1

ERP System Implementation Overview

Never put off until tomorrow what you can do the day after tomorrow
—Mark Twain

INTRODUCTION

Enterprise Resource Planning (ERP) System implementation is both an art and science that consists of planning, implementation, and ongoing maintenance. This methodology is designed to automate the drudgery of implementation and provide organized approaches to problem solving by listing, diagramming, and documenting all steps. Structured methodologies help to standardize and systemize ERP implementation and maintenance by approaching them as an engineering discipline rather than as whims of individual software developers. It is essential to understand structured methodologies in the implementation of ERP systems.

The basic steps of structured methodologies are:

- *Project Definition and Requirement Analysis.* Defining the terms of reference, determining user needs and system constraints, generating a functional specification and a logical model for the best solutions.
- *External Design.* Detailing the design for a selected solution, including diagrams relating all programs, subroutines, and data flow.

- *Internal Design.* Building, testing, installing, and tuning software.
- *Pre-implementation.* Evaluation and acceptance
- *Implementation.* Implementing systems.
- *Post-implementation.* Evaluation of controls and debugging.

This book covers ERP systems, their life cycles, and their major components to aid in understanding of any major ERP, irrespective of brand. It discusses each phase in the ERP life cycle, including the roles of each principal participant, key activities, and deliverables. Particular attention is paid to the audit role, which is the primary focus in succeeding chapters and may have to be adjusted if the other participants in the process do not perform their roles adequately.

When an organization purchases an ERP system, the intent is that the purchased ERP system provides specific functions and benefits. These functions and benefits need to be articulated to ensure that the ERP system performs as desired. This process is called conducting a feasibility analysis. The purpose of the feasibility study is to provide:

- An analysis of the objectives, requirements, and system concepts.
- An evaluation of different approaches for reasonably achieving the objectives.
- Identification of a proposed approach.

The feasibility analysis normally covers:

- *Current working practices.* These are examined in depth, revealing areas in the business where there is duplication of effort, or where procedures instituted in the distant past are carried out even though there is no longer any need for them.
- *Channels of information.* These are examined because the feasibility study is concerned primarily with the input and output information of each internal system. Such a study ignores departmental boundaries and prejudices. When the true information patterns within a business are exposed, it is often possible to reorganize resources so that all relevant data is captured at the point where it can be used for decision.

- *Alternative approaches.* Alternative methods of handling or presenting the data should be considered.
- *Cost factors.* These must be clearly identified and show definite cost savings or related benefits. Existing costs must be examined and used as a basis for comparison. Since this presentation is likely to be related to the information structure rather than to the departmental organization, the new approach may suggest possible improvements that were hidden under the existing system.
- *Supporting services offered*. The training and the systems and programming assistance that will be available during the installation period.
- *Range compatibility.* If the workload expands, can the configuration be increased in power without extensive reprogramming?

Differences and similarities between traditional auditing (i.e., financial, operational and IT auditing) and how they may be integrated in a computerized environment will be discussed. Appropriate ERP/IT control objectives will be defined and correlated as criteria in the ERP system audit.

Integrated Auditing

The term *integrated auditing* came from the IT terminology *integrated data* that then gave rise to the process of Integrated Systems and Systems Integration Processes from which ERP systems emerged. An ERP product can be defined as one that helps automate a company's business process by employing an integrated user interface, an integrated data set, and an integrated code set. So, from a purely business perspective, it is in the auditor's interest to become completely integrated in order to keep pace with technology improvements.

Prior to ERP systems, companies stored important business records in many different departments. Departments used different systems and techniques to manage that information. Information might also have been duplicated many times within an organization without necessarily being identical or similarly up to date. Some of this information might only have been on paper, making it difficult to access

across the organization. For example, a customer might call sales to inquire about the progress of an important order. Instead of answering the question by referring to a shared database, the sales rep would be forced to track down the order by making multiple calls to the company's manufacturing or shipping departments.

In a manufacturing firm, data relating to a product is typically kept by many different departments in the organization:

- A record showing product inventory balance is kept by the Inventory Control Department.
- Its cost and/or standard cost is shown on a record in the Cost Department.
- A record kept by the incentives department shows bonus percentages to be paid to employees for given levels of production.
- In the Finance/Accounting Department, a record of inventory values is kept for manufacturing account purposes.
- Shipping and Receiving maintains records of quantities shipped to customers and receipts of raw material.
- Returns keeps records of inventory returned.
- And the list goes on.

ERP systems originated to serve the information needs of manufacturing companies. Over time they have grown to serve other industries, including health care, financial services, the aerospace industry, and the consumer goods sector. With this growth, ERP systems, which first ran on mainframes before migrating to client/server systems, are now migrating to the Web and include numerous applications.

A vital business objective of ERP systems is to enable each department to know what is happening in the manufacturing plants and to get appropriate data in order to keep records up to date.

The first main characteristic of integrated systems is that they combine separate records relating to the same subject into one related record held in the computer. The new product record contains all items of relevant data that were previously kept in six or more separate records. You can visualize the challenges (i.e., security, confidentiality, accuracy, completeness, and reliability) that this record poses to all the separate departments.

The second feature of integrated systems is that the process of multirecording and transcribing data to update separate records is now replaced by one single input to the computer record. Therefore multi-inputs relating to transactions affecting the product are replaced by one single input to the product record held in the computer. Again, imagine the challenges or controls required to facilitate timely coordination and scheduling of all the processes (manual or otherwise) to be undertaken by the different departments so that the single input to the computer system, also referred to as the single point of entry, is accomplished.

By the preparation of suitable computer programs (e.g., software, applications, utilities, or combinations thereof) all the information needed by the separate departments can be produced when required. This is accomplished by processing the integrated records held on the Product MasterFile or in the Integrated Database. Integrated systems thus link together systems that traditionally have been kept separate and, by their very nature, cut across the conventional departmental boundaries that normally exist in a business.

The unified nature of an ERP system can lead to significant benefits, including fewer errors, improved speed and efficiency, and more complete access to information. With better access to information, employees and managers can gain a better understanding of what is going on in the enterprise so they make better business decisions. For example, an ERP system could let buyers in the Purchasing Department quickly adjust material orders when they see an increase or decrease in customer orders. The result? They will either ensure that orders are met on a timely basis or save on inventory expenses.

With this knowledge, what questions would you ask operational auditors, financial auditors, and IT auditors during the planning and execution of audit engagements?

Integrated auditing should ensure that controls are not duplicated. Effective controls in one department do not result in inefficient controls in another department. The overall objective of controls is to ensure that there is optimal time to market the product. Auditors should promote the enhancement of controls by being able to effectively communicate with each other.

Implementing ERP on platforms is not always easy because of the massive re-engineering process that involves security, quality assurance, and training for members of the organization entrusted to use the ERP systems. In addition to maximizing the operational effectiveness of

the ERP, IT managers, system security officers, and system developers will find themselves challenged on security and control issues.

Audit Objectives in an ERP Environment

The fundamental objectives of an audit of controls do not change in an ERP environment. When evaluating controls over ERP systems, decisions must be made regarding the relevance of operational internal control procedures to Information Technology (IT) controls. Specific control procedures for audit objectives must be tested.

Descriptive material on control procedures and sample compliance tests will be provided. This material will be as detailed as possible and should be read selectively, considering its relevance to the specific environment being audited.

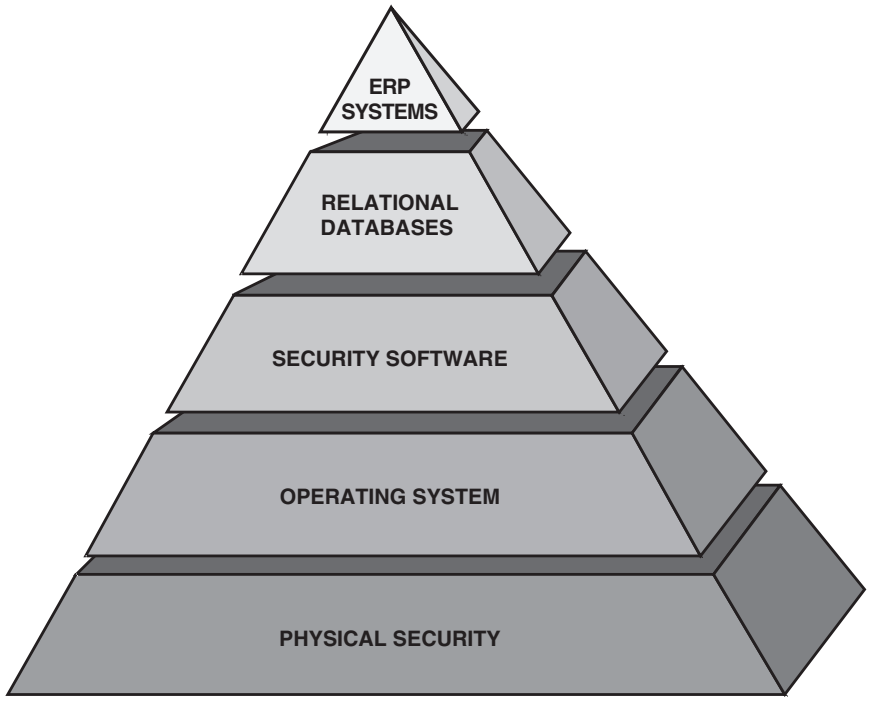
In addition to primary audit responsibilities, auditors should be able to provide advice on effective design of control procedures. Audit should communicate significant weaknesses that come to their attention to the appropriate IT personnel. Auditors should also be alert to weaknesses that require special reviews and be capable of assessing computer systems under development, in addition to the existing systems.

ERP SYSTEM ARCHITECTURE

ERP systems should produce accurate, complete, and authorized information that is supportable and timely. In a computing environment, this is accomplished by a combination of controls in the ERP System, and controls in the environment in which the ERP system operates, including its operating system. Controls are divided into general and application controls. General controls can be further divided into management and environmental controls. Management controls deal with organizations, policies, procedures, planning, and so on. Environmental controls are the operational controls administered through the computer center/computer operations group and the built-in operating system controls.

ERP systems are only as critical as the financial and/or operational sensitivity of the data they process and store. The security of the ERP systems can be thought of as a pyramid (see Exhibit 1.1). The base of the pyramid is the physical security of the hardware—the machine, the

Exhibit 1.1



databases, and the off-line storage media (such as tape or cartridges). The second layer deals with the operating system. The third layer focuses on the security software. This component may have to be included in a mainframe environment by installing a security product such as ACF2 or Top Secret, or it may be included in the operating system such as in the UNIX or AS/400 environment. The purpose is to secure the kernel, the privileged state, and to address spaces of the operating system and the hardware. It is also to ensure that ERP systems do not directly access the operating system and the hardware, which is the cornerstone to any secured operating system. These three layers contribute to the security of the computing environment and are covered in detail in *Auditing and Security, AS/400, NT, Unix, Network and DRP*. If the environment is secure, the ERP will enhance the financial and operational integrity of sensitive transactions in the production data and process. If not, the reverse is inevitable.

At this point, the ERP and the computing environment on which the ERP system operates have been treated separately. In reality, they are not mutually exclusive and independent. The strength of one affects the other. The focus of this book is on ERP systems. Our basis assumes a large networked system which stores, processes, and transmits sensitive data and information.

Enterprise Resource Planning (ERP) is an industry term for the broad set of activities supported by multimodule application software that helps a manufacturer or other business manage the important parts of its business, including product planning, parts purchasing, maintaining inventories, interacting with suppliers, providing customer service, and tracking orders. ERP can also include application modules for the finance and human resources aspects of a business. Typically, an ERP system uses or is integrated with a relational database system. The deployment of an ERP system can involve considerable analysis of business process, employee retraining, and new work procedures.

Unlike legacy systems, which used flat files and traditional IBM Indexed Sequential Access Methods (ISAM) and Virtual Sequential Access Methods (VSAM) for storage of data and information, ERP systems are used with relational databases. A relational database is a collection of data items organized as a set of formally described tables from which data can be accessed or reassembled in many different ways without having to reorganize the database tables.

The standard user and application program interface to a relational database is the structured query language (SQL). SQL statements are used both for interactive queries for information from a relational database and for gathering data for reports.

In addition to being relatively easy to create and access, a relational database has the important advantage of being easy to extend. After the original database creation, a new data category can be added without requiring that all existing applications be modified.

A relational database is a set of tables containing data in predefined categories. Each table (which is sometimes called a relation) contains one or more data categories in columns. Each row contains a unique instance of data for the categories defined by the columns. For example, a typical business order entry database would include a table that described a customer with columns for name, address, phone number, and so on. Another table would describe an order with columns for product, customer, date, sales price, and so on. A user of

the database could obtain a view that fits the user's needs. For example, a branch office manager might like a view or report on all customers that had bought products after a certain date. A financial service manager in the same company could, from the same tables, obtain a report on accounts that needed to be paid.

CHARACTERISTICS OF ERP SYSTEMS

When most people refer to the "core" ERP applications or "modules," they mean the back-office capabilities to manage human resources, accounting and finance, manufacturing, and project-management functions. However, major ERP suites from Oracle, PeopleSoft, and SAP now provide much more—including modules for sales force automation, business intelligence, customer relationship management, and supply chain management.

Although the objectives of our review, evaluation, and testing of the control framework are the same, there are some significant differences between ERP and non-ERP systems. These differences are:

- In ERP systems, certain control procedures leave no documentary evidence of performance. For some other procedures, the evidence of performance is indirect; it may be included in the program logic or in the operator's instructions. Therefore, compliance tests may have to be structured differently in an ERP environment and observation of the client's procedures may become more important.
- In ERP systems, information is often recorded in a form that cannot be read without the use of a computer.
- Financial and business information is often generated automatically by ERP systems based on data previously entered, without further human instructions.
- Errors that might be observed in non-ERP systems may go undetected because of the reduced human involvement in computerized processing. There is a danger that errors in processing may be applied to a large number of transactions without being noticed.
- With proper controls, ERP systems can be more reliable than non-ERP systems. This is because ERP systems subject all data

to the same procedures and controls. Non-ERP systems are subject to random human error. Although computer processing will usually be consistent, errors may still occur; for example, if the computer is incorrectly programmed.

- It is difficult to make changes after an ERP system has been implemented. Therefore, we should be aware of the organization's plans to introduce significant new systems or to make major modifications to existing systems. It is advisable to review new systems or modifications before implementation so that a preliminary assessment can be made of the adequacy of control procedures, in order to ensure an adequate audit trail, and to plan any necessary changes in the audit approach.

ERP systems vary from the simplest, batch-controlled type to complex integrated applications that perform a number of functions simultaneously.

Batch-Controlled Systems

In a computer, a batch job is a program that is assigned to the computer to run without further user interaction. Examples of batch jobs in a PC are a printing request or an analysis of a web site log. In larger commercial computers or servers, batch jobs are usually initiated by a system user. Some are defined to run automatically at a certain time.

In some ERP systems, batch jobs run in the background and interactive programs run in the foreground. In general, interactive programs have priority over batch programs, which run during the time intervals when the interactive programs are waiting for user requests.

The term *batch job* originated when punched cards were the usual form of computer input and the computer operator fed a sequenced batch of cards into the computer. (Hopefully, the output came back the next morning.)

In a typical batch system, user departments periodically submit batches of transactions to the IT department for transcription and processing. Batch totals are normally developed manually, thus setting up control totals which can be reconciled through successive processing stages to the file update report or to the final printed output.

When computers were first introduced, the batch system was predominant. Now many organizations are moving toward the more advanced systems described below.

Online Systems

In computers, interactivity is the dialog that occurs between a human being (or possibly another live creature) and a computer program. (Programs that run without immediate user involvement are not interactive; they are usually called batch or background programs.) Games usually foster a great amount of interactivity. Order-entry applications and many other business applications are also interactive but in a more constrained way, offering fewer options for user interaction.

The World Wide Web offers not only interaction with the browser (the Web application program) but also with the pages that the browser brings up. The implicit invitations called hypertext link to other pages and provide the most common form of interactivity on the Web (which can be thought of as a giant, interconnected application program).

In addition to hypertext, the Web (and many non-Web applications in any computer system) offers other possibilities for interactivity. Any kind of user input, including typing commands or clicking the mouse, is a form of input. Displayed images and text, printouts, motion-video sequences, and sounds are output forms of interactivity.

The earliest form of interaction with computers was indirect, submitting commands on punched cards and letting the computer read and perform the commands. Later computer systems were designed so that average people (not just programmers) could interact immediately with computers, telling them what programs to run. People could interact with word processors (called editors), drawing programs, and other interactive programs. The first interactive human-computer interfaces were input-text sequences called "commands" (as in "DOS commands") and terse one-line responses from the system.

In the late 1970s, the first graphical-user interfaces (GUIs) emerged from the Xerox PARC Lab, found their way into the Apple Macintosh personal computer, and then into Microsoft's Windows operating systems and thus into almost all personal computers available today.

A GUI (usually pronounced GOO-ee) is a graphical-user interface (rather than purely textual user interface) to a computer. As you read this, you are looking at the GUI or graphical user interface of your particular Web browser. The term came into existence because the first interactive user interfaces to computers were not graphical; they were text-and-keyboard oriented and usually consisted of commands you had to remember and computer responses that were infamously brief. The command interface of the DOS operating system (which you can still get to from your Windows operating system) is an example of the typical user-computer interface before GUIs arrived. An intermediate step in user interfaces between the command-line interface and the GUI was the non-graphical, menu-based interface, which let you interact by using a mouse rather than by typing in keyboard commands.

Today's major operating systems provide a graphical-user interface. Applications typically use the elements of the GUI that come with the operating system and add their own graphical-user interface elements and ideas. A GUI sometimes uses metaphors for real-life objects, the desktop, the view through a window, or the physical layout in a building. Elements of a GUI include windows, pull-down menus, buttons, scroll bars, iconic images, wizards, the mouse, and, no doubt, many things that have not been invented yet. With the increasing use of multimedia, sound, voice, motion video, and virtual reality interfaces are likely to become part of the GUI for many applications. A system's graphical-user interface along with its input devices is sometimes referred to as its "look-and-feel."

Online processing permits direct entry of transactions into the computer by user departments, frequently without batch controls. These systems permit the use of controls, such as automatic editing procedures, which can be more effective and instantaneous than batch controls.

There are many varieties of online systems, but they can be divided into three main groups:

- O*nline inquiry, with batch-controlled data entry for transactions, master files and databases.*

  In this approach, the operator at an online terminal can access stored data and learn the status of an account or transaction as of the last update, but cannot change the records. For example,

in an order processing system, the credit manager may be able to access customer accounts and find out their current balance, but the computer files can be updated only by using batch processing. In this situation, our review of internal controls should concentrate on the controls over batch processing.

Although not online, remote job entry (RJE) systems have control implications similar to online systems with batch-controlled data entry. RJE systems use remote facilities for entering batch-processing jobs into a computer from magnetic tape, or magnetic disk. For example, the operator may enter shipping documents together with batch-control information. The program reconciles the batch-control information and prints a message that indicates whether the shipping documents entered are in balance with the control information.

- *Online inquiry, with online data entry, data validation, and data collection. Master file updating from transaction files (online data capture).*

  In this case, individual transactions are entered through remote terminals and transmitted to a central computer where they are edited and validated; and, if accepted, stored in a transaction or daybook file. Later, normally overnight, the transaction file is used to update the master records. The user at the remote terminal has no capability to change the master records directly. These systems should be controlled in a manner similar to batch systems, with input totals accumulated and reconciled with totals calculated by the computer.

- *Online inquiry with online updating of both transaction and master files (real time update).*

  The most complex system to design and control is a real time update system. In these systems, the operator uses the terminal to update the master files directly by entering one transaction at a time. The computer file is said to be "transaction driven." In such a system, there may be few user-input controls. Extensive programmed edit and logging controls are required to protect the computer files from

erroneous or unauthorized transactions. It is also essential for important printed reports, such as control reports, exception reports, or summaries, to be carefully reviewed by a user department employee who is not responsible for entering transactions.

Online Systems Problems

Problems relating to online systems include:

- Data may be lost or altered during transmission of transactions from the user terminal to the computer.
- Many employees may be able to enter transactions through the terminals, thereby increasing the opportunity for erroneous or unauthorized entries.
- Additional procedures may be needed to handle processing malfunctions. The IT department must be able to identify what transactions have and have not been processed. It must be possible to return to a known position (checkpoint) and reprocess transactions from that checkpoint.

Distributed Data Processing

Distributed data processing refers to a network of local computers or minicomputers that are often online to a central computer installation. Typical systems are:

- Local editing of input; primary processing at the central computer; online inquiries to the central computer.
- Local editing of input; processing at the local computer and the central computer; online inquiries to both the local and central computers.
- Local editing of input; primary processing at the local computer on the local master file; consolidation of financial and business data from the local master file at the central computer; online inquiries to the local computer.
- Local editing; processing and file updates by a network of linked computers with no central computer installation.

An important feature of distributed data processing systems is that internal control may be divided among several physical locations or levels of processing. For example, in some systems, local personnel can modify the programs used by the local computers. In other cases the programs are controlled and can be modified only by personnel at the central computer installation.

Integrated Systems

Data processing is integrated when either input or generated data automatically updates the data files used in more than one system. A common example is the input of a customer sales order that automatically generates shipping documents, priced sales invoices, inventory issue instructions, and all related ledger postings.

Where conventional batch-processing systems are integrated, the successive processing steps will be executed by separate subsystems (program suites) in a logical order with batched data being progressively transferred from one program to the next. In these systems, we can usually expect to find a visible trail of "run-to-run" controls which can be reconciled to the original input batch totals. In such systems, it is unusual to find any significant loss of audit trail regarding the control totals, but there may be difficulties in identifying the individual items included in those totals.

In transaction driven systems, however, data files belonging to more than one application may be updated simultaneously by each individual item. For example, a sales transaction may update both account receivable and inventory records. In such systems traditional run-to-run controls do not exist and the potential for loss of audit trail is significant.

Databases

A database is a collection of data used by several different applications. It may be accessed using conventional access methods, or it may be organized and accessed by a database management system (DBMS). The DBMS is normally standard software supplied by either the computer manufacturer or by a software house.

When using a database management system, the data is independent of the application programs. The structure of the data com-

prising the whole database is defined by the DBMS. This structure is generally referred to as a "schema." Each individual application program will usually concern itself only with part of the total schema. The application program's view of the database is called its "sub-schema."

When there is sharing of data among many users, the responsibility for the accuracy of the data must be clearly established. This is often done through the database administrator (DBA), who should be independent of both users and programmers (Key Control Consideration).

The DBA should establish and update a data dictionary/directory system. The dictionary/directory should be concerned with the contents of the database and rules for its updating. In some cases, it is a manual document; in others, computerized. It is a major control tool for management and audit, providing, in effect, a map of information relating to database processing.

Rigorous edit and validation checks should be applied to input data. Since several programs use a data element, a single error may have multiple effect. This is known as "cascading" or "cumulative" error.

Automatic recovery and restart procedures are included in most Database Management Systems (DBMS). Recovery procedures are designed to prevent the loss of the database and of transactions being processed, whereas restart procedures relate to the resumption of computer processing.

Few people would have thought of enterprise resource planning (ERP) in terms of rapid change. ERP systems have been around since the mid-1970s when they first ran on mainframe computers. Enterprises that invested huge sums in these big and complex systems now have elaborate legacy setups that they absolutely depend on to run their companies. Because there is so much at stake, ERP providers and customers introduce changes to technology and deployments gradually to avoid costly mistakes.

Nonetheless, ERP systems *do* change—albeit slowly. People started talking about adapting ERP software to client/server technology at the beginning of the 1980s, but it was not until the late 1990s that more client/server versions shipped than mainframe versions.

Today, ERP is still evolving, adapting to developments in technology and the demands of the market. Four important trends are shaping ERP's continuing evolution: improvements in integration and flexibility; extensions to e-business applications, a broader reach to new users; and the adoption of Internet technologies. Taking a closer look at each will help you understand where ERP is headed.