

LECTURE NOTES

ON

Fundamentals of IT & ERP

2ND SEMESTER

MBA

HI-TECH INSTITUTE OF TECHNOLOGY

INDUSTRIAL ESTATE, KHURDA

Introduction

So what is an ERP system? You always wanted to know but were afraid to ask! An ERP system is a vast information system that manages information about a company's products, customers, suppliers, employees, production facilities, financial balances, etc. It is used by *every* department in an organization and by most employees. The sales department can check prices of products and their inventory levels, enter sales orders from customers, make deliveries, issue invoices to customers and receive payments. The production department uses it to check inventory balances of products, create production orders, manage production schedules, record the receipt of finished or in-progress orders. The marketing department can plan demand and make sales forecasts for the next sales period. The finance department uses it to manage accounts payables, accounts receivables, enter payments made by customers and payments made to suppliers, generate the balance sheet and profit and loss statements at the end of an accounting period. The human relations department tracks all employees in a company, their title, date of joining, department, and salary. Every new employee is added to the system as soon as they join the company. If they quit or are fired, this is also reflected in the system, but their information is still retained for historical records. Figure 1 shows you a list of **resources** used by various departments of a company. Try to distinguish whether the items in the lists refer to a physical or an information resource. You can also think of a resource as an asset of an organization.

<u>Production</u>	<u>Sales</u>	<u>Marketing</u>	<u>Human relations</u>	<u>Accounting</u>
Plants	Customers	plans	Employees	Receipts
Machines	Prospects	Strategy	Departments	Payments
Products	Sales orders	Markets	Units	Receivables
materials	Products	New products	Skills	Payables
Production orders	prices		Contact	Asset values
Purchase orders				Liabilities

Figure 1: Physical and information resources of an organization

An ERP system in effect manages all the resources of an organization or enterprise. Hence it is also called an **enterprise system**. It knows how many manufacturing facilities belong to a company, what are the machines at each facility and their capacities. It is almost like a big brother watching all the activities and transactions in a company. How is this possible? Well, the trick is that every transaction that takes places in a company gets recorded in the ERP system in **real-time**. So, what do we mean by real-time? To explain this, we will digress here to give you a historical perspective of how ERP systems evolved.

A little bit of history (1960s, 1970s, early 1980s)

Companies have been using computers to speed up the performance of their business for more than 50 years. The 1960's were the era of the mainframe. A mainframe was very large could easily cost several million dollars and required a facility as large as a modest size home with controlled climate. Only the largest companies could afford their own mainframes. Most companies shared a mainframe operated by a service bureau by renting time on it. Figure 2 shows the architecture of a mainframe that was accessed by "dumb" terminals, so called because they did not have any computing power in them. They were just like TV screens and were also called CRT (Cathode-ray tube) monitors. A client could send their computer jobs to the service bureau where an operator of the bureau would run the job from the terminal and send the results back to the client. The turn-around time for this job could vary between a few hours and few days. Prior to the advent of CRT's, in early 60's computers were accessed through punched cards. A user would type a program and data for it on special cards and a card reader would read the program and data and send the output to a printer. Both the card reader and printer were directly connected to the main computer. An operator would monitor and give commands to the computer from a keyboard to run programs.



Figure 2: A mainframe connected to "dumb" terminals

The most famous mainframe computer of this time was by far the IBM 360. The most popular programming language for business applications in those days was COBOL. Fortune 500 companies of that time developed many applications using COBOL. Believe it or not many of them are still in use. So mainframes are not dead by any means. They are still used by large businesses. Such systems from the days of yore that still operate alongside newer systems are called **legacy systems**.

Initially, say, in the 1970's, these efforts were at the department level. The sales department might develop a program that generated sales reports every month for the amount of sales they made by product, region, customer and salesperson. At the end of say the month of August, this program would read in all the sales orders data for August (also called the "raw" sales data) and

then sort them in four different ways, by product, region, customer and sales person, and run some simple calculations to produce four different reports. In those days, personal computers or PCs did not exist and so these reports had to be produced at a service bureau which had a large computer. The company would send its raw data to this bureau where a computer operator would enter the data into the computer. Then a program would be run to read this data and produce the reports. A sample sales order form and a sales report are shown in Figure 3.

Sales Order

To:	From:
sunny south	ExitLight
Craper road	Organisation
qbom2s	New York
Mauritius	New York
	United States

Ref S O No: 100002
 Date Ordered: Mar 13, 2007
 Status: Drafted
 R Paid: 0

Description	Qty	Unit Price	Total Price
Black Emergency Light with MR16 Lamps	5	\$ 10.00	\$ 50
Steel Emergency Light S&W 3 Head 12V 12w	4	\$ 10.00	\$ 40
White Emergency Light Self Testing	3	\$ 10.00	\$ 30
Black Emergency Light	2	\$ 10.00	\$ 20
Order Total			\$ 140

(a) A sales order

The screenshot shows a software interface for a sales summary report. At the top, there is a search bar with filters for 'From' (March 7, 2007) and 'To' (March 13, 2007). Below the search bar, there is a table of sales orders. The table has columns for 'Date', 'Entries', 'Items', 'Price', and 'Order Total'. The data is as follows:

Date	Entries	Items	Price	Order Total
September 2006	219	455	\$811.00	\$20,140.00
January 2006	299	299	\$841.00	\$25,030.00
2006	998	1000	\$1540.00	\$15,400.00
December 2006	260	876	\$811.00	\$20,982.00
November 2006	520	895	\$811.00	\$20,860.00
October 2006	499	705	\$811.00	\$20,380.00
September 2006	479	681	\$811.00	\$19,849.00
August 2006	79	881	\$191.00	\$12,119.00
July 2006	61	30	\$411.00	\$4,860.00
June 2006	28	95	\$811.00	\$12,120.00
May 2006	54	80	\$846.11	\$1,410.00
April 2006	84	100	\$400.00	\$1,600.00
2006	1,479	1,938	\$1,188.00	\$19,140.00
2006	2,018	4,188	\$1,800.00	\$36,300.00

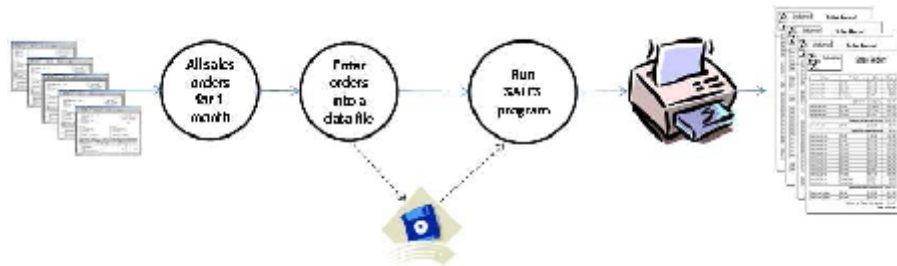
(b) Sales summary report

Figure 3: A sample sales order and a sales summary report

Now imagine yourself as a sales vice president or a regional sales manager in those times. You would receive a report once a month that would show you how your company did in the previous month. So, say on September 7, you got the sales reports for August because it took about a week to gather the August sales orders, send them to the bureau, have the bureau process them and then return them to you. This style of operation where the data is processed in batches, and the information we receive is usually delayed, is called **batch** operation. Why? This is because we are combining the sales orders for one month into one batch and then running the program. If, say, on a whim, on September 15 the sales manager were to wonder how his team was doing

so far in the first half of September, there was no way for her to get this information short of manually collecting the sales orders and collating them by hand which is not feasible in any but the smallest of companies. Tough luck! It is like driving while wearing blinders. On the other hand, when the information we receive about our business is always current it is called operating in **real-time**. Figure 5 gives an example of batch mode operation in our daily lives in a lighter vein. What is the real-time equivalent of this? Also, think about what it would take to switch a business from operating in batch mode to real-time mode.

In the 1980's we discovered spreadsheets like VisiCalc, and, more famously, Excel which we still use today. With a generic spreadsheet program running on a computer, an employee could on her own arrange the raw sales data in a variety of ways and create subtotals and totals by product, region, customer and sales person, etc. This obviated the need to develop a custom program for every application separately. It made employees more productive by giving them more ability to manipulate the raw data and obtain the information relevant to their needs. Of course, this same ability was useful for the staff in other departments such as production planning, marketing, accounting etc. Access to a tool like VisiCalc also meant that the folks working in sales, say, could see their reports more frequently. Rather than waiting for the once a month program run at the service bureau, they could enter, perhaps, a week's worth of sales orders data into a file and then open the file in the Spreadsheet program and play around with it.



(a) Batch mode processing



(b) Real-time processing

Figure 4: Batch v. real-time

This was around the time when personal computers were developed. The first Apple I computer was released in 1979. It would be a while before every salesperson got her own PC, yet this suddenly meant that a company could have a group of PCs that people could share. By the late 1980s, early 1990s it was more common for every employee to have her own PC as the prices

became more affordable. This was the time when early adopters and technology buffs started buying their own home PCs as well.



Figure 5: Another view of Batch v. real-time

The transition to the PC gave more freedom to departments within a company to start “doing their own thing”. Thus, a bright person in a sales department who knew BASIC could write a program to enter sales orders from customers. Rather than taking orders on a paper order form, they could be typed directly into the PC and stored in a file. This removed one step from the process and also the chances of transcription errors when a person entered data from a handwritten order form into the system at a later stage. Further, the data entry program (see Figure 6) could be made "smart" by having it perform validation checks for each field. Thus, a date field could be checked for format ("30-Feb-2013" is not a valid date), a quantity field could only be allowed to have a positive numeric value within a certain range, etc.

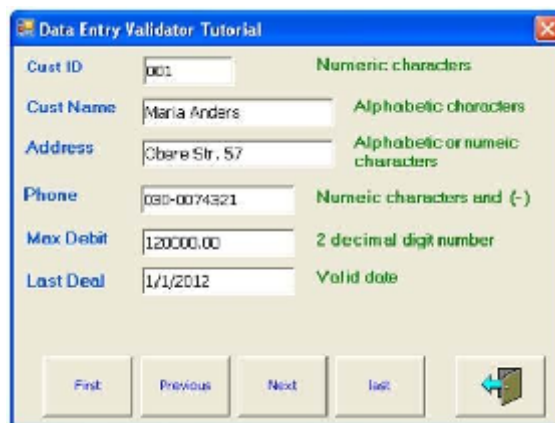


Figure 6: A data entry and validation program

Other departments came up with their own ways of using this computing power at their disposal. The marketing department could do better sales forecasting and planning. The production

department could keep track of their inventory of semi-finished and finished goods, and production orders better, while purchasing could track status of their purchase orders. A simple program could be run to flag late deliveries, or those that were more than one week overdue.

As a result each department had its own information system that helped the department to perform its job better. This gave rise to silos of information systems within the company as we show in Figure 7. Thus, when, say, salesperson Jen received an inquiry from a customer Carly asking, “can you deliver 10 boxes of 5 lb. candy by September 15,” Jen would have to pick up the phone and call Bob in production. Then Bob would have to look up his own system and tell Jen what the inventory level for 5 lb. candy boxes was. This naturally introduced delays in the process because customer Carly would have to wait while Jen got the information from Bob. The natural question to ask next was why Jen couldn’t look up the inventory information directly herself. This would save the additional step of calling Bob and also reduce the customer’s wait time. This posed a major challenge and it was not so easy to fix this issue. Take a moment to pause and reflect on why this was so difficult.

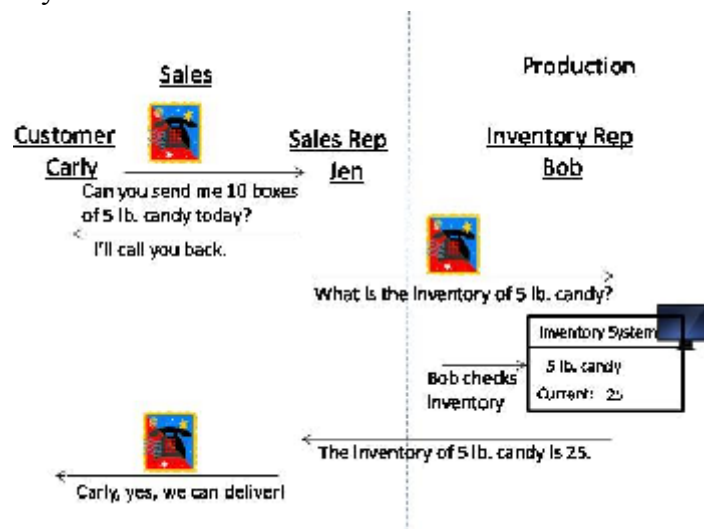


Figure 7: Exchange of information between two departmental silos

Well, each department developed its information system separately from the others. The sales department may have used COBOL language to write a program on an IBM PC, while the production department perhaps used a BASIC language program running on a Mac (see Figure 8). Even more importantly, the computers of the two departments were not connected. They did not know how to exchange data! Thus, departments could exchange data only through good old paper or on a floppy disk, first the 5 1/4” one with a capacity of 250,000 bytes which was actually made from floppy material, and later the smaller 3 1/2” variety with a capacity of 1 million bytes which was not made from floppy material, yet the name had stuck to it. These are now antiques for a computer museum. So when an order inquiry came into the sales department, Jen could do one of two things. One, she could call Bob in production and ask for the latest inventory of 5 lb. candy boxes. Alternatively, she could check the most recent inventory report and then tally all

new orders received since then to make a rough calculation of the likely inventory balance after these previous orders got processed. Let's see the next important development in computing.

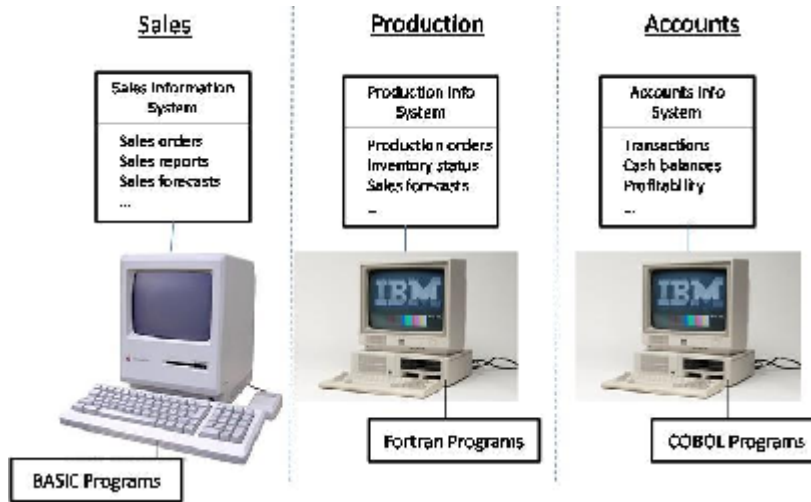


Figure 8: Silos result when each department develops its own information system separately

Local area networks and client-server computing (late 1980's, 1990's)

The next leap in the evolution of information systems came when it became possible to connect two computers together so they could exchange data. This was enabled by the advent of local area networks and the Ethernet standard. The Ethernet protocol for communication was invented by Bob Metcalfe at Xerox Parc labs in 1972, but took a while to become commercialized. By inserting an Ethernet card into a PC, and installing software drivers for it, you could make the PC "network-enabled". Thus, it became possible to connect it to other computers, with similar Ethernet cards and software, on a small area or local network that spanned a lab, a building or even a small campus (see Figure 9). Now it was possible for somebody sitting at the Sales computer to see files on the Production or Marketing computer.

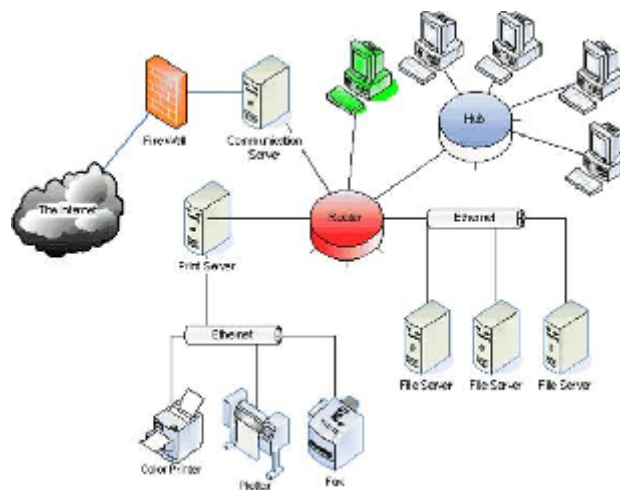


Figure 9: A LAN with PCs, servers, Ethernet cards, routers and hubs

This was a truly major breakthrough in the history of computing and information systems. It meant data could be exchanged between computers but it also meant that if Jen in Sales was looking for inventory information, she would still need to know on which computer it is stored. Ideally, we wanted a program that would allow Jen to access the inventory data in a seamless way (see Figure 10). By seamless we mean that she did not have to know where the data was stored. She would ask for the inventory data and presto it would appear on her computer screen! Then she could respond to customer Carly right away.

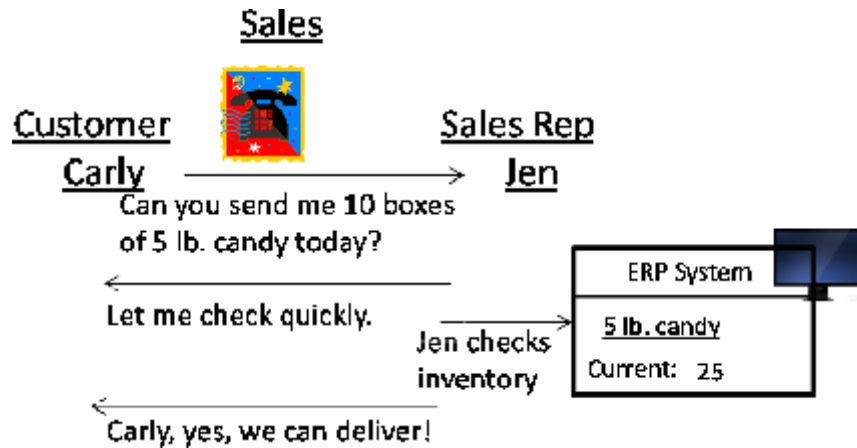
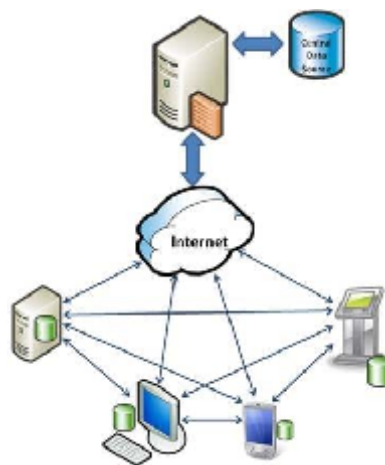
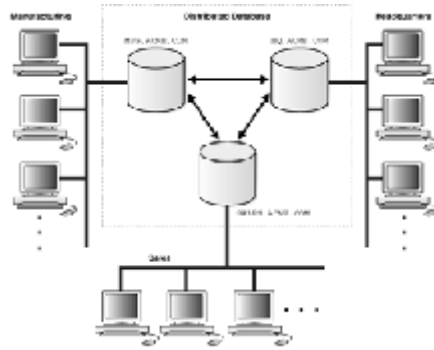


Figure 10: An employee in the sales department can access current inventory herself

The next issue that arose was one of where the data should be stored. Should sales, production, inventory data be stored in separate computers managed by the respective departments or should it be in a central place. See Figure 11. What do you think? Try to make a list of what you think are some pros and cons of the two data arrangements.



(a) Centralized data in one central data store



(b) The data is distributed across three data stores

Figure 11: centralized v. distributed data

This gave rise to the client server technology. A **client** is a computing device available to a user. It can be PC, a Mac, a PDA, a tablet, etc. A **server** is a computer responsible for providing a service. Some examples are:

A **file server** allows you to access files.

A **print server** allows you to select a printer and send your print job to it.

A **database server** allows you to access a database by submitting a query say in SQL. The server will run your query and send you a reply.

A **web server** lets you access web pages. When we “surf the web” by typing a URL in a browser window or by clicking on a link on a page, we are asking for pages from a web server on the internet.

The **client server technology** is based on a request-reply model, where the client sends a “request” to the server and the server handles the request and sends a “reply” or "response" back to the client. Thus, when we surf the web, we are requesting a page from the server and the response is an HTML file that contains the page. On receiving it, your browser knows how to decode the HTML file and display the contents of the file nicely in the browser window.

So how does a client on the internet connect to a server on the internet? Each client and each server has an internet address that is much like a telephone number. So, just like you can place a telephone call on the phone network to anybody worldwide by knowing their number, so too a client device can place a computer-to-computer call on the worldwide internet network by knowing the address of the server computer. We will talk more about the architecture later.

Thus, in time the independent home-grown departmental applications were able to exchange information over the network giving rise to an **integrated information system**. As the need for these systems became clear, they grew to span all resources of an organization. Further, they were standardized into packages called **ERP (Enterprise Resource Planning)** systems.

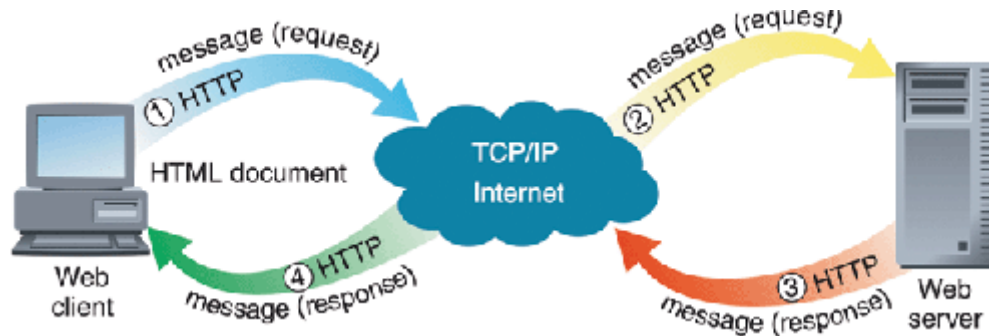


Figure 12: A client interacting with a server

Benefits, Implementation, Vendors

What are the benefits to a company from implementing an ERP system? There are several, and let us consider some of them. First, the seamless integration among different functional areas helps to improve communication, productivity and efficiency. Second, order tracking becomes very simple right from placement through payment. A sales person can tell the exact status of a sales order to a customer at any time. Third, accounts can better manage the revenue cycle from invoice through cash receipt. Fourth, production can manage inter-dependencies of complex bills of materials easily. If, say, the bill of materials (we will explain this term later) of a brake sub-assembly for a car is modified, the changes are incorporated easily. This improves production planning, and helps to shorten production lead times and delivery times across the supply chain of the company. Finally, marketing can make more accurate forecasts which further reduce inventory levels and lead to higher profits for your company.

Let us also consider some disadvantages of ERP systems. First off, implementation costs can be enormous. To implement an ERP system is a multi-year, multi-million dollar effort involving the entire organization. It is fraught with pitfalls and delays that can disrupt the business until the implementation is complete. Second, ERP systems are often too rigid and difficult to adapt to the specific workflow and business process of a company. Most companies end up changing their own processes to accommodate the system which is designed according to "best practices". Third, it is difficult to customize ERP software. While report formats can be changed easily, any non-trivial changes in the business logic require hiring consultants and programmers who are in scarce supply and very pricey too. Finally, remember that bringing a system like this can have a major impact on the power relationships in an organization. What this means is that some people in certain positions lose power while others in other positions gain it. After all, information is power! An information system equalizes the access to information across the organization. Naturally, someone who perceives the new system as a threat to his or her power will do everything to resist such an initiative and to jinx it. Believe it or not, this kind of **organizational politics** can often kill an ERP project!

There are many vendors of ERP systems, the largest two being SAP and Oracle. Let's say a few words here and we'll say more about them in later. **SAP** is the German juggernaut. The company was founded in Germany in 1972. In 2012 their annual revenues were in excess of \$20 Billion. SAP has the largest market share in the ERP market. A large number of Fortune 500 companies use SAP. Their flagship product for a several decades has been SAP R/3. There are over 9,000 implementations of R/3 at over 6,000 companies. In recent years, SAP has introduced new products like Business Objects One, SAP Hanna, etc. They provide both generic solutions that apply to a range of industries and also industry specific solutions for segments like insurance, banking, healthcare, etc. In recent years they have introduced many other products.

Oracle was founded as a database company in the 1970s by the charismatic Larry Ellison. It is still the largest database company; however, they were late to the game of enterprise systems. They entered this space by acquiring PeopleSoft in 2005, which had already acquired J.D. Edwards. PeopleSoft was a vendor of human resources, financial management and supplier relationship management software. J.D. Edwards had strengths in project management, manufacturing and financial software. Oracle integrated the software of these companies into its own products. Oracle has gained market share in the last decade and is a close rival to SAP now. Oracle's 2012 revenue was \$37 Billion but a large part of it came from database products.

Both SAP and Oracle make software for large customers with revenues upwards of \$1 Billion per year. Other vendors like Microsoft Great Plains, SYSPRO and Banner make software for smaller organizations.

The SAP R/3 software is very large and consists of modules for specific application areas as shown in Figure 13. A customer organization may decide to implement these modules in a phased manner. For instance, they may start with the basic modules like Sales & Distribution, Material management and Production planning and Financial accounting. Once these modules are successfully rolled out and are running in a stable manner, in phase 2 they could introduce Human resources, Fixed asset management and Quality management.

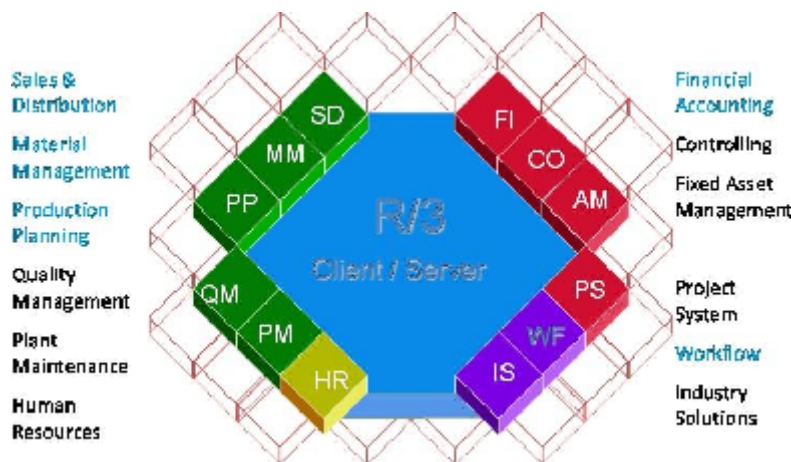


Figure 13: Various modules of SAP (source: SAP)

Summary of chapter

An ERP or enterprise system is defined by Thomas Davenport as a comprehensive commercial software package, designed to support and integrate organizational processes across functional boundaries. They are built on a common database and are used to execute the complete range of transactions in an organization. As packages, they are designed to offer generic functionality for use by many companies, not custom-built solutions for a specific organization. To accommodate organizational variation, these systems are somewhat configurable, but by no means infinitely malleable. Unlike simpler systems that may only affect an department, an enterprise system (ES) spans an entire organization.

Some key issues and concepts that we have discussed in this article are as follows.

1. ERP systems are real-time systems. Legacy systems are batch mode systems. This means that new information pertaining to sales orders, deliveries received and made, payments received or made, etc. is entered into the system as soon as it happens. In a legacy system such information is “batched” and then input into the system at certain intervals, like once a day, once a week or once a month.
2. ERP systems store all the information in a common database. In legacy systems the information was scattered across multiple databases, some in the sales database, some in the production database, etc.
3. ERP systems are integrated systems. Legacy systems were not integrated. This means that all the information of the company is accessible from one application on one device in a seamless way. Legacy systems evolved as multiple independent applications that could not exchange information.
4. ERP systems allow development of cross-functional processes. In legacy systems this is not possible. Most customer facing processes in a company require participation of several departments or functions. Legacy systems made it very hard for information to flow across different applications. In ERP systems this is possible since any department can access the information produced by any other department.

I hope this gives you some historical perspective and an overview of the main features of an ERP system. Actual work in an organization is done through business processes or work routines that connect various activities in a logical manner as we shall discuss next.