PEARSON NEW INTERNATIONAL EDITION

Enterprise Resource Planning

Mary Sumner

First Edition

ALWAYS LEARNING™

PEARSON®

Pearson New International Edition

Enterprise Resource Planning
Mary Sumner
First Edition

PEARSON®

◆ TRADITIONAL SYSTEMS DEVELOPMENT

The traditional systems development life cycle included the phases of problem definition, feasibility study, systems analysis, systems design, detailed design, implementation, and maintenance (see Table 1). In systems analysis, the analyst undertakes a detailed analysis of the current system, using tools and techniques, such as process models and data models. Using these models, the systems designer analyzes bottlenecks, duplication of effort, inconsistencies, and other problems with the current system.

The fundamental approach used in traditional systems development is to analyze the current system's shortcomings and to develop a "new" system, which builds in changes in processes and data that will support the firm's business requirements. The rationale is that automating the current system is counterproductive because the current system may have problems, including redundant processes, insufficient data,

TABLE 1 Information Systems Design: Traditional Approach

Step	*Activities*	*Tools and Techniques*
Problem definition	Identify problems with the current system	Interviewing and data collection
Feasibility study	Assess the need for a systems project, including technical, economic, and management feasibility	Preliminary cost analysis
Systems analysis	Undertake a detailed analysis of the current system, including processes, information flows, and work organization	Logical process models—present system; Logical data models—present system; Organization charts (functional hierarchy diagrams)
Systems design	Development of objectives for the new system; re-engineering of processes and information	Logical process models—proposed system; logical data models—proposed system; organization charts—proposed system
Detailed design	Design of specifications for the proposed system	Program design specifications output design; input design Database design; forms design
Implementation	Software implementation; training end-users; development of reporting systems; design of controls and security	Coding; testing; documentation
Maintenance	Ongoing technical support; ongoing upgrades and enhancements	

and inefficient workflow. The systems design process provides an opportunity to re-engineer or re-invent the current system prior to automating it. The systems design process seeks to assure logical database design prior to detailed design, during which the specifications for the physical system are developed (e.g., output design, input design). Once the physical design specifications are set, then the system is programmed, tested, and implemented.

The problem with the traditional systems development life cycle is that it takes too much time and costs too much. The traditional life cycle follows a "waterfall" approach, in which there is a sequence of steps, starting with planning and analysis to be followed by design, detailed design, and implementation. Since the mid-1980s, companies have been seeking faster methods of developing information systems.

◆ NEW APPROACHES TO SYSTEMS DEVELOPMENT

Three different approaches designed to speed up the building of information systems were prototyping, end-user development, and software packages. In the late 1980s, prototyping was introduced as a methodology for obtaining user requirements more quickly and accurately. Using prototyping, systems designers could show "models" of systems documents (e.g., reports, screens) to end-users to get a better idea of their requirements. While this approach enabled end-users to specify their requirements, it did not necessarily speed up the systems implementation process, including coding, testing, and debugging.

End-user development was another approach introduced in the mid- to late-1980s and is still relevant for many applications. Equipped with spreadsheets and database packages, like Microsoft Excel and Access, end-users constructed their own information systems. While this approach worked effectively for local departmental applications, it was not appropriate for the development of large-scale production systems requiring quality assurance, security, documentation, backup, and live production.

In the late 1980s, software packages became more prevalent. They offered economies of scale in development, enhancement, and maintenance, and many companies moved toward purchasing commercial off-the-shelf software. ERP systems are large-scale, integrated commercial off-the-shelf software packages that support the entire value chain of business functions.

◆ THE ERP SYSTEMS DEVELOPMENT PROCESS

The ERP systems design process is different from the traditional systems development process. The ERP systems development process includes planning, requirements analysis, design, detailed design, implementation, and maintenance (see Table 2).

Planning starts with a needs assessment, which provides a business justification for the purchase of the software. This needs assessment phase is important because of the major investment in an ERP system and its business impact. The requirements analysis phase of an ERP project involves specifying the business processes to be supported by the ERP package. Most ERP vendors offer "best practices," which are models of functions supported by the ERP system.

TABLE 2 ERP Systems Design Process

Step	*Activities*	*Tools and Techniques*
Planning	Conduct a needs assessment; provide a business justification, based upon the difference between the existing system and the proposed system	Interviewing; cost justification
Requirements analysis	Analyze current business processes and specify the processes to be supported; select the ERP system	Use best practices models to see what the company can gain by implementing the new system
Design	Re-engineer business processes around the best practices model of the ERP system or customize the software	Use the ERP methodology's best practices or customize
Detailed design	Choose standard models, processes, inputs, and outputs (e.g., customer lists, vendor lists)	Interactive prototyping
Implementation	Configure the system; migrate data from the old system to the new system; develop interfaces; implement reporting systems; conduct testing; implement controls, security; train end-users	Work with vendors to correct any "bugs" in the software; clean processes and data Use reporting tools
Maintenance and continuous improvement	Provide technical support; provide upgrades and enhancements	Add enhanced functionality to existing modules

In the design phase of the project, the project sponsors introduce the best practices, which the ERP system supports. This entails re-engineering business processes to fit the software. This is fundamentally different from the traditional systems development approach, in which the systems designer defines new business requirements and implements software to conform to these requirements. One of the fundamental design decisions in implementing an ERP package is whether to re-engineer the organization's business processes to fit the software or to customize the software to fit the organization's business practices.

PLANNING: MAKING THE BUSINESS CASE FOR ERP

The business justifications for ERP include tangible and intangible benefits, including inventory reduction, operating cost reductions, overdue accounts collection, process improvement, and reduction in cycle times (Ross, Vitale, and Willcocks, 2003). The technology and business rationales for ERP packages are illustrated in Table 3.

TABLE 3 Rationales for ERP

Technology Rationales	*Business Rationales*
Ability to use timely operational data	Inventory cost reductions
Ability to integrate systems, instead of maintaining many separate systems	IT cost reductions
Ability to enhance systems without incurring the time and cost of custom development and modifications	Personnel cost reductions
Ability to implement new features, such as Customer Relationship Management (CRM)	Increased profitability
Access to on-line, real-time operational data	Productivity improvement
Reduction in the time and cost of systems development and maintenance	Better cash management

REQUIREMENTS ANALYSIS

Requirements analysis activities involve (1) analyzing business processes and (2) specifying the processes to be supported by the ERP package. Since the company is buying into the vendor's view of best practices, it is important to select a system which fits with the organization's goals and competitive strategy (Umble, Haft, and Umble, 2003). Most vendors offer best practices for specific industries, such as the chemical industry and oil industry.

The process of selecting the best ERP system entails working through a checklist of activities (see Table 4) (Umble et al., 2003).

Aside from the business issues, there are a number of technology factors to consider in selecting an ERP vendor and an ERP system (see Table 5).

DESIGN: RE-ENGINEERING VERSUS CUSTOMIZING

The fundamental decision in ERP systems design is re-engineering versus customizing. In the re-engineering approach, the team selects a commercial off-the-shelf ERP and re-engineers business processes to fit the package. In the customizing approach, the team selects a commercial ERP and customizes the ERP to meet unique requirements (see Table 6).

An in-depth analysis of the trade-offs between re-engineering and customizing an ERP system illustrates several important factors. Re-engineering the business to fit the software can disrupt the organization because this represents changes in procedures, work flows, and data. However, customizing an ERP can make upgrading to newer versions difficult since vendor-supplied versions will be based upon vanilla versions of the software (see Table 7).

ALTERNATIVE ERP DESIGN OPTIONS

ERP systems can be designed using various approaches. A complete vanilla ERP package is easiest to implement because the organization can follow the vendor-prescribed

TABLE 4 Selecting an ERP System

Steps	*Activities*	*Complete Y/N*
Create the vision	Develop corporate objectives for ERP	
Create a feature/function list	Use a team who are familiar with company processes; map current processes to the new best practices	
Create a software candidate list	Narrow the field based on size of company, industry type; talk to existing buyers in the industry	
Narrow the field to four to six serious candidates	Conduct preliminary analysis of strengths and weaknesses; determine goodness of fit	
Create the Request For Proposal (RFP)	Develop a list of features and functions	
Review the proposals	Consider strengths and weaknesses of each proposal	
Select two or three finalists	Have the finalists demonstrate their packages	
Select the winner	Consider numerous factors (e.g., supplier support, closeness of fit, technological risk)	
Justify the investment	Conduct a cost-benefit analysis; tangible benefits include improved material control, reduced costs, increased productivity, increased on-time deliveries, improved customer service, inventory reduction, and elimination of redundant databases; intangible benefits include reduced cost, higher morale, and improved communications; make a GO, NO GO decision	
Negotiate the contract	Participate in contract review	
Run a pre-implementation pilot	Have the cross-functional team review the pilot	
Validate the justification	Involve the cross-functional team in a final GO, NO GO decision	

TABLE 5 Technology Factors to Consider in Selecting an ERP System

Technology Factors	*Questions*
Cost of technology	What are the start-up and recurring costs?
Installation	What consulting assistance is offered (time, cost)?
User interfaces	What interfaces are supported?
Upgradability	What is the frequency of upgrades?
Computing environment	What is the computing environment?
Personnel requirements	What expertise is needed for design and implementation (business analysts, consultants)?

TABLE 6 Re-engineering vs. Customizing

	Pros	***Cons***
Customizing approach	Supports unique business processes; strategic processes are maintained	An ERP may not support these unique business processes; re-inventing the wheel; customization is difficult, since modules are integrated; difficult to upgrade the software to newer versions, since upgrades are based on vanilla versions
Re-engineering approach	Is supported by an ERP solution; takes advantage of shared or generic processes within industries (e.g., industry templates); best practices may represent improved process changes; documents best practices; works well when there is minimal organizational change	Does not support strategic or unique business processes; resistance occurs when there is extensive organizational change

TABLE 7 Detailed Comparison: Re-engineering vs. Customizing Approach

	Re-engineering Approach	***Customizing Approach***
Re-engineering business processes	Supports re-engineering processes to fit the software system's best practices	Re-engineering is independent of the tool being implemented (e.g., its models, processes, outputs)
Organizational fit	Works well with minimal organizational change, but extensive re-engineering may disrupt the organization	May disrupt the organization less because software is designed to support current methods of work organization and structure
Evolution	Evolution depends upon vendor upgrades and enhancements to the system	Evolution can support unique user requirements
Timeliness	Software is available and ready to implement	May involve lengthy systems development activities
Cost	Implementation is cost-effective	May involve extensive cost of custom implementation
Requirements	Puts boundaries on the design; designs conform with business models and best practices	Provides greater flexibility for meeting unique requirements; not constrained by the tools' best practices; no boundaries for the design
Competitiveness	Other firms have access to the same design	Do not have to use software to which everyone in the industry has access
Fit	Requirements will be supported by an ERP system	Unique requirements may not be supported by an ERP system
External consulting	More of a turnkey approach, particularly using a vanilla implementation	May entail the expense of much external consulting