UNIT 1
System Development Fundamentals

LH- 9HRS

Er. Rolisha Sthapit

SYSTEM ANALYSIS AND DESIGN (SAD)

CONTENTS (LH-9HRS)

1.1 The system development environment

Introduction, Modern Approach of System Analysis and Design, Information System and its Types, Developing Information systems and the System Development Life Cycle, The Heart of System Development Process, The Traditional Waterfall SDLC, Approaches for Improving Development, CASE tools, Rapid Application Development, Service -Oriented Architecture, Agile Methodologies, eXtreme Programming, Object -Oriented Analysis and Design

1.2 The origins of software

Introduction, System Acquisition, Reuse

1.3 Managing the Information Systems Project

Introduction, Managing Information Systems Project, Representing and Scheduling Project, Using Project Management Software Plans

1.1 The system development environment

Introduction, Modern Approach of System Analysis and Design, Information System and its Types, Developing Information systems and the System Development Life Cycle, The Heart of System Development Process, The Traditional Waterfall SDLC, Approaches for Improving Development, CASE tools, Rapid Application Development, Service - Oriented Architecture, Agile Methodologies, eXtreme Programming, Object -Oriented Analysis and Design

Introduction: Data Vs Information

- **Data** is raw unprocessed facts and figures that have no context or purposeful meaning.
- Data are raw facts about the organization and its business transactions.
- Most data items have little meaning and use by themselves.
- **Information** is processed data that has meaning .
- Information is data that has been refined and organized by processing and purposeful intelligence.

- Data is used as input for the computer system. Information is the output of data.
- Data is unprocessed facts figures. Information is processed data.
- Data doesn’t depend on Information. Information depends on data.
- Data is not specific. Information is specific.
- Data is a single unit. A group of data which carries news and meaning is called Information.
- Data doesn’t carry a meaning. Information must carry a logical meaning.
- Data is the raw material. Information is the product.

System

- A system is an orderly grouping of interdependent components linked together according to a plan to achieve a specific objective.
- A system is a collection of components (subsystems) that work together to realize some objective. For example, the library system contains librarians, books, and periodicals as components to provide knowledge for its members.
- A set of detailed methods, procedures and routines established or formulation to carry out specify activity, perform a duty or solve a problem.

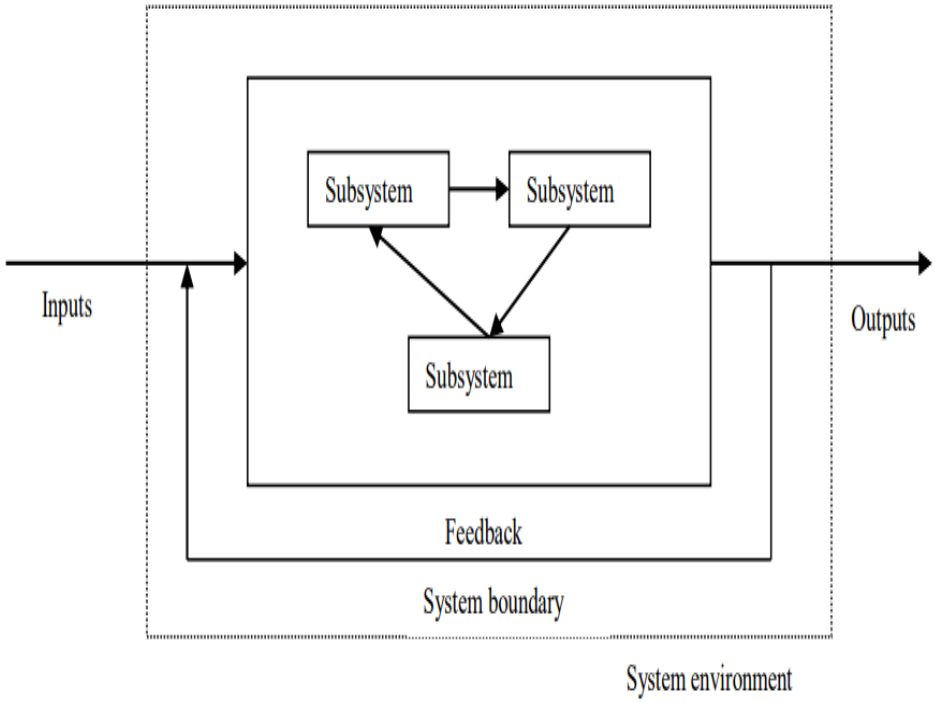
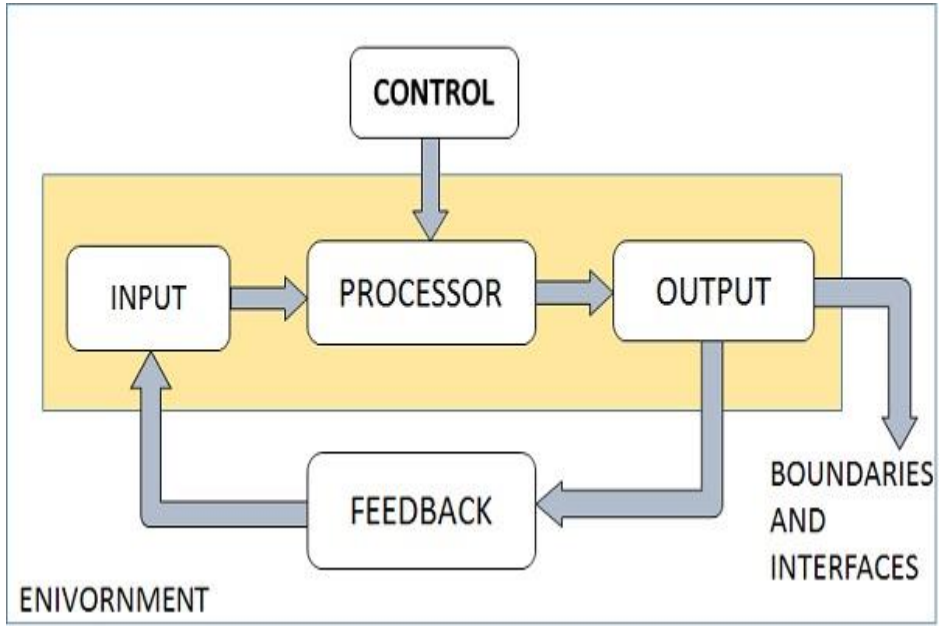


Fig: Basic System Model

Elements of System

Every system has three activities or functions. These activities are input, processing and output.

- **Input:** It involves capturing and assembling elements that enter the system to be processed. Inputs to the system are anything to be captured by the system from its environment. For example, raw materials.
- **Processing:** It involves transformation processes that convert input to output. For example, a manufacturing process.
- **Output:** It involves transferring elements that have been produced by a transformation process to their ultimate destinations. Outputs are the things produced by the system and sent into its environment. For example, finished products.



The system also includes other two additional activities. These activities include feedback and control.

- **Feedback:** It is data about the performance of a system. It is the idea of monitoring the current system output and comparing it to the system goal. Any variation from the goal are then fed back in to the system and used to adjust it to ensure that it meets its goal. For example, data about sales performance is feedback to a sales manager.
- **Control**: It involves monitoring and evaluating feedback to determine whether a system is moving toward the achievement of its goals. The control function then makes necessary adjustments to a system's input and processing components to ensure that it produces proper output. For example, a sales manager exercises control when reassigning salespersons to new sales territories after evaluating feedback about their sales performance

Theoretical approaches to systems have introduced many generalized principles. **Goal setting** is one such principle. It defines exactly what the system is supposed to do. There are principles concerned with system structure and behavior. **System boundary** is one such a principle. This defines the components that make up the system.

Anything outside the system boundary is known as **system environment.** A system can be made up of any number of **subsystems.** Each subsystem carries out part of the system function i.e. part of the system goal.

The subsystems communicate by passing messages between themselves. Several systems may share the same environment. Some of these systems may be connected to one another by means of a shared boundary, or **interface.** A system that interacts with other systems in its environment is called **open system**. Finally, a system that has the ability to change itself or environment in order to survive is called an **adaptive system**.

Characteristics of System

- Organization
- Interaction
- Interdependence
- Integration
- Central Objective
- **Organization-**It implies structure and order. It is the arrangement of components that helps to achieve objectives.
- **Interaction-**It refers to manner in which each component functions with other components of the system. In an organization, for example, purchasing must interact with production, advertising with sales, etc.

- **Interdependence-** It means that parts of the organization or computer system depend on one another. They are coordinated and linked together according to a plan. One subsystem depends on the input of another subsystem for proper functioning.
- **Integration-** It refers to the completeness of system. It is concerned with how a system is tied together. It is more than sharing a physical part or location. The parts of a system work together within the system even though each part performs a unique function.
- **Central Objective-**Objective may be real or stated. All the components work together to achieve that particular objective.

Why System Analysis ?

- The dissection of a system into its component pieces to study how those component pieces interact and work.

  (1) The survey and planning

  (2) The study and analysis

  (3) The definition

What is System Design ?

- The process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements is called System Design.

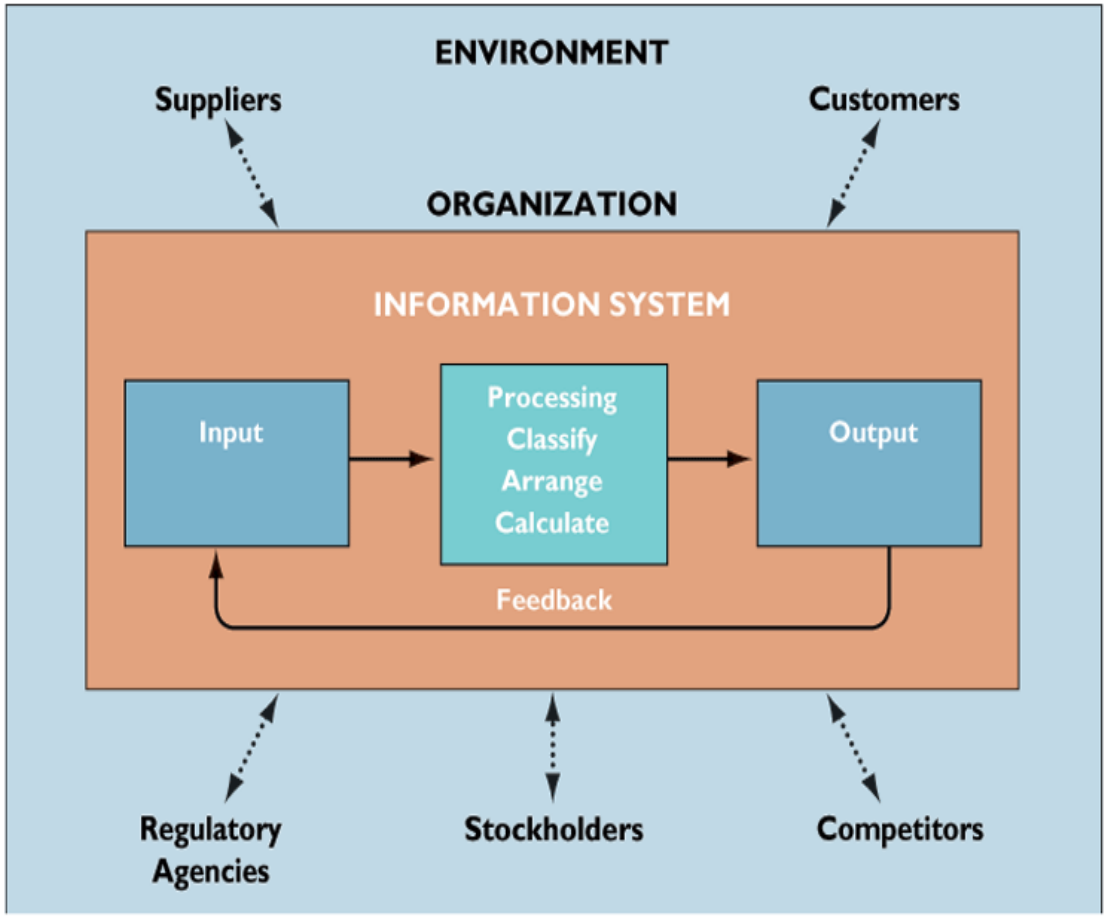
Need for System Analysis and Design

- Installing a system without proper planning leads to great user dissatisfaction and frequently causes the system to fall into disuse.
- Lends structure to the analysis and design of information systems.
- A series of processes systematically undertaken to improve a business through the use of computerized information systems

Information System

- In a simplest sense, a system that provides information to people in an organization is called **information system (IS).**
- Information systems in organizations capture and manage data to produce useful information that supports an organization and its employees, customers, suppliers and partners. So, many organizations consider information system to be the essential one.
- Information systems produce information by using data about significant people, places, and things from within the organization and/or from the external environment to make decisions, control operations, analyze problems, and create new products or services.
- **Information** is the data shaped into a meaningful form.
- **Data**, on the other hand, are the collection of raw facts representing events occurring in organizations or the environment before they have been organized and arranged into a form that people can understand and use.

- The three activities to produce information in an information system are input, processing, and output.
- **Input** captures or collects row data from within the organization or from its external environment for processing.
- **Processing** converts these raw data into the meaningful information.
- **Output** transfers this information to the people who will use it or to the activities for which it will be used.
- Information systems also require **feedback**, which is used to monitor the current information system output and compare it to the system goal.



- The two types of information systems are **formal and informal**.
- **Formal information** systems are based on accepted and fixed definitions of data and procedures for collecting, storing, processing, disseminating, and using these data with predefined rules.
- **Informal information systems**, in contrast, relay on unstated rules.
- Formal information systems can be **manual** as well as **computer based.**
- **Manual information** systems use paper-and-pencil technology.
- In contrast, **computer-based information systems (CBIS)** relay on computer hardware and software for processing and disseminating information.

- **Information Technology:**

Information technology is a contemporary term that describes the combination of computer technology (hardware and software) with telecommunications technology (data, image, and voice networks). Combination of hardware and software products and services that companies use to manage, access, communicate, and share information

- **Systems analysis:**

Systems analysis is a problem-solving technique that decomposes a system into its component pieces for the purpose of studying how well those component parts work and interact to accomplish their purpose.

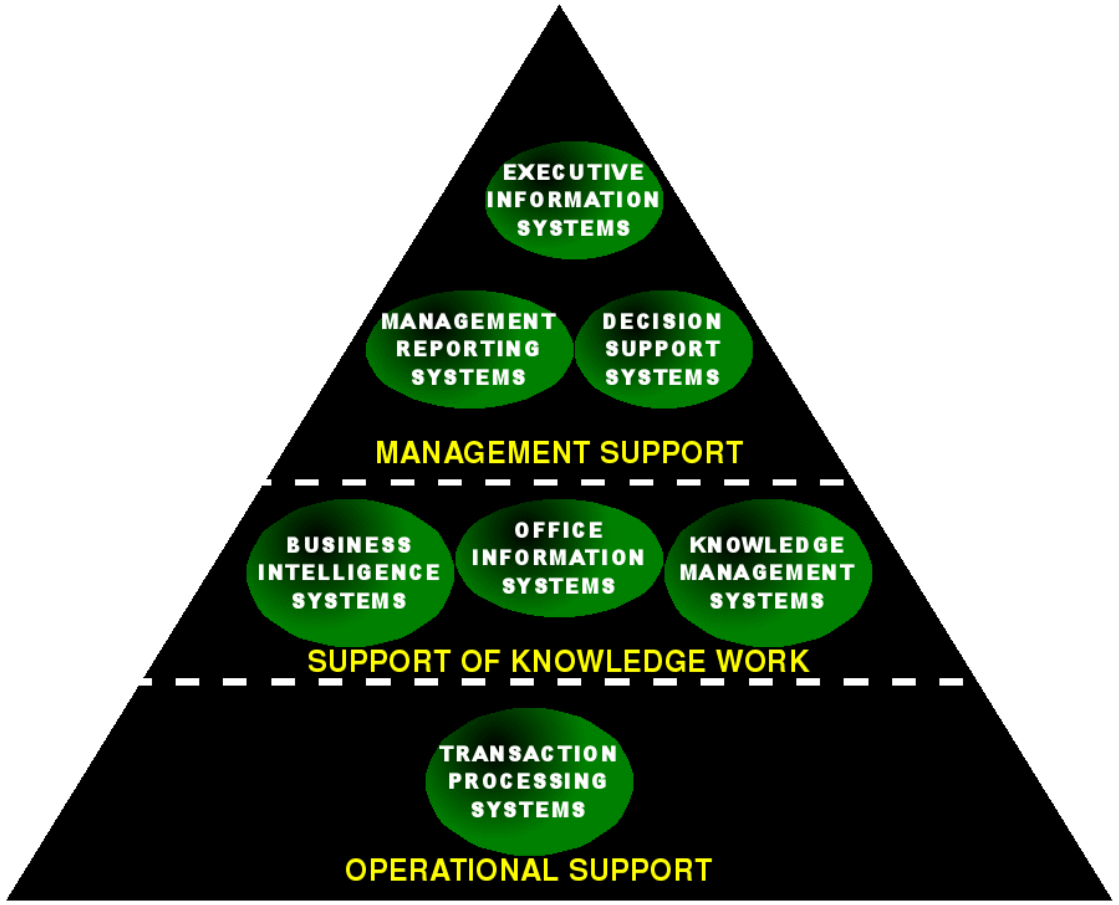
- **Information systems analysis:**

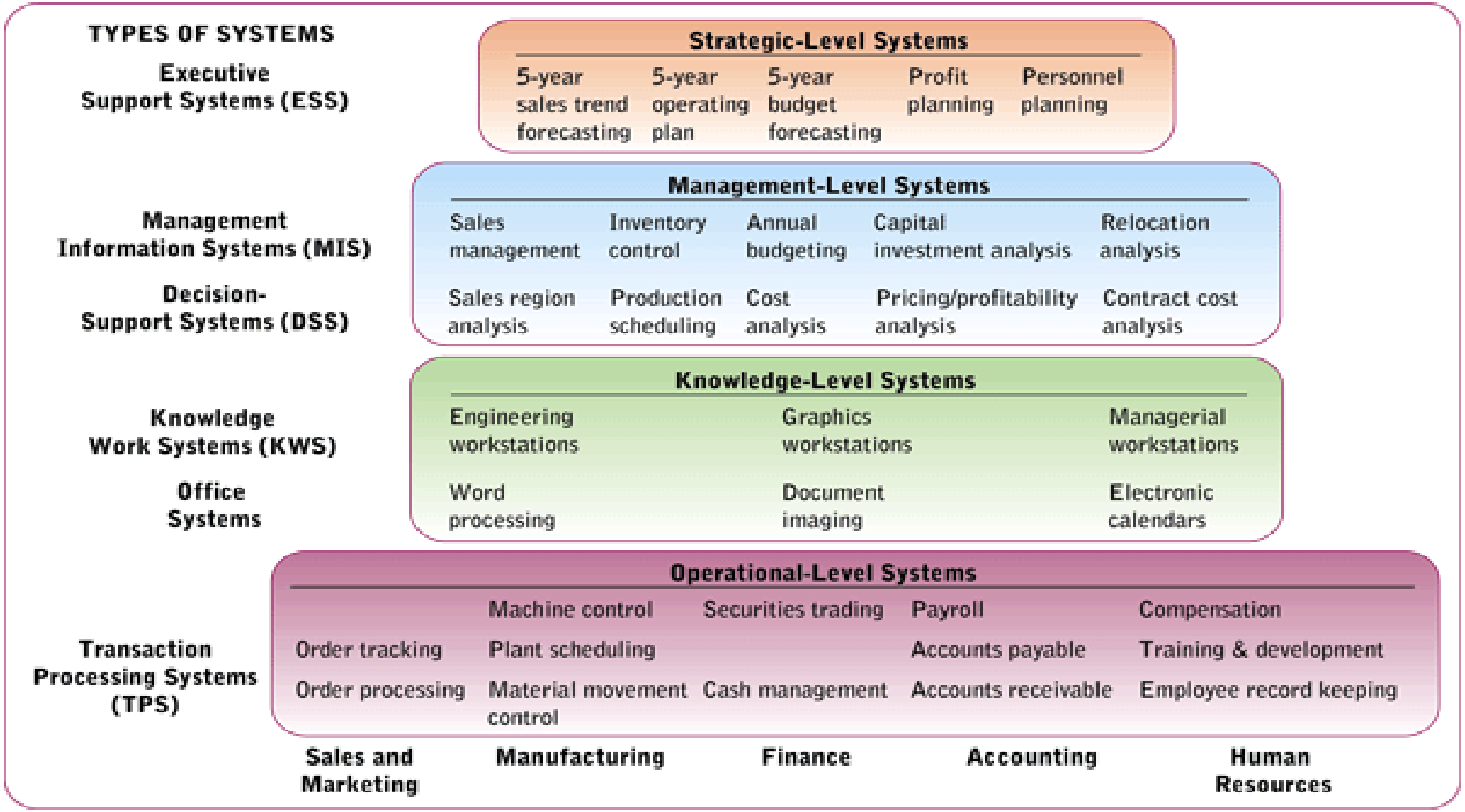
Information systems analysis is defined as those development phases in a project that primarily focus on the business problem, independent of any technology that can or will be used to implement a solution to that problem.

Types of Information System

- In practice there are several classes of information systems in organizations. Each class serves the needs of different types of users. These are:

1. Transaction processing system (TPS)
2. Management information system (MIS)
3. Decision support system (DSS)
4. Executive information system (EIS)
5. Expert system
6. Communication and collaboration system
7. Office automation system.




TYPES OF SYSTEMS
Executive Support Systems (ESS)
Management Information Systems (MIS)
Decision-Support Systems (DSS)
Knowledge Work Systems (KWS)
Office Systems
Transaction Processing Systems (TPS)
Strategic-Level Systems
5-year sales trend forecasting
5-year operating plan
5-year budget forecasting
Profit planning
Personnel planning
Management-Level Systems
Sales management
Inventory control
Annual budgeting
Capital investment analysis
Relocation analysis
Sales region analysis
Production scheduling
Cost analysis
Pricing/profitability analysis
Contract cost analysis
Knowledge-Level Systems
Engineering workstations
Graphics workstations
Managerial workstations
Word processing
Document imaging
Electronic calendars
Operational-Level Systems
Machine control
Securities trading
Payroll
Compensation
Order tracking
Plant scheduling
Accounts payable
Training & development
Order processing
Material movement control
Cash management
Accounts receivable
Employee record keeping
Sales and Marketing
Manufacturing
Finance
Accounting
Human Resources

Transaction Processing Systems(TPS)

- These are the computerized systems that perform and records the daily routine transactions necessary to conduct business. These systems serve the operational level of the organization. Some examples include sales order entry, hotel reservation systems, payroll, employee record keeping, and shipping.
- Transaction processing systems are central to a business. TPS failure for a few hours can cause a firm's demise and perhaps other firms linked to it. Managers need TPS to monitor the status of internal operations and the firm's relations with external environment. TPS are also major producers of information for the other types of systems.
- Online transaction processing systems (OLTPS) is an interactive data processing system that involves a direct connection between TPS programs and users. As soon as a single transaction is entered into a computer system, the program interacts immediately with the user for that transaction. It is often known as the live system where there is no time lag between data creation and its processing. A good example of this system is online ticket reservation system.

- TPS is a type of IS that manages data created in everyday operations. This includes storing, formatting, processing, retrieving, and creating some new aggregate data.
- Examples: purchasing transactions, sales orders, sales transactions, payroll, employee data, inventory
- Records daily, routine activities
- Serves supervisory level of management

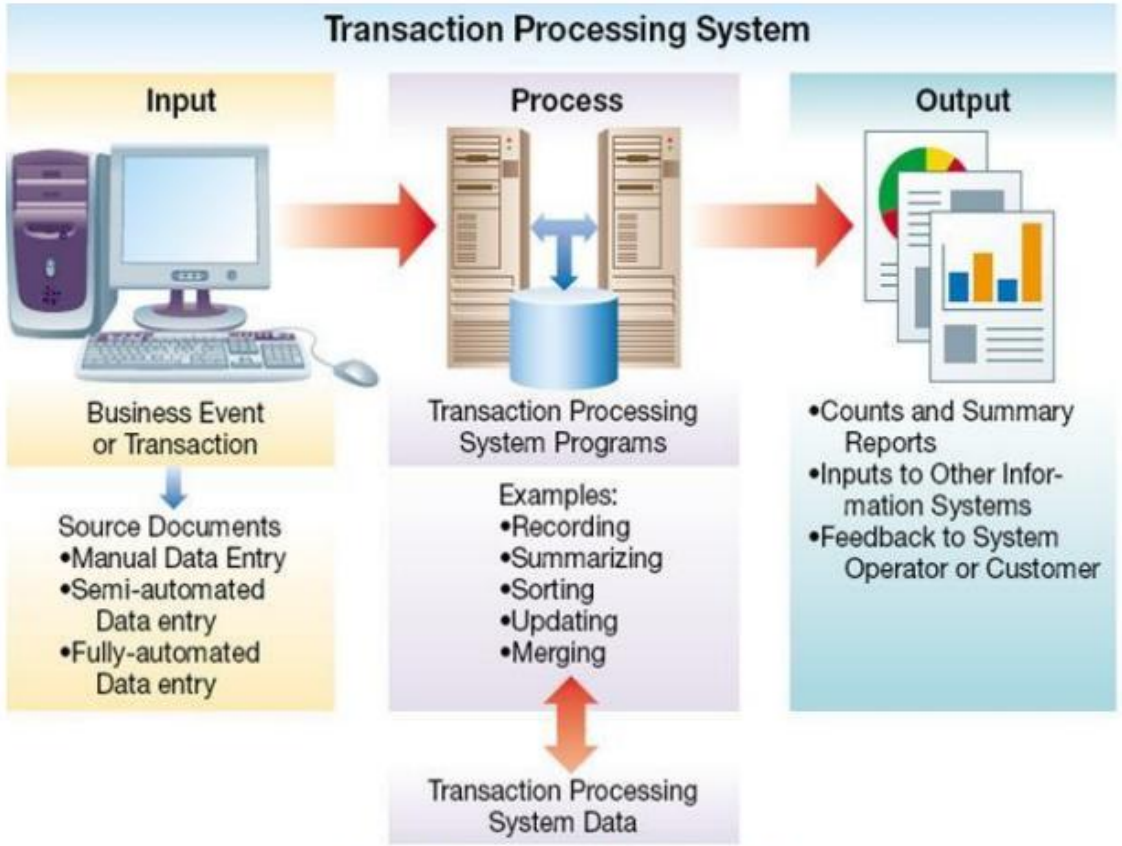


Figure: Transaction Processing Systems

	TYPE OF TPS SYSTEM				
	Sales/ marketing systems	Manufacturing/ production systems	Finance/ accounting systems	Human resources systems	Other types (e.g., university)
Major functions of system	Sales management	Scheduling	Budgeting	Personnel records	Admissions
	Market research	Purchasing	General ledger	Benefits	Grade records
	Promotion	Shipping/receiving	Billing	Compensation	Course records
	Pricing	Engineering	Cost accounting	Labor relations	Alumni
	New products	Operations		Training	
Major application systems	Sales order information system	Materials resource planning systems	General ledger	Payroll	Registration system
	Market research system	Purchase order control systems	Accounts receivable/payable	Employee records	Student transcript system
	Pricing system	Engineering systems	Budgeting	Benefit systems	Curriculum class control systems
		Quality control systems	Funds management systems	Career path systems	Alumni benefactor system

Example: TPS includes Airline Reservation Systems, Banking Systems, job scheduling systems and queue monitoring system on a typical shop floor, Processing of orders etc.

Management Information Systems (MISs)

- These are the information systems at the management level of an organization and serve management-level functions like planning, controlling, and decision-making. These systems provide reports that are usually generated on a predetermined schedule and appear in prearranged format.
- Typically, these systems use internal data provided by the transaction processing systems. These systems are used for structured decision-making and in some cases for semi-structured decision making as well.
- Salary analysis and sales reporting are the examples in which MIS can be used.
- Management information system, or MIS, broadly refers to a computer-based system that provides managers with the tools to organize, evaluate and efficiently manage departments within an organization.

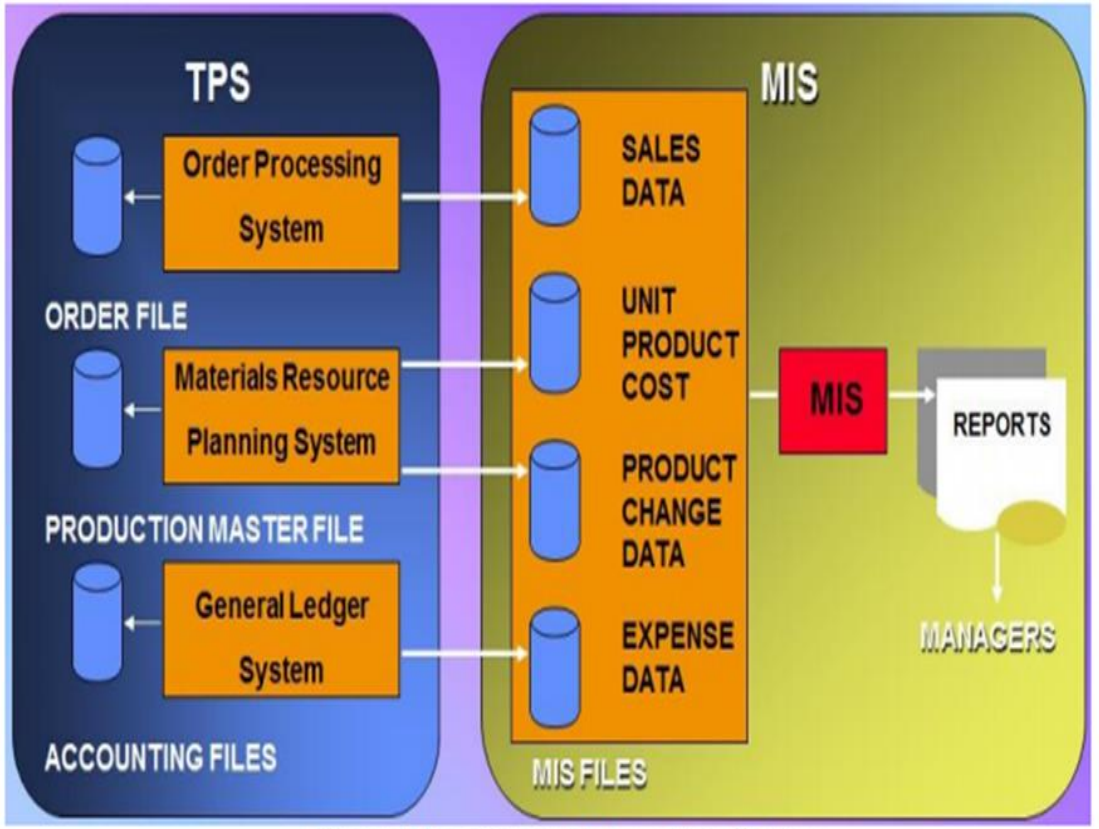


Figure: Management Information System

- MIS convert data from internal and external sources into information for managers.
- The source of data for an MIS usually comes from numerous databases. These databases are usually the data storage for Data Processing Systems.
- MIS summarise and report on the organisation's basic operations.
- MIS produce reports for managers interested in historic trends on a regular basis.
  - MIS operate at the tactical level
  - *Example: Annual budgeting*

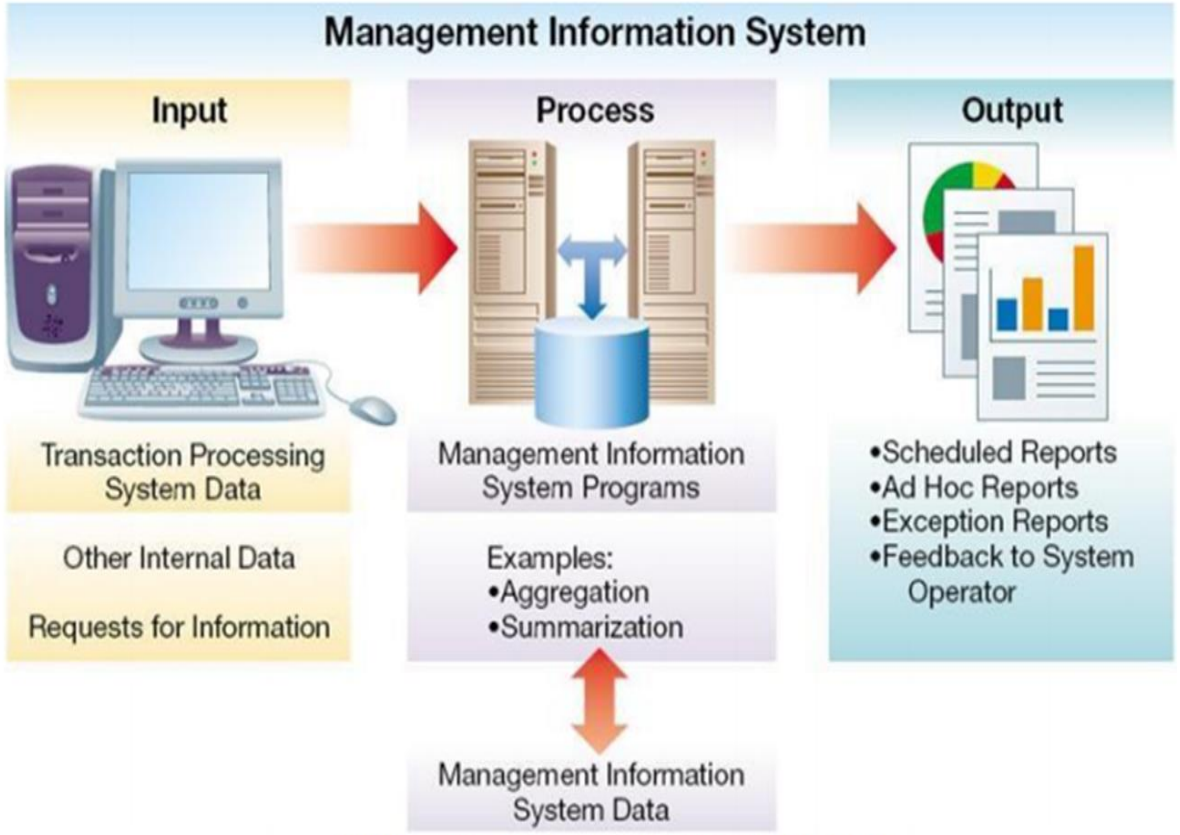


Figure: Management Information Systems

Properties of MIS

- **Management-oriented:** The basic objective of MIS is to provide information support to the management in the organization for decision making.
- **Management directed:** When MIS is management-oriented, it should be directed by the management because it is the management who tells their needs and requirements more effectively than anybody else.
- **Integrated:** It means a comprehensive or complete view of all the subsystems in the organization of a company.
- **Common database:** This is the basic feature of MIS to achieve the objective of using MIS in business organizations.

- **Computerized:** MIS can be used without a computer. But the use of computers increases the effectiveness and the efficiency of the system.
- **User friendly/Flexibility:** An MIS should be flexible.
- **Information as a resource:** Information is the major ingredient of any MIS.
- **Common data flows:** The integration of different subsystems will lead to a common data flow which will further help in avoiding duplicacy and redundancy in data collection, storage and processing.
- **Heavy planning-element:** The preparation of MIS is not a one or two day exercise. It usually takes 3 to 5 years and sometimes a much longer period.

Decision Support Systems (DSSs)

- These systems also serve at the management level of the organization. These systems combine data and sophisticated analytical models or data analysis tools to support semi structured and unstructured decision-making.
- These systems use internal information from TPS and MIS, and often information from external sources, such as current stock prices or product prices of competitors.
- DSS have more analytical power than other systems. Contract cost analysis is an example in which DSS can be used.
- A decision support system (DSS) is an information system application that provides its users with decision-oriented information whenever a decision-making situation arises. When applied to executive managers, these systems are sometimes called executive information systems (EIS).

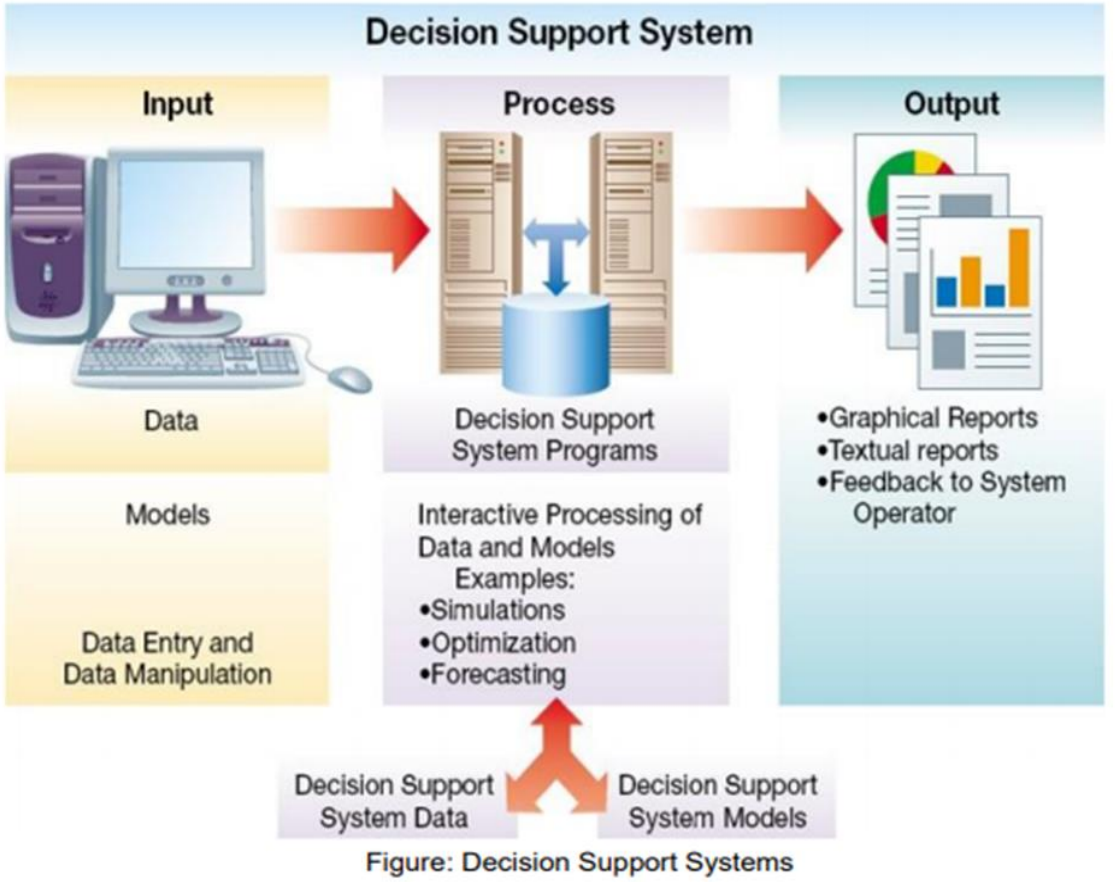


Figure: Decision Support Systems

- A Decision Support System (DSS) is an interactive computer-based system or subsystem intended to help decision makers use communications technologies, data, documents, knowledge and/or models to identify and solve problems, complete decision process tasks, and make decisions.
- Decision Support System is a general term for any computer application that enhances a person or group's ability to make decisions.
- Also, Decision Support Systems refers to an academic field of research that involves designing and studying Decision Support Systems in their context of Use

Some common DSS are:

Area	Common DSS Models
Accounting	Cost analysis, discriminant analysis, break-even analysis, auditing, tax computation and analysis, depreciation methods, budgeting
Corporate Level	Corporate planning, venture analysis, mergers and acquisitions
Finance	Discounted cash flow analysis, return on investment, buy or lease, capital budgeting, bond refinancing, stock portfolio management, compound interest, after-tax yield, foreign exchange values
Marketing	Product demand forecast, advertising strategy analysis, pricing strategies, market share analysis, sales growth evaluation, sales performance
Personnel	Labor negotiations, labor market analysis, personnel skills assessment, employee business expense, fringe benefit computations, payroll and deductions
Production	Product design, production scheduling, transportation analysis, product-mix inventory level, quality control, learning curve, plant location, material allocation, maintenance analysis, machine replacement, job assignment, material requirement planning
Management Science	Linear programming, decision trees, simulation, project evaluation and planning, queuing, dynamic programming, network analysis
Statistics	Regression and correlation analysis, exponential smoothing, sampling, time-series analysis, hypothesis testing

Table 1-1 Systems Development for Different IS Types

IS Type	*IS Characteristics*	*Systems Development Methods*
Transaction processing system	High-volume, data capture focus; goal is efficiency of data movement and processing and interfacing different TPSs	Process orientation; concern with capturing, validating, and storing data and with moving data between each required step
Management information system	Draws on diverse yet predictable data resources to aggregate and summarize data; may involve forecasting future data from historical trends and business knowledge	Data orientation; concern with understanding relationships among data so data can be accessed and summarized in a variety of ways; builds a model of data that supports a variety of uses
Decision support system	Provides guidance in identifying problems, finding and evaluating alternative solutions, and selecting or comparing alternatives; potentially involves groups of decision makers; often involves semi-structured problems and the need to access data at different levels of detail	Data and decision logic orientations; design of user dialogue; group communication may also be key, and access to unpredictable data may be necessary; nature of systems requires iterative development and almost constant updating

Executive Information Systems (EISs)

- These systems are also called executive support systems (ESSs) and serve the strategic level of the organization. These systems are designed to address unstructured decision making through advanced graphics and communication. It is used by top management.
- These systems incorporate data about external events such as new tax laws or competitors, but they also draw summarized information from internal MIS and DSS.
- These systems are not designed to solve a specific problem but they provide generalized computing and telecommunication capacity that can be applied to a changing array of problems. 5-year operating plan is an example in which EIS can be used.
- Another special characteristic of an ESS is its drill down capability, which is the ability of the system to provide information at any level of detail desired by the decision maker. For example, the CEO of a company may want the monthly sales of Product X for the entire company. Next, the CEO may want a breakdown of sales figures on a regional basis or on a store wide basis. The drill down facility can provide both.

It is a user friendly, interactive system, and almost intuitive to use; it has excellent menus and graphic capabilities designed to meet the information needs of top management engaged in long range planning, crisis management, and other strategic decisions.

Such systems assist in the making of decisions that require an in depth understanding of the firm and of the industry in which the firm operates.

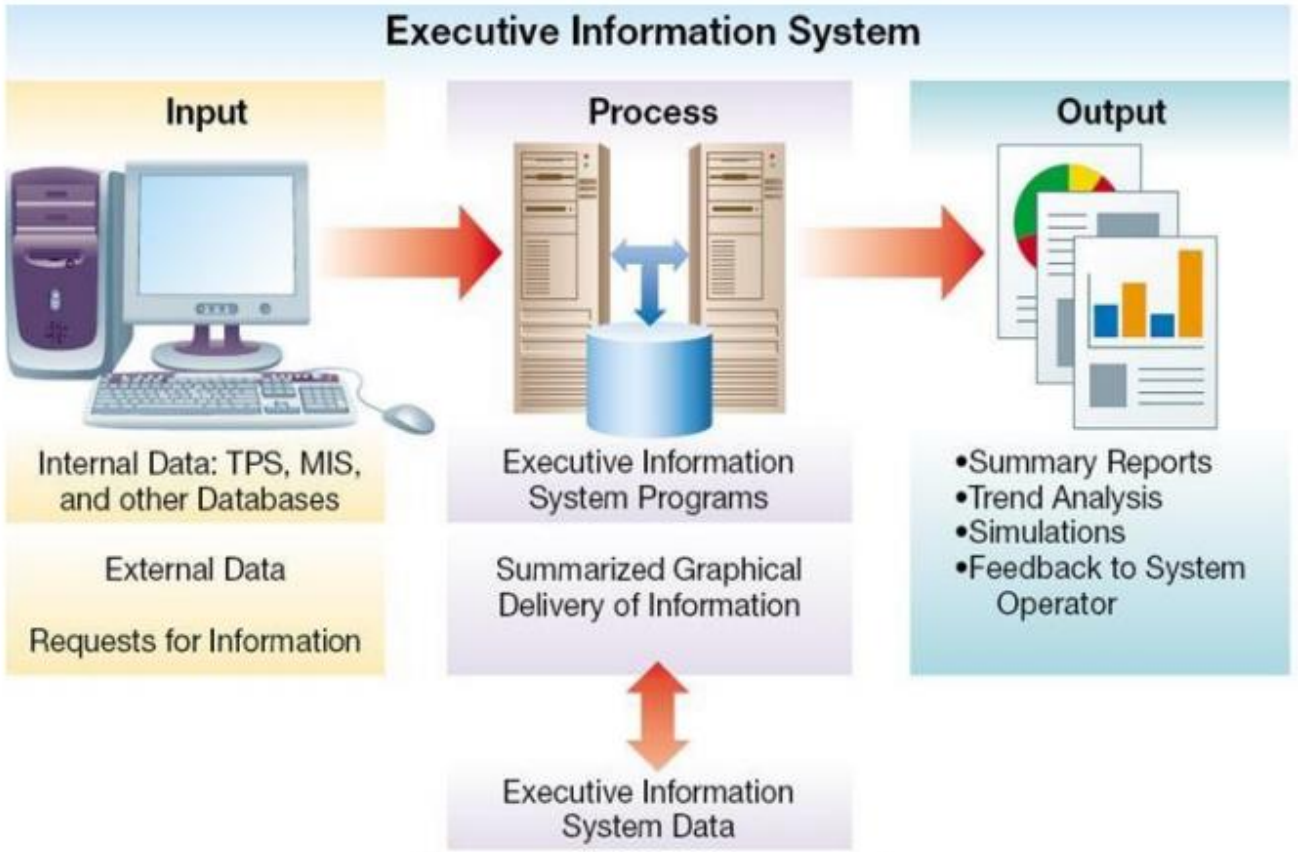


Figure: Executive Support Systems

Expert Systems

- An expert system is an extension of DSS that captures and reproduces the knowledge and expertise of an expert problem solver or decision maker and then simulates the “thinking” or “actions” of that expert.
- These systems imitate the logic and reasoning of the experts within their respective fields. Expert systems are implemented with artificial intelligence (AI) technology that captures, stores, and provides access to the reasoning of the experts.
- An expert system is a system that employs human knowledge captured in a computer to solve problems that ordinarily require human expertise.(Turban)
- An expert system is a computer program that tries to emulate human reasoning. It does this by combining the knowledge of human experts and then, following a set of rules, draws inferences.

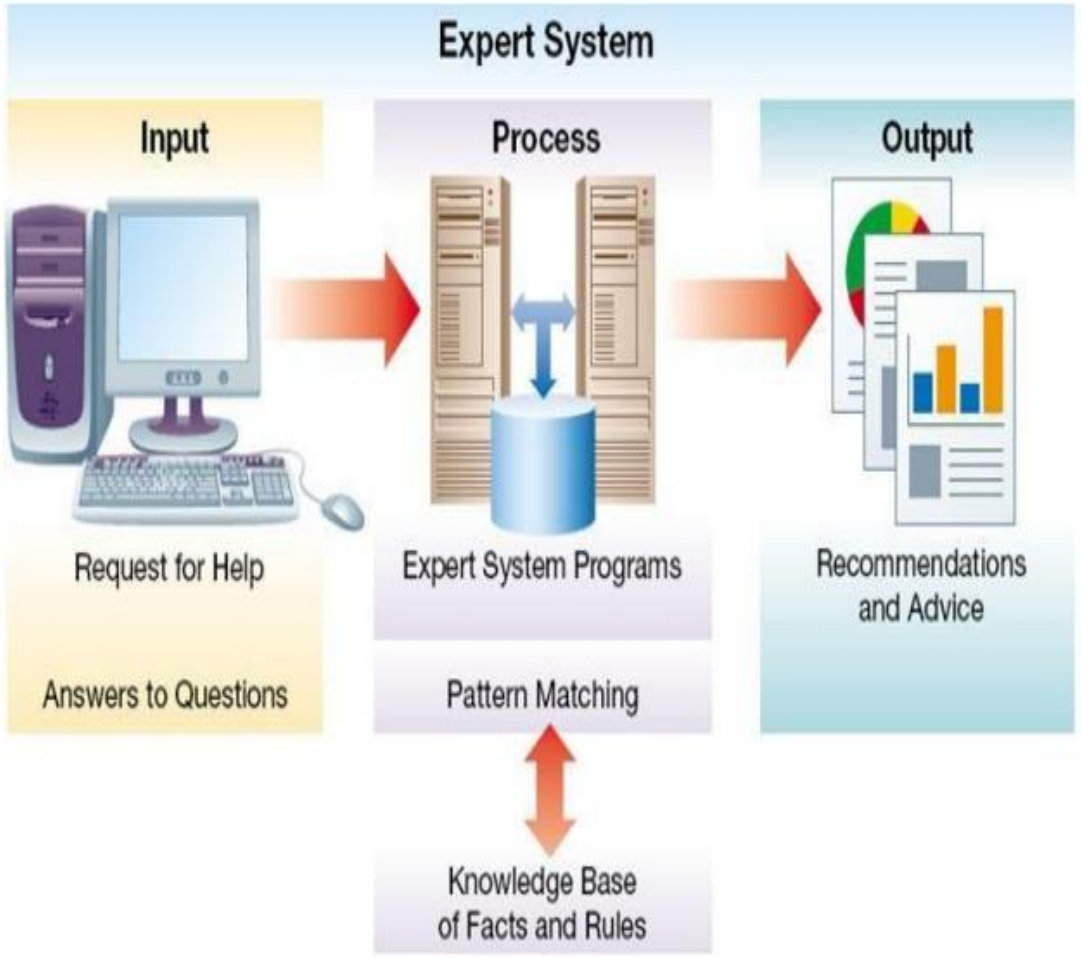
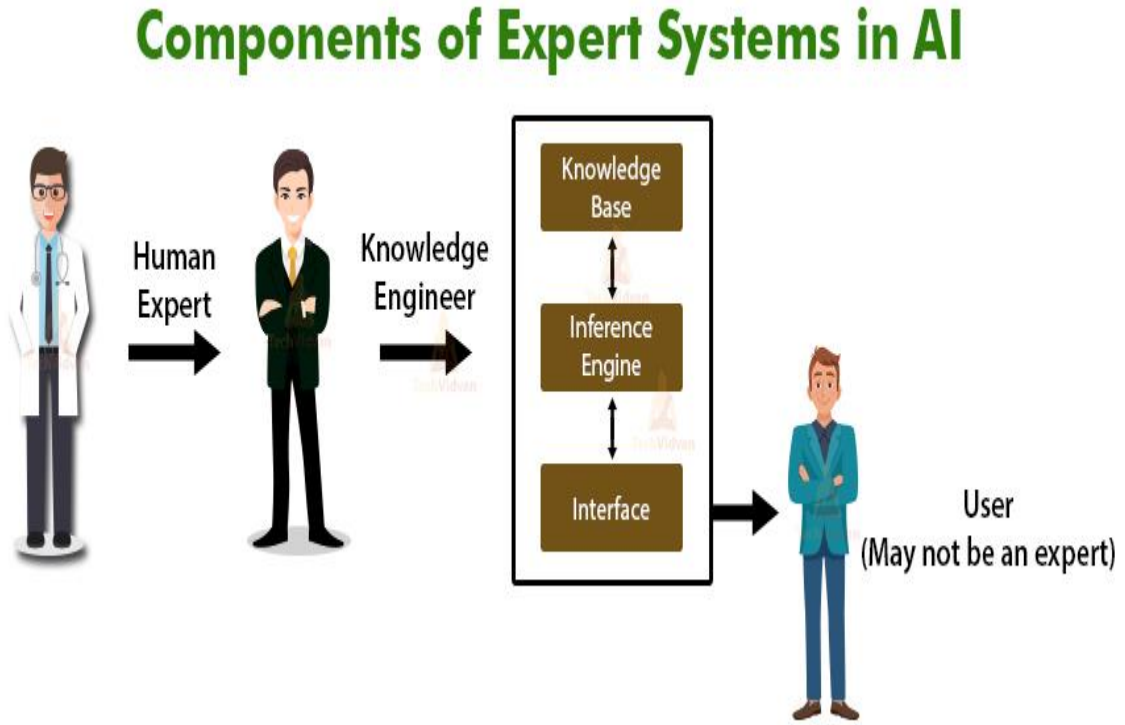


Figure: Expert Systems

- An expert system is made up of three parts:
  - A **knowledge base** stores all of the facts, rules and information needed to represent the knowledge of the expert.
  - An **inference engine** interprets the rules and facts to find solutions to user queries.
  - A **user interface** allows new knowledge to be entered and the system queried.



Communication and Collaboration Systems

- These systems enable more effective communications between workers, partners, customers and suppliers to enhance their ability to collaborate. These systems use network technology that allows companies to coordinate with other organizations across great distances.
- These systems create new efficiencies and new relationships between an organization, its customers and suppliers, and business partners redefining organizational boundaries.

Office Automation Systems

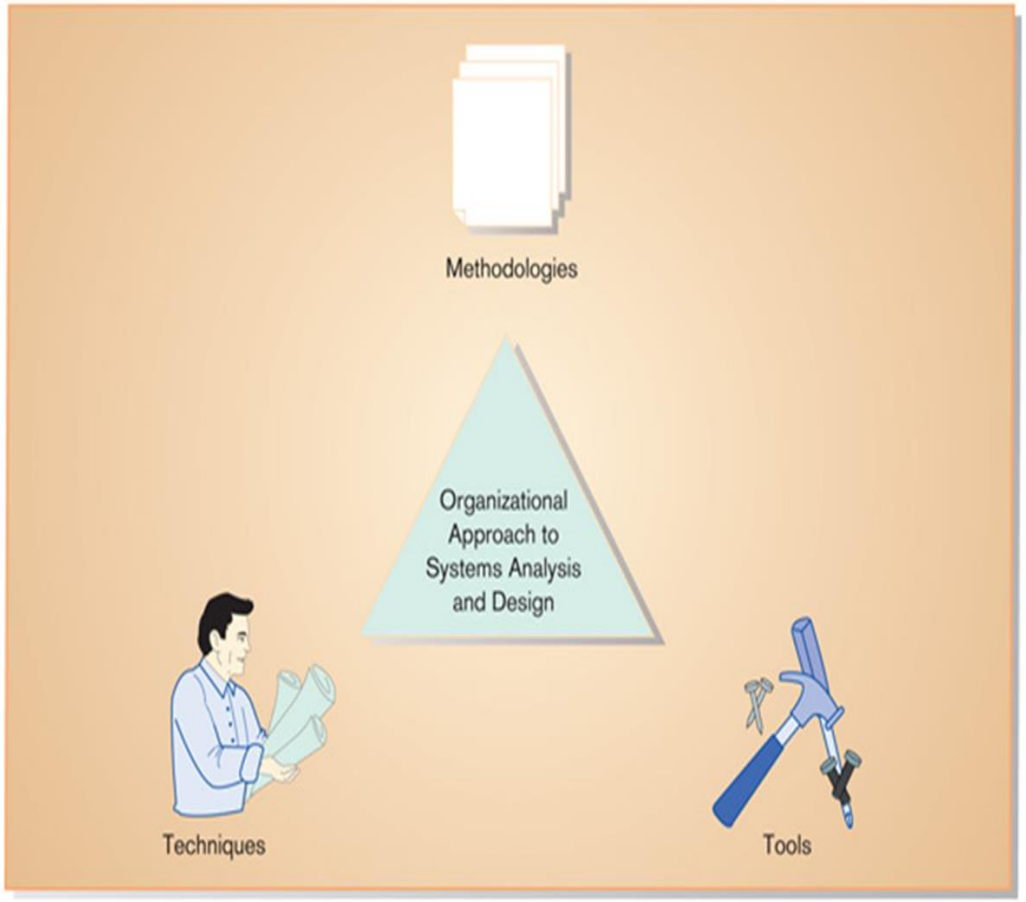
- Office automation (OA) is more than word processing and spreadsheet applications.
- Office automation systems support the wide range of business office activities for improved work flow and communication between workers, regardless of whether or not those workers are located in the same office.
- Office automation functions include word processing, spreadsheet applications, electronic mails, work group computing, fax processing, work flow management etc.
- Office automation systems can be designed to support both individuals and work groups.
- **Personnel information systems** are those designed to meet the needs of a single user. They are designed to boost an individual's productivity.
- **Work group information systems**, on the other hand, are designed to meet the needs of a work group. They are designed to boost the group's productivity.

Systems Analysis and Design

- **System analysis and design** is a complex, challenging, and simulating organizational process that a team of business and systems professionals uses to develop and maintain computer-based information systems. It is an organizational improvement process.
- Information systems are built and rebuilt for organizational benefits.
- **System Analysis:** Process of gathering and interpreting facts, diagnosing problems, and using the facts to improve the system.
- **System Design**: Process of planning a new system to replace or complement the old.
- Analysis specifies what the system should do and design states how to achieve the objective.

- An important result of system analysis and design is application software i.e. software designed to support organizational functions or processes such as inventory management, payroll, or mark-sheet analysis.
- In addition to application software, the total information system includes the hardware and systems software on which the application software runs, documentation and training materials, the specific job roles associated with the overall system, controls and the people who use the software along with their work methods.
- In systems analysis and design, we use various **methodologies, techniques and tools** that have been developed, tested, and widely used over the years to assist people during system analysis and design.

- Methodologies are comprehensive, multistep approaches to systems development that will guide your work and influence the quality of your final product: the information system. Methodologies use a standard set of steps. A methodology adopted by an organization will be consistent with its general management style. Most methodologies incorporate several development techniques.
- Techniques are particular processes that will help to ensure that your work is well thought-out, complete, and comprehensible to other on the project team. Techniques also provide support for a wide range of tasks like conducting interviews, planning and managing the activities in a system development project, diagramming the system's logic, and designing the reports that the system will generate.
- Tools are typically computer programs that make it easy to use and benefit from the techniques and to faithfully follow the guidelines of the overall development methodology.



Importance of Systems Analysis and Design

- Systems analysis and design is the collection of important activities that takes place when new information systems are being built or existing ones are changed. All the activities are needed to build good information systems.
- The systems developed by using systems analysis and design activities fulfill the requirements of organizations' personnel.
- Furthermore, we can develop information systems easily and rapidly because there are lots of supporting methodologies, tools, and techniques. The information system can be built in the most effective way.
- The systems also fit into an existing environment and will be very easy to use and maintain. By following the activities involved in systems analysis and design, we can develop high quality information system within allocated budget and time.

Information System Stakeholders

- A stakeholder is any person who has an interest in an existing or proposed information system. She/he may be technical or non-technical and internal or external worker.
- Stakeholders are also called information workers. An information worker involves in creating, collecting, processing, distributing and using information.
- There are six groups of stakeholders and each group has a different role in the same information system. But in practice, any individual person may play more than one role.
- For example, a system analyst may also work as a system designer. The six groups are:

1. System owners
2. System users
3. System designers
4. System builders
5. System analysts and project managers
6. Information technology vendors and consultants.

(1)System owners

- System owners are the information system's sponsors and chief advocates. They are usually responsible for funding the project of development, operate, and maintain the information system. They are interested with-how much will the system cost? And how much value or what benefit will the system return to the business? Every information system has one or more system owners. They usually come from the ranks of managers to supervisors.

(2)System Users

- These are the people who use or are affected by the information system on a regular basis. They are concerned with the system's functionality related with their jobs and the system's ease of learning and use. A system user may capture, validate, enter, respond, store and exchange data and information. System users are also called clients. To know business requirements, discussions with most users need to be kept.

(3)System Designers

- These are technology specialists who translate system users' business requirements and constraints into technical solutions. These are interested in information technology choices and the design of systems within the constraints of the chosen technology. They design the computer database, inputs, outputs, screens, networks, and programs that will meet the system users' requirements. These designs guide the construction of the final system.

(4)System Builders

- These are also technology specialists who construct information systems and components based on the design specifications generated by the system designer.

(5)Systems Analysts and Project Managers

A. Systems Analyst:

- Although, many people in organizations are responsible for systems analysis and design, in most organizations the systems analyst has the primary responsibility. The primary role of a systems analyst is to study the problems and needs of an organization in order to determine how people, methods and information technology can best be combined to bring about improvements in the organization. System analysts identify and validate problems and needs and ensure that the technical solution fulfills these problems and needs. Systems analysts study the system and identify and validate its problems and needs for system owners and users and ensure that the technical solution fulfills the business needs.

B. Project Manager:

- To build a good information system and applications all the stakeholders must work together as a team. Teams require leadership. For this reason, usually one or more of these stakeholders takes on the role of project manager to ensure that systems are developed on time, within budget and acceptable quality. So, project manager is responsible for planning, monitoring, and controlling projects with respect to schedule, budget, deliverables, customer satisfaction, technical standards and system quality.

(6)Information Technology Vendors and Consultants

- Most information systems are dependent on information technology that must be selected, installed and customized, integrated into business, and technically supported. This technology is developed, sold, and supported by IT vendors. Similarly, many businesses rely on external consultants to help them develop or acquire information systems and technology. The use of consultants may be driven by the need for specialized knowledge or skills or by an immediate need to complete a project.

A Modern Approach to Systems Analysis and Design

- **1950s:** All applications had to be developed in machine language or assembly language. They had to be developed from scratch because due to the absence of software industry.
- **1960s:** Smaller, faster, less expensive computers, beginning of the software industry, use in-house development.
- **1970s:** Realized how expensive to develop customized information system for every application , started development of database management system.
- **1980s:** The software industry expended greatly, CASE(computer aided software engineering) tools.
- Started writing application software in oop languages, graphics were used, developed less software in-house and bought more from software vendors.
- **1990s:** focus on system integration, GUI(Graphical user interface) applications, client/server platforms, Internet.
- The new century: Web application development, wireless PDAs (personal digital assistants, eg pocket PCs), ASP(application service provider).

Process-Oriented Approach

- Focus is on flow, use and transformation of data in an information system
- Involves creating graphical representations such as data flow diagrams and charts
- Data are tracked from sources, through intermediate steps and to final destinations
- Natural structure of data is not specified
- Disadvantage: data files are tied to specific applications

Data-Oriented Approach

- Depicts ideal organization of data, independent of where and how data are used
- Data model describes kinds of data and business relationships among the data
- Business rules depict how organization captures and processes the data

The Systems Analyst As A Modern Business Problem Solver

Why do businesses need Systems Analysts?

- The system analyst bridges the communications gap between those who need the computer and those who understand the technology

Systems analysts

- Systems analysts are people who understand both business and computing.
- Systems analysts study business problems and opportunities and then transform business and information requirements of the business into the computer-based information systems and computer applications that are implemented by various technical specialists including computer programmers

A formal definition:

- A systems analyst facilitates the study of the problems and needs of a business to determine how the business system and information technology can best solve the problem and accomplish improvements for the business.
- The product of this activity may be improved business processes, improved information systems, or new or improved computer applications frequently all three.
- When information technology is used, the systems analyst is responsible for: the efficient capture of data from its business source the flow of that data to the computer the processing and storage of that data by the computer the flow of useful and timely information back to the business and its people

Roles of the System Analyst

- The analyst plays a key role in information systems development projects.
- Must understand how to apply technology to solve business problems.
- Analyst may serve as change agents who identify the organizational improvement.

Preparing Career as a Systems Analyst

System analysts are the key individuals in the information system development process. To succeed as a system analyst, you will need to develop the following skills.

- **Working Knowledge of Information Technology:** This is the technical skill. The analyst must be aware of both existing and emerging information technology. Such knowledge can be acquired by college courses, seminars and training programs.
- **Computer Programming Experience and Expertise:** This is also a technical skill needed by systems analysts. Most system analyst need to be proficient in one or more high level programming language.
- **General Knowledge of Business Processes and Terminology:** Most of the systems today are business related and the systems analysts must be able to communicate with business experts to gain understanding of their problems and needs. So, this skill is must. To develop this skill, the system analyst should have knowledge about the courses like accounting, finance, business law and ethics, economics, manufacturing, marketing, operations management, human resource management, organizational behavior etc.

- **General Problem Solving Skill:** The systems analyst must be able to take a large business problem, break down that problem into its component parts, analyze the various aspects of the problem, and then assemble into an improved system to solve the problem. To develop this skill, a system analyst should have knowledge about critical thinking and reasoning.
- **Good Interpersonal Communication Skill:** To know the user requirements, an analyst must be able to communicate orally and in writing. To develop this skill, the courses like business and technical writing, business and technical speaking, interviewing and listening will be effective.
- **Good Interpersonal Relations Skill:** The systems analysts should interact with all the stakeholders in the information system development project. To do this they must have this skill. To improve this skill, the analyst should have knowledge about the courses like teamwork, principles of persuasion, managing change and conflict, and leadership.
- **Flexibility and Adaptability:** No two projects are alike. So, a successful system analyst must learn to be flexible and to adapt to unique challenges and situations.

- **Character and Ethics:** The system analyst should have strong character and a sense of right and wrong. This is needed to hide the sensitive and confidential facts and information of an organization.
- **System Analysis and Design Skill**: All systems analysts should know concepts and principles, tools, and techniques of information systems development.

Developing Information Systems and System Development Life Cycle (SDLC)

- Most organizations use a standard set of steps, called a **systems development methodology** to develop and support their information systems.
- It is a standard process followed in an organization to conduct all the steps necessary to analyze, design, implement, and maintain information systems. And **systems development life cycle (SDLC)** is the traditional methodology used to develop, maintain, and replace information systems.
- It includes different phases as shown in the figure below. This representation of SDLC is sometimes referred to as the **waterfall model** or classic life cycle.

- Phases in SDLC:
  - Planning
  - Analysis
  - Design
  - Implementation
  - Maintenance

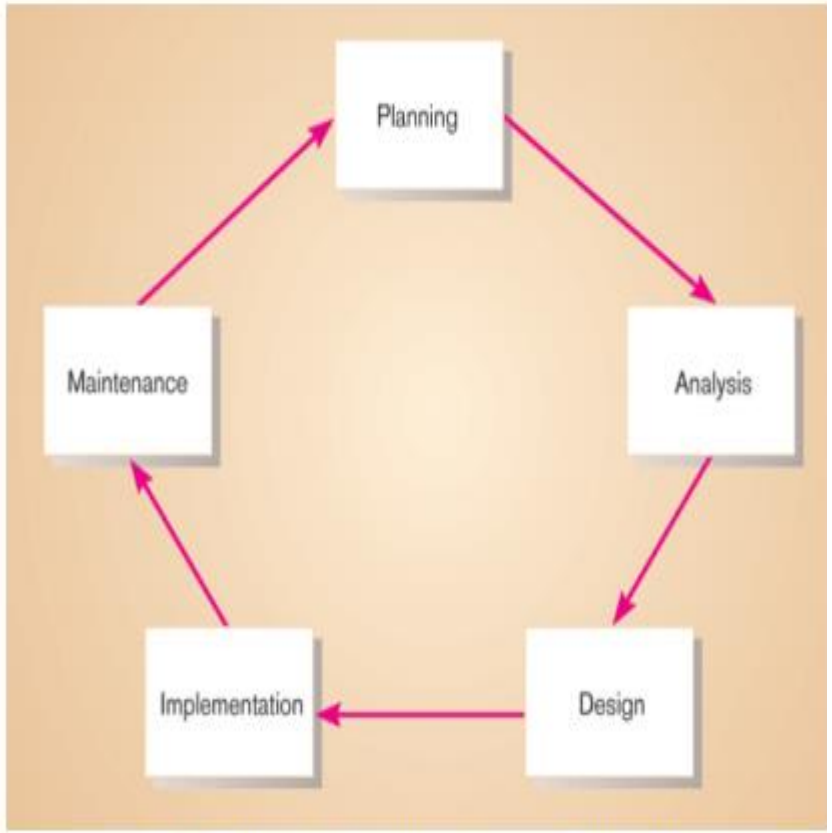
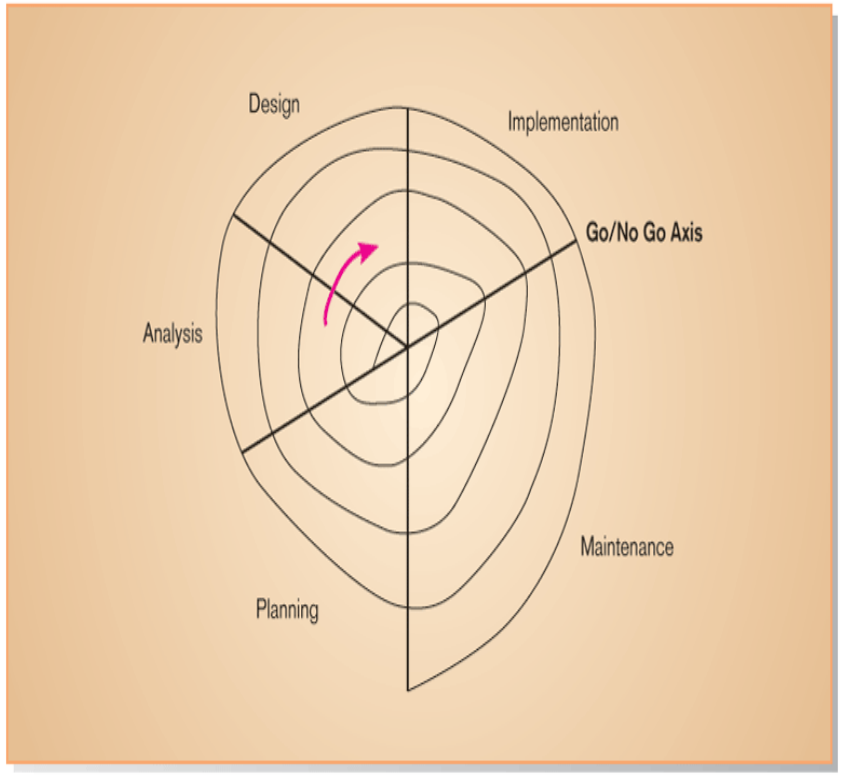


Fig: The systems development life cycle

Figure 1-4 Evolutionary model SDLC



- The first phase is called **planning**.
- In this phase, someone identifies the need for a new or enhanced system.
- These needs are then analyzed, prioritized and arranged into a plan for the IS department.
- Here, a potential information systems project is explained and an argument for continuing or not continuing with the project is presented; a detailed plan is also developed for conducting the remaining phases or the SDLC for the proposed system.
- The next phase is called **analysis**.
- During this phase, the analyst studies the current system and proposes alternative replacement systems.
- Here, the analyst thoroughly studies the organization's current procedures and the information systems used to perform organizational tasks.

- The analyst work with users to determine what the users want from a proposed system.
- The analyst carefully studies any current systems, manual and computerized, that might be replaced or enhanced as part of this project.
- The analyst studies the requirements and structures them according to their interrelationships and eliminates any redundancies; generates alternative initial designs to match the requirements; compare these alternatives to determine which best meets the requirements within the cost, labor, and technical levels the organization is willing to commit to the development process.
- The output of this phase is a description of the recommended alternative solution. Once the recommendation is accepted by owners, you can begin to make plans to acquire any hardware and system software necessary to build or operate the system as proposed

- The next phase is called **design**. During this phase, you convert the description of the recommended alternative solution into logical and then physical system specification.
- Here, you must design all aspects of the system form input and output screens to reports, databases, and computer processes.
- **Logical** design is the part of the design process that is independent of any specific hardware or software platform. Theoretically, the system could be implemented on any hardware and systems software.
- **Physical** design is the part of the design phase in which the logical specifications of the system form logical design are transformed into technology-specific details from which all programming and system construction can be accomplished.

- The next phase is called **implementation**.
- In this phase, the information system is coded, tested, installed, and supported in the organization.
- During coding, programmers write the programs that make up the information system.
- During testing, programmers and analysts test individual programs and the entire system in order to find and correct errors.
- During installation, the new system becomes a part of the daily activities of the organization. Implementation activities also include initial user support such as the finalization of documentation, training programs, and ongoing user assistance

- The final phase of SDLC is called **maintenance**.
- In this phase, information system is systematically repaired and improved.
- When a system is operating in an organization, users sometimes find problems with how it works and often think of better ways to perform its functions.
- Also the organization’s needs with respect to the system change over time.
- In maintenance, you make the changes that users ask for and modify the system to reflect changing business conditions.

TABLE 1-1 Products of SDLC Phases

Phase	Products, Outputs, or Deliverables
Planning	Priorities for systems and projects; an architecture for data, networks, and selection hardware, and information systems management are the result of associated systems
	Detailed steps, or work plan, for project
	Specification of system scope and planning and high-level system requirements or features
	Assignment of team members and other resources
	System justification or business case
Analysis	Description of current system and where problems or opportunities exist, with a general recommendation on how to fix, enhance, or replace current system
	Explanation of alternative systems and justification for chosen alternative
Design	Functional, detailed specifications of all system elements (data, processes, inputs, and outputs)
	Technical, detailed specifications of all system elements (programs, files, network, system software, etc.)
	Acquisition plan for new technology
Implementation	Code, documentation, training procedures, and support capabilities
Maintenance	New versions or releases of software with associated updates to documentation, training, and support

The Heart of System Development Process

Figure 1-8 The analysis–design–code–test loop

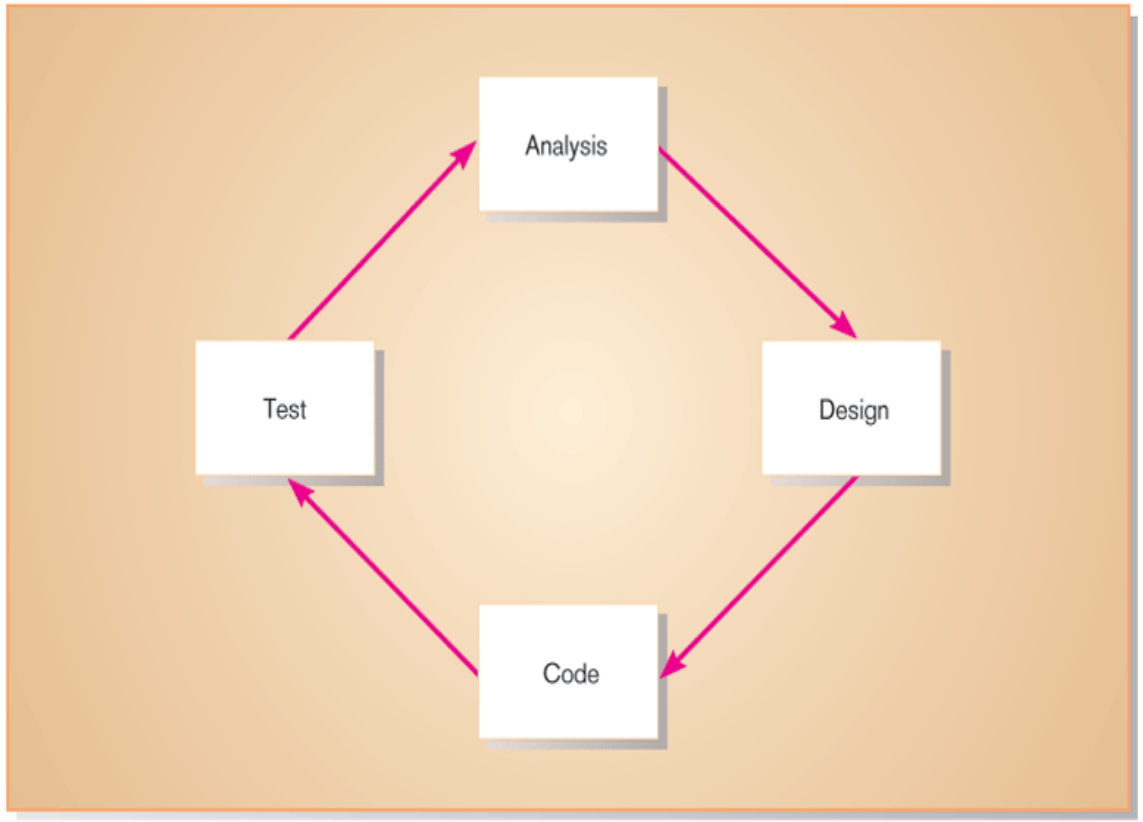
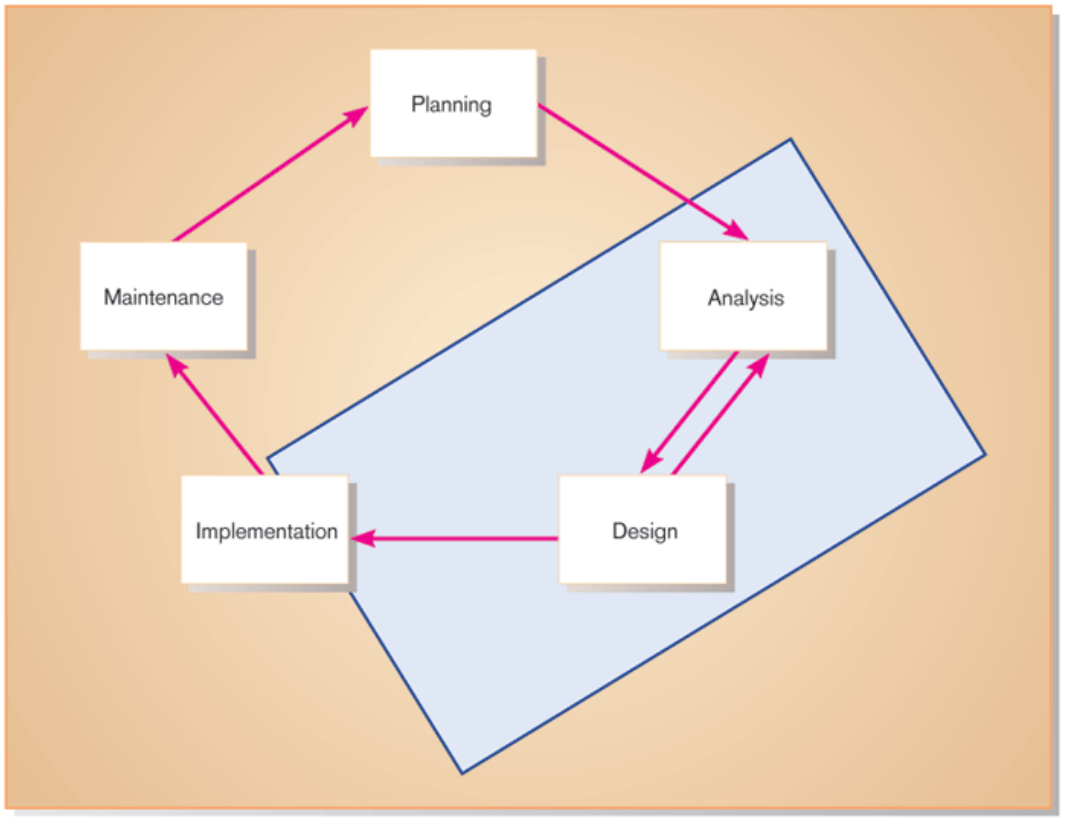


Figure 1-9 The heart of systems development



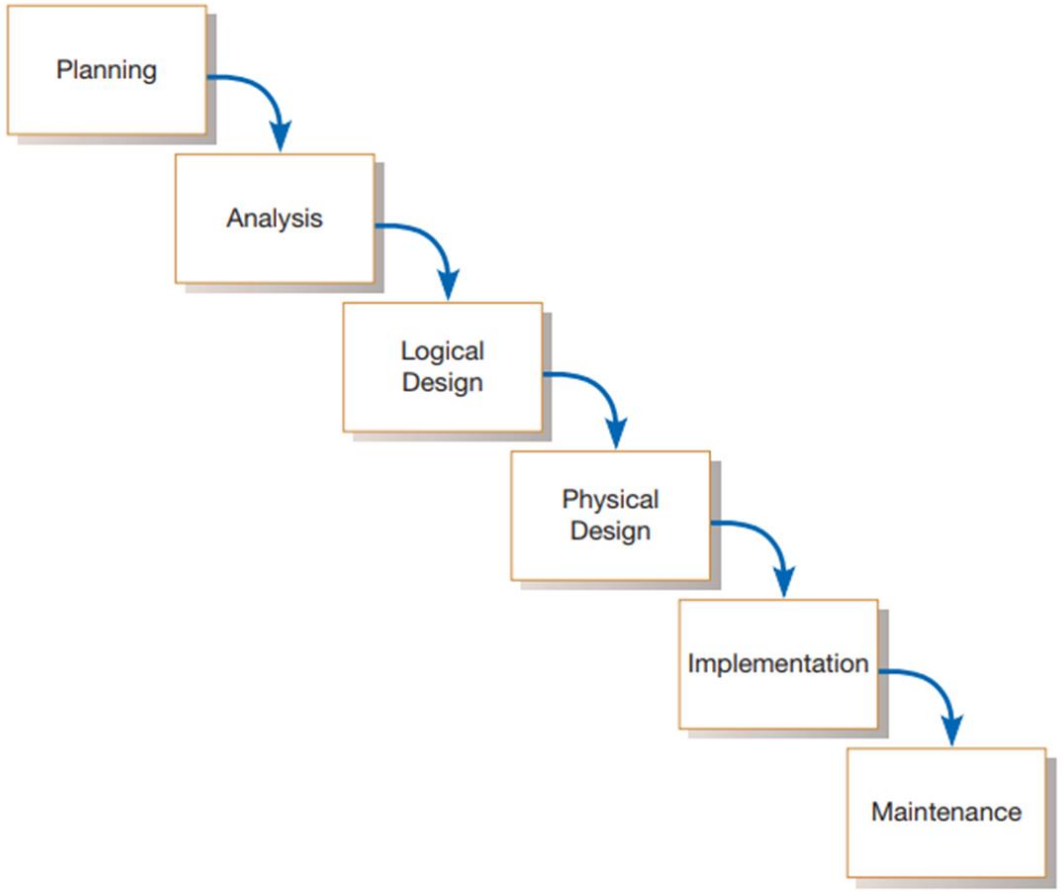
Current practice combines analysis, design, and implementation into a single iterative and parallel process of activities

- The heart of system development is analysis-design-implementation.
- After collecting the system requirements, they are thoroughly analyzed by experts.
- After analyzing them properly, the design for implementation is done by keeping a stress on meeting the requirements.
- As a next step, the system design is implemented with the help of information from previous stages, so that the system meets the expected goals.

The Traditional Waterfall SDLC

- Waterfall model (sometimes called classic life cycle or the linear sequential model) is the oldest and the most widely used paradigm for information systems development.
- While it does have weaknesses, it is significantly better than a haphazard approach. This model is suitable for the projects in which user requirements are certain and precise.
- Note how the flow of the project begins in the planning phase and from there runs “downhill” to each subsequent phase, just like a stream that runs off a cliff
- At the completion of each phase, a milestone has been reached and a document is produced to be approved by the stakeholders before moving to the next phase; painstaking amounts of documentation and signoffs through each part of the development cycle is required.
- The pure waterfall lifecycle consists of several non-overlapping stages, as shown in the following figure.
- The model begins with planning & analysis and continues with logical design, physical design, implementation (i.e. coding, testing) and maintenance.

- The waterfall model is a linear, sequential approach to the system development life cycle (SDLC) that is popular in software engineering and product development.
- The waterfall model emphasizes a logical progression of steps.



Waterfall model can be used when:

- Requirements are not changing frequently
- Application is not complicated and big
- Project is short
- Requirement is clear
- Environment is stable
- Technology and tools used are not dynamic and is stable
- Resources are available and trained

Advantages of Waterfall Model

- Simple and easy to understand and use.
- Easy to manage due to the rigidity of the model- each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

Problems with Waterfall Approach

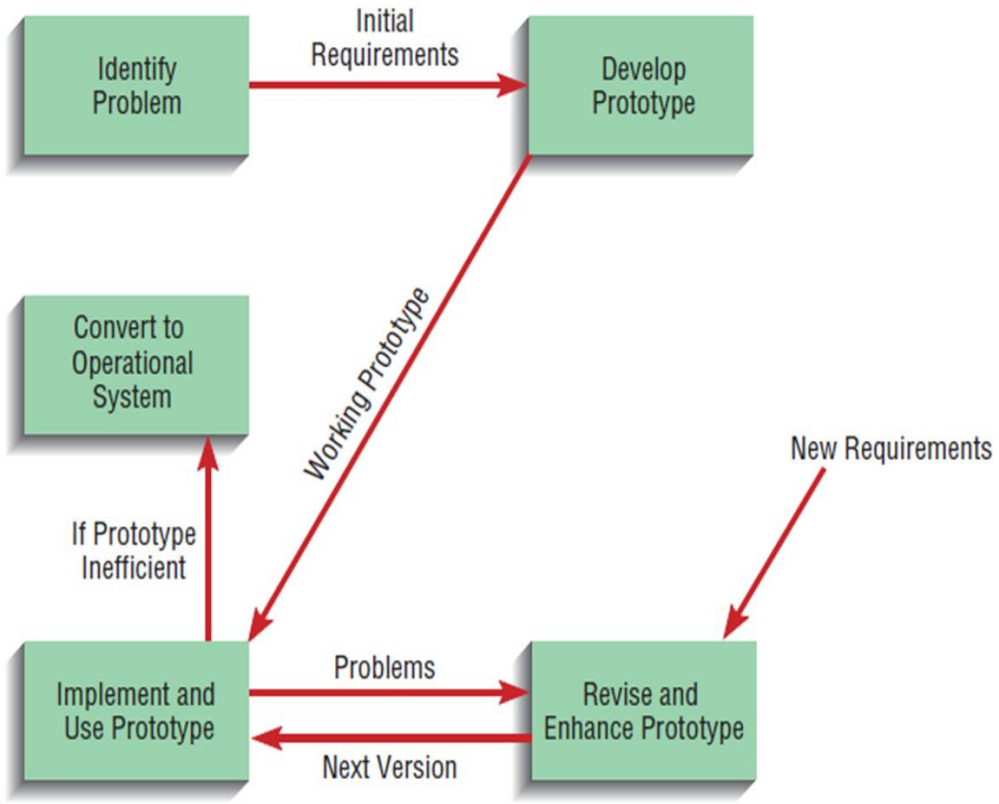
- System requirements “locked in” after being determined (can't change).
- Limited user involvement (only in requirements phase).
- Too much focus on milestone deadlines of SDLC phases to the detriment of sound development practices.
- High amounts of risk and uncertainty.
- Poor model for long and ongoing projects.
- Cannot accommodate changing requirements.
- Difficult to measure progress within stages.
- Adjusting scope during the life cycle can end a project.

Different Approaches to Improving Information Systems Development

- Several different approaches have been developed in the continuous effort to improve the systems analysis and design process.
- The two important approaches are prototyping and joint application development (JAD).

Prototyping Approach

- Prototyping is a form of rapid application development (RAD).
- Prototyping is a rapid, iterative, and incremental process of systems development in which requirements are converted to a working system that is continually revised through close work between the development team and the users.
- We can build a prototype with any computer language or development tool, but special prototyping tools have been developed to simply the process.
- A prototype can be developed with some fourth-generation language (4GL), with the query and screen and report design tools of a database management system, and with tools called computer-aided software engineering (CASE) tools.
- In prototyping, the analyst works with users to determine the initial or basic requirements for the system. The analyst then quickly builds a prototype.



When the prototype is completed, the users work with it and tell the analyst what they like and do not like about it. The analyst uses this feedback to improve the prototype and takes the new version back to the users. This iterative process continues until the users are relatively satisfied with what they have seen. Ideally, the prototype serves as a mechanism for identifying information system requirements. In this case, we throw away the prototype (also called throwaway prototype) after identifying requirements. The actual information system is developed with an eye toward quality and maintainability based on the requirements.

Advantages of prototypes

- Potential for changes to the system early in the development.
- Opportunity to stop developing a non-working system.
- Possibility of developing system that closely addresses users' requirements and expectations.
- Flexibility in design.
- The developed prototype can be reused by the developer for more complicated projects in the future.
- Missing functionalities can be easily figured out.

Disadvantages of prototypes:

- Prototyping is difficult to manage.
- Increase cost
- It is a time consuming if customer asks for changed in prototype.
- The methodology may increase the system complexity as scope of the system may expand beyond original plans.
- Customer may get confused in the prototypes and real systems.
- The invested effort in the preparation of prototypes may be too much if not properly monitored.

Joint Application Design (JAD)

- **Users, Managers and Analysts work together for several days**
- **System requirements are reviewed**
- Structured process involving users, analysts, and managers.
- Several-day intensive workgroup sessions.
- Purpose: to specify or review system requirements.
- Joint application design (JAD) is a group-based method for collecting user requirements and creating system designs.
- It is used within the systems analysis and design stages of the SDLC.
- Unlike the traditional SDLC, where the analysts interview individual users of the new information system to understand their needs JAD has a meeting in which all users meet simultaneously with analysts.
- During the meeting, all users jointly define and agree upon systems requirements.

Phases of JAD Model

Since you have become familiar with the JAD concept, it is time to know about its phases and how the model's design and development approach works:

1. **Define Specific Objectives**: The facilitator, in partnership with stakeholders, set all the objectives as well as a list of items which is then distributed to other developers and participants to understand and review. This objective contains elements like the scope of this projected system, its potential outcome, technical specification required, etc.

2. **Session Preparation**: The facilitator is solely responsible for this preparation where all relevant data is collected and sent to other members before time. For better insight, research carried out to know about the system requirement better and gather all the necessary information for development.

3. **Session Conduct**: Here the facilitator is accountable to identify those issues which have to be working out for making the system error-free. Here the facilitator will serve as a participant but will not have a say regarding any information.

4. **Documentation**: After the product is developed, the records and published documents are put forward into the meeting so that the stakeholders and consumers can approve it through the meeting.

Advantages:

- Saves time
- Greater support for, and acceptance of new systems
- Produces higher quality systems
- Easier implementation
- Lower training costs

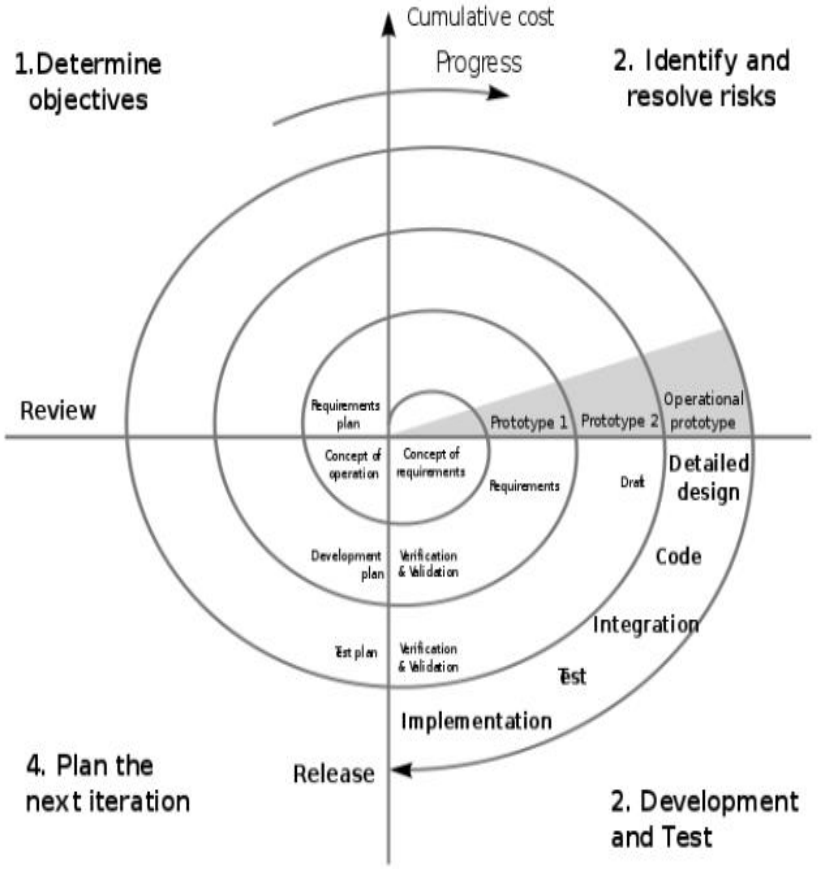
Disadvantages:

- Very difficult to get all users to JAD meetings
- All the problems that may be caused by any group process

Spiral Approach

- The spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model.
- This Spiral model is a combination of iterative development process model and sequential linear development model i.e. the waterfall model with a very high emphasis on risk analysis.
- It allows incremental releases of the product or incremental refinement through each iteration around the spiral.
- The exact number of loops of the spiral is unknown and can vary from project to project. Each loop of the spiral is called a Phase of the software development process.

- The functions of these four quadrants are discussed below-
- **Objectives determination and identify alternative solutions:** Requirements are gathered from the customers and the objectives are identified, elaborated and analyzed at the start of every phase. Then alternative solutions possible for the phase are proposed in this quadrant.
- **Identify and resolve Risks:** During the second quadrant all the possible solutions are evaluated to select the best possible solution. Then the risks associated with that solution is identified and the risks are resolved using the best possible strategy. At the end of this quadrant, Prototype is built for the best possible solution



- **Develop next version of the Product:** During the third quadrant, the identified features are developed and verified through testing. At the end of the third quadrant, the next version of the software is available.
- **Review and plan for the next Phase:** In the fourth quadrant, the Customers evaluate the so far developed version of the software. In the end, planning for the next phase is started

Advantages

- **Risk Handling:** The projects with many unknown risks that occur as the development proceeds, in that case, Spiral Model is the best development model to follow due to the risk analysis and risk handling at every phase.
- **Good for large projects:** It is recommended to use the Spiral Model in large and complex projects.
- **Flexibility in Requirements:** Change requests in the Requirements at later phase can be incorporated accurately by using this model.
- **Customer Satisfaction:** Customer can see the development of the product at the early phase of the software development and thus, they familiar with the system by using it before completion of the total product.

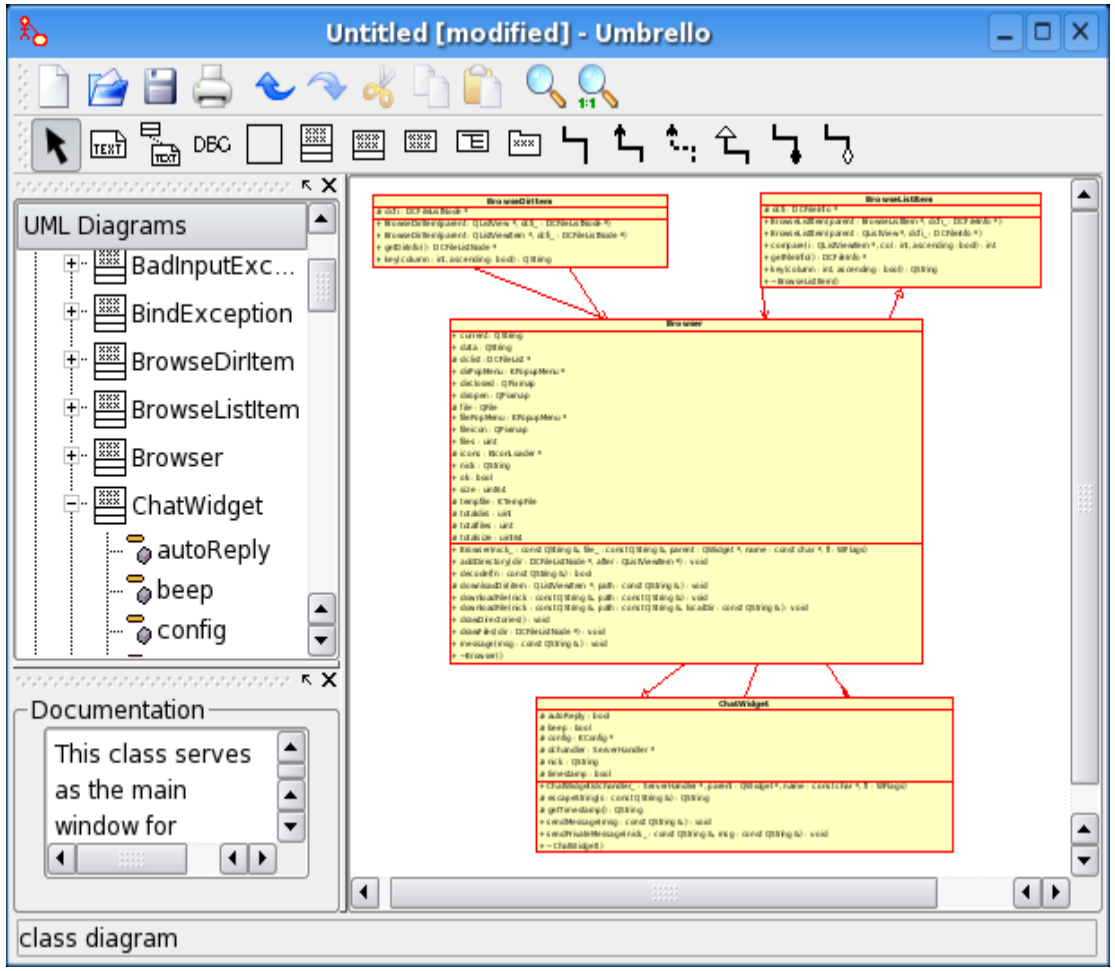
Disadvantages

- **Complex:** The Spiral Model is much more complex than other SDLC models.
- **Expensive:** Spiral Model is not suitable for small projects as it is expensive.
- **Too much dependable on Risk Analysis:** The successful completion of the project is very much dependent on Risk Analysis. Without very highly experienced expertise, it is going to be a failure to develop a project using this model.
- **Difficulty in time management:** As the number of phases is unknown at the start of the project, so time estimation is very difficult.

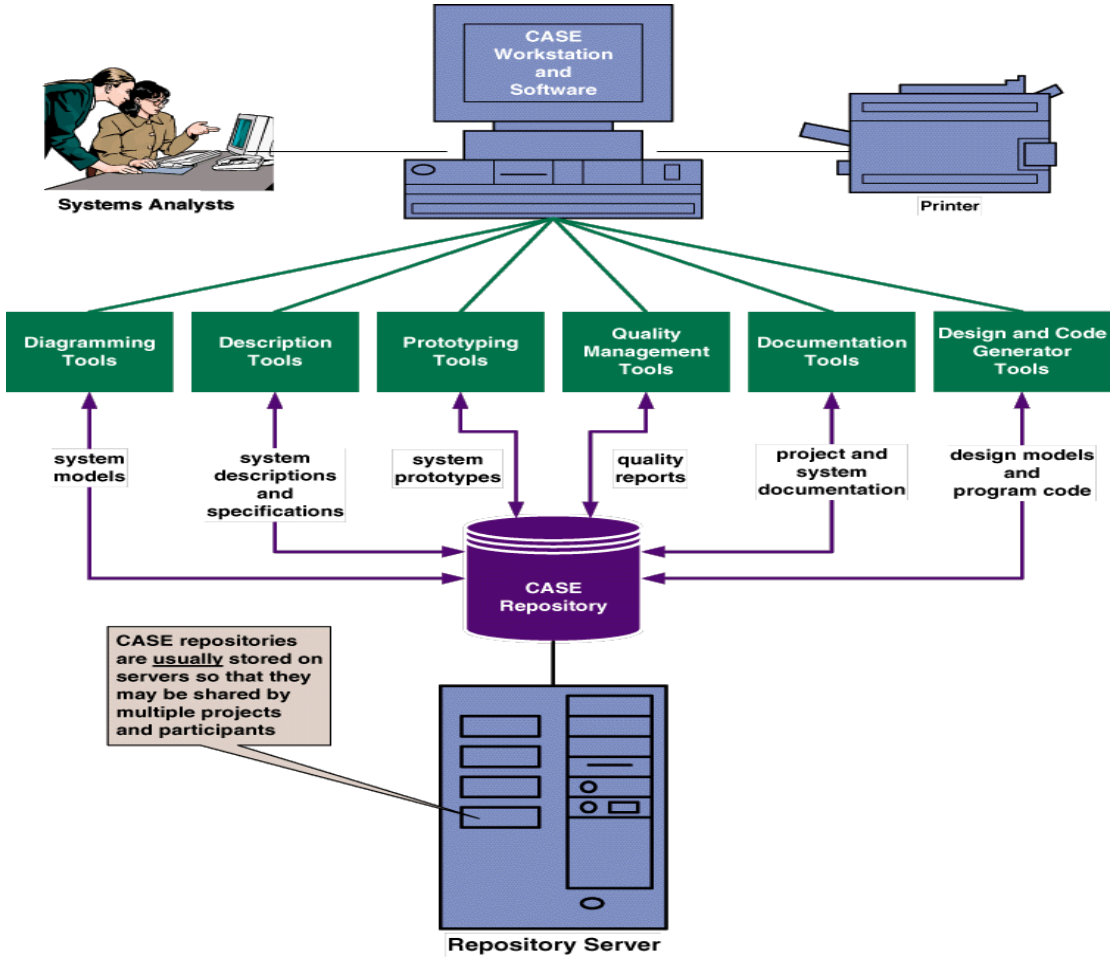
CASE (Computer-Aided Software Engineering) TOOLS

- **Computer-aided software engineering (CASE)** refers to automated software tools used by systems analysts to develop information systems.
- These tools can be used to automate or support activities throughout the systems development process with the objective of increasing productivity and improving the overall quality of systems
- *Computer-Aided Software Engineering* (CASE) tools are automated software packages that help to automate activities in the SDLC.
- CASE tools aim to enforce an engineering-type approach to the development of software systems.
- CASE tools range from simple diagramming tools to very sophisticated programs to document and automate most of the stages in the SDLC.
  - CASE tools used since the early 1990s

- **Computer-Aided Software Engineering (CASE) tools**
  - Diagramming tools enable graphical representation.
  - Computer displays and report generators help prototype how systems “look and feel”.
  - Analysis tools automatically check for consistency in diagrams, forms, and reports.
  - Central repository for integrated storage of diagrams, reports, and project management specifications.
  - Documentation generators standardize technical and user documentation.
  - Code generators enable automatic generation of programs and database code directly from design documents, diagrams, forms, and reports.



- The case for using CASE tools:
  - Improve quality of systems developed.
  - Help to increase the productivity of systems analysts.
  - Improve communication within the development team and with users.
  - Encourage an integrated approach to the SDLC.
  - Improve the management of the project.
  - Particularly helpful for systems maintenance



Types of CASE tools

There are three types of CASE tools:

- Upper CASE
  - Support analysis and design
- Lower CASE
  - Support programming and implementation
- Integrated CASE
  - Combines both upper and lower CASE

- Upper CASE tools
  - Create and modify the system design.
  - Information about the project stored in the CASE repository (reports, diagrams, files)
  - Support modelling of how the system fits into the organisation.
  - Analysis reports show incomplete parts and errors in the system design e.g. balance between process and data models.
- Lower CASE Tools
  - Generate source code and reduce need for systems programming.
  - Time for maintenance is reduced because test and debug are eliminated.
  - Once mastered, promote the re-use of existing documentation and components.

- Examples of CASE Tools:
  - Diagramming – for representing processes, data and control structures graphically. (analysis)
  - CASE repository – holds information required to create, modify and evolve the system. (analysis, design, implementation)
  - Form and report generators – automate generation of forms and reports to aid prototyping. (design, implementation, RAD, XP)
  - Code generators – automate generation of source code from diagrams and forms. (design, implementation)
  - Project management – aid in the planning, tracking, controlling and reporting of project management. (planning)
  - Document generator – create standard reports based upon the contents of the CASE repository. (analysis, design, implementation)
  - CASE analysis tools – help to identify problems of inconsistency, redundancy, and omissions. (more likely in analysis and design)

TABLE 1-2 Examples of CASE Usage within the SDLC

SDLC Phase	Key Activities	CASE Tool Usage
Project identification and selection	Display and structure high-level organizational information	Diagramming and matrix tools to create and structure information
Project initiation and planning	Develop project scope and feasibility	Repository and documentation generators to develop project plans
Analysis	Determine and structure system requirements	Diagramming to create process, logic, and data models
Logical and physical design	Create new system designs	Form and report generators to prototype designs; analysis and documentation generators to define specifications
Implementation	Translate designs into an information system	Code generators and analysis, form and report generators to develop system; documentation generators to develop system and user documentation
Maintenance	Evolve information system	All tools are used (repeat life cycle)

Advantages	Disadvantages
Produce systems with a longer effective operational life	Produce initial systems that are more expensive to build and maintain
Produce systems that more closely meet user needs and requirements	Require more extensive and accurate definition of user needs and requirements
Produce systems with excellent documentation	May be difficult to customize
Produce systems that need less systems support	Require more training of maintenance staff
Produce more flexible systems	May be difficult to use with existing systems

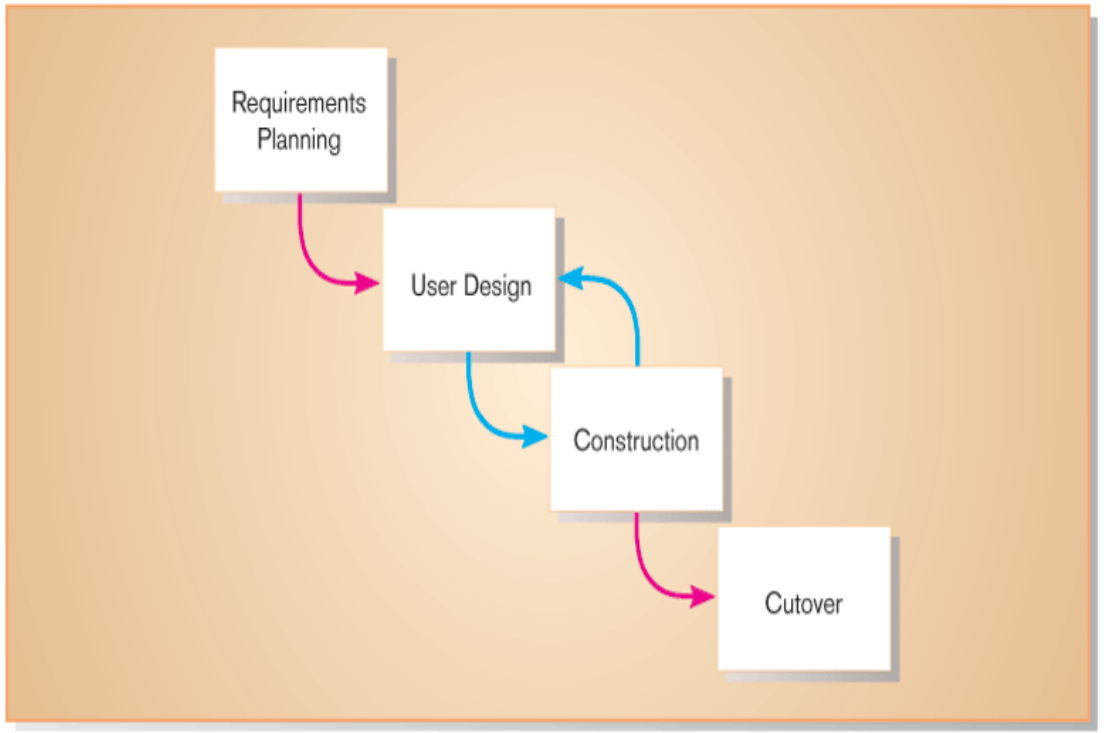
Rapid Application Development (RAD) Approach

- The **RAD (Rapid Application Development)** model is based on prototyping and iterative development with no specific planning involved. The process of writing the software itself involves the planning required for developing the product.
- Rapid Application Development focuses on gathering customer requirements through workshops or focus groups, early testing of the prototypes by the customer using iterative concept, reuse of the existing prototypes (components), continuous integration and rapid delivery.

Rapid Application Development (RAD) model has the following phases −

- **Requirements Planning phase** − In the requirements planning phase, a workshop needs to be conducted to discuss business problems in a structured manner.
- **User Description phase** − In the User Description phase, automated tools are used to capture information from users.
- **Construction phase** − In the Construction phase, productivity tools, such as code generators, screen generators, etc. are used inside a time-box, with a "Do until Done" approach.
- **Cut Over phase** − In the Cut over phase, installation of the system, user acceptance testing and user training are performed.

Figure 1-12 RAD life cycle



Different phases of RAD model includes

Phases of RAD model	Activities performed in RAD Model
Business Modeling	• On basis of the flow of information and distribution between various business channels, the product is designed
Data Modeling	• The information collected from business modeling is refined into a set of data objects that are significant for the business
Process Modeling	• The data object that is declared in the data modeling phase is transformed to achieve the information flow necessary to implement a business function
Application Generation	• Automated tools are used for the construction of the software, to convert process and data models into prototypes
Testing and Turnover	• As prototypes are individually tested during every iteration, the overall testing time is reduced in RAD.

When to use RAD Methodology?

- When a system needs to be produced in a short span of time (2-3 months)
- When the requirements are known
- When the user will be involved all through the life cycle
- When technical risk is less
- When there is a necessity to create a system that can be modularized in 2-3 months of time
- When a budget is high enough to afford designers for modeling along with the cost of automated tools for code generation

Advantages and Disadvantages of SDLC RAD Model

Advantages	Disadvantages
• Flexible and adaptable to changes	• It can't be used for smaller projects
• It is useful when you have to reduce the overall project risk	• Not all application is compatible with RAD
• It is adaptable and flexible to changes	• When technical risk is high, it is not suitable
• It is easier to transfer deliverables as scripts, high-level abstractions and intermediate codes are used	• If developers are not committed to delivering software on time, RAD projects can fail
• Due to code generators and code reuse, there is a reduction of manual coding	• Reduced features due to time boxing, where features are pushed to a later version to finish a release in short period
• Due to prototyping in nature, there is a possibility of lesser defects	• Reduced scalability occurs because a RAD developed application begins as a prototype and evolves into a finished application
• Each phase in RAD delivers highest priority functionality to client	• Progress and problems accustomed are hard to track as such there is no documentation to demonstrate what has been done
• With less people, productivity can be increased in short time	• Requires highly skilled designers or developers

Service Orientated Architecture

- It is modern and new concept of software development .
- It make individual SOA services are unassociated or loosely coupled to another. Each service executes one action.
- Each service can be used in other application within the organization or even in other organizations.
- We can say that service-oriented architecture is simply a group of services that can be called upon to provide specific functions.
- Rather than including calls to other services, a service can use certain defined protocols so that it can communicate with other services.

Advantages

- SOA allows reuse the service of an existing system alternately building the new system.
- It allows plugging in new services or upgrading existing services to place the new business requirements.
- It can enhance the performance, functionality of a service and easily makes the system upgrade.
- SOA has capability to adjust or modify the different external environments and large applications can be managed easily.
- The companies can develop applications without replacing the existing applications.
- It provides reliable applications in which you can test and debug the independent services easily as compared to large number of code.

Disadvantages

• SOA requires high investment cost (means large investment on technology, development and human resource).
• There is greater overhead when a service interacts with another service which increases the response time and machine load while validating the input parameters.
• SOA is not suitable for GUI (graphical user interface) applications which will become more complex when the SOA requires the heavy data exchange.

Agile Methodologies

- Agile methodologies, it argues that software development methodologies adapted from engineering generally do not fit with real world software development. In civil engineering requirements tends to be well understood, construction become very predictable.
- Motivated by recognition of software development as fluid, unpredictable, and dynamic.
- Three key principles
  - **Adaptive rather than predictive.**
  - **Emphasize people rather than roles.**
  - **Self-adaptive processes, as software is developed, the process used should be refined and improved of course after reviewed by people working on the project.**

  - **Agile is not for every project, it is for:**

    - unpredictable or dynamic requirement.
    - responsible and motivated developers.
    - customers who understand the process and will get involved.

- Agile Methodologies share iterative development (Martin, 1999). Iterative development focuses on the frequent production of working versions of a system that have a subset of the total number of required features. Iterative development provides feedback to customers and developers alike.
- Systems analyst or tester or manager, are not as important as the individuals who fill those roles. Fowler argues that the application of engineering principles to systems development has resulted in a view of people as interchangeable units instead of a view of people as talented individuals, each bringing something unique to the development team.
- The Agile Methodologies promote a self-adaptive software development process. As software is developed, the process used to develop it should be refined and improved. Development teams can do this through a review process, often associated with the completion of iterations. The implication is that, as processes are adapted, one would not expect to find a single monolithic methodology within a given corporation or enterprise. Instead, one would find many variations of the methodology, each of which reflects the particular talents and experience of the team using it.

Agile Manifesto

TABLE 1-3 The Agile Manifesto

The Manifesto for Agile Software Development

Seventeen anarchists agree:

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- *Individuals and interactions* over processes and tools.
- *Working software* over comprehensive documentation.
- *Customer collaboration* over contract negotiation.
- *Responding to change* over following a plan.

That is, while we value the items on the right, we value the items on the left more. We follow the following principles:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Businesspeople and developers work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Continuous attention to technical excellence and good design enhances agility.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Simplicity—the art of maximizing the amount of work not done—is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

—Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas (*www.agileAlliance.org*)

(*Source: http://agilemanifesto.org/* © 2001, the above authors. This declaration may be freely copied in any form, but only in its entirety through this notice.)

The Agile Methodologies group argues that software development methodologies adapted from engineering generally do not fit with real-world software development.

TABLE 1-4 Five Critical Factors That Distinguish Agile and Traditional Approaches to Systems Development

Factor	Agile Methods	Traditional Methods
Size	Well matched to small products and teams. Reliance on tacit knowledge limits scalability.	Methods evolved to handle large products and teams. Hard to tailor down to small projects.
Criticality	Untested on safety-critical products. Potential difficulties with simple design and lack of documentation.	Methods evolved to handle highly critical products. Hard to tailor down to products that are not critical.
Dynamism	Simple design and continuous refactoring are excellent for highly dynamic environments but a source of potentially expensive rework for highly stable environments.	Detailed plans and Big Design Up Front, excellent for highly stable environment but a source of expensive rework for highly dynamic environments.
Personnel	Requires continuous presence of a critical mass of scarce experts. Risky to use non-agile people.	Needs a critical mass of scarce experts during project definition but can work with fewer later in the project, unless the environment is highly dynamic.
Culture	Thrives in a culture where people feel comfortable and empowered by having many degrees of freedom (thriving on chaos).	Thrives in a culture where people feel comfortable and empowered by having their roles defined by clear practices and procedures (thriving on order).

(*Source:* Boehm, Barry; Turner, Richard, *Balancing Agility and Discipline: A Guide for the Perplexed,* 1st Ed., © 2004. Reprinted and electronically reproduced by permission of Pearson Education, Inc.)

eXtreme Programming (XP)

- eXtreme Programming is an approach to software development put together by Beck and Andres (2004). It is distinguished by

- its short cycles
- incremental planning approach
- focus on automated tests written by programmers and customers to monitor the development process.
- reliance on an evolutionary approach to development that lasts throughout the lifetime of the system.
- use of two-person programming teams.
- Planning, analysis, design, and construction are all fused into a single phase of activity.

- Under this approach, coding and testing are intimately related parts of the same process. The programmers who write the code also develop the tests. The emphasis is on testing those things that can break or go wrong, not on testing everything.
- Code is tested very soon after it is written. The overall philosophy behind eXtreme Programming is that the code will be integrated into the system it is being developed for and tested within a few hours after it has been written. If all the tests run successfully, then development proceeds. If not, the code is reworked until the tests are successful.
- Another part of eXtreme Programming that makes the code-and-test process work more smoothly is the practice of pair programming. All coding and testing is done by two people working together to write code and develop tests. Beck says that pair programming is not one person typing while the other one watches; rather, the two programmers work together on the problem they are trying to solve, exchanging information and insight and sharing skills.

- Compared to traditional coding practices, the advantages of pair programming include:

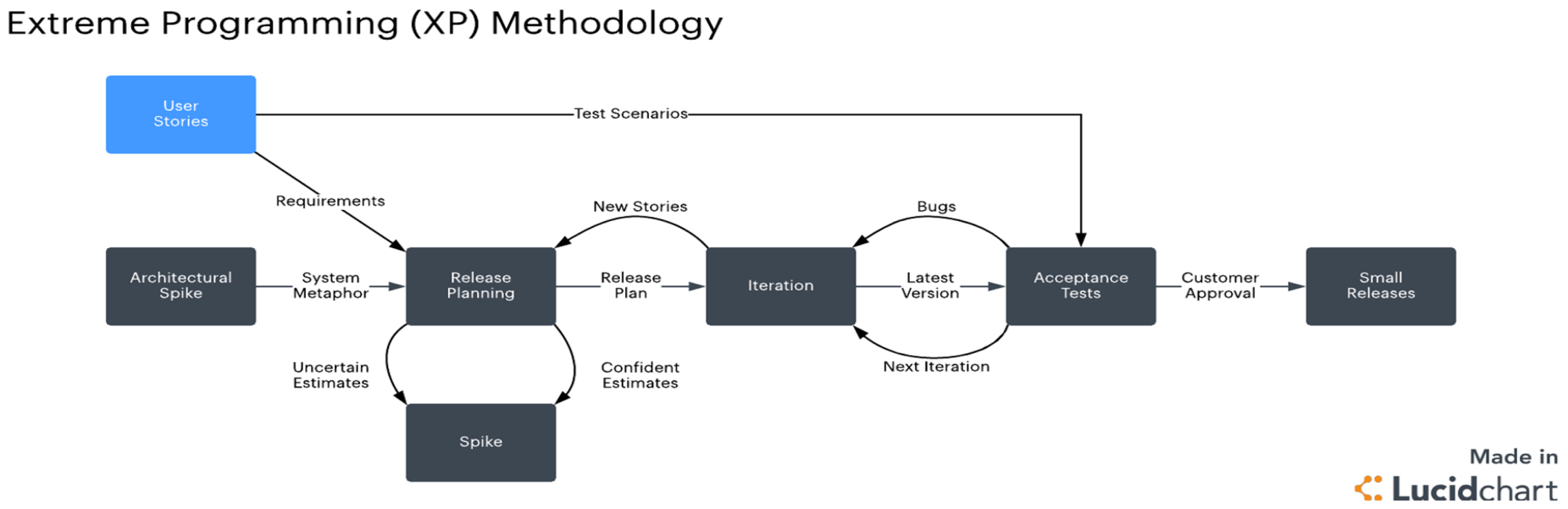
(1) more (and better) communication among developers,

(2) higher levels of productivity,

(3) higher-quality code, and (

(4) reinforcement of the other practices in eXtreme Programming, such as the code andtest discipline (Beck & Andres, 2004).

Although the eXtreme Programming process has its advantages, just as with any other approach to systems development, it is not for everyone and is not applicable to every project.



Why is it called “Extreme?”

- Extreme Programming takes the effective principles and practices to extreme levels.
- Code reviews are effective as the code is reviewed all the time.
- Testing is effective as there is continuous regression and testing.
- Design is effective as everybody needs to do refactoring daily.
- Integration testing is important as integrate and test several times a day.
- Short iterations are effective as the planning game for release planning and iteration planning.

Object- Oriented Analysis and Design

- Object-oriented analysis and design (OOAD) is a Systems development methodologies and techniques based on objects rather than data or processes.
- The object-oriented approach combines data and processes (called methods) into single entities called objects.
- Objects usually correspond to the real things an information system deals with, such as customers, suppliers, contracts, and rental agreements.
- The goal of OOAD is to make systems elements more reusable, thus improving system quality and the productivity of systems analysis and design.
- The object-oriented approach to systems development shares the iterative development approach of the Agile Methodologies.

- One of the most popular realizations of the iterative approach for object-oriented development is the Rational Unified Process (RUP), which is based on an iterative, incremental approach to systems development.
- RUP has four phases: inception, elaboration, construction, and transition

Phases of OOAD

- In the **inception phase**, analysts define the scope, determine the feasibility of the project, understand user requirements, and prepare a software development plan.
- In the **elaboration phase,** analysts detail user requirements and develop a baseline architecture. Analysis and design activities constitute the bulk of the elaboration phase.
- In the **construction phase**, the software is actually coded, tested, and documented.
- In the **transition phase**, the system is deployed, and the users are trained and supported. As is evident from Figure, the construction phase is generally the longest and the most resource intensive.
- The elaboration phase is also long, but less resource intensive.
- The transition phase is resource intensive but short.
- The inception phase is short and the least resource intensive.

1.2 The origins of software

Introduction, System Acquisition, Reuse

Introduction

- There are various sources of software for organizations.
- There are criteria to evaluate software from different sources.
- The impact of reuse on software development.
- If you wanted to write application software, you did it in-house, and you wrote the software from scratch. Today there are many different sources of software and firms that produce software, rather than in the information systems department of a corporation. But for those of you who do go on to work in a corporate information systems department, the focus is no longer exclusively on in-house development.

System Acquisition

- Internal corporate information systems departments now spend a smaller and smaller proportion of their time and effort on developing systems from scratch. Companies continue to spend relatively little time and money on traditional software development and maintenance. Instead, they invest in packaged software, open-source software, and outsourced services.
- Organizations today have many choices when seeking an information systems.

Outsourcing

- If one organization develops or runs a computer application for another organization, that practice is called outsourcing. Outsourcing includes a spectrum of working arrangements.
- At one extreme is having a firm develop and run your application on its computers—all you do is supply input and take output.
- A common example of such an arrangement is a company that runs payroll applications for clients so that clients do not have to develop an independent in-house payroll system. Instead, they simply provide employee payroll information to the company, and, for a fee, the company returns completed paychecks, payroll accounting reports, and tax and other statements for employees. For many organizations, payroll is a very cost-effective operation when outsourced in this way.

- Outsourcing is big business. Some organizations outsource the information technology (IT) development of many of their IT functions at a cost of billions of dollars. Most organizations outsource at least some aspect of their information systems activities.
- One example of the extent of outsourcing is Shell Oil. In 2008, Shell signed outsourcing contracts with EDS, T-Systems, and AT&T worth $3.2 billion USD.
- Biggest outsourcing companies are IBM and EDS.
- Some reasons for outsourcing include

➢ freeing up internal resources,

➢ increasing the revenue potential of the organization,

➢ reducing time to market,

➢ increasing process efficiencies, and

➢ outsourcing noncore activities.

Advantages:

- Reduced operating expenses
- Flexibility
- Exposure to ne talent
- Focusing on core business processes
- Downsize the business by using outsourcing services
- Managing resources
- Time-saving

Disadvantages:

- Lowered Quality of Service
- Miscommunication
- Risk

Sources of Software

- We can group the sources of software into six major categories:
- Information technology services firms,
- Packaged software producers,
- Enterprise-wide solutions,
- Cloud computing vendors,
- Open-source software,
- In-house developers

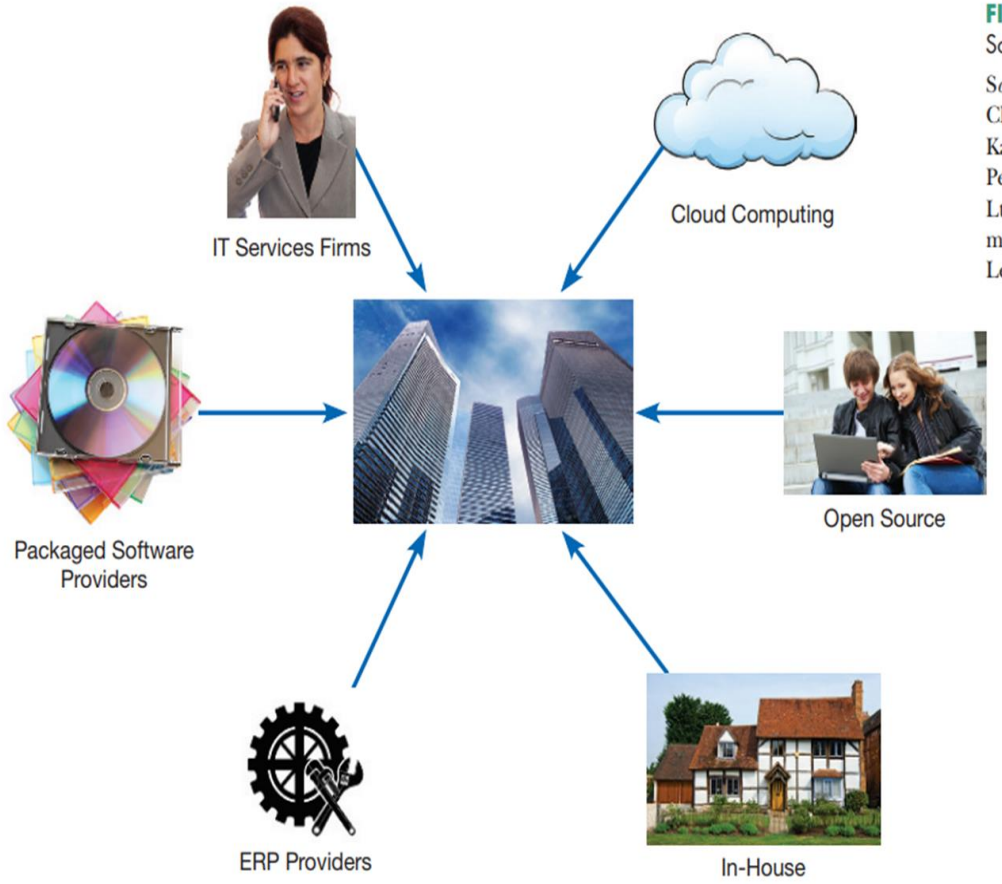


FIGURE 2-1
Sources of application software

Sources: Middle: Paulista/Fotolia, Clockwise starting with upper left: Kamira/Shutterstock; Amit John/ Pearson India Education Services Pvt. Ltd; Dmitry Kalinovsky/Shutterstock; mubus/Fotolia; grgroup/Fotolia; Le Do/Shutterstock

Information Technology Services Firms

- If a company needs an information system but does not have the expertise or the personnel to develop the system in-house the company will likely consult an information technology services firm.
- IT services firms help companies develop custom information systems for internal use, or they develop, host, and run applications for customers, or they provide other services.
- Many of the leading software companies in the world specialize in services, which include custom systems development. These firms employ people with expertise in the development of information systems. Their consultants may also have expertise in a given business area.
- For example, consultants who work with banks understand financial institutions as well as information systems. Consultants use many of the same methodologies, techniques, and tools that companies use to develop systems in-house.

Packaged Software Producers

- The growth of the software industry has been phenomenal since its beginnings in the mid-1960s.
- Some of the largest computer companies in the world are companies that produce software exclusively.
- A good example is **Microsoft**, probably the best-known software company in the world. Almost 87 percent of Microsoft's revenue comes from its software sales, mostly for its Windows operating systems and its personal productivity software, the Microsoft Office Suite.
- The packaged software development industry serves many market segments.
- Their software offerings range from general, broadbased packages, such as productivity tools, to very narrow, niche packages, such as software to help manage a day care center.
- Software companies develop software to run on many different computer platforms, from microcomputers to large mainframes.
- The companies range in size from just a few people to thousands of employees.

- Software companies develop what are sometimes called **prepackaged** or **off-the-shelf systems.** Microsoft's Word and Intuit's Quicken, QuickPay, and QuickBooks are popular examples of such software.
- Software companies develop software to run on many different computer platforms, from microcomputers to large mainframes.
- The companies range in size from just a few people to thousands of employees.
- Software companies consult with system users after the initial software design has been completed and an early version of the system has been built. The systems are then tested in actual organizations to determine whether there are any problems or if any improvements can be made. Until testing is completed, the system is not offered for sale to the public.

- Some off-the-shelf software systems cannot be modified to meet the specific, individual needs of a particular organization. Such application systems are sometimes called **turnkey systems.** The producer of a turnkey system will make changes to the software only when a substantial number of users ask for a specific change.
- However, other off-the-shelf application software can be modified or extended, by the producer or by the user, to more closely fit the needs of the organization. Even though many organizations perform similar functions, no two organizations do the same thing in quite the same way. A turnkey system may be good enough for a certain level of performance, but it will never perfectly match the way a given organization does business.

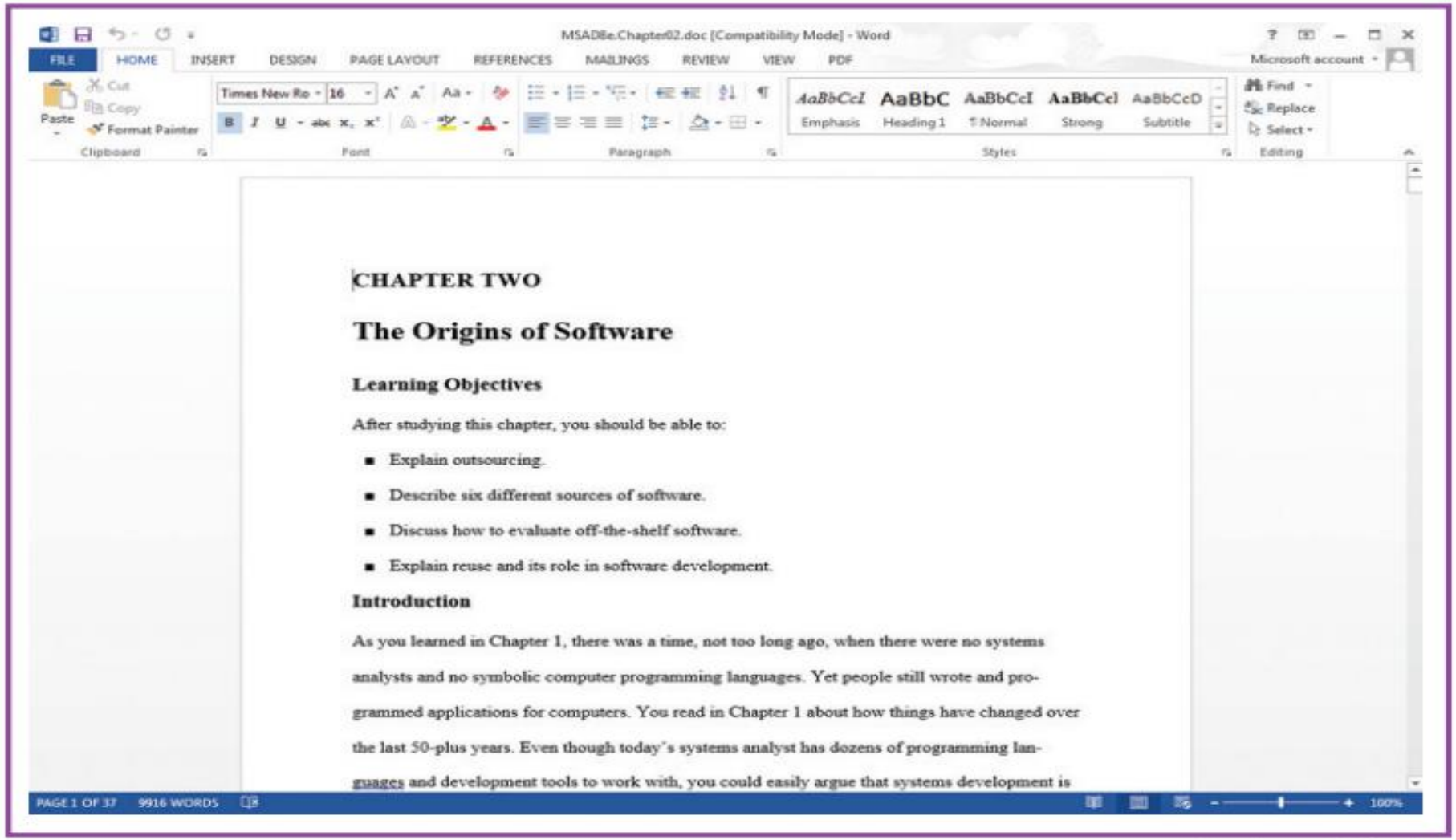


Figure: A document created in Microsoft's Word

Enterprise Solutions Software

- Many firms have chosen complete software solutions, called enterprise solutions or enterprise resource planning (ERP) systems, to support their operations and business processes.
- These ERP software solutions consist of a series of integrated modules.
- Each module supports an individual, traditional business function, such as accounting, distribution, manufacturing, or human resources.
- The traditional approach would use different systems in different functional areas of the business, such as a billing system in accounting and an inventory system in the warehouse.
- The difference between the modules and traditional approaches is that the modules are integrated to focus on business processes rather than on business functional areas.

- Using enterprise software solutions, a firm can integrate all parts of a business process in a unified information system.
- All aspects of a single transaction occur seamlessly within a single information system, rather than as a series of disjointed, separate systems focused on business functional areas.
- The benefits of the enterprise solutions approach include a single repository of data for all aspects of a business process and the flexibility of the modules.
- A single repository ensures more consistent and accurate data, as well as less maintenance.
- The modules are flexible because additional modules can be added as needed once the basic system is in place.

- Added modules are immediately integrated into the existing system.
- However, there are disadvantages to enterprise solutions software. The systems are very complex, so implementation can take a long time to complete.
- A system that integrates individual traditional business functions into a series of modules so that a single transaction occurs seamlessly within a single information system rather than several separate systems is called Enterprise Resource Planning (ERP) systems

Cloud Computing

- Another method for organizations to obtain applications is to rent them or license them from third-party providers who run the applications at remote sites. Users have access to the applications through the Internet or through virtual private networks.
- The application provider buys, installs, maintains, and upgrades the applications. Users pay on a per-use basis or they license the software, typically month to month.
- Although this practice has been known by many different names over the years, today it is called cloud computing.
- Cloud computing refers to the provision of applications over the Internet, where customers do not have to invest in the hardware and software resources needed to run and maintain the applications.

- A well-known example of cloud computing is **Google Apps**, where users can share and create documents, spreadsheets, and presentations.
- Another well-known example is **Salesforce.com,** which provides customer relationship management software online.
- Cloud computing encompasses many areas of technology, including software as a service (often referred to as SaaS), which includes Salesforce.com, and hardware as a service, which includes Amazon Web Services and allows companies to order server capacity and storage on demand.
- Taking the cloud computing route has its advantages. The top three reasons for choosing to go with cloud computing, all of which result in benefits for the company, are

(1) freeing internal IT staff,

(2) gaining access to applications faster than via internal development,

(3) achieving lower cost access to corporate-quality applications.

- IT managers do have some concerns about cloud computing, however. The primary concern is over **security.** Concerns over security are based on storing company data on machines one does not own and that others can access.
- In fact, the top two reasons for not using cloud services are concerns about unauthorized access to proprietary information and unauthorized access to customer information.
- Another concern is **reliability.** Some warn that the cloud is actually a network of networks, it is vulnerable to unexpected risks due to its complexity.

Open Source Software

- Open-source software is unlike the other types of software you have read about so far. Open-source software is different because it is freely available, not just the final product but the source code itself.
- It is also different because it is developed by a community of interested people instead of by employees of a particular company.
- Open-source software performs the same functions as commercial software, such as operating systems, e-mail, database systems, web browsers, and so on. Some of the most well-known and popular open-source software names are **Linux,** an operating system; **mySQL**, a database system; and **Firefox,** a web browser.
- Open source also applies to software components and objects.
- Open source is developed and maintained by communities of people, and sometimes these communities can be very large.

- If the software is free, you might wonder how anybody makes any money by developing open-source software. Companies and individuals can make money with open source in two primary ways:

(1) by providing maintenance and other services or

(2) by providing one version of the software free and selling a more fully featured version.

- Some open-source solutions have more of an impact on the software industry than others. Linux, for example, has been very successful in the server software market, where it is estimated to have as much as 36 percent of the market share (W3Techs, 2015). In the desktop operating systems market, Linux has about 1 percent market share. Other open-source software products, such as mySQL, have also been successful, and open source's share of the software industry seems destined to continue to grow.

In-House Development

- In-house development has become a progressively smaller piece of all systems development work that takes place in and for organizations. Internal corporate information systems departments now spend a smaller and smaller proportion of their time and effort on developing systems from scratch.
- In-house development can lead to a larger maintenance burden than other development methods, such as packaged applications.
- A study by Banker, Davis, and Slaughter found that using a code generator as the basis for in-house development was related to an increase in maintenance hours, whereas using packaged applications was associated with a decrease in maintenance effort.
- Of course, in-house development need not entail (to make something necessary) development of all of the software that will constitute the total system. Hybrid solutions involving some purchased and some in-house software components are common. If you choose to acquire software from outside sources, this choice is made at the end of the analysis phase. . The choice between a package and an external supplier will be determined by your needs, not by what the supplier has to sell.

TABLE 2-1 Leading Software Firms and Their Development Specializations

Specialization	Example Firms or Websites
IT Services	Accenture Computer Sciences Corporation (CSC) IBM HP
Packaged Software Providers	Intuit Microsoft Oracle SAP AG Symantec
Enterprise Software Solutions	Oracle SAP AG
Cloud Computing	Amazon.com Google IBM Microsoft Salesforce.com
Open Source	SourceForge.net

TABLE 2-2 Comparison of Six Different Sources of Software Components

Producers	When to Go to This Type of Organization for Software	Internal Staffing Requirements
IT services firms	When task requires custom support and system can't be built internally or system needs to be sourced	Internal staff may be needed, depending on application
Packaged software producers	When supported task is generic	Some IS and user staff to define requirements and evaluate packages
Enterprise-wide solutions vendors	For complete systems that cross functional boundaries	Some internal staff necessary but mostly need consultants
Cloud computing	For instant access to an application; when supported task is generic	Few; frees up staff for other IT work
Open-source software	When supported task is generic but cost is an issue	Some IS and user staff to define requirements and evaluate packages
In-house developers	When resources and staff are available and system must be built from scratch	Internal staff necessary though staff size may vary

- For each criterion, an explicit comparison should be made between the software package and the process of developing the same application in-house.
- The most common criteria include the following:

➢ Cost

➢ Functionality

➢ Vendor support

➢ Viability of vendor

➢ Flexibility

➢ Documentation

➢ Response time

➢ Ease of installation

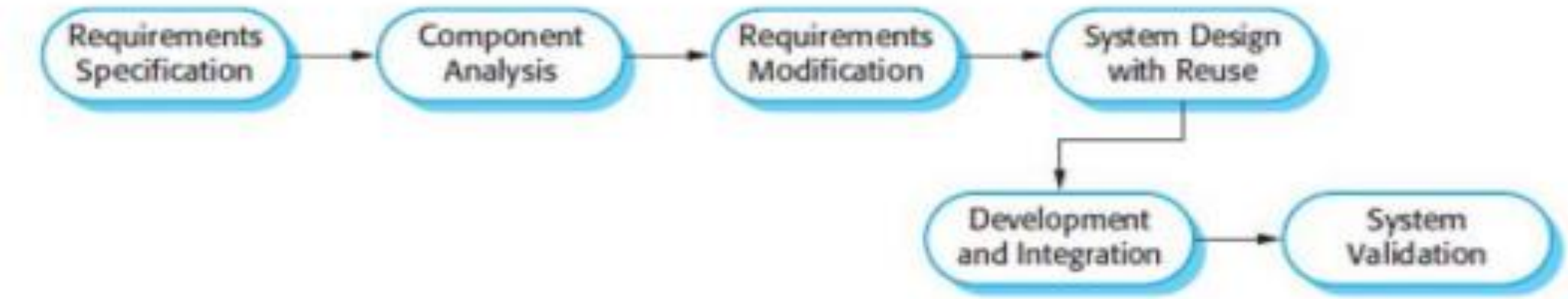
Reuse

- Reuse is the use of previously written software resources in new applications.
- Because so many bits and pieces of applications are relatively generic across applications, it seems intuitive that great savings can be achieved in many areas if those generic bits and pieces do not have to be written a new each time they are needed.
- Reuse should increase programmer productivity because being able to use existing software for some functions means they can perform more work in the same amount of time.
- Reuse should also decrease development time, minimizing schedule overruns.
- Because existing pieces of software have already been tested, reusing them should also result in higher-quality software with lower defect rates, decreasing maintenance costs.
- Some of the components that can be reuse are Source code, design and interfaces, user manuals, software documentation, software requirement specifications and many more.

Advantages:

- Less effort,
- Time-saving,
- Reduce cost,
- Increase software productivity,
- Utilize fewer resources,
- Leads to a better quality software.

Stages of Re-use Oriented Software Engineering



- **Requirement Specification:** First of all specify the requirements. This will help to decide that we have some existing software components for the development of software or not.
- **Component Analysis:** Helps to decide that which component can be reused where.
- **Requirement Modification:** If the requirements are changed by the customer, then still existing components are helpful for use or not.
- **Development:** Existing components are matching with new software or not.
- **Integration:** can we integrate the new systems with existing components?
- **System Validation:** To validate the system that it can be accepted by the customer or not.

1.3 Managing the Information Systems Project

Introduction, Managing Information Systems Project, Representing and Scheduling Project, Using Project Management Software Plans

Introduction

- We will focus on the systems analyst's role as **project manager** of an information systems project. Throughout the SDLC, the project manager is responsible for initiating, planning, executing, and closing down the systems development project.
- Project management is arguably the most important aspect of an information systems development project. Effective project management helps to ensure that systems development projects meet customer expectations and are delivered within budget and time constraints.
- We will then be provided with an understanding of the project manager's role and the project management process.
- The discussion then turns to techniques for reporting project plans using Gantt charts and network diagrams.
- The project will conclude with a discussion of the use of commercially available project management software that can be used to assist with a wide variety of project management activities

- Project is a planned undertaking of a series of related activities to reach an objective that has a beginning and an end.
- **Objective of a project**

– Solve a business problem (develop a MIS)

– Take advantage of a business opportunities (develop BIS)

– Other non rational reason: spend existing available resources, training and enhancing skills of employees

- **Where do projects come from?**

– There is no standard and answer varies from organization to organization

- Several projects may be submitted and need selection by filling a "**Systems Service Requests"**

- Project management (PM) may be the most important aspect of systems development.
- **Effective PM helps to ensure**
    - The meeting of customer expectations.
    - The satisfying of budget and time constraints.
- The nature of projects has changed from custom development to implementing packaged software and data warehousing.
- PM skills are difficult and important to learn.

Project Management:

- Activities include planning work, assessing risk, estimating resources required, organizing the work, assigning tasks, controlling project execution, reporting progress, analyzing results
- Five major variables

❖Scope

❖Time

❖Cost

❖Quality

❖Risk

MANAGING THE INFORMATION SYSTEMS PROJECT

- **Project**

A planned undertaking of related activities to reach an objective that has a beginning and an end

- Project management is an important aspect of the development of information systems and a critical skill for a systems analyst.
- The focus of project management is to ensure that systems development projects meet customer expectations and are delivered within budget and time constraints.
- The **project manager** is a systems analyst with a diverse set of skills—management, leadership, technical, conflict management, and customer relationship—who is responsible for initiating, planning, executing, and closing down a project.

- As a project manager, your environment is one of continual change and problem solving.
- In some organizations, the project manager is a very experienced systems analyst, whereas in others, both junior and senior analysts are expected to take on this role, managing parts of a project or actively supporting a more senior colleague who assumes the project manager role.
- Understanding the project management process is a critical skill for your future success.
- Creating and implementing successful projects requires managing the resources, activities, and tasks needed to complete the information systems project.
- Project management is a controlled process of initiating, planning, executing, and closing down a project.

Deciding on Systems Projects

- **System Service Request (SSR)**
  - A standard form for requesting or proposing systems development work within an organization
- **Feasibility study**
  - A study that determines whether a requested system makes economic and operational sense for an organization

Pine Valley Furniture
System Service Request

REQUESTED BY Juanita Lopez DATE October 1, 2014

DEPARTMENT Purchasing, Manufacturing Support

LOCATION Headquarters, 1-322

CONTACT Tel: 4-3267 FAX: 4-3270 e-mail: jlopez

TYPE OF REQUEST

[X] New System
[ ] System Enhancement
[ ] System Error Correction

URGENCY

[ ] Immediate – Operations are impaired or opportunity lost
[ ] Problems exist, but can be worked around
[X] Business losses can be tolerated until new system installed

PROBLEM STATEMENT

Sales growth at PVF has caused greater volume of work for the manufacturing support unit within Purchasing. Furthe r, more concentration on customer service has reduced manufacturing lead times, which puts more pressure on purchasing activities. In addition, cost-cutting measures force Purchasing to be more aggressive in negotiating terms with vendors, improving delivery times, and lowering our investments in inventory. The current modest systems support for Manufacturing/Purchasing is not responsive to these new business conditions. Data are not available, information cannot be summarized, supplier orders cannot be adequately tracked, and commodity buying is not well supported. PVF is spending too much on raw materials and not being responsive to manufacturing needs.

SERVICE REQUEST

I request a thorough analysis of our current operations with the intent to design and build a completely new information system. This system should handle all purchasing transactions, support display and reporting of critical purchasing data, and assist purchasing agents in commodity buying.

IS LIAISON Chris Martin (Tel: 4-6204 FAX: 4-6200 e-mail: cmartin)

SPONSOR Sal Divario, Director, Purchasing

TO BE COMPLETED BY SYSTEMS PRIORITY BOARD

[ ] Request approved Assigned to
Start date
[ ] Recommend revision
[ ] Suggest user development
[ ] Reject for reason

FIGURE:
System Service Request for Purchasing Fulfillment System with name and contact information of the person requesting the system, a statement of the problem, and the name and contact information of the liaison and sponsor

Project Management Activities

Skills of Project Manager

TABLE 3-1 Common Activities and Skills of a Project Manager

Activity	Description	Skill
Leadership	Influencing the activities of others toward the attainment of a common goal through the use of intelligence, personality, and abilities	Communication; liaison between management, users, and developers; assigning activities; monitoring progress
Management	Getting projects completed through the effective utilization of resources	Defining and sequencing activities; communicating expectations; assigning resources to activities; monitoring outcomes
Customer relations	Working closely with customers to ensure that project deliverables meet expectations	Interpreting system requests and specifications; site preparation and user training; contact point for customers
Technical problem solving	Designing and sequencing activities to attain project goals	Interpreting system requests and specifications; defining activities and their sequence; making trade-offs between alternative solutions; designing solutions to problems
Conflict management	Managing conflict within a project team to assure that conflict is not too high or too low	Problem solving; smoothing out personality differences; compromising; goal setting
Team management	Managing the project team for effective team performance	Communication within and between teams; peer evaluations; conflict resolution; team building; self-management
Risk and change management	Identifying, assessing, and managing the risks and day-to-day changes that occur during a project	Environmental scanning; risk and opportunity identification and assessment; forecasting; resource redeployment

Phases of Project Management Process

- Phase 1: Initiation
- Phase 2: Planning
- Phase 3: Execution
- Phase 4: Closedown

Phase 1: Initiation

- The first phase of the project management process in which activities are performed to assess the size, scope, and complexity of the project and to establish procedures to support later project activities.
- During project initiation, the project manager performs several activities to assess the size, scope, and complexity of the project and to establish procedures to support subsequent activities.
- Depending on the project, some initiation activities may be unnecessary and some may be very involved.
- The types of activities you will perform when initiating a project are summarized as follows:

Activities During Project Initiation

1. **Establishing the project initiation team:** This activity involves organizing project team members to assist in accomplishing the project initiation activities.
2. **Establishing a relationship with the customer:** A thorough understanding of your customer builds stronger partnerships and higher levels of trust.
3. **Establishing the project initiation plan:** This step defines the activities required to organize the initiation team while it is working to define the goals and scope of the project.
4. **Establishing management procedures:** Successful projects require the development of effective management procedures.
5. **Establishing the project management environment and project workbook**: The focus of this activity is to collect and organize the tools that you will use while managing the project and to construct the project workbook. Diagrams, charts, and system descriptions provide much of the project workbook contents. Thus, the project workbook serves as a repository for all project correspondence, inputs, outputs, deliverables, procedures, and standards established by the project team.
6. **Developing the project charter**. The project charter is a short (typically one page), high-level document prepared for the customer that describes what the project will deliver and outlines many of the key elements of the project.

Phase 2: Planning

- The second phase of the project management process that focuses on defining clear, discrete activities and the work needed to complete each activity within a single project.
- The next step in the project management process is project planning. Research has found a positive relationship between effective project planning and positive project outcomes.
- Project planning involves defining clear, discrete activities and the work needed to complete each activity within a single project.
- It often requires you to make numerous assumptions about the availability of resources such as hardware, software, and personnel.
- In actual fact, you often have to construct longer-term plans that are more general in scope and nearer term plans that are more detailed. The repetitive nature of the project management process requires that plans be constantly monitored throughout the project and periodically updated (usually after each phase), based upon the most recent information.

Activities During Project Planning

1. Describing Project Scope, Alternatives, and Feasibility
2. Dividing the Project into Manageable Tasks
3. Estimating Resources and Creating a Resource Plan
4. Developing a Preliminary Schedule
5. Developing a Communication Plan
6. Determining Project Standards and Procedures
7. Identifying and Assessing Risk
8. Creating a Preliminary Budget
9. Developing a Project Scope Statement
10. Setting a Baseline Project Plan

1. Describing project scope, alternatives, and feasibility

- The purpose of this activity is to understand the content and complexity of the project. Within systems development methodology, one of the first meetings must focus on defining a project's scope.
- Although project scope information was not included in the SSR developed by Chris and Juanita, it was important that both shared the same vision for the project before moving too far along.
- During this activity, you should reach agreement on the following questions:
    - What problem or opportunity does the project address?
    - What are the quantifiable results to be achieved?
    - What needs to be done?
    - How will success be measured?
    - How will we know when we are finished?
- After defining the scope of the project, your next objective is to identify and document general alternative solutions for the current business problem or opportunity. You must then assess (to judge or decide value or importance) the feasibility of each alternative solution and choose which to consider during subsequent SDLC phases.

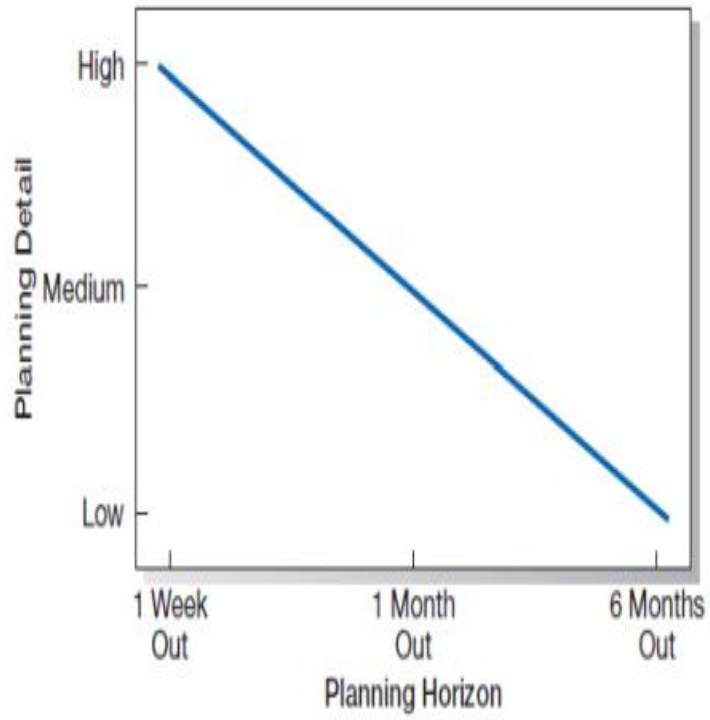


FIGURE 3-8
Level of project planning detail should be high in the short term, with less detail as time goes on

At the start of project, there is no idea of doing work, every thing is unknown so detailed plan is needed to do every step. But after some time ,that is week or month, people gets idea of project scheduling and can do some steps on the basis of previous step or his/her own creativity.

2. Dividing the project into manageable tasks:

- This is a critical activity during the project planning process. Here, you must divide the entire project into manageable tasks and then logically order them to ensure a smooth evolution between tasks.
- **Work Breakdown Structure (WBS)**

  - Division of project into manageable and logically ordered tasks and subtasks

- **Scheduling Diagrams**

  - **Gantt chart**: horizontal bars represent task durations

  - **Network diagram:** boxes and links represent task dependencies

Developing a Preliminary Schedule

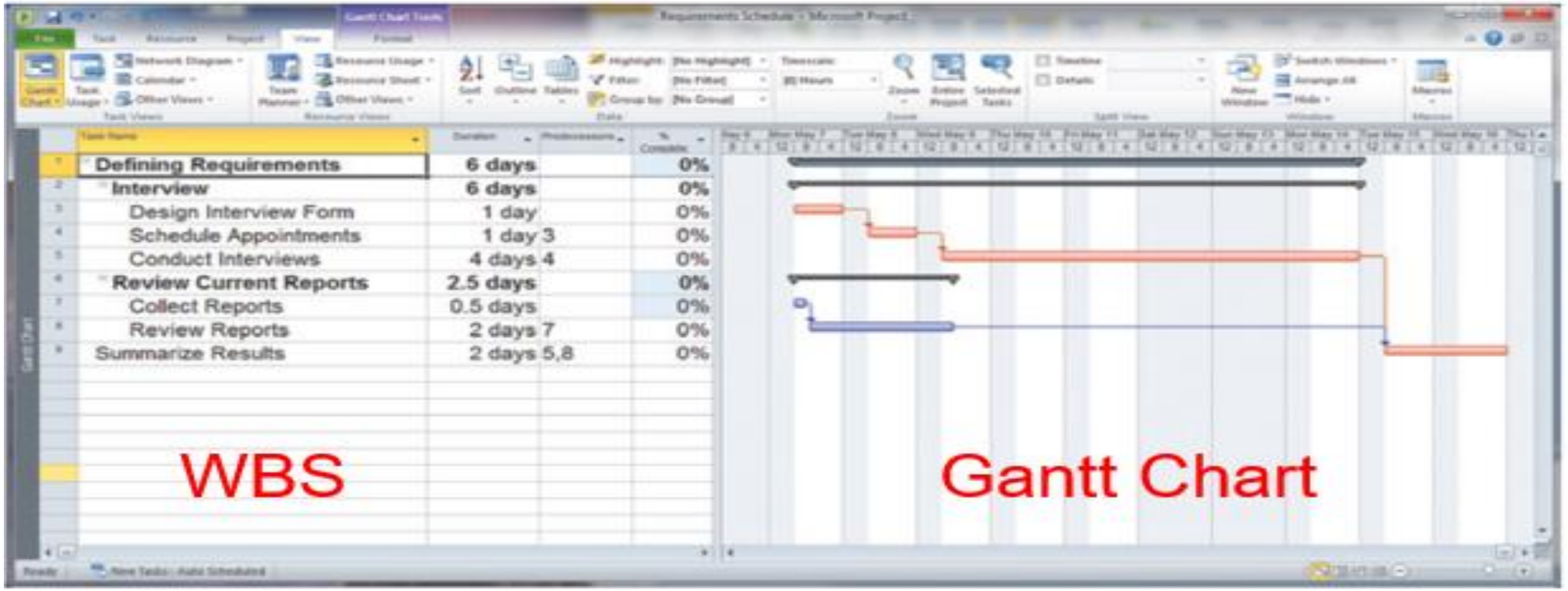


FIGURE 3-10
Gantt chart showing project tasks, duration times for those tasks, and predecessors

3. Estimating resources and creating a resource plan

- The goal of this activity is to estimate resource requirements for each project activity and to use this information to create a project resource plan. The resource plan helps assemble and deploy resources in the most effective manner.
- For example, you would not want to bring additional programmers onto the project at a rate faster than you could prepare work for them. Project managers use a variety of tools to assist in making estimates of project size and costs.
- The most widely used method is called COCOMO (constructive cost model), COCOMO predict human resource requirements for basic, intermediate, and very complex systems.

4. Developing a preliminary schedule

- During this activity, you use the information on tasks and resource availability to assign time estimates to each activity in the work breakdown structure. These time estimates will enable you to create target starting and ending dates for the project.
- Target dates can be revisited and modified until a schedule is produced that is acceptable to the customer. Determining an acceptable schedule may require that you find additional or different resources or that the scope of the project be changed. The schedule may be represented as a Gantt chart or as a network diagram.

Scheduling Diagrams Network Diagram

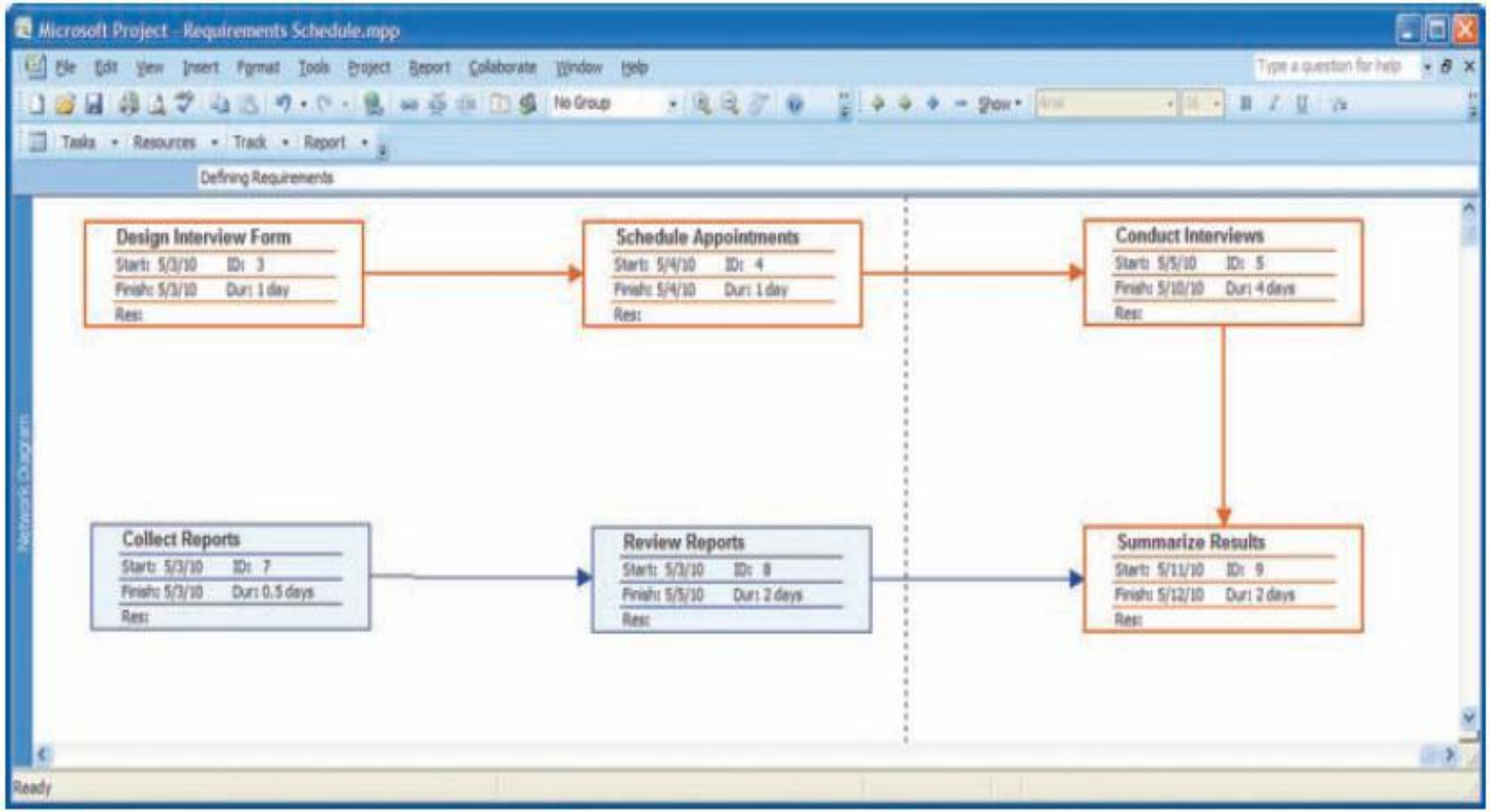


FIGURE:

- A network diagram illustrating tasks with rectangles (or ovals) and the relationships and sequences of those activities with arrows (*Source:* Microsoft Corporation.)

5. Developing a communication plan

- The goal of this activity is to outline the communication procedures among management, project team members, and the customer. The communication plan includes when and how written and oral reports will be provided by the team, how team members will coordinate work, what messages will be sent to announce the project to interested parties, and what kinds of information will be shared with vendors and external contractors involved with the project.
- It is important that free and open communication occur among all parties with respect to proprietary information and confidentiality with the customer
- When developing a communication plan, numerous questions must be answered in order to assure that the plan is comprehensive and complete, including the following:

- Who are the stakeholders for this project?
- What information does each stakeholder need?
- When, and at what interval, does this information need to be produced?
- What sources will be used to gather and generate this information?
- Who will collect, store, and verify the accuracy of this information?
- Who will organize and package this information into a document?
- Who will be the contact person for each stakeholder should any questions arise?
- What format will be used to package this information?
- What communication medium will be most effective for delivering this information to the stakeholder?
- Once these questions are answered for each stakeholder, a comprehensive communication plan can be developed. In this plan, a summary of communication documents, work assignments, schedules, and distribution methods will be outlined.

6. Determining project standards and procedures

- During this activity, you will specify how various deliverables are produced and tested by you and your project team. For example, the team must decide which tools to use, how the standard SDLC might be modified, which SDLC methods will be used, documentation styles (e.g., type fonts and margins for user manuals), how team members will report the status of their assigned activities, and terminology.
- Setting project standards and procedures for work acceptance is a way to ensure the development of a high-quality system. Also, it is much easier to train new team members when clear standards are in place.
- Organizational standards for project management and conduct make the determination of individual project standards easier and the interchange or sharing of personnel among different projects feasible.

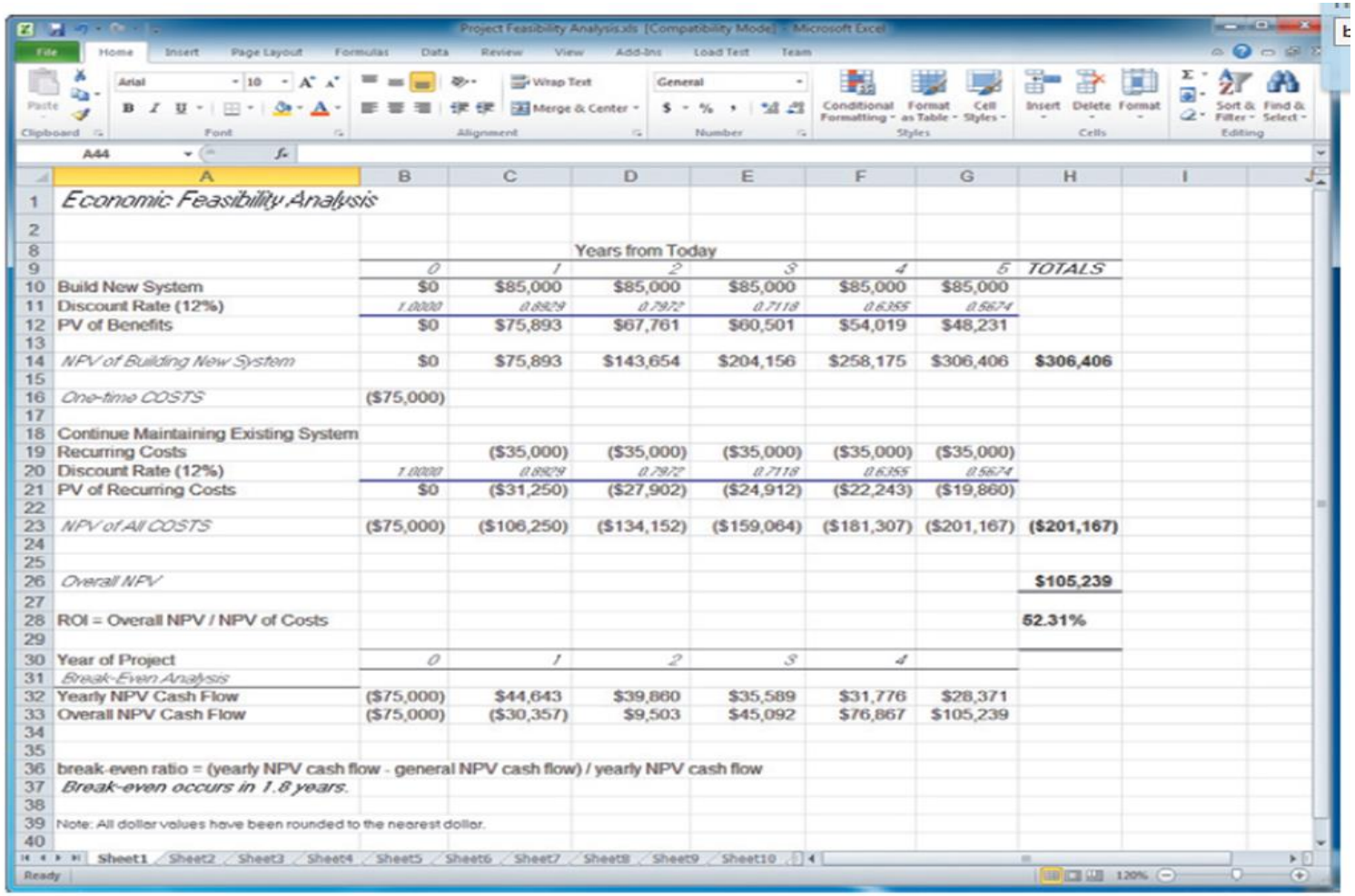
7. Identifying and assessing risk

- The goal of this activity is to identify sources of project risk and estimate the consequences of those risks. Risks might arise from the use of new technology, prospective users' resistance to change, availability of critical resources, competitive reactions or changes in regulatory actions due to the construction of a system, or team member inexperience with technology or the business area. You should continually try to identify and assess project risk.

8. Creating a preliminary budget

- During this phase, you need to create a preliminary budget that outlines the planned expenses and revenues associated with your project. The project justification will demonstrate that the benefits are worth these costs.

Developing a Preliminary Budget



	A	B	C	D	E	F	G	H
1	*Economic Feasibility Analysis*							
2								
8				Years from Today				
9		*0*	*1*	*2*	*3*	*4*	*5*	*TOTALS*
10	Build New System	$0	$85,000	$85,000	$85,000	$85,000	$85,000	
11	Discount Rate (12%)	*1.0000*	*0.8929*	*0.7972*	*0.7118*	*0.6355*	*0.5674*	
12	PV of Benefits	$0	$75,893	$67,761	$60,501	$54,019	$48,231	
13								
14	*NPV of Building New System*	$0	$75,893	$143,654	$204,156	$258,175	$306,406	**$306,406**
15								
16	*One-time COSTS*	($75,000)						
17								
18	Continue Maintaining Existing System							
19	Recurring Costs		($35,000)	($35,000)	($35,000)	($35,000)	($35,000)	
20	Discount Rate (12%)	*1.0000*	*0.8929*	*0.7972*	*0.7118*	*0.6355*	*0.5674*	
21	PV of Recurring Costs	$0	($31,250)	($27,902)	($24,912)	($22,243)	($19,860)	
22								
23	*NPV of All COSTS*	($75,000)	($106,250)	($134,152)	($159,064)	($181,307)	($201,167)	**($201,167)**
24								
25								
26	*Overall NPV*							**$105,239**
27								
28	ROI = Overall NPV / NPV of Costs							**52.31%**
29								
30	Year of Project	*0*	*1*	*2*	*3*	*4*		
31	*Break-Even Analysis*							
32	Yearly NPV Cash Flow	($75,000)	$44,643	$39,860	$35,589	$31,776	$28,371	
33	Overall NPV Cash Flow	($75,000)	($30,357)	$9,503	$45,092	$76,867	$105,239	
34								
35								
36	break-even ratio = (yearly NPV cash flow - general NPV cash flow) / yearly NPV cash flow							
37	*Break-even occurs in 1.8 years.*							
38								
39	Note: All dollar values have been rounded to the nearest dollar.							
40								

FIGURE : A financial cost and benefit analysis for a systems development project (*Source:* Microsoft Corporation.)

Spreadsheet software is good for this.

9. Developing a Project Scope Statement

- An important activity that occurs near the end of the project planning phase is the development of the Project Scope Statement. Developed primarily for the customer, this document outlines work that will be done and clearly describes what the project will deliver. The Project Scope Statement is useful to make sure that you, the customer, and other project team members have a clear understanding of the intended project size, duration, and outcomes.

10. Setting a Baseline Project Plan

- Once all of the prior project planning activities have been completed, you will be able to develop a Baseline Project Plan. This baseline plan provides an estimate of the project's tasks and resource requirements and is used to guide the next project phase—execution. As new information is acquired during project execution, the baseline plan will continue to be updated.

Phase 3: Execution

- The third phase of the project management process in which the plans created in the prior phases (project initiation and planning) are put into action.
- Project execution puts the Baseline Project Plan into action.
- Within the context of the SDLC, project execution occurs primarily during the analysis, design, and implementation phases.
- **Execution Phases:**
    1. Execute baseline project plan.
    2. Monitor progress against baseline plan.
    3. Manage changes in baseline plan.
    4. Maintain project workbook.
    5. Communicate project status.

1. Executing the Baseline Project Plan:

- This means that you initiate the execution of project activities, acquire and assign resources, orient and train new team members, keep the project on schedule, and ensure the quality of project deliverables. This is a formidable task, but a task made much easier through the use of sound project management techniques.

2. Monitoring project progress against the Baseline Project Plan :

- While you execute the Baseline Project Plan, you should monitor your progress. If the project gets ahead of (or behind) schedule, you may have to adjust resources, activities, and budgets. Monitoring project activities can result in modifications to the current plan. Measuring the time and effort expended on each activity will help you improve the accuracy of estimations for future projects. It is possible, with project schedule charts such as Gantt charts, to show progress against a plan, and it is easy with network diagrams to understand the ramifications (the possible results of an action)of delays in an activity.

- Monitoring progress also means that the team leader must evaluate and appraise (to examine someone or something in order to judge their qualities, success or needs)each team member, occasionally change work assignments or request changes in personnel, and provide feedback to the employee's supervisor.

3. Managing changes to the Baseline Project Plan

- You will encounter pressure to make changes to the baseline plan. Numerous events may initiate a change to the Baseline Project Plan, including the following possibilities:

➢ A slipped completion date for an activity

➢ A bungled (to do something wrong, in a careless or stupid way) activity that must be redone

➢ The identification of a new activity that becomes evident later in the project

➢ An unforeseen change in personnel due to sickness, resignation, or termination

Monitoring Progress with a Gantt Chart

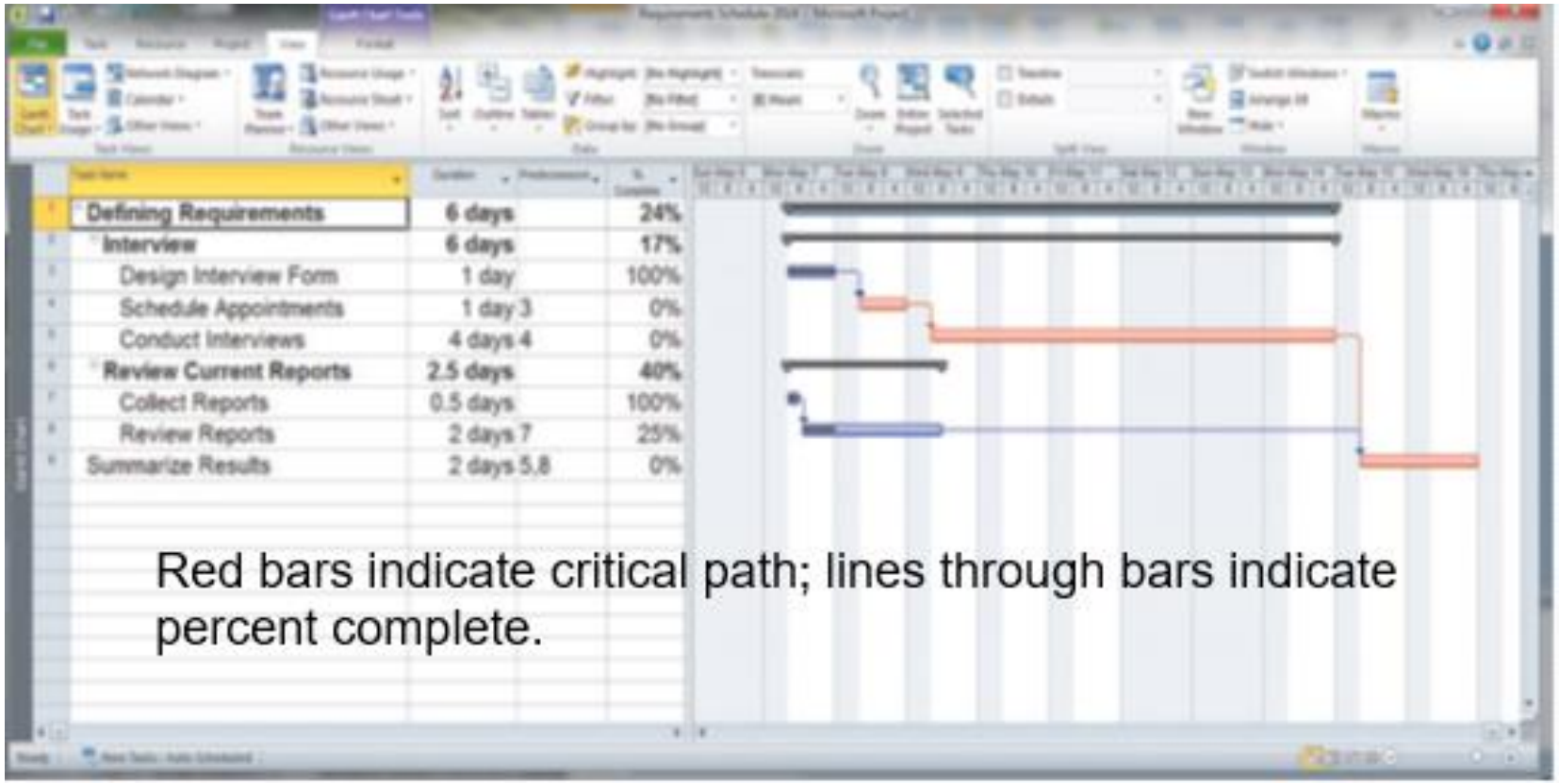


FIGURE :
Gantt chart with tasks 3 and 7 completed and task 8 partially completed (*Source:* Microsoft Corporation.)

4. Maintaining the project workbook

- As in all project phases, maintaining complete records of all project events is necessary. The workbook provides the documentation new team members require to assimilate (to take in, fit into, or become similar) project tasks quickly. It explains why design decisions were made and is a primary source of information for producing all project reports.

5. Communicating the project status

- The project manager is responsible for keeping all stakeholders—ystem developers, managers, and customers—abreast (describes two or more people who are next to each other and moving in the same direction) of the project status.

Phase 4: Closedown

- The final phase of the project management process that focuses on bringing a project to an end.
- The focus of project closedown is to bring the project to an end.
- Projects can conclude with a natural or unnatural termination.
- A **natural termination** occurs when the requirements of the project have been met—the project has been completed and is a success.
- An **unnatural termination** occurs when the project is stopped before completion.
- **Activities During Close down:**
    - Close down the project.
    - Conduct post-project reviews.
    - Close the customer contract.

- **Closing-down the project**

- Inform all members about the project end during a review meeting

- **Conducting post-project review**

- Set a review meeting with management and customers to assess project' strengths and weakness

- Develop new idea for new projects

- **Closing the customer contract**

- Stop funding and further new projects

When does a project end?

- If requirements have been all met (normal end)
- If all objectives have been successfully achieved
- Customers' need are not any more valid in the customer business
- environment; state-of-the-art technology is available on the market)
- Running out of money further new projects

Representing and Scheduling Project

- A project manager has a wide variety of techniques available for depicting and documenting project plans.
- These planning documents can take the form of graphical or textual reports, although graphical reports have become most popular for depicting project plans.
- The most commonly used methods are **Gantt charts** and **network diagrams**.
- Because Gantt charts do not (typically) show how tasks must be ordered (precedence) but simply show when a task should begin and when it should end, they are often more useful for depicting relatively simple projects or subparts of a larger project, showing the activities of a single worker, or monitoring the progress of activities compared to scheduled completion dates.
- A network diagram shows the ordering of activities by connecting a task to its predecessor and successor tasks.
- Sometimes a network diagram is preferable; other times a Gantt chart more easily shows certain aspects of a project.

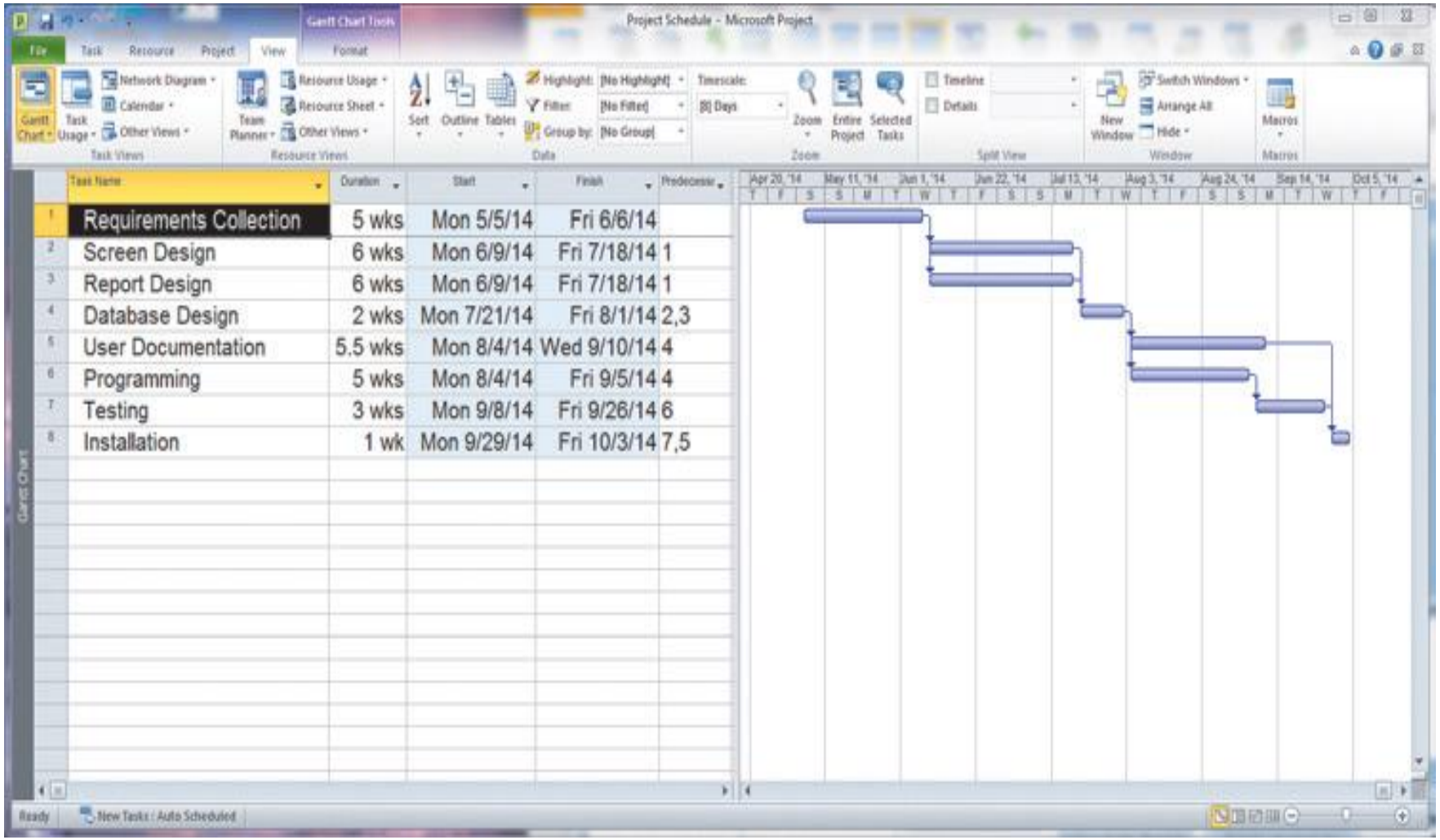


Figure: Graphical diagrams that depict project plans
(a) A Gantt chart

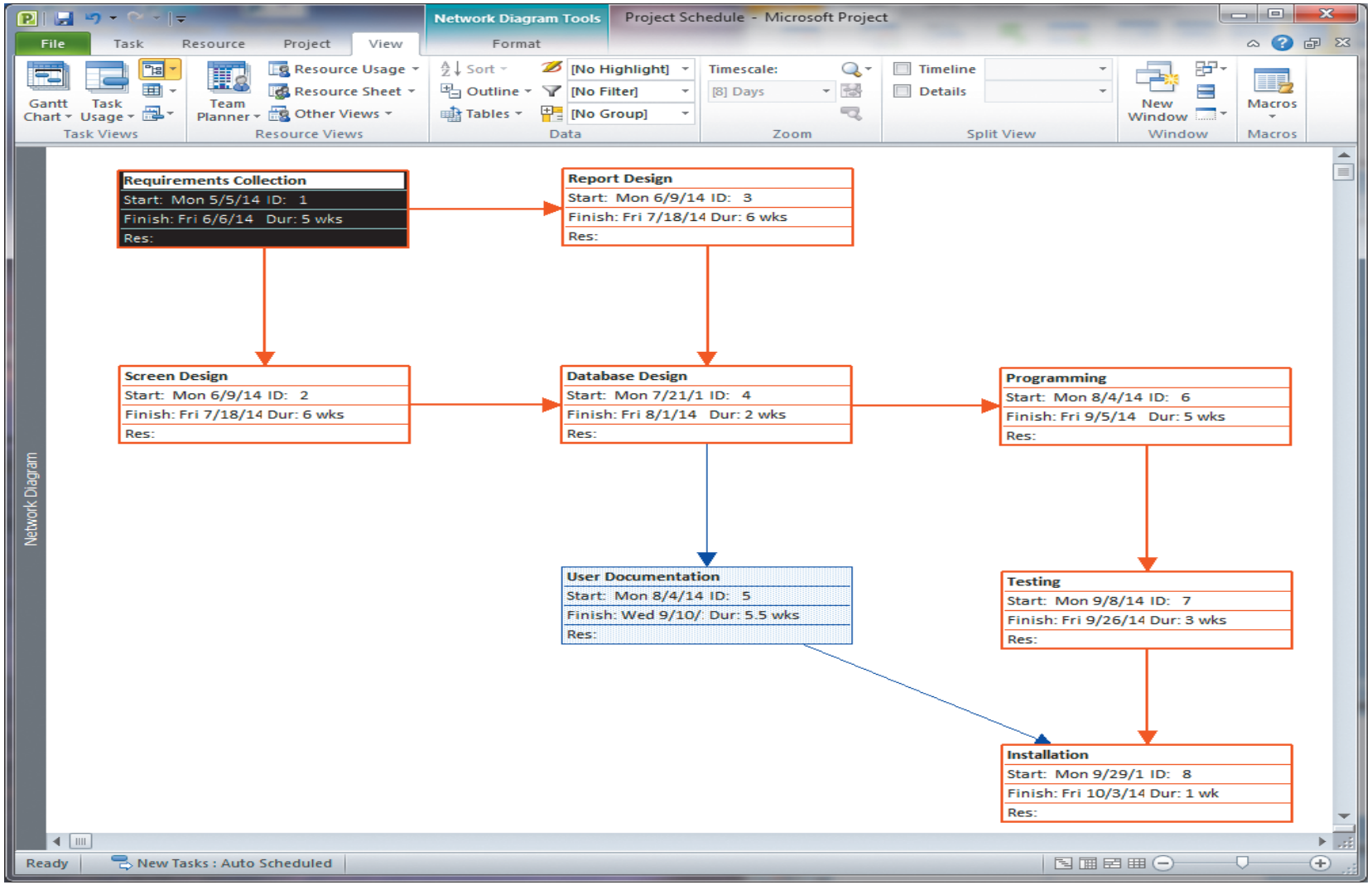


Figure: Graphical diagrams that depict project plans
(b) A network diagram

Gantt Charts vs. Network Diagrams

- Here are the key differences between these two charts:
- Gantt charts visually show the duration of tasks, whereas a network diagram visually shows the sequence dependencies between tasks.
- Gantt charts visually show the time overlap of tasks, whereas a network diagram does not show time overlap but does show which tasks could be done in parallel.
- Some forms of Gantt charts can visually show slack time available within an earliest start and latest finish duration. A network diagram shows this by data within activity rectangles.

- A project manager will periodically review the status of all ongoing project task activities to assess whether the activities will be completed early, on time, or late.
- If early or late, the duration of the activity can be updated.
- Once changed, the scheduled start and finish times of all subsequent tasks will also change.
- Making such a change will also alter a Gantt chart or network diagram used to represent the project tasks.

Representing Project Plans:

- Project scheduling and management require that time, costs, and resources be controlled. Resources are any person, group of people, piece of equipment, or material used in accomplishing an activity.
- Network diagramming is a critical path scheduling technique used for controlling resources.
- A critical path refers to a sequence of task activities whose order and durations directly affect the completion date of a project.
- A network diagram is one of the most widely used and best known scheduling methods.
- You would use a network diagram when tasks
    - are well defined and have a clear beginning and end point,
    - can be worked on independently of other tasks,
    - are ordered, and
    - serve the purpose of the project

- A major strength of network diagramming is its ability to represent how completion times vary for activities.
- Because of this, it is more often used than Gantt charts to manage projects such as information systems development, where variability in the duration of activities is the norm.
- Network diagrams are composed of circles or rectangles representing activities and connecting arrows showing required work flows

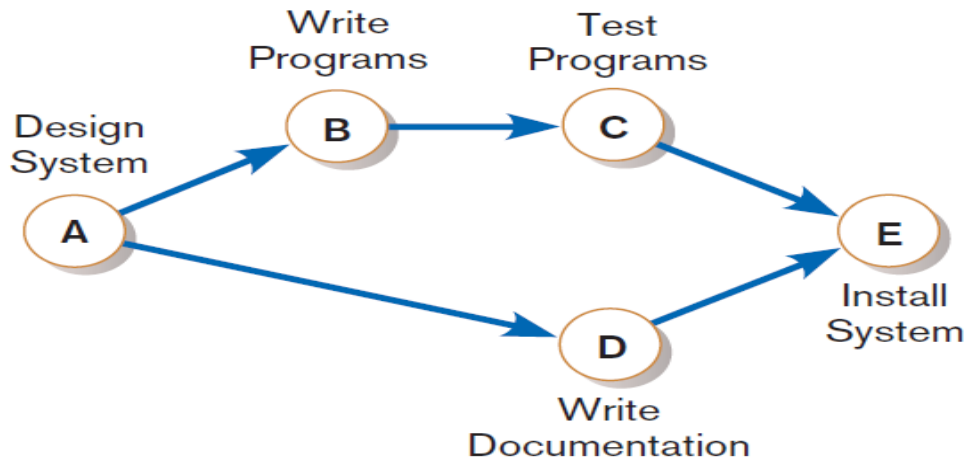


Figure
A network diagram showing activities (represented by circles) and sequence of those activities (represented by arrows)

Calculating Expected time durations using PERT:(Program Evaluation Review Technique)

- One of the most difficult and most error-prone activities when constructing a project schedule is the determination of the time duration for each task within a work breakdown structure.
- It is particularly problematic to make these estimates when there is a high degree of complexity and uncertainty about a task.
- **PERT (Program Evaluation Review Technique)** is a technique that uses optimistic, pessimistic, and realistic time estimates to calculate the expected time for a particular task.
- This technique can help you to obtain a better time estimate when there is some uncertainty as to how much time a task will require to be completed.

- The **optimistic (o)** and pes**simistic (p)** times reflect the minimum and maximum possible periods of time for an activity to be completed.
- The **realistic (r)** time, or most likely time, reflects the project manager's "best guess" of the amount of time the activity actually will require for completion.
- Once each of these estimates is made for an activity, an **expected time (ET)** can be calculated.

$$ET = \frac{o + 4r + p}{6}$$

where

ET = expected time for the completion for an activity
o = optimistic completion time for an activity
r = realistic completion time for an activity
p = pessimistic completion time for an activity

Example

- For example, suppose that your instructor asked you to calculate an expected time for the completion of an upcoming programming assignment. For this assignment, you estimate an optimistic time of two hours, a pessimistic time of eight hours, and a most likely time of six hours. Using PERT, the expected time for completing this assignment is 5.67 hours. Commercial project management software such as Microsoft Project assists you in using PERT to make expected time calculations. Additionally, many commercial tools allow you to customize the weighting of optimistic, pessimistic, and realistic completion times.

Using project management software

- A wide variety of automated project management tools is available to help you manage a development project.
- New versions of these tools are continuously being developed and released by software vendors.
- Most of the available tools have a set of common features that include the ability to define and order tasks, assign resources to tasks, and easily modify tasks and resources.
- Project management tools are available to run on IBM-compatible personal computers, the Macintosh, and larger mainframe and workstation-based systems.
- These systems vary in the number of task activities supported, the complexity of relationships, system processing and storage requirements, and, of course, cost.
- When using this system to manage a project, you need to perform at least the following activities:
    - Establish a project starting or ending date.
    - Enter tasks and assign task relationships.
    - Select a scheduling method to review project reports.

- Project management software helps project managers (PMs) and teams collaborate and meet goals on time while managing resources and cost. Functions may include task distribution, time tracking, budgeting, resource planning, team collaboration, and many more.
- People also refer to project management software as Task Management Software or Project Portfolio Management (PPM).
- Project management software covers a range of platforms, each with a slightly different mix of functionality. It’s crucial that the vendor you select makes your projects easier to manage and doesn’t add unneeded complexity. The transition should be as smooth as possible.
- The three major pillars of project management are planning, tracking, and collaboration.

Project Start Date

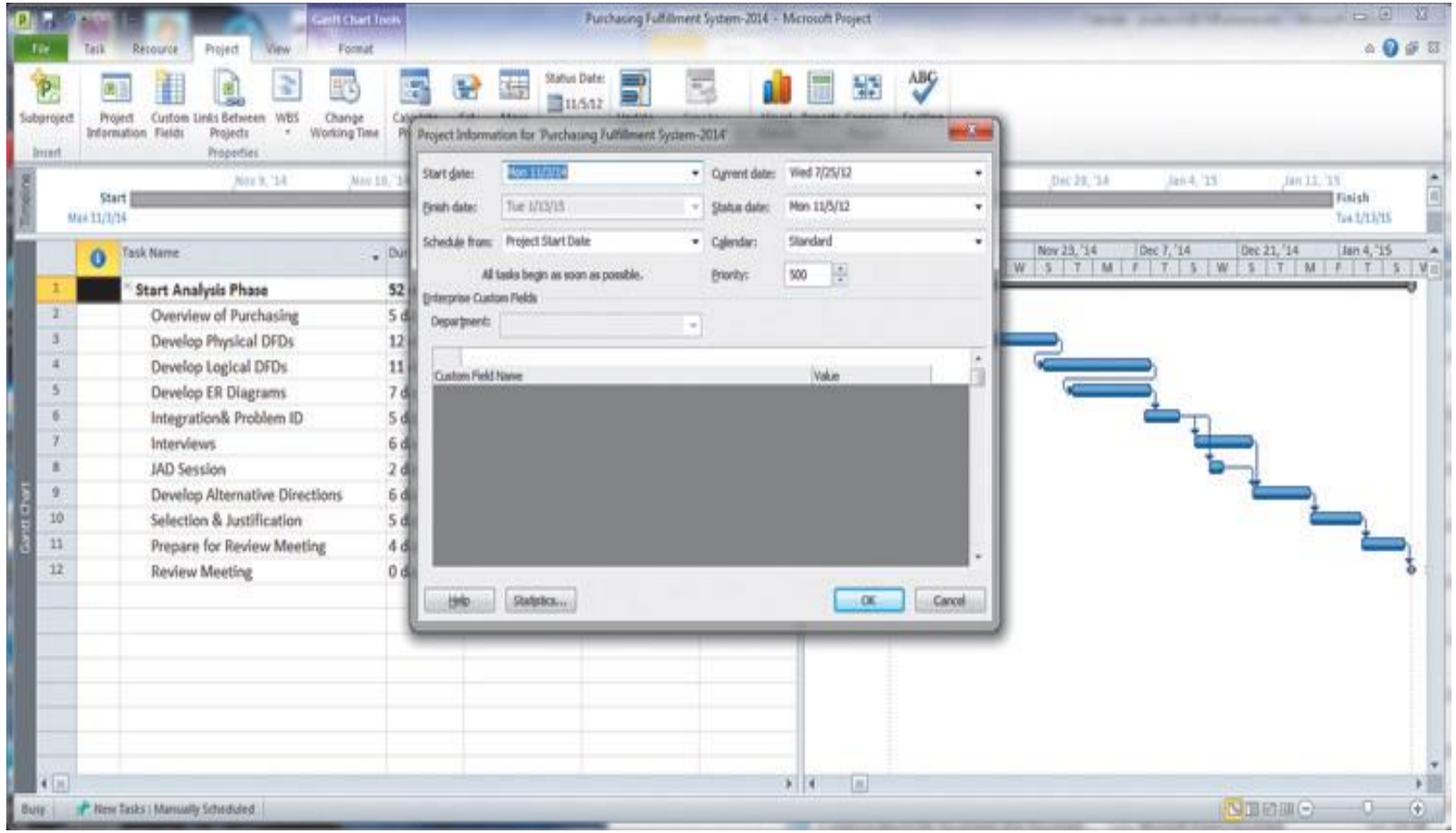
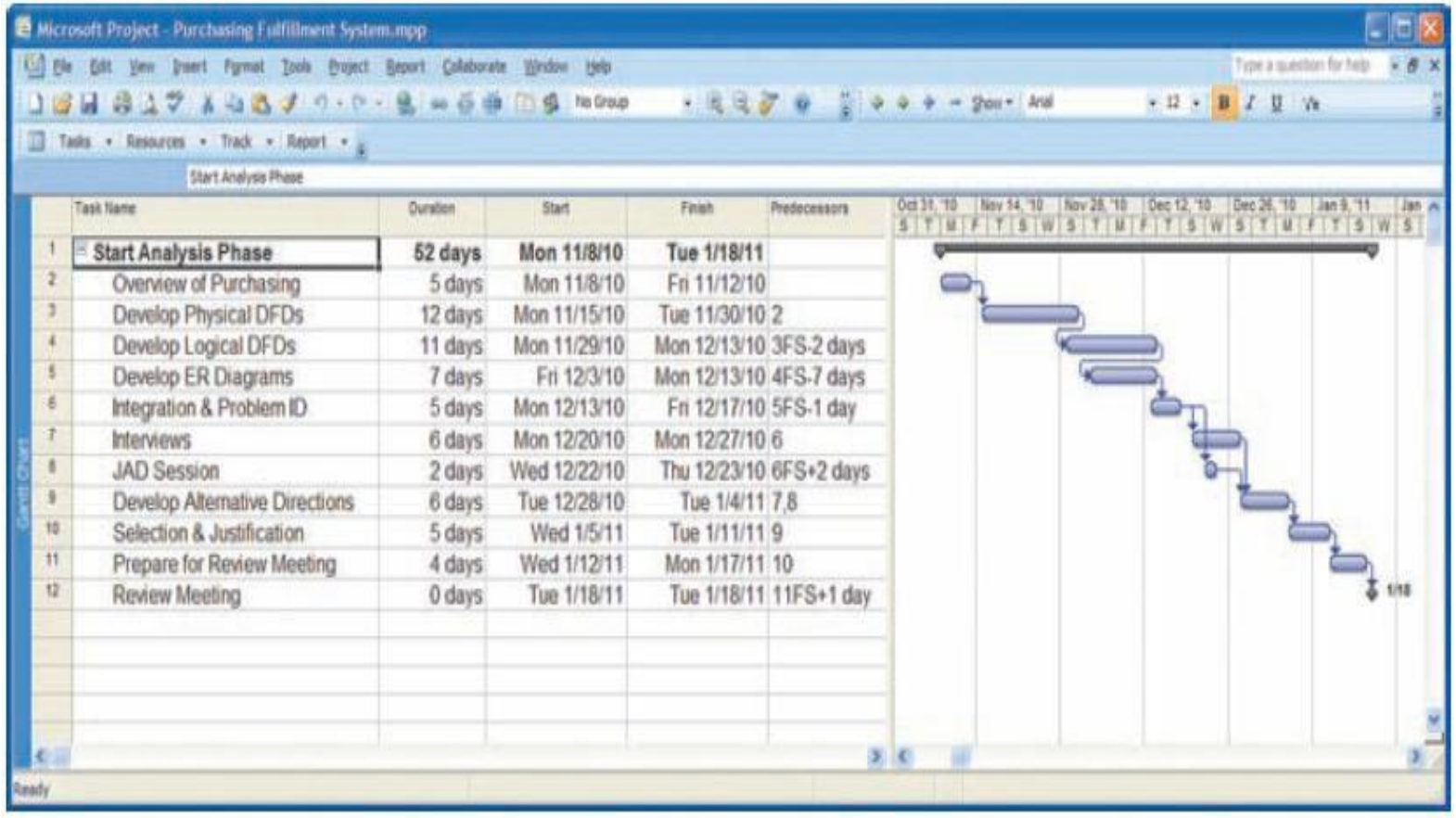


FIGURE
Establishing a project starting date in Microsoft Project for Windows (*Source:* Microsoft Corporation.)

Entering Tasks



	Task Name	Duration	Start	Finish	Predecessors
1	Start Analysis Phase	52 days	Mon 11/8/10	Tue 1/18/11	
2	Overview of Purchasing	5 days	Mon 11/8/10	Fri 11/12/10	
3	Develop Physical DFDs	12 days	Mon 11/15/10	Tue 11/30/10	2
4	Develop Logical DFDs	11 days	Mon 11/29/10	Mon 12/13/10	3FS-2 days
5	Develop ER Diagrams	7 days	Fri 12/3/10	Mon 12/13/10	4FS-7 days
6	Integration & Problem ID	5 days	Mon 12/13/10	Fri 12/17/10	5FS-1 day
7	Interviews	6 days	Mon 12/20/10	Mon 12/27/10	6
8	JAD Session	2 days	Wed 12/22/10	Thu 12/23/10	6FS+2 days
9	Develop Alternative Directions	6 days	Tue 12/28/10	Tue 1/4/11	7,8
10	Selection & Justification	5 days	Wed 1/5/11	Tue 1/11/11	9
11	Prepare for Review Meeting	4 days	Wed 1/12/11	Mon 1/17/11	10
12	Review Meeting	0 days	Tue 1/18/11	Tue 1/18/11	11FS+1 day

FIGURE
Entering tasks and assigning task relationships in Microsoft project for Windows (*Source:* Microsoft Corporation.)

Viewing Network Diagram

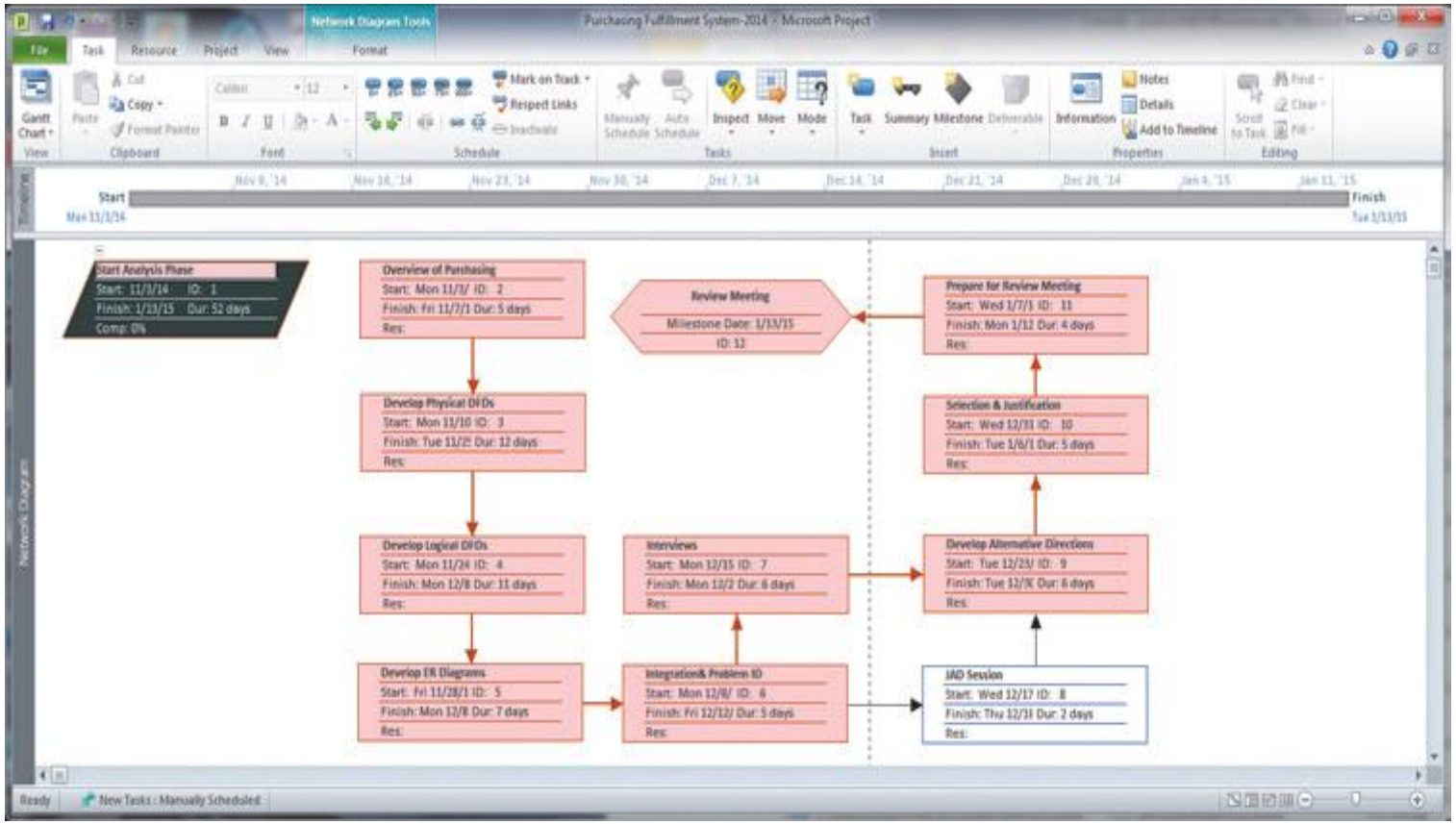


FIGURE:
Viewing project information as a network diagram in Microsoft Project for Windows (*Source:* Microsoft Corporation.)

Hexagon shape indicates a milestone.

Red boxes and arrows indicate critical path (no slack).

Viewing Gantt Chart

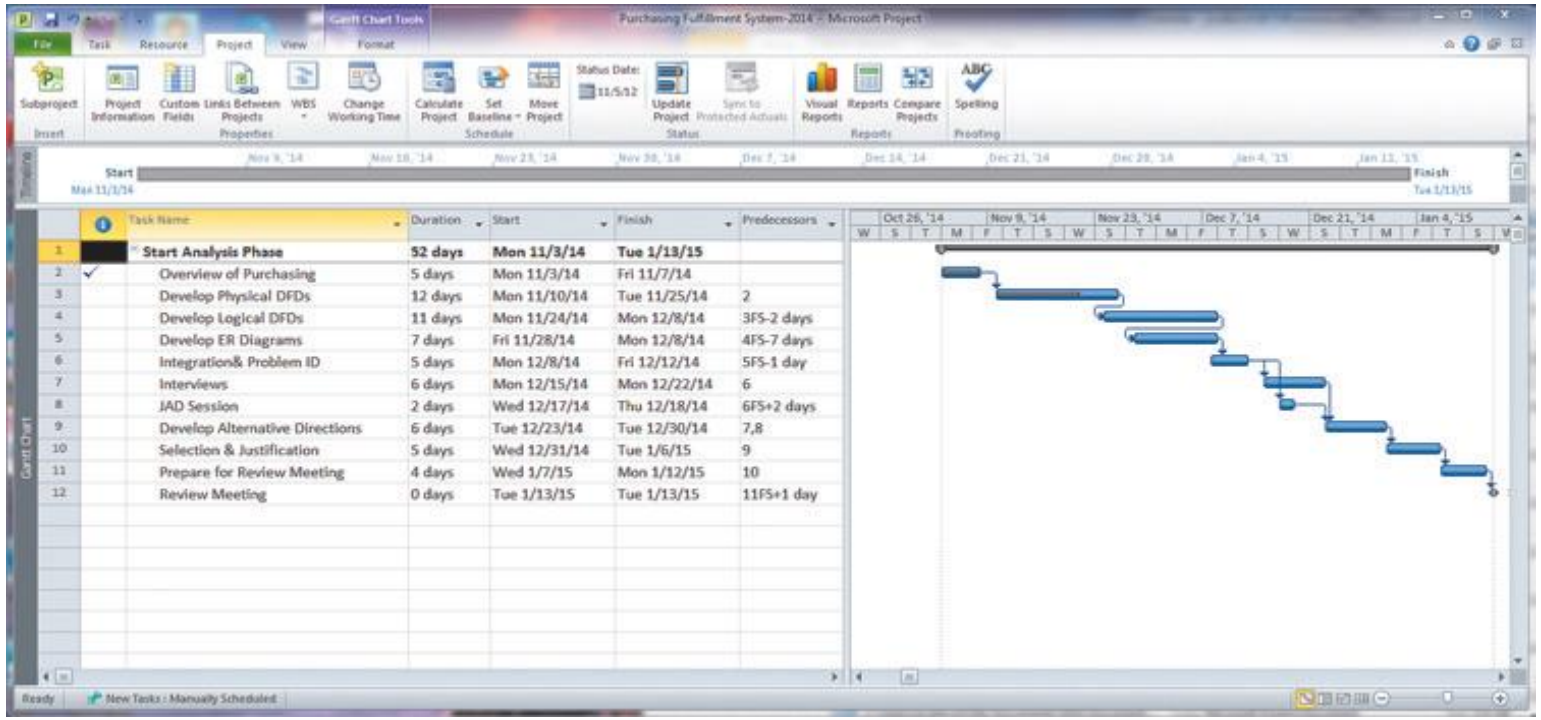


FIGURE
Gantt chart showing progress of activities (right frame) versus planned activities (left frame)

Black line at top indicates a summary activity (composed of subtasks).
Diamond shape indicates a milestone.