CHAPTER

4

Development Life Cycle

LEARNING OBJECTIVES

After reading this chapter, you should be able to:

- Review the systems development life cycle (SDLC).
- Examine the problems and alternatives with SDLC.
- Know the key issues in ERP implementation strategy.
- Understand ERP implementation life cycle.
- Examine the rapid implementation methodologies.
- Compare and contrast SDLC and ERP life cycles.
- Examine the role of people like top management, consultants, and subject matter experts (SMEs) in the ERP life cycle.
- Understand the importance of the project management office and the project organization to a successful ERP implementation.
- Know the components of a project organization and the roles and responsibilities of each.

CASE 4-1

Opening Case

Of Men and Mice: An ERP Case Study

Source: Based on the article by Katz, D. (2001). Of Men and Mice: An ERP Case Study, *CFO.com*, March 21.

Jackson Laboratory is a nonprofit, independent, world-renowned genetic research institute founded in 1929. Located in Bar Harbor, Maine, it had a budget of $80 million and 1,200 employees, including 32 in IT. Jackson Laboratory decided to install an ERP system with a $5 million budget and a one-year time frame. Despite the installation challenges, the project's actual cost was close to the budget and took only about six months longer than expected.

Jackson Lab's major installation challenge was the integration of its unique mouse-development functions into Oracle's ERP system. One of the problems faced by Jackson stemmed from an internal HR issue (i.e., the risk that the action or inaction of the software provider would hinder the implementation). Jackson Lab coped with these challenges by modifying the ERP system to accommodate its business process, placing special emphasis on training, seeking a fixed-fee contract with Oracle, and purchasing a surety bond to reduce project risk. The surety bond was issued by an entity on behalf of a second party, guaranteeing that the second party would fulfil an obligation or series of obligations to a third party. In the event that the obligations are not met, the third party would recover its losses via the bond. Every year about $3 billion worth of surety bonds are generated by construction projects compared with a mere $8 million for IT (mostly for governmental contracts); however, there is insufficient commonality and standardization in the IT industry on the bonds. A surety bond works well only for a fixed-fee contract because it provides the benchmarks needed to frame a bond price.

Jackson Lab selected an integrated ERP suite from Oracle rather than a best-of-breed option. The Oracle applications suite included modules for process manufacturing, accounting, e-procurement, and HR, among others. Their biggest challenge was modification of the Oracle Process Manufacturing (OPM) module to accommodate the lab's unique business processes of raising and distributing mice. The OPM module was designed for companies that mix ingredients together to produce such products as bread or beer, not for a lab environment.

The implementation team chose a phased-implementation approach instead of a big bang approach. The first phase initially went live in February, including the management of production capacity, accounts receivable, some general-ledger functions, and the purchasing of manufacturing material; in April, they launched other modules including accounting for research grants, the rest of general-ledger functions, accounts payable, and fixed assets. For the second phase, which began in June, the remaining modules including process management, human resources, payroll, labor distribution, and a grant filing application were installed.

Jackson faced personnel problems during ERP installation when the best and brightest employees were involved in the implementation process, leaving them shorthanded to do the everyday work. In addition, Jackson's IT staff lacked experience with

ERP, only one person had some experience in installing an ERP. Further cost overruns resulted from training, an especially big-cost item. The time-and-materials basis contract would have increased the risk of overtime and going over budget because vendors and consultants have an interest in quoting low and seeing the work grow as the project proceeds. There is a natural competitiveness between the buyer and the ERP vendor. The vendor benefits by placing a "veil of complexity" over their work; the buyer wants to get the system up and running with the least amount of work and customization. The service-level agreements generally tend to be very complex because a much clearer definition of roles and responsibilities between client and service provider is needed. From a consultant's and vendor's perspective, a high (>25 percent) contingency is quite reasonable depending on the nature of the work, whereas this is too much from the buyer's perspective.

PREVIEW

This chapter will discuss the systems development process as it applies to ERP applications. As you saw in the opening case, organizations like Jackson Lab can spend significant dollars beyond their initial estimates to purchase and implement an ERP—and in addition change their business processes to fit the mold of the purchased system. The ERP implementation's success depends significantly on redesigning processes rather than customizing the technology to fit that process. Customization is expensive. Overall it increases the support fees paid for upgrades and prevents organizations from taking advantage of rapid implementations. To overcome these problems, Jackson Lab used several strategies to reduce the risk. First, they negotiated a *fixed-fee contract* with Oracle rather than use the more commonly used *time-and-materials contract.* Second, they bought a *surety bond* from Gladwyne, a risk management consulting firm. This strategy protected Jackson Lab from the project's cost overruns caused by IT staff being forced to solve the technological challenges with "out-of-the-box thinking." Third, it spent considerable time and effort on *changing the business processes and training* its employees with the new processes. Finally, they got better *cooperation from the vendor*, Oracle, by negotiating a fixed-fee contract and a favorable service-level agreement. One additional strategy was that Jackson Lab should have built a time buffer into their ERP implementation life cycle to avoid the pressure of delivering the system on a predetermined timeline.

In general, there are various technical and organizational challenges in implementing ERP depending on the organization, scope of implementation, business processes, and skill level of the people using these applications. The purpose of this chapter is to make you knowledgeable about the ERP life cycle process and to alert you to the implementation challenges by looking at the experiences of other organizations. The chapter begins with a brief overview of the system development life cycle (SDLC). SDLC provides useful guidelines to the ERP implementation process. Next, it discusses the key phases of the ERP life cycle with emphasis on roadblocks in each phase and solutions available to overcome these roadblocks, surveys the different life cycle methodologies and accelerators for ERP implementation, and discusses the key differences between SDLC and ERP life cycles. Throughout the discussions, the chapter provides hints on what roles you should play as an end user and discusses the implications for managers.

SYSTEMS DEVELOPMENT LIFE CYCLE

The process of developing new information systems is often called the system development life cycle. It basically includes a systematic process of planning, designing, and creating an information system for organizations. Even though the process of developing a system for individual or personal use can be simple, the task can become very complex when the system has to support thousands of business processes for several hundred users both inside and outside an organization. For complex systems development projects (e.g., ERP), it is often better to have a structured methodology to avoid mishaps and coordinate the design and development tasks properly among the members of a large systems development team.

SDLC uses a *systems approach* for problem solving that basically states that complex problems need to be broken up into smaller manageable problems using a systems' hierarchy, and then developing a solution for each problem within the hierarchy. It provides a structured top-down problem identification and bottom-up solution process for managing complex problems. The structured or phased approach is designed to catch problems at an early stage before they become a major risk to the system implementation process. The SDLC process requires both technical and nontechnical problem-solving skills; therefore, the development team must understand technology, as well as the organization's business processes, culture, and people (or potential end users of this system). For example, a component of an HR system must capture organizational policy on health care benefits and retirement and the process of deducting the premiums from the payroll checks. Every organization will have some variations that need to be accurately captured and processed by the new system. Capturing these processes and then implementing them in a new system can be difficult for a person with an IT background only; therefore, the development team must be composed of people with a wide variety of IT and business skills for the project to be successful.

Traditional SDLC

In the early days of systems development, very few of these projects were successful in the first attempt. There were many reasons for the early failures, chief among them being lack of experience. This led to the systems approach, which we described earlier, and a structured SDLC methodology. The SDLC consists of tasks that are divided into phases or stages as shown in Figure 4-1. Please read systems analysis and design books for complete details on SDLC.

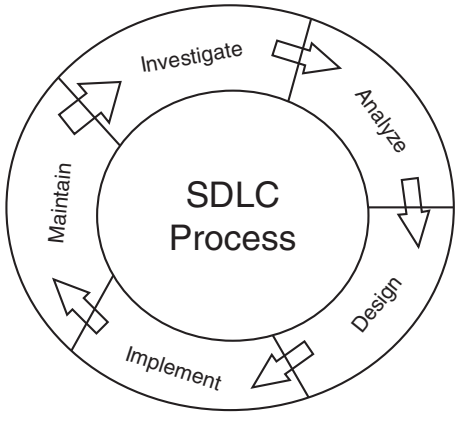

FIGURE 4-1 Traditional SDLC Methodology

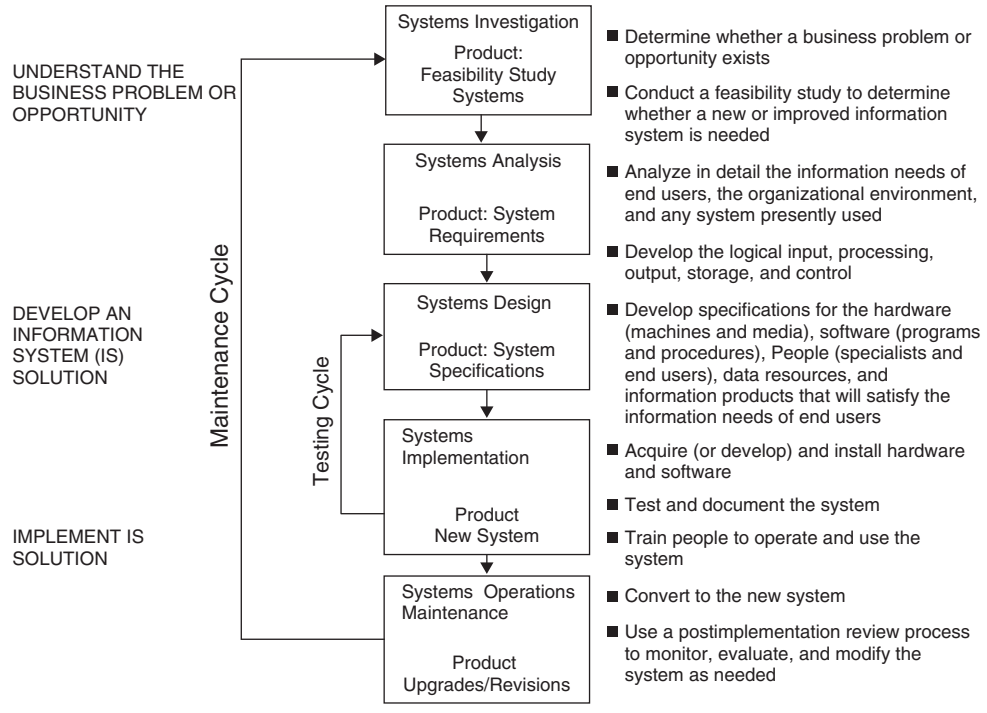

FIGURE 4-2 SDLC Approach

Figure 4-2 provides a summary of the traditional SDLC methodology. The SDLC process begins when someone in the organization identifies a need, or investigation phase, for a new system and ends the implementation phase where the new system is installed and running in the organization.

Rapid SDLC Approaches

The SDLC process has several problems, even though it is rigorous in making sure that the new system is complete and successful in the organization. First, developing a new system is time consuming and tedious. In many cases the new system is outdated by the time it is developed. Second, the cost associated with the SDLC process is very high. The cost of recruiting the development team and involving other members of the organization in the development process can be very expensive. Finally, all information systems do not require such a rigorous SDLC process. For example, the SDLC would be overkill for a small-scale decision-making application; therefore, over the years organizations have used rapid approaches to SDLC that are quicker and less expensive shortcuts to this process. These are called Rapid SDLC approaches.

One rapid development approach is *prototyping* (Figure 4-3). This approach does not go through the analysis and design phases; instead, it implements a skeleton or a prototype of the actual system with a focus on input (i.e., user interface) and output (i.e., screen displays and reports generated with dummy data). The idea is to demonstrate the system functionality as soon as possible to the users and to get their feedback on the prototype. Their feedback is incorporated into the new system and demonstrated back to the users. This approach has proven to be very

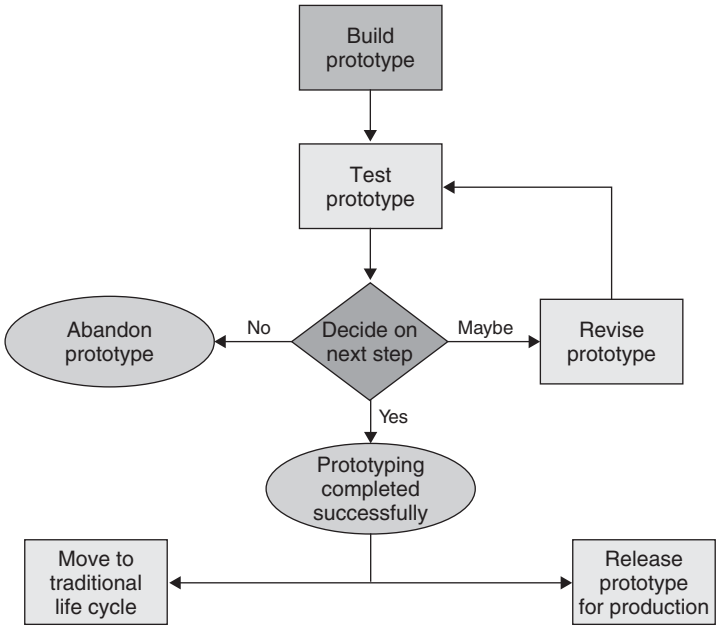

FIGURE 4-3 Prototype Development

effective with user-interactive systems because the prototype is eventually converted into a full-scale system. In ERP implementations, many companies install a sandbox system to expose users to the system functionality. ERP sandboxes replicate at least the minimal functionality needed to get user feedback before implementing a full-scale system. The goal of sandboxing is similar to that of prototyping.

Another rapid development approach is *end-user development* (EUD), which lets the end users create their own applications. This process became popular in the 1980s with the advent of personal computers (PCs). In this process the users are trained by the IT staff or professional trainers to develop customized applications (e.g., a small decision-making application with an Excel spreadsheet or a departmental employee tracking system with an Access database). Several other customized approaches have similarly been developed over the years to circumvent the exhaustive SDLC. EUD is applicable in ERP for designing custom reports from the ERP system.

ERP IMPLEMENTATION LIFE CYCLE

ERP applications are prepackaged software developed by commercial software vendors and custom installed for organizations to automate and integrate the various business processes. Although an ERP is packaged software, it is very different from PC-based software packages (e.g., Microsoft Office or other software) that you may have purchased for personal use as shown in Table 4-1. These are complex software packages costing millions of dollars to develop and maintain that automate hundreds of business processes in an organization. Furthermore, these applications are mission critical (i.e., if they fail or break down, the organization will stop functioning). For example, without these systems a bank would not be able to service its customers for withdrawals or deposits, and a manufacturing company would not be able to assemble and ship their products. Hershey, Corp.,

TABLE 4-1 Differences Between ERP and Other Software Packages

	ERP Software	Other Packaged Software
Software Cost Significance to Organization	Millions of dollars Mission critical	Hundreds to thousands Support or productivity improvement
Installation Time	One to several years	Almost instantly
Change Management Strategy	Requires significant change management strategy from beginning to end for success; business process change, training, communications, etc.	Requires some training and support
Implementation Costs	Requires in-house employee time, consultants, and vendor support in millions of dollars	Requires little or no consulting support or vendor technical support

experienced this problem in real life when they implemented SAP/R3 in the late 1990s when their supply-chain distribution was disrupted, causing a big dent in their holiday sales. Any breakdown of an ERP application can therefore be very disruptive and cost millions of dollars to the organization.

A rigorous ERP life cycle process, though expensive and time consuming, is therefore recommended to ensure success. ERP systems can be deployed in a big bang or phased approach, either of which initiates the stages of a system development life cycle. According to the staged system implementation model,[1] the life cycle consists of four phases—adaptation, acceptance, routinization, and infusion. Adaptation is similar to system investigation, whereas acceptance is similar to system analysis where user requirements are analyzed and accepted by the team before proceeding to design and implementation. Additionally, routinization is where the ERP system is either customized or business processes are changed to assimilate the system in the organization. Once operational, the infusion or maintenance and evaluation phase gets started where recurring problems are fixed and new features are sought for next implementation life cycle.

ERP Implementation Plan

An ERP implementation plan is used to create a roadmap or blueprint to meet cost, scope, and time constraints of an implementation. There are many different ERP implementation methodologies promoted by different vendors and consultants. The appropriateness of the plan depends, in part, on the project, the company, and the reasons for the implementation.

Following are three major implementation plan choices:

1. *Comprehensive.* A comprehensive ERP integration plan is the most expensive, lengthy, and costly approach. It involves implementation of the full functionality of the ERP software in addition to industry-specific modules. Implementing the full functionality requires a high level of business process reengineering (BPR) with major changes in the business processes and customization of legacy systems.

[1] Kwon, T. & Zmud, R. (1987). Unifying the Fragmented Models of Information Systems Implementation, in: R. J. Boland & R. A. Hirschheim (Eds.), *Critical Issues in Information Systems Research* (pp. 227–252z). Chichester, UK: Wiley.

2. *Middle of the Road.* A middle-of-the-road ERP implementation plan involves some changes in the core ERP modules and a significant amount of BPR. The middle-of-the-road approach is not as expensive as the comprehensive approach or as straightforward as the vanilla approach.
3. *Vanilla.* A vanilla ERP implementation plan utilizes core ERP functionality and exploits the best practice business processes built into the software. A company following a vanilla implementation will have to simply align their business processes to the ERP system, rather than modify the software. By eliminating or minimizing the required BPR, the project's costs and time required for the implementation are minimized.

ERP Implementation Methodology

Methodology refers to a *systematic* approach to solving a business problem. ERP methodology builds on the theory that an enterprise can maximize its returns by maximizing the utilization of its fixed supply of resources. Information technology, with its increasing computer power and the ability to correlate pieces of information, has proven to be the best tool for business problem solving. Like SDLC, an ERP development life cycle provides a systematic approach to implementing ERP software in the changing but limited-resource organizational environment.

There are many different vendor-driven methodologies or approaches that use traditional ERP development life cycle or rapid ERP life cycles (e.g., Total Solution, FastTrack, Rapid-Re, Accelerated SAP (ASAP), and business integration methodology (BIM)). Implementation methodologies are similar in their overall approach with the differences coming primarily in the staging of the process steps and formality of structure. The traditional ERP life cycle accomplishes one stage at a time and requires formal milestone approvals prior to moving to the next stage. In a rapid ERP life cycle, once a company commits to the implementation, employees are empowered to make the decisions to keep the project moving forward. They also allow flexibility and quicker feedback loops to accommodate rapid corrections as shown in Figure 4-4.

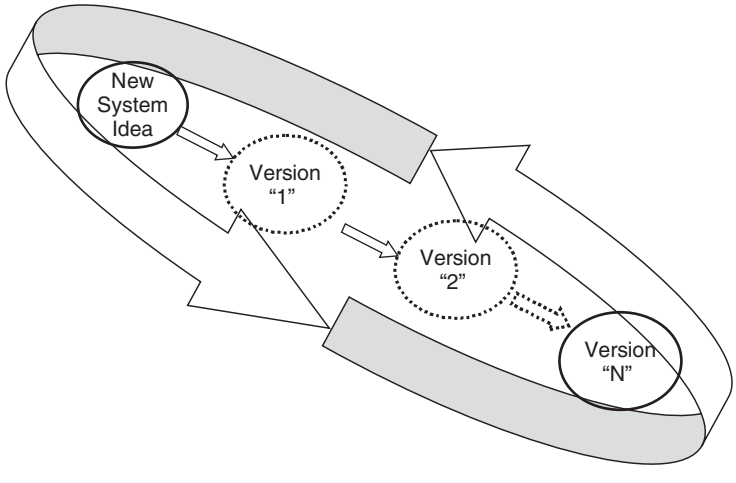

FIGURE 4-4 Rapid Application Development Process

Traditional ERP Life Cycle

Like the traditional SDLC, which we discussed earlier, the traditional ERP life cycle approach has a deliverable at the end of each stage (e.g., a report with supporting documents) that is reviewed by management and upon which a decision is made either to continue with the project or not. End-user or people involvement is critical in both SDLC and ERPLC; however, there are other variations to the traditional SDLC process. The emphasis in ERP implementation is on customizing the software as well as on changing the organization's business processes, rather than determining the user requirements for developing new applications (as in the traditional SDLC). This may seem like a small deviation, but it requires a major change in the thinking process as well as team composition and skill level of people involved in the development process. Furthermore, the ERP life cycle, as shown in Figure 4-5, iterates at a much faster pace than in the traditional SDLC.

The traditional ERP life cycle includes the following major stages:

Stage 1. *Scope and commitment stage.* This is similar to the investigation stage in SDLC discussed earlier. In addition to conducting the feasibility study, however, one of the first steps is to develop a scope of ERP implementation within the resource and time requirement. A number of task parameters or characteristics of the ERP implementation

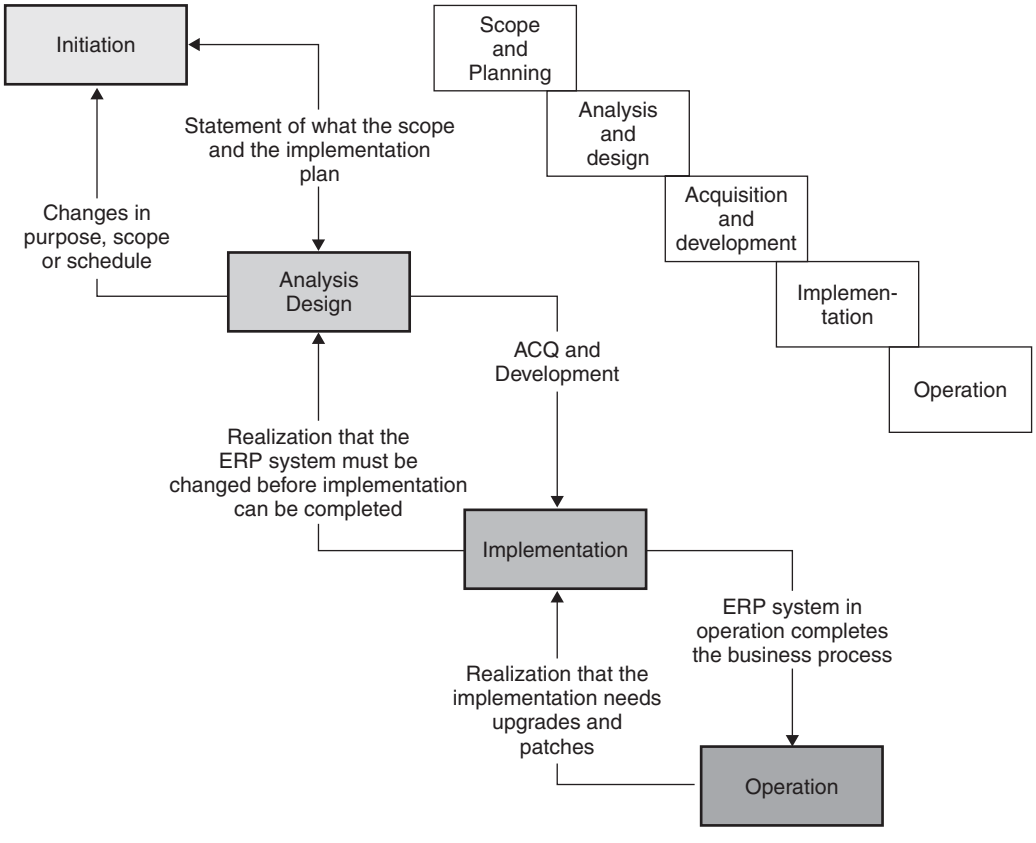

FIGURE 4-5 Traditional ERP Life Cycle

need to be defined at the planning stage. How large will the ERP system scope be in terms of departmental or functional coverage? Develop a long-term vision for the new system and a short-term implementation plan and top management's commitment for both the vision and implementation plan. The composition and the structure of the implementation team, the role of external consultants both in terms of time and scope, and the role of internal employees, including the subject matter experts (SMEs) who will provide the knowledge to embed business rules and input for interface and report design, are other key factors to be considered at this stage. Vendor selection is another key activity toward the end of this stage. Although no decisions should be made on the ERP software, vendor information must be reviewed and choices could be narrowed by testing alternative software and developing a business case for the project. A number of items need to be assessed and established to create the boundaries and scope. Table 4-2 lists the key decisions to be made for each type of scope.

Stage 2. *Analysis and design stage.* In addition to analysis of user requirements, the ERP team has first to make a decision on the software and decide on consultants and SMEs. Another key activity is to map the differences between the current business process and the embedded process in the ERP software or gap analysis and to develop a long-term plan on whether to change the business processes of the organization or to customize the ERP software to support existing processes. Using the gap analysis, the team must develop a design that among other things includes a change management plan, a list of embedded processes, user interface screens, and reports in the ERP software that will need customization, design of these changes, and a process of involving subject matter experts in the design. Other activities include creating plans for data conversion, system conversion, and training. For a system to be successful, the team must develop a detailed change management strategy and plan for the release of the new system. By the end of this stage, the team usually has a sandbox or prototype of the ERP software installed that is accessible to the entire implementation team, consultants, and SMEs.

Stage 3. *Acquisition and development stage.* This stage is similar to the acquisition and testing stage of traditional SDLC. The organization has to purchase the license for the production version of the software and build the production version of the system, which

TABLE 4-2 List of Scopes and Commitments

Scope Type	Description/Key Decision Points
Gap Analysis	Gap analysis is the evaluation of the functions provided by the ERP system compared with the operational processes necessary to run your business
Physical Scope	Establishes which sites will be addressed, the geographical locations of the sites, and the number of users.
BPR Scope	Will the current processes be refined, replaced, or eliminated. What users, departments, sites will be affected?
Technical Scope	How much modification will be done to the ERP software? What processes will be utilized as is and which will be customized?
Resource Scope	How much time and budget is allocated for the project?
Implementation Scope	Which modules should be implemented? How should the modules be connected to the existing system?

is eventually to be made available to the end users. The entire production platform must be configured and built with the necessary hardware, network, security, software, database, and real production data. The tasks identified in the gap analysis are executed at this stage. These include customization of embedded software rules, data in the database tables, input screens, and reports that come with the ERP system. While the technical team is working on the installation, the change management team works with end users on implementing the changes in business processes and preliminary training with the sandbox version of the software. The data team similarly works on migrating data from the old system to the new system. This can be an extremely difficult task when the old system is a legacy application using a nonrelational database. Data mapping, missing data, and data dictionary design are the major tasks for data conversion. Finally, the ERP system needs to be configured with proper security, implement the authentication and authorization policy for accessing the system, and contain other modifications as recommended by the design plan.

Stage 4. *Implementation stage.* The focus for this stage is on installing and releasing the system to the end users (i.e., "Go-Live") and on monitoring the system release to the end users. This production platform is a mirror of the development version of the system. Errors found in the production version have to go through the help desk or support staff. Any changes made to the development version are then retested and migrated to the production system as regularly scheduled updates. System conversion is a major activity for the new system and needs to be managed carefully. There are four basic conversion approaches, which are visually represented in Figure 4-6. The first

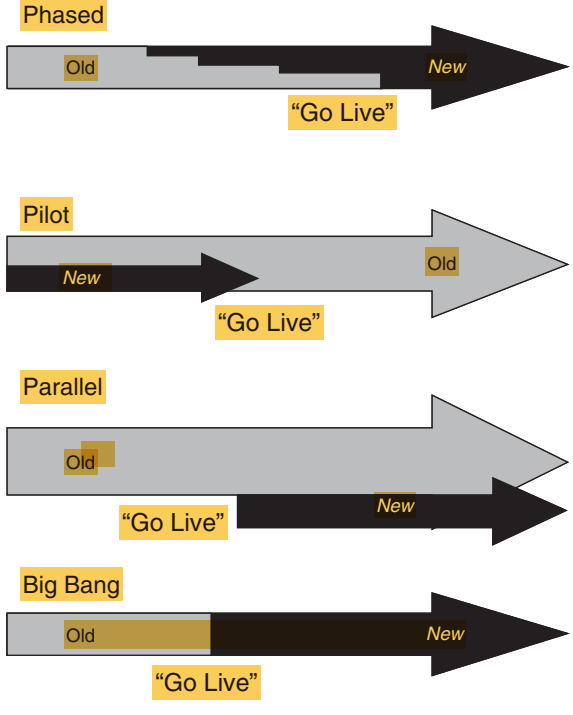

FIGURE 4-6 ERP Conversion Approaches

approach, *phased*, is a gradual movement of the company from the existing legacy system(s) to the ERP implementation. This approach can take a significant amount of time, but can also be the least disruptive to the company. The second approach, *pilot*, implements a small version of the final system. This pilot system is used to ensure that the final system is appropriate. It is the equivalent of a test drive in that the system is used, but only by select areas, and its impact can be managed more closely. The third approach, *parallel*, has the most up-front cost because the ERP system is implemented and used in conjunction with the legacy system. This approach is best used when risk of ERP failure is of significant concern. The final approach, *direct cutover* or big bang, is the highest-risk approach but the most straightforward and clean. The company moves from the legacy system directly and immediately to ease the ERP system. This approach has the least amount of up-front costs because systems are not duplicated or run concurrently for any length of time. Training end users on how to use the new system is another important activity. Training is generally part of the change management strategy designed to ease the transition to the post-implementation environment. Feedback received from system usage needs to be funneled to the post-implementation team for ongoing system support, including upgrades and patches, as well as to make adjustments to the change management strategy.

Stage 5. *Operation stage.* This is often managed by the operation team with assistance from the implementation team. Knowledge transfer is the major activity as support for the new system is migrated to the help desk and support staff. Some implementation team members are very often hired as support staff. The other major activities are ongoing training of new users to the system as ERP modules are released, as well as to take a fresh look at the change management strategy. The team has to monitor user feedback from training and actual system usage carefully and make the necessary adjustments to the change management approach. Another key activity is management of new releases of the software, installation of patches and upgrades to the system, and managing the software contract with the ERP vendor.

A summary of ERP life cycle phases is shown in Figure 4-7.

ROLE OF CHANGE MANAGEMENT Change management (CM) plays an important role throughout the ERP life cycle. System failures often occur when the attention is not devoted to this from the beginning stages. A vision for CM needs to be articulated from the first stage and then revised, monitored, and implemented on a constant basis. A major role of the SMEs and other internal users working with the team is to guide the implementation team on all the activities of change management, including guidance on what processes need changing, customization of business rules in ERP software, input screen design, report design, and training and communications plan for the end users affected by the new system. Support of the top management as well as skills of the change management team are essential for successful implementation. Change management strategy and activities are discussed in detail elsewhere in this book.

Rapid ERP Life Cycles

ERP implementations are usually very long. They usually start with a long requirements-gathering phase, followed by designs, and implementations. That means that significant amounts of time (months to years) could go by between the time the requirement is given and the time it is

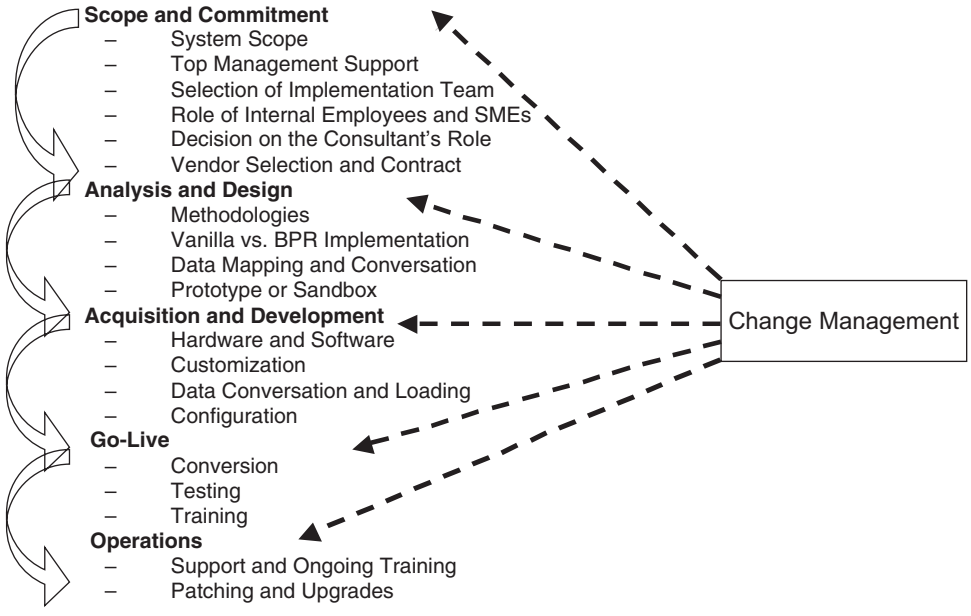

FIGURE 4-7 ERP Life Cycle Phases Summary

actually implemented. As businesses grow and move quickly, there is a high chance that the requirement will change during those months or years of time that pass.

Consultants play an important role in rapid implementation of ERP systems. They provide different methodologies and techniques for rapid or accelerated implementation. This is an area where the use of experienced consultants can best be leveraged as they bring knowledge of techniques and approaches that have worked well with other organizations. Scripts and wizards provided by consultants can help automate some of the more common tasks that occur during an implementation. These include migration of data, identification of duplicate data, and other standard tasks.

This section will give you a sample of methodologies offered by ERP consulting firms. The appropriate implementation model may vary based on company, culture, software, budget, and the purpose of the implementation, but previous implementation experience of the program management and consultants will likely be the largest driving factor in determining the best approach. With that said, the high commitment of resources required in the first stages of the implementation and the time and cost saving claimed by rapid implementation approaches have drastically increased the number of rapid implantations. Some examples of methodologies used in implementing ERP systems follow.

TOTAL SOLUTION ERP packages have increasingly become indispensable to run businesses; yet, ERP implementations often fail because these solutions are highly integrated, demand cross-functional collaboration, and require significant change management. Although ERP solutions have matured and stabilized over the years, they are still difficult to implement and manage. Ernst & Young, LLP, have developed a systematic way of approaching systems reengineering called the *Total Solution.* The Total Solution approach has five components:

1. *The value proposition.* Building the business case for an ERP solution. The key decision to be made before any process can begin is to make sure that the ERP solution makes sound business sense. The following questions should be answered before the process is started:
   - Is the investment in new technology justified?
   - Does the ERP solution match the company's objectives?
   - Does management understand what change means, and does that change have full support?
   - What is the framework for making decisions?
   - What milestones will measure the project's progress?
   - Is value being delivered throughout the process?
2. *Reality check.* Assessing an organization's readiness for change. Since many people oppose change, it's something that needs to be anticipated. Status quo is easy. Change is not. The following questions therefore need to be asked:
   - Is the organization ready for change?
   - Are there any "hidden agendas?" If so, how will they be managed?
   - Is everybody on board with the nature, scope, and pace of the change?
   - What are management's expectations?

   The answers to these questions will adjust the implementation approach. Knowing the answers up front helps to avoid a possibility that the change is incompatible with the client's expectations.
3. *Aligned approach.* Setting the right expectations that deliver both short-term and long-term value. Short-term as well as long-term benefits are equally important to any project's success. Even if change is discomforting for some, it is easier to accept when progress is visible. In this approach, the following tasks are performed:
   - Evaluate alternatives to a comprehensive reengineering project.
   - Craft a "best-fit" approach that allows the implementation to proceed in well-defined modules.
   - Communicate expected results to management. Keep communicating throughout the project so no surprises surface at the end. This approach helps keep the entire project on time, on budget, and on management's agenda for success.
4. *Success dimension.* Getting the right blend of people, skills, methods, and management in the team. The key to any project's success is having the right mix of people, skills, methods, and management (i.e., people with diverse skills in process management, change management, knowledge management, and industry skills). Teamwork is very important to a project's success.
5. *Delivering value.* Measuring results and celebrating success. A project that does not show measurable results throughout the process is going to flounder. People will lose enthusiasm and the expectation of a new way of doing business becomes just another broken promise. Total Solution methodology makes sure that every project pays continuous "value dividends" all along the way and helps to minimize the risk of change.

FASTTRACK Whether your business objective involves global reengineering, process improvement, or software replacement, Deloitte & Touche Consulting Group's FastTrack implementation methodology can enhance and accelerate ERP software implementations. The *FastTrack* approach developed by Deloitte & Touche is based on a matrix of five phases and five focus areas:

Phases Designed to reflect and integrate decisions regarding business redesign, organizational change and performance, training , process and systems integrity, client–server technologies and technical architecture.

Stage 1. *Scoping and planning:* Project definition and scope. Project planning is initiated.
Stage 2. *Visioning and targeting:* Needs assessment. Vision and targets identified. As-is modeling.
Stage 3. *Redesign:* To-be Modeling. Software design and development.
Stage 4. *Configuration:* Software development. Integration test planning.
Stage 5. *Testing and delivery:* Integration testing. Business and system delivery.

Areas In addition, it identifies five areas (groups) as an individual thread to be woven into a cohesive fabric through its five-phase work plan. The areas and a list of the functions performed are as follows:

Stage 1. *Project management* (project organization, risk management, planning, monitoring, communications, budgeting, staffing, quality assurance).
Stage 2. *Information technology architecture* (hardware and network selection, procurement, installation, operations, software design, development, installation).
Stage 3. *Process and systems integrity* (security, audit control).
Stage 4. *Change leadership* (leadership, commitment, organizations design, change readiness, policies and procedures, performance measurements).
Stage 5. *Training and documentation* (needs assessment, training design and delivery for project team, management, end users, operations, and help desk. Scripting of end-user and operations documentation).

Rapid-Re Gateway, a consulting firm in New York, has developed an ERP life cycle methodology called *Rapid-Re.* The five-stage, 54-step modular methodology is customized to the needs of each project because that is what happens in practice. Individual projects skip, rearrange, or recombine tasks to meet their needs or give greater or lesser emphasis to some tasks.

Stage 1. *Preparation.* Mobilize, organize, and energize the people who will perform the reengineering project.
Stage 2. *Identification.* Develop a customer-oriented process model of the business.
Stage 3. *Vision.* Select the processes to reengineer and formulate redesign options capable of achieving breakthrough performance.
Stage 4. *Solution.* Define the technical and social requirements for the new processes and develop detailed implementation plans.
Stage 5. *Transformation.* Implement the reengineering plans. In an ideal project, stages one and two consider all key processes within a company and conclude with a step that sets priorities for the processes to reengineer. The other stages are executed repeatedly for each process selected for reengineering.

ACCELERATED SAP (ASAP) The ASAP roadmap is a detailed project plan by SAP that describes all activities in an implementation. It includes the entire technical area to support technical project management, and addresses such concerns as interfaces, data conversions, and authorizations earlier than do most traditional implementations.

The ASAP roadmap consists of five phases—project preparation, business blueprint, realization, final preparation, and go-live—and supports continuous change.

Phase 1. *Project preparation.* Proper planning and assessing organizational readiness is essential. Determine if there is a

- full agreement that all company decision makers are behind the project
- clear project objectives
- efficient decision-making process
- company culture that is willing to accept change

ASAP's project estimator can be used to guide the project team through a series of predefined questions and to drive interviews with senior executives and key operating managers about their expectations of R/3 and the speed of its deployment.

Phase 2. *Business blueprint.* The engineer delivers a complete toolkit of predefined business processes. During the business blueprint phase R3's broad scope is narrowed to fit the industry-specific processes. Using questionnaires and the models from the business engineer, the business processes are documented to reflect the future vision of the business. Industry templates further accelerate the process by predefining industry best-business practices. The result is a comprehensive blueprint of the business. During this phase, training begins on R3's integrated business systems. Level 2 hands-on training provides a step-by-step education of R3 business process skills. The business blueprint is a visual model of your business' future state. It will allow the project team to define the scope clearly and only to focus on the R3 processes needed to run the business.

Phase 3. *Realization.* Based on the business blueprint, a two-step process is begun to configure the R3 system. First, the baseline system will be configured. Second, the system is fine-tuned to meet all of the business process requirements. Because the initial configuration is based on the blueprint, the baseline system gives a real-worldview of how the business transactions will actually run.

Phase 4. *Final preparation.* In this phase, the R3 system is fine-tuned. Necessary adjustments are made in order to prepare the system and the business for production start-up. Final systems tests are conducted, and end-user training is completed. Initial audit procedures are developed.

Phase 5. *Go-live and support.* In this phase, procedures and measurements are developed to review the benefits of the R3 investment on an ongoing basis. SAP support and services are provided to ensure that the system continues to run smoothly. The online service system (OSS) provides electronic support using a remote connection. The implementation assistant provides answers for most of the questions that may arise. It is an easy-to-use repository of information defining what to do, who should do it, and how long it should take.

ASAP provides examples, checklists, or templates as samples for things such as a cutover plan. They are used as a starting point to avoid "reinventing the wheel." ASAP calls these things "Accelerators."

BUSINESS INTEGRATION METHODOLOGY (BIM) The BIM, developed by Accenture Systems in the 1990s, is targeted for full-scale ERP projects that diagnose business integration needs, design business strategies and architectures, deliver one or more business capabilities to meet those needs, and ensure that the value of those capabilities can be sustained over time. This includes the strategic planning, delivery, and operation of technologies, processes, facilities, and human performance. To achieve business integration, a team must define and implement a

comprehensive set of changes to an organization, spanning improvements to business processes, technology, and human performance, all aligned with an organization's overall strategy.

This methodology is best suited for full life cycle projects where organizations expect to involve either custom-built solutions or a blend of custom and packaged components. In addition, it is intended for use on medium to large projects that implement a full life cycle custom-built BI solution and comprises the BIM content areas that are relevant to custom solution planning, delivery, and operations:

The planning phase. The objective of this phase is to help an organization define appropriate strategies and approaches for achieving an enduring competitive advantage and building stakeholder value. The planning phase defines new and improved business capabilities to support the organization's strategies and creates detailed plans to help the organization effectively and efficiently implement changes—and realize and sustain value—during the delivering and operating phases.

The delivering phase (aka the standard or custom route). This phase translates the business architecture into a specific business capability. A business capability is the combination of human performance, business process, and technology that collectively creates value by improving business performance. The delivering phase defines a cross-competency approach for taking each business capability from blueprint to deployment.

The managing phase. This phase directs, coordinates, and monitors the activities outlined in the other three phases, in order to achieve improved business results. This phase determines whether the proposed business values were achieved; the projects and change journey were effectively managed; there was an ongoing alignment of context, content, and course of action; the necessary levels of ownership, sponsorship, commitment, and leadership were achieved; and the program sponsor or stakeholder expectations were met or exceeded.

The operating phase. This phase operates the new business capabilities that were created in the delivering phase. Operating is based on the definitions of sourcing strategies, service providers, and customers, which were established in the planning phase. The work in this phase must meet the formal service targets and metrics established in earlier phases, and it must provide feedback for improvements based on measurements of actual performance against those targets.

AGILE DEVELOPMENT The agile development methodology has gained popularity over the past decade for traditional software development. This success has helped this methodology migrate to ERP implementations. Here is a key reason for this success:

> *At the core of any agile approach is an assumption that whatever the requirements might be at the beginning of a project, they won't be the same at the end of the project. The longer the project, the more truth there is in this assumption. To mitigate this situation, agile methodologies start with smaller sets of requirements, they start small and deliver functionality incrementally in a series of releases. No single release covers all requirements, but every release delivers more than the previous one.*[2]

[2] *Is agile ERP implementation possible?* (March 11, 2009). Retrieved on October 2010 from http://community.dynamics.com/product/nav/navnontechnical/b/navigateintosuccess/archive/2009/03/11/is-agile-erp-implementation-possible-63.aspx

Therefore, the key problem that agile methodology tries to solve is changing and unclear requirements. By developing smaller portions of the ERP system and releasing them to users, the users will be able to provide feedback quickly on how this system meets their needs. This also means that if a requirement is missing or wrong, it can be corrected quickly (often in one iteration, which may be between two and six weeks), rather than requiring a long and expensive process. Another key aspect is that new components are not built upon building blocks that are wrong. In a traditional waterfall implementation, if the first requirement implemented is wrong, an entire ERP system could be implemented upon that basis. This means that if one thing needs to be changed, it could cause the entire system, or at least a large portion of the system, to be changed. This is very expensive and time consuming to undertake and should be avoided at all costs.

An agile methodology has many different types of implementations. Two of the most popular implementations are Scrum and extreme programming (XP). In Scrum, the methodology states the following: "Instead of providing complete, detailed descriptions of how everything is to be done on the project, much is left up to the team. This is done because the team will know best how to solve its problem."[3] The Scrum methodology works in an agile iterative methodology; however, it empowers the team (including a Scrum master, product owner, and team members) to make decisions that will help deliver a successful product deliverable. "Extreme Programming (XP) is successful because it stresses customer satisfaction. Instead of delivering everything you could possibly want on some date far in the future this process delivers the software you need as you need it."[4] In XP, an agile iterative approach is valued because it gives working software to the customers quickly and incorporates their feedback quickly to build the best product possible.

OTHERS There are industry-specific rapid implementation approaches (e.g., those available from the Cobre Group's consulting firm called *Implementation Accelerator*). This accelerator facilitates conversions and upgrades by providing a tool to use for data mapping, workflow analysis, project planning, and end-user training.

Another example is from the Chemical Industry Data Exchange. They have an implementation accelerator that is divided into phases: plan, assess, enable, test, and go-live. Each phase has specific tools, templates, and real-world suggestions contributed by members.

ERP Life Cycle Vs. SDLC

Because of their prepackaged nature, ERP applications generally do not require the traditional SDLC process; however, that does not mean they can be bought from the vendor, installed (i.e., a PC-based software package), and used *as is* immediately. ERP packages are complex with embedded business processes in all major functional areas of business. In addition, they represent best practices by industry or area of business. These processes and functional activities are generic in nature and must therefore be adapted for the specific requirements of the company. This is not an easy course of action. It requires a thorough understanding of the business process of the company, data requirements, informational flows, system access and security, integration with existing software applications, and compatibility with current hardware systems of the company. The ERP life cycle is often therefore as rigorous as is the traditional SDLC life cycle;

---

[3] *Introduction to Scrum—an agile process.* (2010). Retrieved on October 2011 from http://www.mountaingoatsoftware.com/topics/scrum

[4] Wells, D. (September 29, 2009). *Extreme Programming: A Gentle Introduction.* Retrieved on October 2010 from http://www.extremeprogramming.org/

TABLE 4-3 Comparing and Contrasting SDLC with ERPLC

	SDLC	ERP Life Cycle
Goal	Develop a new system to support the organization requirements	Implement a packaged system to support the organization requirements
Analysis	Evaluate user needs through observations and interviews and create system specifications	Vendor analysis and evaluation of business process changes due to the implementation
Design	Develop new system architecture, user interface, and reporting tools	Installation and customization plan of ERP software, data conversion, and change management strategies
Implementation	Acquire hardware, software, develop applications, installation, testing, training, and conversion	"Go-Live" conversion or releasing the system to the users, training, and support
Consultant Role	Technical support mainly during design and implementation	Change management, process change, and technical support from beginning to end
Management Role	Some oversight and support	Significant oversight and involvement—especially in change management
End-User Role	Focus group providing input during the various stages with most involvement during implementation stage	Multiple groups such as SMEs, advance users, and self-service users are part of implementation team with continuous involvement
Operations	Maintains, updates, and provides technical support	Maintains, updates, upgrades, and monitors change management strategy

however, there are also differences due to the prepackaged nature of the software as shown in Table 4-3. Some of the key differences are as follows:

- SDLC does not mention software acquisition until the fourth stage, whereas in the ERP life cycle, the ERP software must be selected at a very early stage of the implementation process. One of the key early decisions in the ERP life cycle is software or vendor selection. ERP vendors have traditionally embedded the best practices and business rules in their software. Some vendors specialize in certain industries. Understanding the ERP software's functionality and the embedded business processes are therefore crucial for successful implementation. A good match between the company's business process and software's embedded functionality means quicker implementation and millions of dollars saved in implementation costs.
- In SDLC the new application is custom designed based on the user requirements as determined from the feasibility study and analysis. On the other hand, in the ERP life cycle the new application is bought by the organization and users are asked to change their business process and policy to take advantage of the best practices embedded in the ERP software. The emphasis in ERP life cycle is more toward reengineering organizational process and change management to improve productivity and create efficiencies with the help of embedded functionality of the ERP software.
- Another difference is in the role of external consultants in the ERP life cycle. In traditional SDLC, the consultant's role is limited to IT hardware, software, and training. Most of the team is made up of people from inside the organization. Consultants play a very important role right from beginning to end during ERP installations advising the organization on software vendor selection, business process reengineering, software installation, and change management.

There are similarities between the ERP life cycle and SDLC. For example, the feasibility stage in the ERP life cycle is similar to the SDLC. ERP implementation requires scoping the project requirements and conducting a proper feasibility study from operational, economic, technical, and strategic perspectives like any other system. The new ERP system must be strategically aligned with an organization's long-term strategy and vision. Top management support will be available only when the new system fulfills the long-term vision of the company. With top management support there is also a long-term commitment of resources for the project. The ERP implementation life cycle is an expensive long-term investment for the company with a return on investment that is intangible and spread over a long period of time. In addition to the feasibility, other similarities occur at the conversion stage. The company can either go for a phased approach or a big bang conversion that replaces the old system with the new on a fixed date and time. ERP implementations similarly require extensive data conversion, proper software testing and quality assurances, end-user training, and post-implementation IT support in terms of installing software patches and product upgrades.

PROJECT MANAGEMENT

ERP projects take on their own organizational duties and job functions separate and apart from the day-to-day business functions. A clear project plan and reporting structure will better ensure that the project receives the attention and accountability needed to be successful. Figure 4-8 shows a sample organization structure for an ERP implementation.

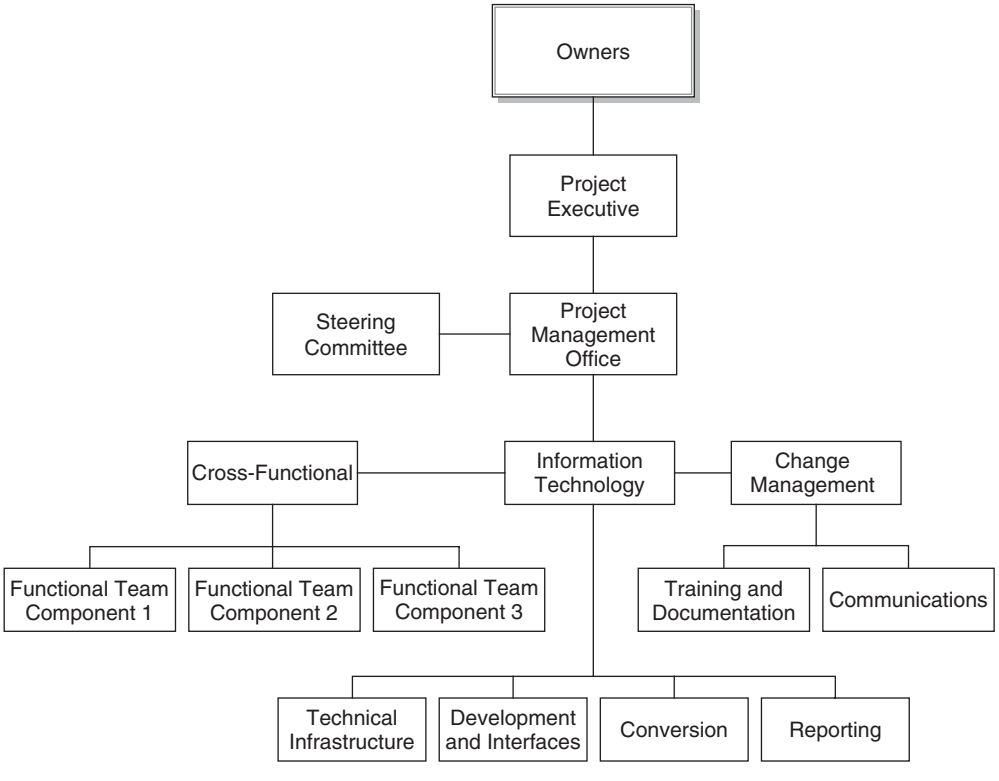

FIGURE 4-8 Project Organization

The organizational structure must coincide with the project governance. The project owners, a project steering committee, and project executive must develop the hierarchy and determine who is responsible for each system implementation component. A number of staff is most often assigned to the implementation from existing business organizations. It is recommended that organizational structures for the functional and technical staff on the project report to project leads and not to their home departments. The ERP implementation project should provide "backfill" staff to departments that have assigned staff to the project to ensure business can continue without interruption and free up staff assigned to the project to work full time on the project without also having to continue with duties from the home department.

As is often the case, ERP implementation project organizations are created just for the project; however, this is changing. Companies are finding more and more that a modified project organization is needed to support the ERP and ensure ongoing progress. Many businesses now have a project management organization within IT to provide the project management necessary for company projects. The functional, technical, and change management staff for the project will likely consist of existing staff from the business, new hires, and consultants. The creation of the project organization and having them work together as a team takes effort and should not be overlooked. The staff in each area must be skilled to accomplish the tasks assigned to them and to develop a strong sense of project teamwork for the overall success of the ERP implementation. The development of the team is the responsibility of the project management office along with the leads in each area.

The organization structure is divided into several areas, the first of which is the project management office. This group can consist of many staff, and their goal is to ensure that the project is moving forward and meeting the implementation goals and objectives. In major ERP implementations, this group will consist of the project executive, project manager(s), project schedulers(s), and administrative staff. The project office works closely with the leads in each area to measure progress. Project measurement, even though at times seemingly tedious, is the only way to identify if a project is meeting deadlines and staying on schedule. Although anecdotal or qualitative data are good, quantitative data provide insights to the project that would not surface if not in place.

The next level within the project organization is the leads that manage a project team. The leads are a direct link to the staff working on the project. They provide the input to management and coordinate team activities. Lead positions are sometimes the most difficult positions to fill on a project. The combination of management skills and in-depth knowledge of a functional or technical area is not easily found.

The teams are next, and they are broken out into several logical areas: The *functional* team(s) is usually broken out by ERP functional components with a team of knowledgeable staff in each area. The key to this area is an overall lead to ensure that the cross-functional issues are being addressed. It is important to remember that ERP systems will cross functional boundaries. It is the cross-functional lead's responsibility to ensure that the data and procedures that cross organizational boundaries are addressed during the implementation. The technical area will consist of an *infrastructure* team that installs and implements the hardware and software on which the system will run, including servers, operating systems, network gear, and operational software.

The *development* team will work with the ERP system to develop or modify the software to meet the goals of the system. This team will vary in size depending on the decision to modify or not modify the system. The *conversion* team is critical. Conversion needs both knowledgeable legacy system staff on it and a knowledgeable ERP staff to convert the legacy data to the new system accurately. Conversion is usually the most complex and time-consuming activity within the project (i.e., critical path). The last area is reporting. The retrieving of data and reporting information out of

the new system has a steep learning curve. The *reporting* team will need to develop a reporting framework and initial set of reports that need to be included in the system implementation. As with most conversions from a legacy system to an ERP there will be hundreds, sometimes thousands, of reports that have been written for a variety of purposes. Identifying and prioritizing the reports is a job in itself. You will find that many of the reports will not be necessary in the ERP system so prioritizing reports and even eliminating reports from the inventory for a Go-live is essential to ensure that the company's needs are being met with the implementation. As the implementation progresses, the reporting team will need to continue to develop reports and/or provide the tools and training necessary to easily retrieve data from the system.

Last, the *change management* team will be the glue that provides the link back to the company and keeps the project in the forefront. Change management will develop both the training plan and a communications plan for the project. Their role is to provide project implementation information to key areas within the organization. Change management should use all communication methods available to them to ensure the message is getting out. This includes e-mail, Web, letters, and presentations. Presentations are especially helpful because they are much more interactive.

PROJECT ROLES AND RESPONSIBILITIES

Identifying and describing roles and responsibilities for project staff is necessary to ensure there is accountability within the project. Project staff will be made up of a variety of people from different parts of the organization along with external staffing (consultants). Defining roles, often used as job descriptions on a project, will be the responsibility of the project management office. Each member of the project team will need to know what is expected of them, who they will report to, and what they will be evaluated on. Table 4-4 is an example of project roles along with the defined activities for each and the skills to fill that role. Note that there will be more detail to the information in the table in Appendix A in this chapter.

IMPLICATIONS FOR MANAGEMENT

ERPs are becoming more and more ubiquitous in the business landscape as they become critical to the long-term positioning and success of today's businesses. As companies look to ERP systems, there is a steady stream of painful and, more times than not, unsuccessful attempts at implementations. The strategy used for the implementation along with the ancillary decisions as to what implementation accelerators to use and which, if any, third-party applications to select all factor into the overall approach. There are several areas that positively increase the chances of implementation success.

First and foremost, it is critical to have solid top management commitment. ERP implementations typically address fundamental business operations. If senior management is not committed to the project, it will eventually lose backing and fail.

Given the complexities of ERP implementations, it is also important to have strong and experienced program management. Program management is the glue that keeps the project together and provides leadership. If the project does not have a strong leader, it can flounder and not achieve goals within the boundaries of scope, schedule, and cost. Along with experienced program management, it is important to have experienced ERP consultants. By using people who have previously implemented the technology, the project leverages the knowledge gained and starts further up the learning curve.

TABLE 4-4 Summary of Roles for ERP Implementation

Project Role Title	Role Definition
Owners	The owners will consist of senior management in the company. The chair is empowered to make decisions when the owners cannot reach consensus. The owners determine overall policy, budget, and scope of the project. The owners meet when needed at the call of the chair.
Project Executive	The project executive oversees project activities, provides broad project oversight, resolves policy level issues, and ensures that the project stays within scope. The project executive also builds consensus on business process changes that impact the business and provides project status updates (as needed) to the owners. The project executive works with the application steward and project manager(s) to establish overall project direction, review and evaluate project progress, and ensure appropriate user involvement for the duration of the project.
Project Manager(s)	The project manager manages the day-to-day aspects of the project, ensures that the project plan is being followed, and keeps both team members and the project executive aware of the status of the project. This responsibility includes overall management of the project to ensure that all tasks are completed on a timely basis, in a quality fashion, and in accordance with the approved project plan. The project manager serves as the primary liaison between the project team and project executive.
Functional Module Leads	The module leads provide leadership and overall direction for the implementation at the module level, ensuring the quality of deliverables and adherence to the project plan and milestones.
Technical Infrastructure Team Lead	The technical infrastructure lead is responsible for overall technical infrastructure implementation including organizing and directing the efforts of the team, coordinating technical team meetings, and reporting on progress to project executive and project manager(s). The technical infrastructure lead must understand all of the technologies well enough to ensure that there will be no "gaps" in the solution and that all of the technologies will be integrated.
Development Lead	The development lead is responsible for managing the development team in the design and implementation of all modifications including reports, interfaces, online changes, and batch programs.
Change Management Lead	The change management lead coordinates the overall change management effort, including training, communication, and campus readiness activities. The change management lead is responsible for developing strategies and detailed work plans, monitoring progress, and resolving change management issues.
Conversion Lead	The conversion lead coordinates the overall conversion effort. The conversion lead is responsible for developing strategies and detailed work plans, monitoring progress, and resolving issues.

TABLE 4-4 Continued

Project Role Title	Role Definition
Reporting Lead	The reporting lead coordinates the overall reporting effort. The reporting lead is responsible for developing strategies and detailed work plans, monitoring progress, and resolving issues.
System Test Lead	The system test lead coordinates the overall system testing effort. The system test lead is responsible for developing strategies and detailed work plans, monitoring progress, and resolving issues.
Module Team Members	Module team members are responsible for analyzing requirements and converting them into solutions. Module team members provide direction and application knowledge with respect to business process design, configuration, testing, training, and implementation. Module team members are composed of the functional and technical resources from the project and campuses and report to the module leads.
Technical Infrastructure Team Members	The technical infrastructure team members are responsible for supporting the module teams throughout the design and implementation of the ERP software. The technical infrastructure team members are responsible for designing and building the necessary architecture components to supplement the delivered ERP technical solution.
Development Team Members	Development team members are responsible for developing the designed solutions necessary to meet business requirements. Development team members provide direction and ERP technical knowledge to the client team with development of modifications and interfaces.
Change Management Team Members	Change management team members are responsible for completing the tasks related to the training development and delivery.
Conversion Team Members	Conversion team members are responsible for designing and developing conversion programs (temporary and permanent) necessary to convert the legacy system data into the ERP database.
Reporting Team Members	Reporting team members are responsible for designing and developing the reports needed to support the system.
System Test Team Members	System test team members are responsible for completing the tasks related to system test effort.
Subject Matter Experts (SMEs)	Subject matter experts are responsible for ensuring that business-specific requirements are addressed in the design, built, and the system is tested. SMEs provide coordination and facilitation of communications between the project team and the organization. SMEs coordinate and prioritize functional requirements. SMEs provide leadership and functional expertise in support of the implementation with specific knowledge in one or more business processes.
Project Administrative Support	Project administrative support is responsible for providing administrative to the project.

In order to reduce the chances of unexpected and unpleasant surprises, it is a good heuristic to minimize the type and number of customizations that are implemented. Any change is a chance for unexpected and unwelcome surprises and increases the chance of risks becoming reality. It is also important to empower team members. The more each member of the team can do, the greater the amount of work that is performed. As part of empowerment, however, it is important to keep a focus on processes to ensure that the right activities are occurring, the right groups are involved, and a methodology is being followed.

Along with the actual implementation, it is critical to emphasize training and change management. The implementation is not successful if the system is not used to its fullest extent. It is also important to have significant and strong post-implementation support. The job is not done once the system goes live. ERP applications, when updated and upgraded regularly, can last for a long time and provide enormous benefits and returns for the entire organization. Finally, effective and frequent communication will keep everyone on the same page and give the greatest chance of issues and problems being identified early, one hopes, along with preventing problems before they are allowed to affect the project.

Summary

- This chapter reviews the systems development life cycle—both traditional and alternative approaches—and points out the benefits and limitations of the traditional and the newer approaches. Reviewing the five phases of the SDLC approach (i.e., investigation, analysis, design, implementation, and maintenance) provides the necessary background to understand the ERP life cycle methodologies and see why the SDLC approaches cannot be used without changes for ERP implementation.
- The ERP life cycle has variations from the SDLC process due to various reasons; however, the key reason is that organizations buy ERP as prepackaged software and then have to customize them as well as change their company's business processes to implement these systems. Because ERP systems are complex systems that impact a large number of users in the organization, the implementation team will need a proper installation and change management plan.
- One of the first steps is choosing an appropriate implementation strategy. There are three routes for the company: comprehensive, vanilla, or middle-of-the-road strategy. Comprehensive will take longer and require more resources as opposed to vanilla, which can be quick but may or may not help improve the company's operations. Most organizations may choose a middle-of-the-road strategy because it will allow them to maximize their returns on the ERP investment.
- There are various ERP methodologies. In addition to the traditional ERP implementation life cycle there are rapid implementation methodologies developed by ERP consulting firms. These are Total Solution, FastTrack, Rapid-Re, ASAP, BIM, and others. These implementation methodologies are similar, in the sense that they allow you to choose from among the implementation strategies discussed earlier, with the differences coming primarily in the staging of the process steps and formality of structure.
- Consultants play an important role in rapid implementation of ERP systems. Rapid or accelerated implementation approaches are very popular and require the use of experienced consultants to leverage the knowledge of techniques that have worked well with other organizations. Scripts and wizards