective, customer preferences in models or luxury options, or whether repeat customers or Subaru owners return again to become buyers. The dealership is wasting dollars by not being able to channel its advertising and promotions more precisely. By not fully understanding their customers, the overall business performance of both Ace and Subaru may be negatively impacted.

A new system with a database of customers and prospects could help Ace make better sales and marketing decisions. The system could help Ace find the most important sources of customers, better allocate advertising and promotional budgets, and identify trends in customer preferences and demographics. If Ace automated this information, it would be more useful to Subaru in helping it stock dealers' inventory with models and options that are the most popular. The company could also have more accurate numbers on customer acquisition costs,

Key information requirements include:

- Identifying where potential customers are obtaining information about the dealership
- Identifying repeat customers or Subaru owners
- Identifying prospects and buyers
- Identifying what car models and options customers are purchasing
- Identifying how much is being financed

Some organizational changes will be required. Sales associates will need to make sure that customer and prospect information has been entered into the database. Managers will need to learn how to make use of the information in the database to help them run their dealership.

For the solution, please see the files named Ch09_Ace_Auto.mdb and Ch09_Ace_Auto_Solution.mdb, which are in the Chapter 9 folder. The Microsoft Access file is an example solution file that represents one of many alternative database designs that would satisfy Ace's requirements.

This particular design simplifies information on options to prevent the design from becoming too complicated for students. The base price of the car is assumed to be the list price. The total price paid by the customer is the list price (which includes the cost of luxury options) minus the discount offered by the dealer at the time of sale. A VIN number is used to identify the model of interest to the showroom visitor, which assumes that all models of interest to showroom visitors are actual vehicles with VIN numbers that are in dealer inventory. The VIN number of interest during the visit may not necessarily be the VIN number identifying the vehicle the customer actually purchased. The Customer_Prospect table contains information on both customers and prospects, including everyone who has visited the dealer, with customers identified when this table is joined to the Orders table.

ACHIEVING OPERATIONAL EXCELLENCE: REDESIGNING BUSINESS PROCESSES FOR WEB PROCUREMENT

Software skills: Web browser software

Business skills: Procurement

You are in charge of purchasing for your firm and would like to use the Acklands-Grainger.com (<http://www.acklandsgrainger.com>) B2B e-commerce site for this purpose. Find out how to place an order for painting supplies by exploring the Catalog, Order Form, and Repair Parts Order capabilities of this site. Do not register at the site. Describe all the steps your firm would need to take to use this system to place orders online for 30 gallons of paint thinner. Include a diagram of what you think your firm's business process for purchasing should be and the pieces of information required by this process.

In a traditional purchase process, whoever is responsible for making the purchase fills out a requisition form and submits it for approval based on the company's business rules. When the requisition is approved, a purchase order with a unique purchase order identification number is sent to the supplier. The purchaser might want to browse supplier catalogs to compare prices and features before placing the order. The purchaser might also want to find out if the items to be purchased are available. If the purchasing firm were an approved customer, that company would be granted credit to make the purchase and would be billed for the total cost of the items purchased and shipping after the order was shipped after the order was placed. Alternatively, the purchasing company might have to pay for the order in advance or pay for the order using a credit card. Multiple payment options might be possible.

How might this process have to change in order to make purchases electronically from the AcklandsGrainger site?

The general steps to place an order at AcklandsGrainger.com are as follows: Login, go to the online catalog section, and then go to the Repair Parts Centre. Complete a form for quantity, manufacturer part number or description, manufacturer/brand, and the manufacturer model number. Select a payment method, and then click the checkout icon. The students diagrams will vary based on their experience with flowcharting and diagramming tools. Multiple payment options need to exist for AcklandsGrainger to protect themselves, and to provide convenience for all types of customers.

CASE STUDY: LEMONTONIC AND NET: A MATCH MADE ON THE WEB

1. Write a systems analysis report about the Lemontonic instant messaging system.

Lemontonic identified four challenges in the online dating industry that IM could address: connectivity, control, comfort, and convenience. Lemontonic would use the Web and IM together to provide a realistic and enjoyable way for people to meet online.

What do you think may have been the problems with previous similar systems?

Problems with other systems included that people did not have control over who communicated with them, and they could not communicate instantly. Other online dating systems used email instead of IM. They did not allow, as does Lemontonic, features of IM including the ability to chat, share videos and websites after saying "hello".

What management, organization, and technology factors do you think caused the problems?

Most of the factors were technological: the old tools did not allow as much user control over the communications, nor were they as convenient. The VP of Lemontonic realized that new technology (IM) could introduce convenience, comfort, and control to the

connectivity-mix. Now users use the online tools to mimic face-to-face dating experiences (such as watching a video together). In addition the platform was scaleable to allow customer numbers to grow.

What do you think was the impact of these problems?

Impact of the problem was that users did not subscribe to the old dating systems. Now, there are over 60 000 members using the service. The new system supported a viable business idea. Without the technology, there would be no business.

What are the objectives and information requirements of a new systems solution?

The new system will allow people to meet new people and conduct an online date with selected individuals. The company charges a fee for subscribers, so the more users the more revenue the company will generate.

Information requirements for the new system would require that the data are secure and confidential. Subscribers would need to be able to access a database, and search for compatible members using a number of search criteria. The company would need information on credit cards, user profiles, use statistics, success rates, etc.

2. As part of your report, diagram the new member subscription process for joining Lemontonic. How should this process be improved?

The signing up process includes entering data on:

- Email address (and password creation)
- Name
- Gender
- Age
- Location

Students may identify that you can see who is online (pictures and location) without subscribing. This may compromise their sense of confidentiality.

3. Describe what you think the role of end users and technical specialists should have been in analyzing the Lemontonic situation and developing a solution.

The company would have beta tested the system with potential clients (end users) to ensure it was user friendly, and satisfied any concerns with confidentiality and control of information.

The technical specifications would need to be very specific and precise, given the need to maintain data security and privacy.

The technical specialists must work closely with the end users, both clients and the business owners. These clients know what they want in a dating service, and the owners

know what they need in terms of information from the system. Without the end user assistance, the technical specialists would find it much harder to build a system that would meet all of the needs outlined in the case. The technical specialists are trained in how to design a system. Too little discussion and communication between the techies and the non-techies will be apparent through design flaws and a poorly implemented project. Understand where both sides are coming from, and you'll do a better job of getting them to work together. You can never have too *much* communication.