System Analysis and Design

Er. Arjun Sharma Poudel

System Analysis and Design

Systems development is systematic process which includes phases such as

1) planning,
2) analysis,
3) design,
4) deployment and
5) maintenance.

Systems Analysis

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components.

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

Analysis specifies **what the system should do.**

Systems Design

It is a process of planning a new business system or replacing an existing system by defining its components or modules to satisfy the specific requirements. Before planning, you need to understand the old system thoroughly and determine how computers can best be used in order to operate efficiently.

System Design focuses on **how to accomplish the objective of the system**.

System Analysis and Design (SAD) mainly focuses on –

- Systems
- Processes
- Technology

Definition of System

- A system is an orderly grouping of interdependent components linked together according to a plan to achieve a specific objective.

- Eg: Digestive System ,Public Transportation System , Computer System

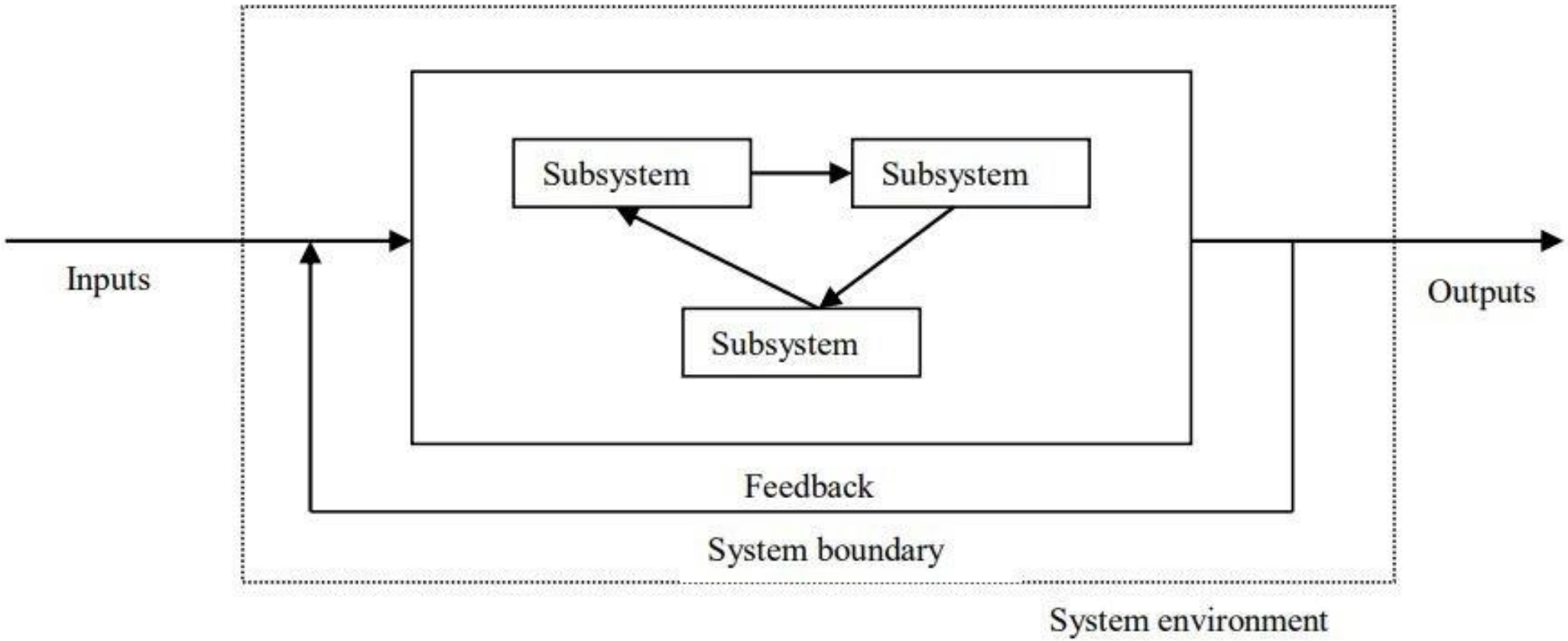
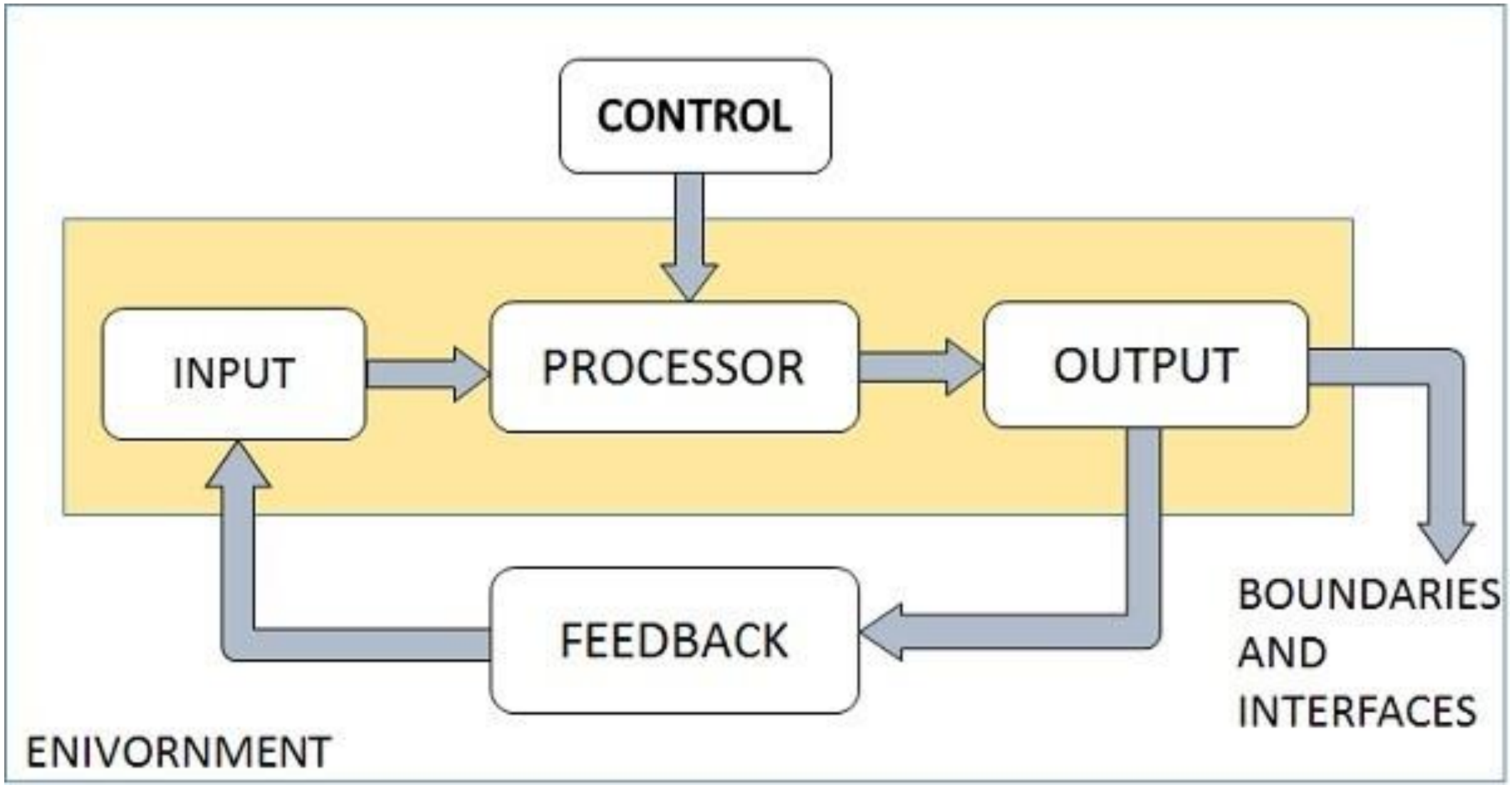


Fig: Basic System Model


CONTROL
INPUT
PROCESSOR
OUTPUT
FEEDBACK
BOUNDARIES
AND
INTERFACES
ENIVORNMENT

- 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

Characteristics of a System

- Organization
- Interaction
- Interdependence
- Integration
- Central Objective

- Organization-It implies structure and order.
- Interaction-It refers to manner in which each component functions with other components of the system.
- Interdependence-Units/parts are dependent on each other.
- Integration-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.

The Systems Development Environment

- Information Systems Analysis and Design (ISAD)
  - Complex organizational process
  - Used to develop and maintain computer based information systems.
  - Used by a team of business and systems professionals.
  - An organizational improvement process (respond to and anticipate problems).
  - Uses technology (Internet, WWW marketing, online business, eBay, Amazon.com etc).

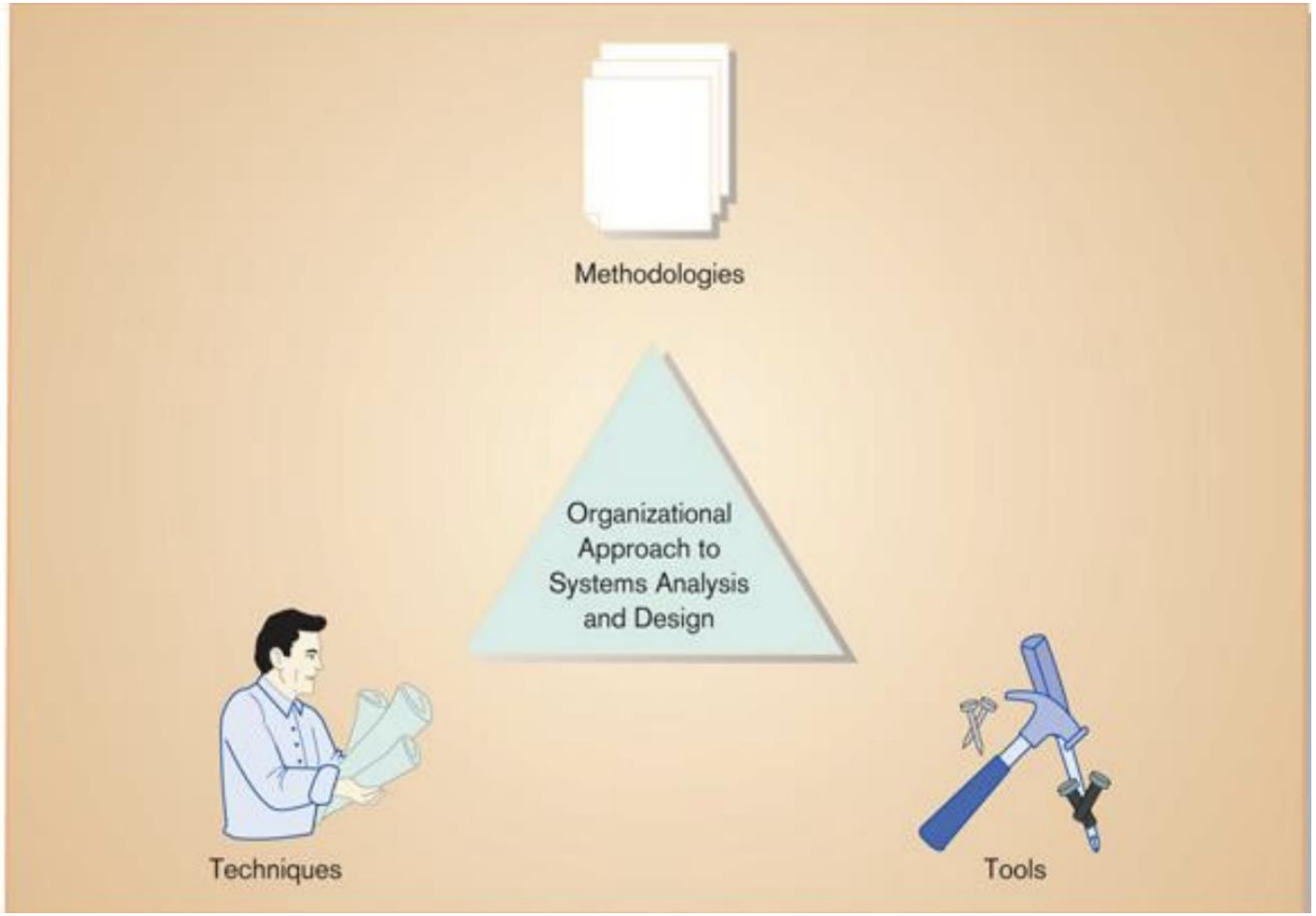


Figure 1 An organizational approach to systems analysis and design is driven by methodologies, techniques, and tools

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).

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

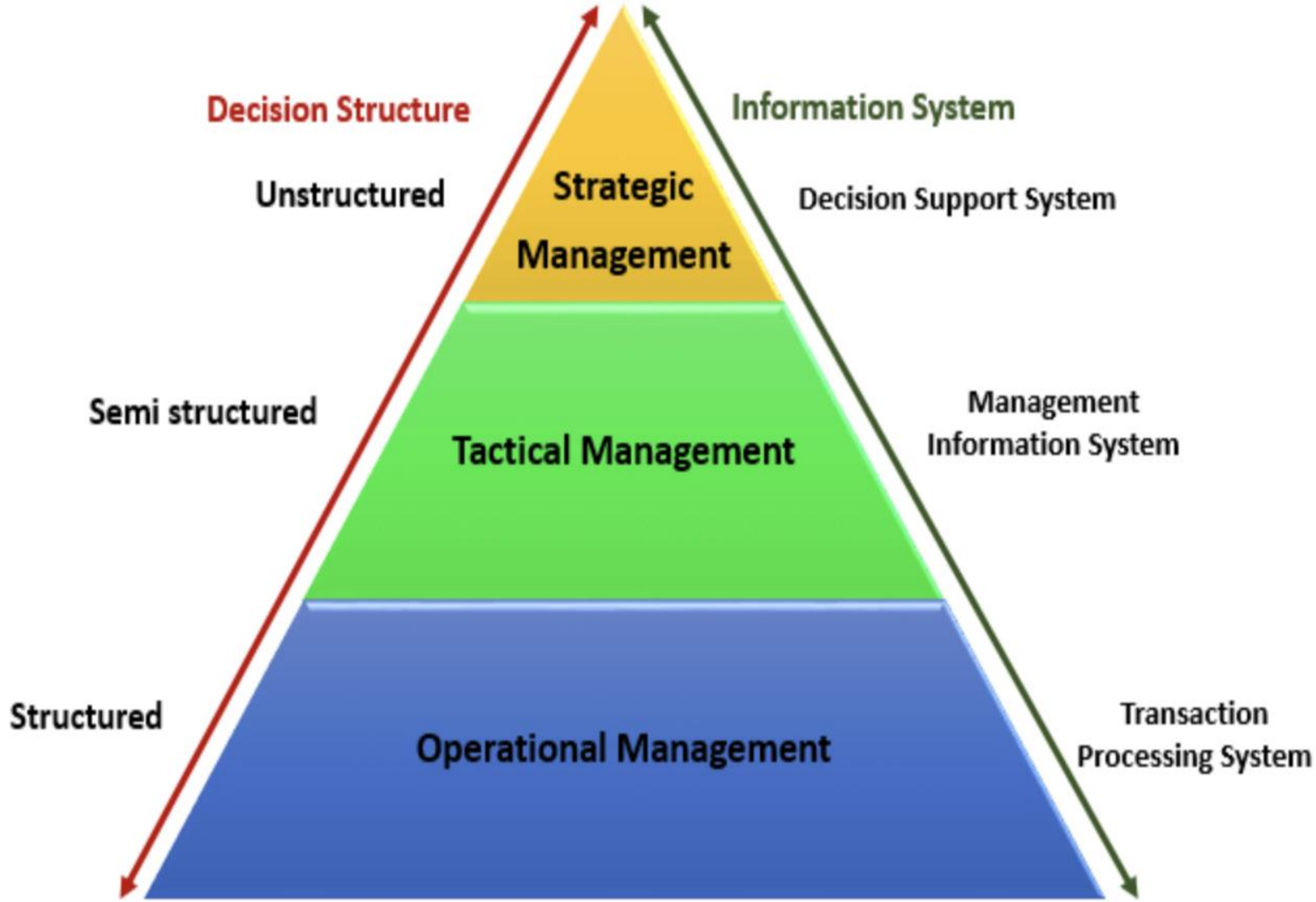
- Installing a system without proper planning leads to great user dissatisfaction and frequently causes the system to fall into disuse.
- Lends structure to the analysis and design of information systems.
- A series of processes systematically undertaken to improve a business through the use of computerized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

What is an Information System?

- A system that provides information to people in an organization is called information system (IS).
- An information system is an arrangement of people, data, processes, interfaces, networks, and technology that interact for the purpose of supporting and improving both day-to-day operations in a business (sometimes called data processing), as well as supporting the problem solving and decision making needs of management (sometimes called information services).

Types of Information System(IS)

- 1. Transaction Processing System (TPS)
- 2. Management Information System (MIS)
- 3. Decision Support System (DSS)
- 4. Experts System


Decision Structure
Unstructured
Semi structured
Structured
Information System
Decision Support System
Management
Information System
Transaction
Processing System
Strategic
Management
Tactical Management
Operational Management

Transaction Processing System (TPS)

- Transaction Processing System are information system that processes data resulting from the occurrences of business transactions
- Their objectives are to provide transaction in order to update records and generate reports i.e. to perform store keeping function
- The transaction is performed in two ways: **Batching processing** and **Online transaction processing**.

- The main objective of a transaction processing system is to answer routine questions such as;
- **How printers were sold today?**
- **How much inventory do we have at hand?**
- **What is the outstanding due for John Doe?**

- Examples of transaction processing systems include;
- **Point of Sale Systems** – records daily sales
- **Payroll systems** – processing employees salary, loans management, etc.
- **Stock Control systems** – keeping track of inventory levels
- **Airline booking systems** – flights booking management

Management Information System (MIS)

- Management Information System is designed to take relatively raw data available through a Transaction Processing System and convert them into a summarized and aggregated form for the manager, usually in a report format. It reports tending to be used by middle management and operational supervisors.
- Management Information Systems (MIS) are used by tactical managers to monitor the organization's current performance status. The output from a transaction processing system is used as input to a management information system.

- Many different types of report are produced in MIS. Some of the reports are a summary report, on-demand report, ad-hoc reports, etc
- The MIS system analyzes the input with routine algorithms i.e. aggregate, compare and summarizes the results to produced reports that tactical managers use to monitor, control and predict future performance.
- For example, input from a point of sale system can be used to analyze trends of products that are performing well and those that are not performing well. This information can be used to make future inventory orders i.e. increasing orders for well-performing products and reduce the orders of products that are not performing well.

- MIS systems provide the information needed to make the structured decision and based on the experience of the tactical managers, they make judgment calls i.e. predict how much of goods or inventory should be ordered for the second quarter based on the sales of the first quarter
- Examples of management information systems include;
- **Sales management systems** – they get input from the point of sale system
- **Budgeting systems** – gives an overview of how much money is spent within the organization for the short and long terms.
- **Human resource management system** – overall welfare of the employees, staff turnover, etc.

Decision Support System (DSS)

- Decision support systems are used by senior management to make non-routine decisions. Decision support systems use input from internal systems (transaction processing systems and management information systems) and external systems.
- The main objective of decision support systems is to provide solutions to problems that are unique and change frequently. Decision support systems answer questions such as;
- What would be the impact of employees' performance if we double the production lot at the factory?
- What would happen to our sales if a new competitor entered the market?

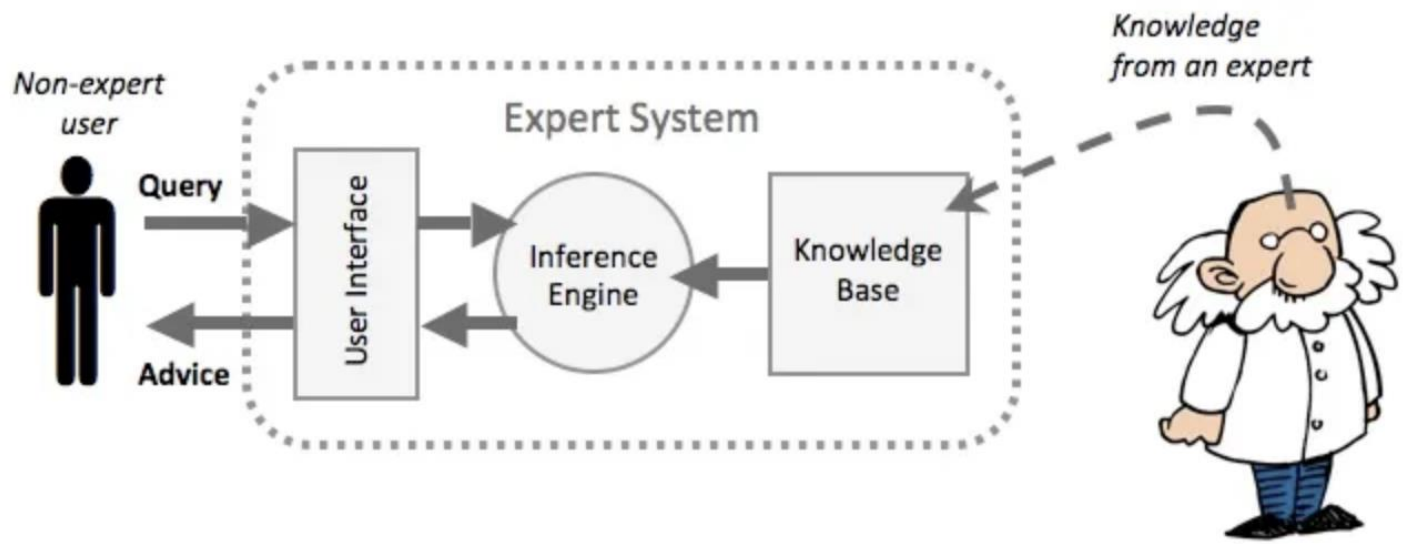
- Decision support systems use sophisticated mathematical models, and statistical techniques (probability, predictive modeling, etc.) to provide solutions, and they are very interactive.

- Examples of decision support systems include;
- **Financial planning systems** – it enables managers to evaluate alternative ways of achieving goals. The objective is to find the optimal way of achieving the goal. For example, the net profit for a business is calculated using the formula Total Sales less (Cost of Goods + Expenses). A financial planning system will enable senior executives to ask what if questions and adjust the values for total sales, the cost of goods, etc. to see the effect of the decision and on the net profit and find the most optimal way.
- **Bank loan management systems** – it is used to verify the credit of the loan applicant and predict the likelihood of the loan being recovered.

Experts System

- Experts systems include expertise in order to aid managers in diagnosing problems or in problem-solving. These systems are based on the principles of artificial intelligence research.
- Experts Systems is a knowledge-based information system. It uses its knowledge about a specify area to act as an expert consultant to users.
- Knowledgebase and software modules are the components of an expert system. These modules perform inference on the knowledge and offer answers to a user's question

Expert Systems



Expert System

Executive Management System(EIS)

- Senior managers of an organization use the EIS. Therefore, it must be easy to use so that executives can use it without any assistance.
- EIS can do trend analysis, exception reporting and have drill-down capabilities.
- The results are usually presented in a graphical form tailored to the executive's information needs

Systems Analysis and Design

- A method used by companies to create and maintain systems that perform basic business functions
- Main goal is to improve employee efficiency by applying software solutions to key business tasks
- A structured approach must be used in order to ensure success

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.
- An important (but not the only) 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.
- To be effective, both techniques and tools must be consistent with an organizations system development methodology. These make easy for system developers to conduct the steps in methodology.

FIGURE 1-1
An organizational approach to systems analysis and design is driven by methodologies, techniques, and tools

Sources: Top: Mitarart/Fotolia; Left: Lev/Fotolia; Right: PaulPaladin/Fotolia

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.

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.

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.

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.

System Builders

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

Systems Analysts

- 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

Project Managers

- 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.

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

What is a Systems Analyst?

- 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 For a Career as a Systems Analyst(Or, Skills Needed)

- Working Knowledge of Information Technology
- Programming Experience and Expertise
- General Business Knowledge
- Problem-Solving Skills
- Communications Skills
- Interpersonal Relations Skills
- Flexibility and Adaptability
- Character and Ethics
- Systems Analysis and Design Skills

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.

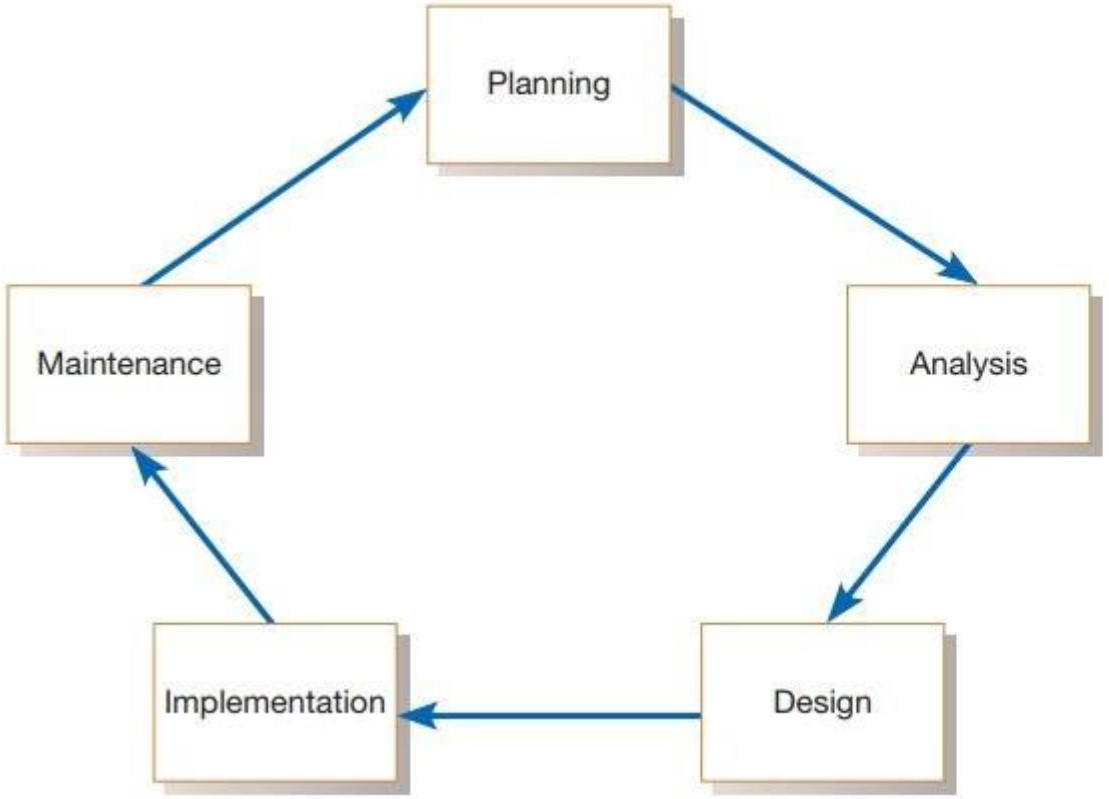


FIGURE 1-2
Systems development life cycle

- 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

Traditional Waterfall SDLC

- Waterfall 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

Traditional Waterfall SDLC

- 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.

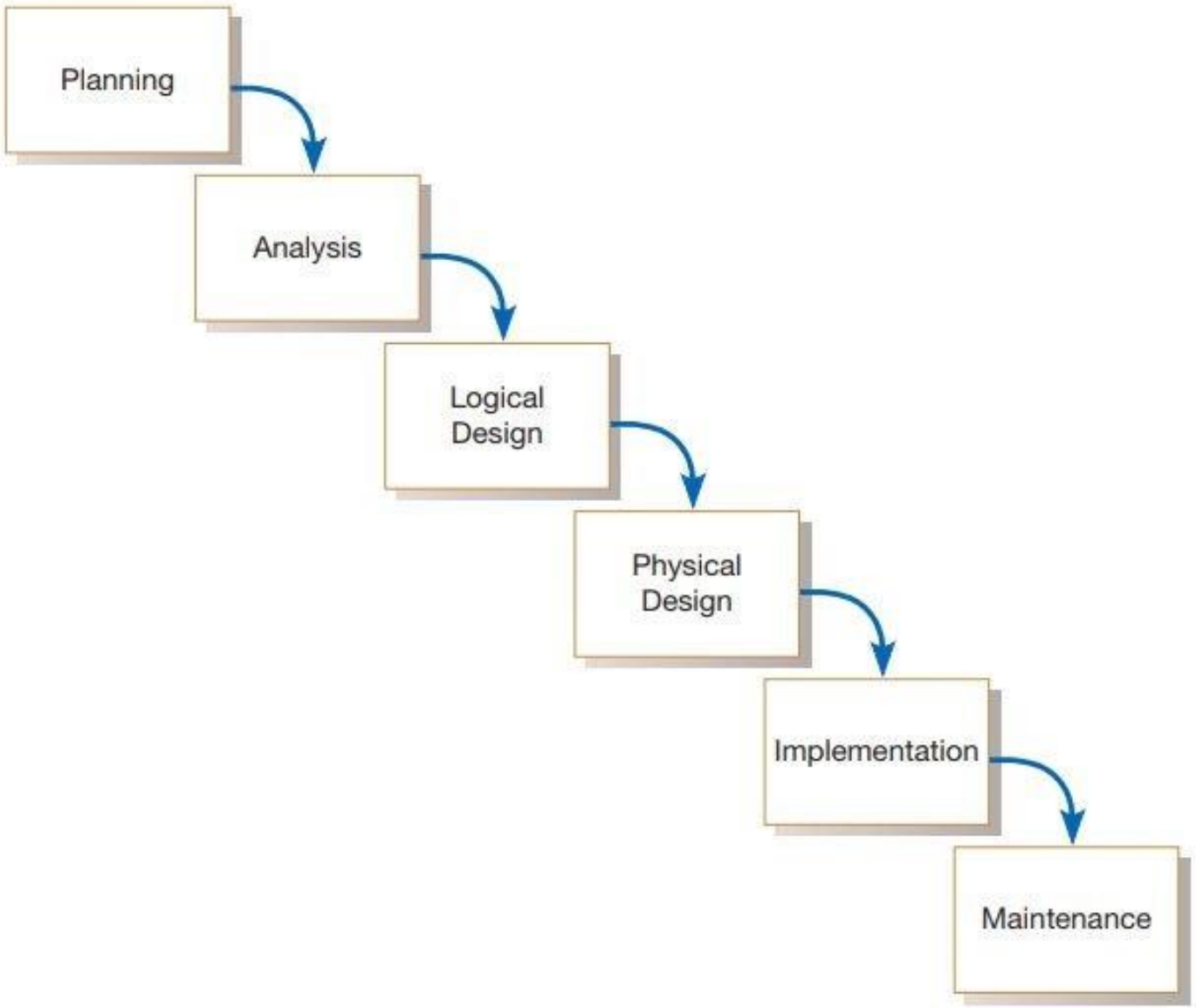


FIGURE 1-10
Traditional waterfall SDLC

- The problems that are sometimes encountered with the linear sequential model are:
- Changes can cause confusion as the project team proceeds.
- It is often difficult for the customer to state all requirements explicitly. The linear sequential model requires this and makes difficulty to respond to changing customer requirements.
- A working version of the system will be available to customers late in the project time-span. A major blunder, if undetected until the working program is reviewed, can be disastrous.
- The linear nature of the classic life cycle leads to “blocking states” in which some project team members must wait for other members of the team to complete dependent tasks.
- User involvement is limited.

Assignment

- When can we use waterfall model?
- Advantages and disdvantages of Waterfall Model.

Different Approaches to Improving Information Systems Development

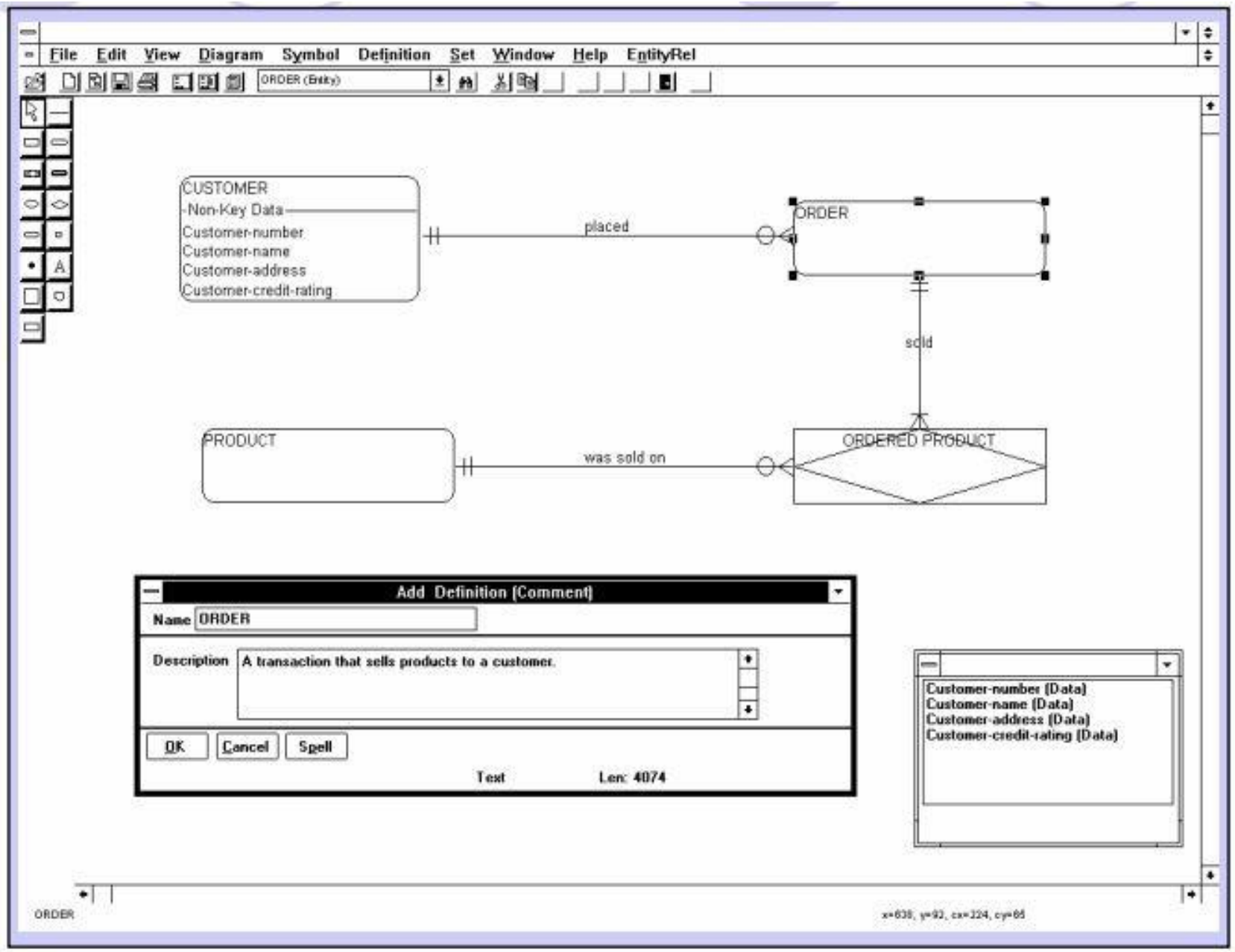
Computer-Aided Systems Engineering (CASE)

- Computer-aided systems engineering (CASE) is the application of information technology to systems development activities, techniques, and methodologies.
- CASE tools are programs (software) that automate or support one or more phases of a systems development life cycle.
- The technology is intended to accelerate the process of developing systems and to improve the quality of the resulting systems.

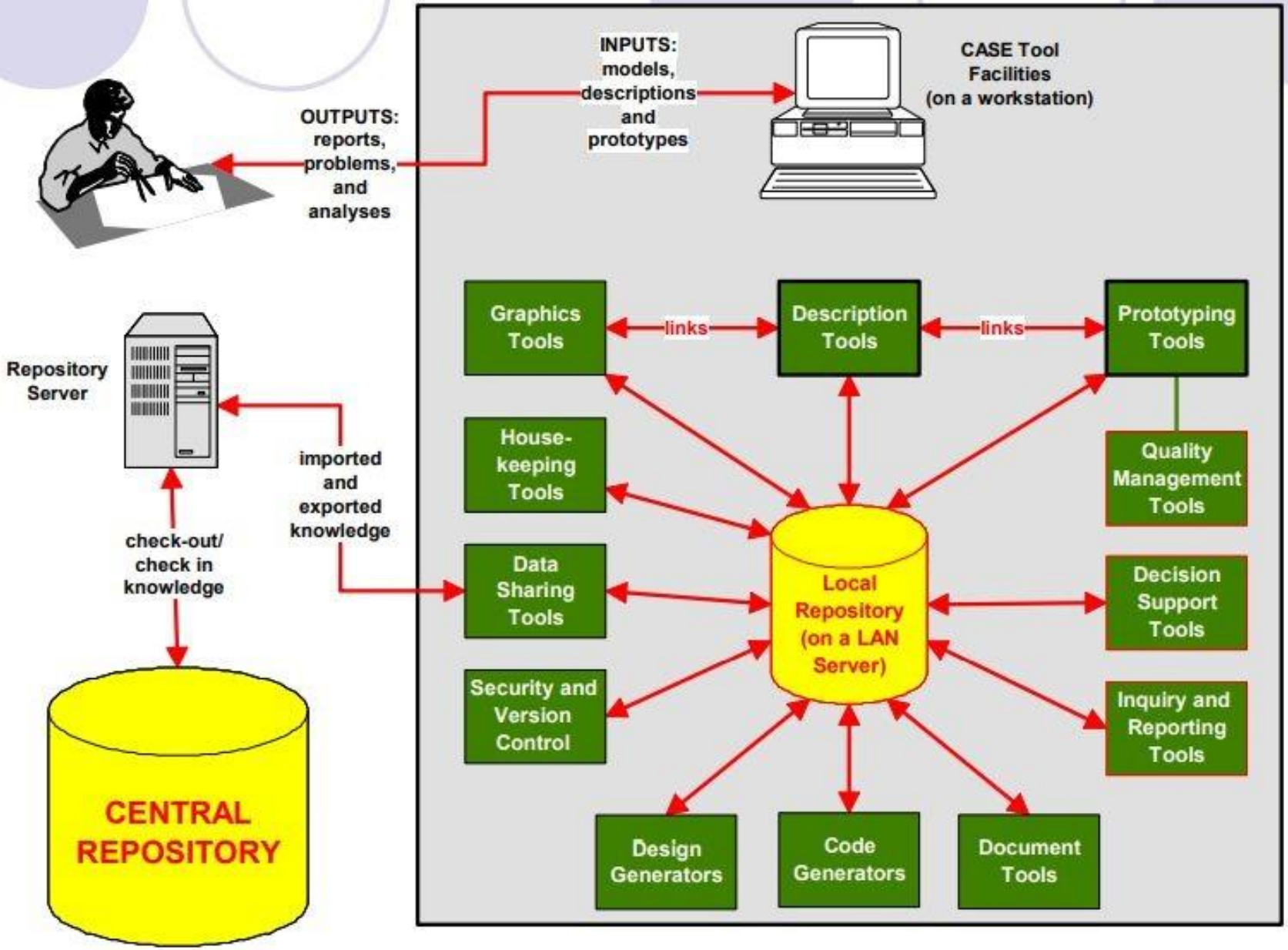
- CASE tools are classified according to which phases of the life cycle they support.
- The term **upper-CASE** describes tools that automate or support the 'upper' phases of systems development – the survey, study, definition, and design phases.
- The term **lower-CASE** describes tools that automate or support the 'lower' phases of systems development – detailed design, construction, and implementation (and also support).

- At the center of any CASE tool's architecture is a developer's database called a CASE repository.
- CASE repository is a system developer's database where developers can store system models, detailed description and specification, and other products of system development.
- It is also called dictionary or encyclopedia.
- Around the CASE repository is a collection of tools or facilities for creating system models and documentation.

- These facilities generally include:
- Diagramming tools –These tools are used to draw system models.
- Dictionary tools – These tools are used to record, delete, edit, and output detailed documentation and specification.
- Design tools – These tools are used to construct system components including system inputs and outputs. These are also called prototyping tools.
- Documentation tools – These tools are used to assemble, organize, and report on system models, descriptions and specifications, and prototypes.
- Quality management tools – These tools are used to analyze system models, descriptions and specifications, and prototypes for completeness, consistency, and conformance to accepted rules of methodologies.
- Design and code generator tools – These tools automatically generate database designs and application programs or significant portions of those programs.



A CASE Tool Screen



CASE Architecture

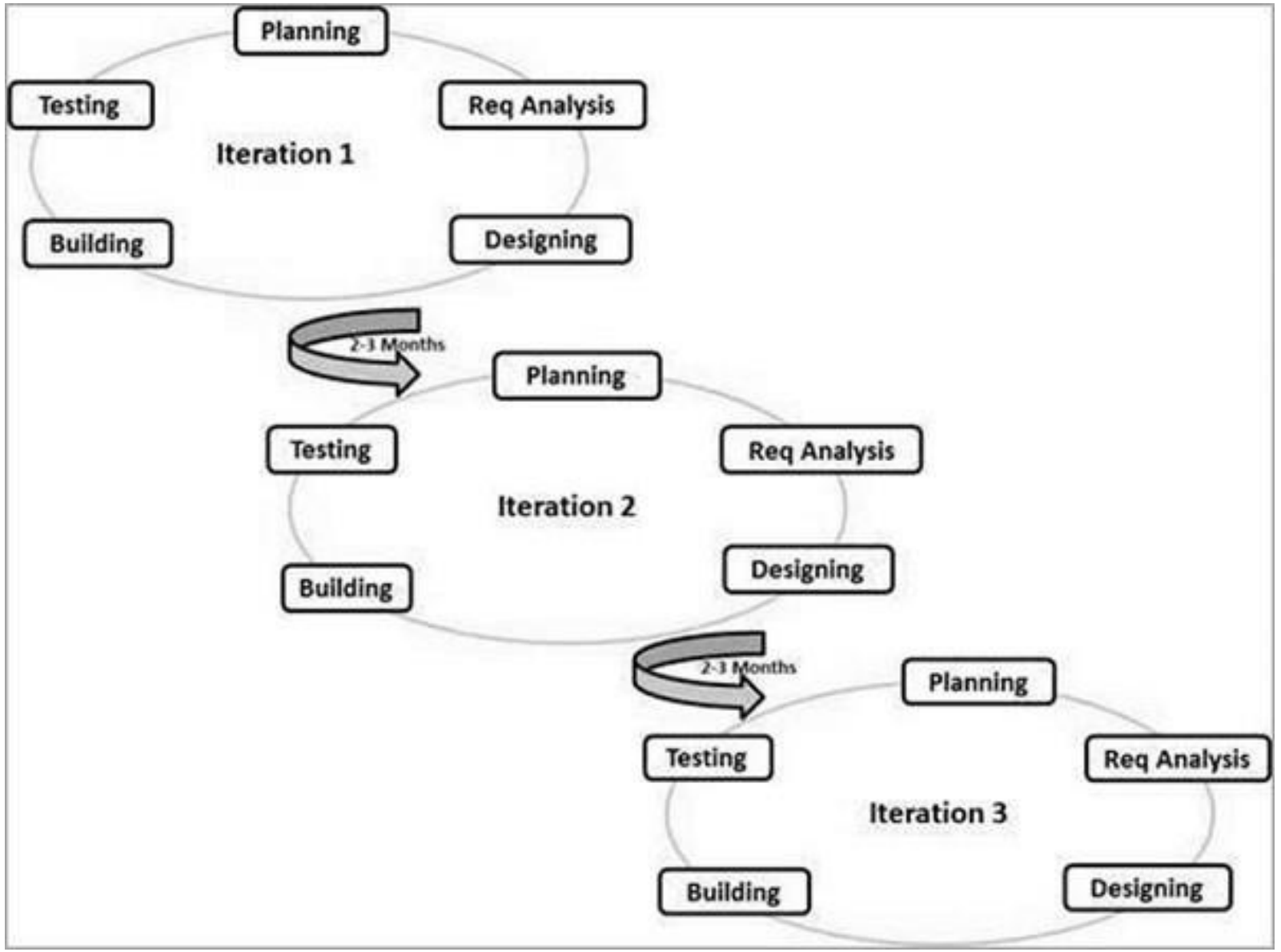
Assignment

- Advantages and disadvantages of CASE Tools.

Agile Model

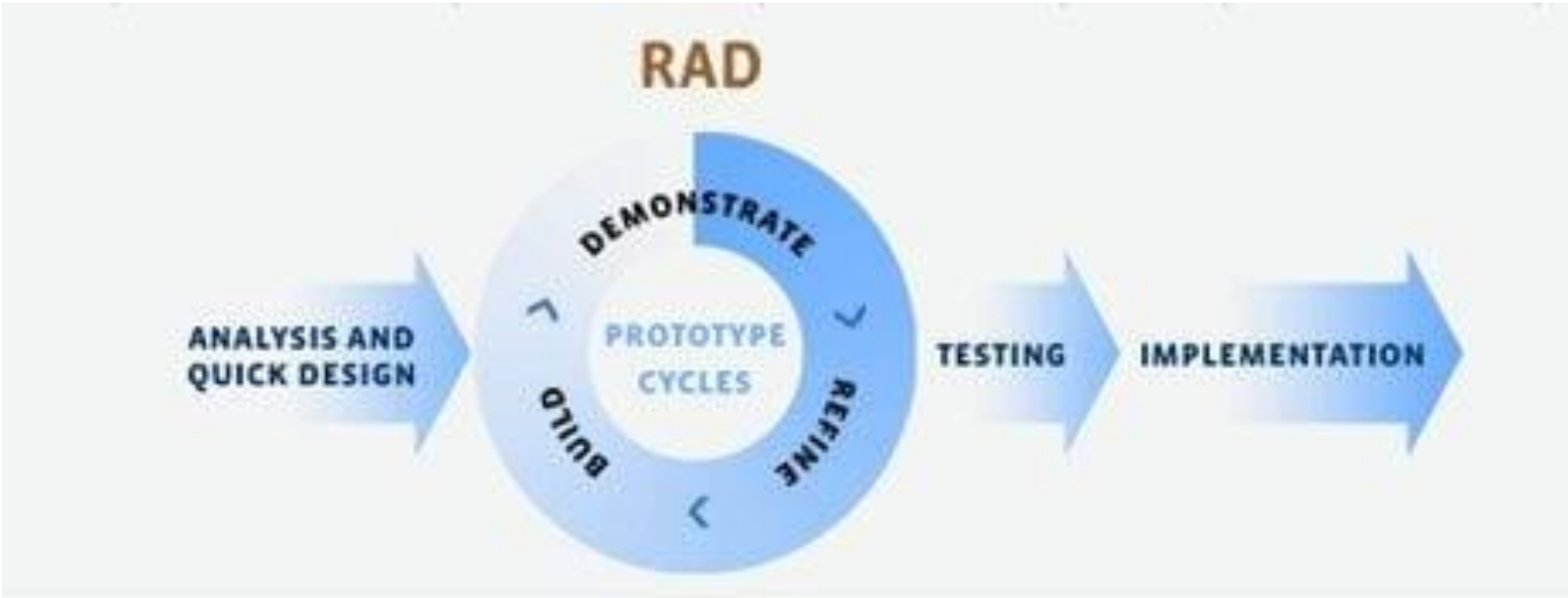
- Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.
- Agile Methods break the product into small incremental builds. These builds are provided in iterations.
- Each iteration typically lasts from about one to three weeks.
- Three key principles
- • Adaptive rather than predictive.
- • Emphasize people rather than roles.
- • Self-adaptive processes.

- Every iteration involves cross functional teams working simultaneously on various areas like −
- Planning
- Requirements Analysis
- Design
- Coding
- Unit Testing and
- Acceptance Testing.
- At the end of the iteration, a working product is displayed to the customer and important stakeholders


Planning
Testing
Req Analysis
Iteration 1
Building
Designing
2-3 Months
Planning
Testing
Req Analysis
Iteration 2
Building
Designing
2-3 Months
Planning
Testing
Req Analysis
Iteration 3
Building
Designing

RAD (Rapid Application Development)

- 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.


RAD
ANALYSIS AND QUICK DESIGN
DEMONSTRATE
REFINE
BUILD
PROTOTYPE CYCLES
TESTING
IMPLEMENTATION

- A rapid application development (or RAD) approach based on iterative prototyping.
- This strategy designs and constructs the system as a series of prototypes to which the system users react.
- The prototyping process is as follows:
- Step 1. - Define the base-level scope of the first (or next) version of the system.
- Step 2. - Define, design, construct, and load the database.

- Step 3. - Define, design, and construct the inputs. Demonstrate this prototype to the system users. (*Repeat step 3 until the system users are satisfied. If necessary, return to step 1 to add new requirements to the database design*.)
- Step 4. - Define, design, and construct the outputs. Demonstrate this prototype to the system users. (*Repeat step 4 until the system users are satisfied. If necessary, return to step 1 to add new database requirements, or step 2 to add new input requirements*.)

- Step 5. - Define, design, and construct the interface. Demonstrate this prototype to the system users. (*Repeat step 5 until the system users are satisfied. If necessary, return to step 1, 2, or 3 to add new database, input, or output requirements, respectively*.)
- Step 6. - Design and construct any missing system controls such as security, backup, recovery, etc.
- Step 7. - Implement this version of the system.
- Step 8. - Go to step 1 to begin the RAD cycle for the next version of the system.

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

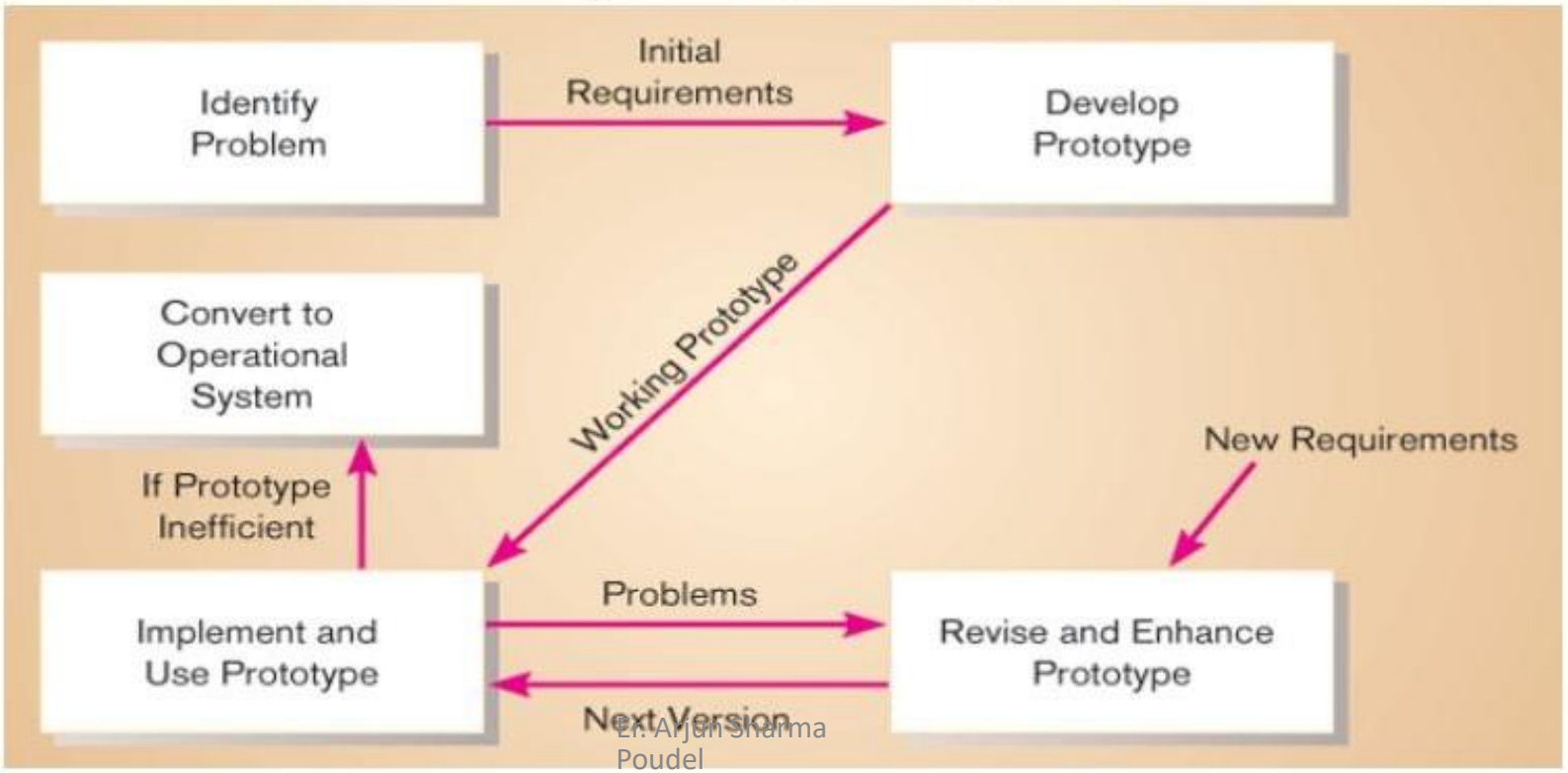
Advantages and Disadvantages of SDLC RAD Model

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

Prototyping

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.

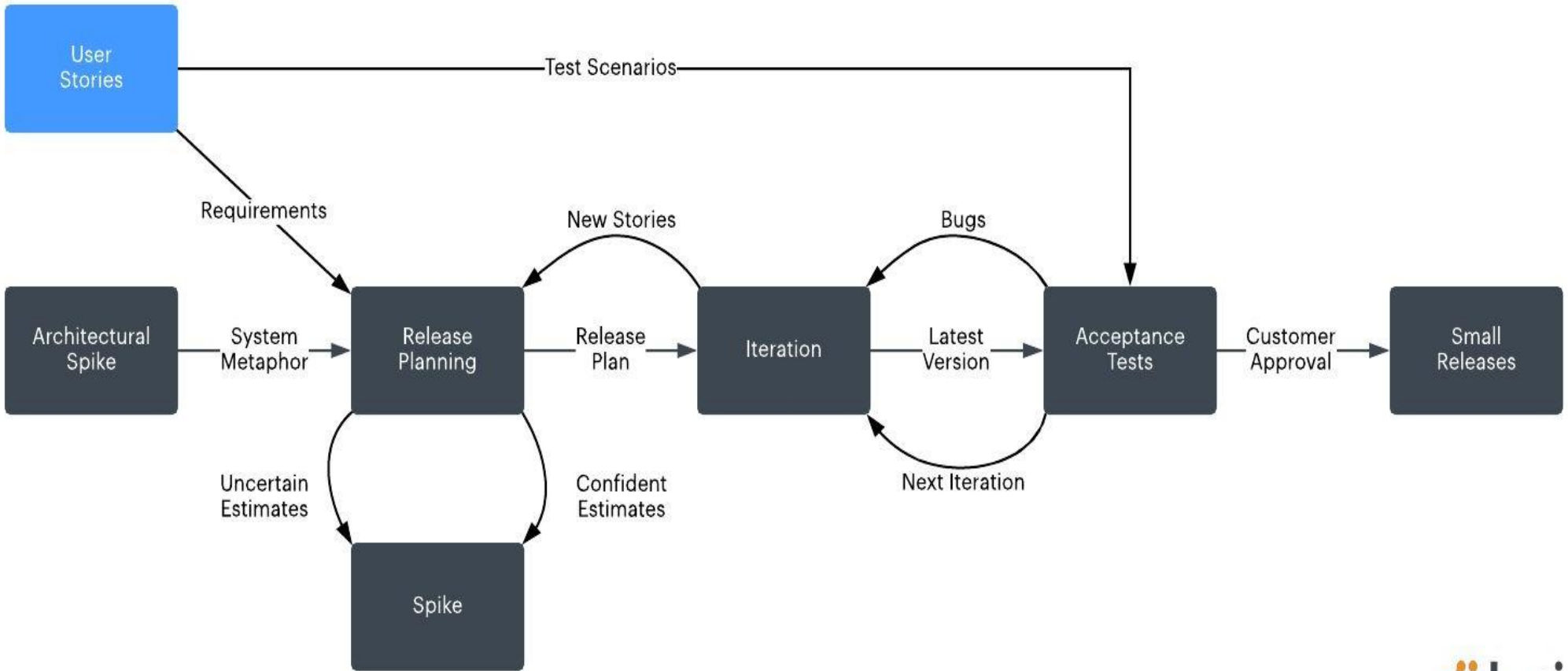


In prototyping, the analyst works with users to determine the initial or basic requirements for the system. The analyst then quickly builds a prototype. When the prototype is completed, the users work with it and tell the analyst what they like and do not like about it. The analyst uses this feedback 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.

Extreme Programming(XP)

- XP is a lightweight, efficient, low-risk, flexible, predictable, scientific, and fun way to develop a software.
- e**X**treme **P**rogramming (XP) was conceived and developed to address the specific needs of software development by small teams in the face of vague and changing requirements.
- Extreme Programming is one of the Agile software development methodologies.
- It provides values and principles to guide the team behavior. The team is expected to self-organize.

Extreme Programming (XP) Methodology



Made in Lucidchart

- Some of the good practices that have been recognized in the extreme programming model and suggested to maximize their use are given below:
- **Code Review:** Code review detects and corrects errors efficiently. It suggests pair programming as coding and reviewing of written code carried out by a pair of programmers who switch their works between them every hour.
- **Testing:** Testing code helps to remove errors and improves its reliability. XP suggests test-driven development (TDD) to continually write and execute test cases. In the TDD approach test cases are written even before any code is written.

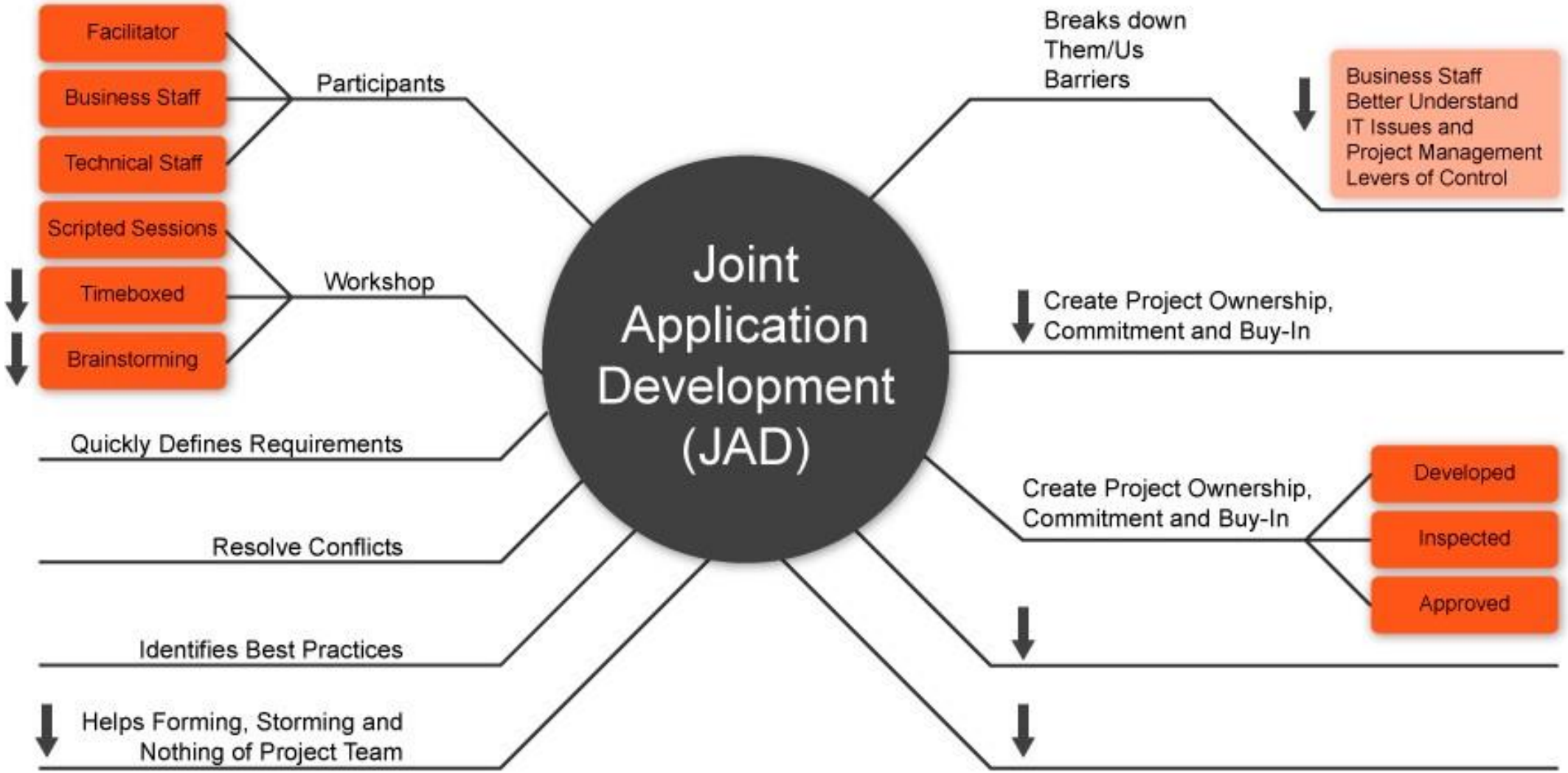
- **Incremental development:** Incremental development is very good because customer feedback is gained and based on this development team come up with new increments every few days after each iteration.
- **Simplicity:** Simplicity makes it easier to develop good quality code as well as to test and debug it.
- **Design:** Good quality design is important to develop a good quality software. So, everybody should design daily.
- **Integration testing:** It helps to identify bugs at the interfaces of different functionalities. Extreme programming suggests that the developers should achieve continuous integration by building and performing integration testing several times a day.

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.

Joint Application Development (JAD)

- Joint Application Development (JAD) is a management process that helps IT professionals to interact more effectively with users in order to develop information and technology solutions that really work.
- The **purpose** of JAD is to define the boundaries of the project, develop a solution, and monitor the project before it is completed.
- JAD is usually a strictly delineated 3-6 month project.
- For large-scale projects, it is recommended to apply an incremental approach to the project and use JAD for each increment.


Joint Application Development (JAD)
Facilitator
Business Staff
Technical Staff
Participants
Scripted Sessions
Timeboxed
Brainstorming
Workshop
Quickly Defines Requirements
Resolve Conflicts
Identifies Best Practices
Helps Forming, Storming and Nothing of Project Team
Breaks down Them/Us Barriers
Business Staff Better Understand IT Issues and Project Management Levers of Control
Create Project Ownership, Commitment and Buy-In
Create Project Ownership, Commitment and Buy-In
Developed
Inspected
Approved

- It is used for collecting information system requirements and reviewing system designs.
- It is a structured process in which users, managers, and analysts work together for several days in a series of intensive structured meetings run by a JAD session leader to specify or review system requirements.
- Here, people work together to agree on system requirements and design details, time and organizational resources are better managed.
- Group members are more likely to develop a shared understanding of what the IS is supposed to do.

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.

Benefits of Using JAD Model

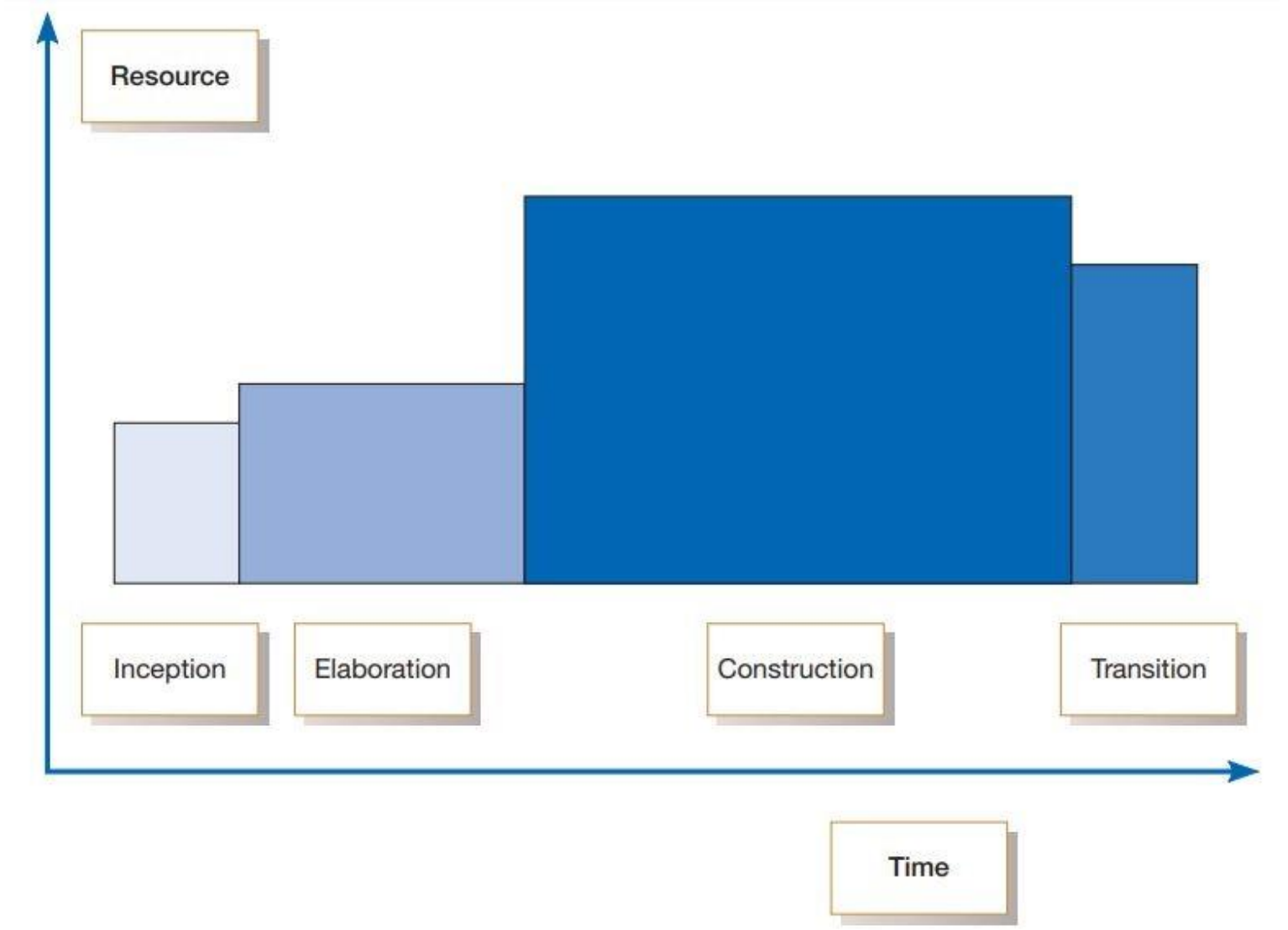
- **Improved Delivery Time**: The time required for developing a product using JAD model is lesser and efficient than that of other traditional models.
- **Cost Reduction**: Efficiently analyzing the requirements and facts with business executives and stakeholders will make less effort to develop the system and hence less cost will be required for the entire development process.
- **Better Understanding**: Since the entire requirement is analyzed by business executives, followed by a cautious choice of developers and team member who can professionally interact with each other better usually helps in understanding the product development better.
- **Improved Quality**: Since all the key decision makers and stakeholders of the project are involved in the development of the project so there is the least chance of error and hence the product quality becomes better and more accurate.

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.

- Service oriented architecture

The Origins of Software

After studying this chapter, you should be able to

- Explain outsourcing
- Describe six different sources of software
- Explain reuse and its role in software development

System Acquisition

- Outsourcing: If one organization develops or runs a computer application for another organization, that practice is called outsourcing.

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, and in-house developers

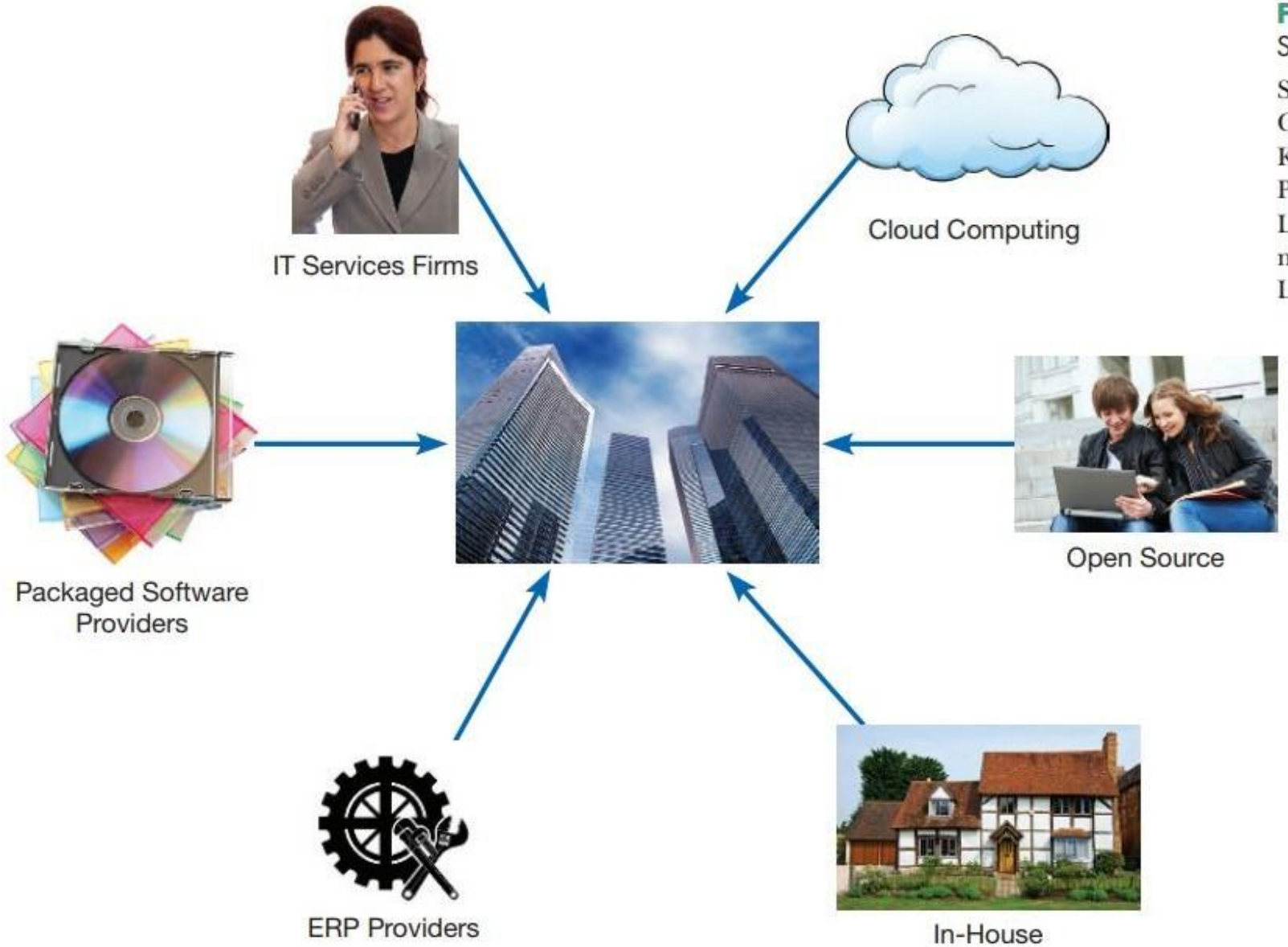


FIGURE 2-1
Sources of application software

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

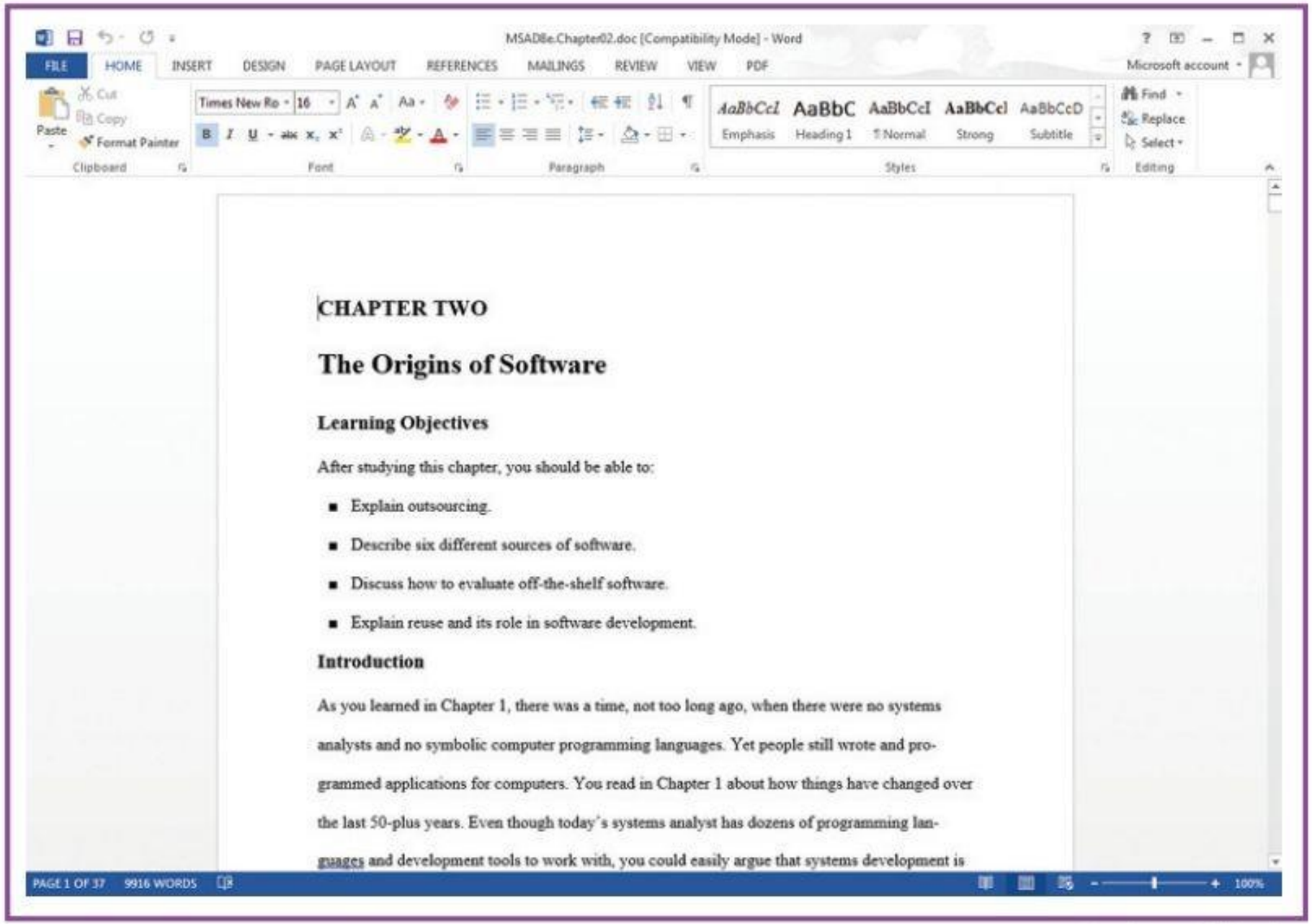
Information Technology Services Firms

- If a company needs an information system but does not have the expertise or the personnel to develop the system in-house, and a suitable off-the-shelf system is not available, 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.
- These firms employ people with expertise in the development of information systems. Their consultants may also have expertise in a given business area.

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.


MSAD8e.Chapter02.doc [Compatibility Mode] - Word
FILE
HOME
INSERT
DESIGN
PAGE LAYOUT
REFERENCES
MAILINGS
REVIEW
VIEW
PDF
Microsoft account
Paste
Cut
Copy
Format Painter
Times New Ro
16
Find
Replace
Select
Clipboard
Font
Paragraph
Styles
Editing
Emphasis
Heading 1
Normal
Strong
Subtitle
CHAPTER TWO
The Origins of Software
Learning Objectives
After studying this chapter, you should be able to:
Explain outsourcing.
Describe six different sources of software.
Discuss how to evaluate off-the-shelf software.
Explain reuse and its role in software development.
Introduction
As you learned in Chapter 1, there was a time, not too long ago, when there were no systems analysts and no symbolic computer programming languages. Yet people still wrote and programmed applications for computers. You read in Chapter 1 about how things have changed over the last 50-plus years. Even though today's systems analyst has dozens of programming languages and development tools to work with, you could easily argue that systems development is
PAGE 1 OF 37
9916 WORDS
100%

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.

- 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

- Cloud computing is the provision of computing resources, including applications, over the Internet, so customers do not have to invest in the computing infrastructure needed to run and maintain the resources.
- 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.

- A well-known example of cloud computing is Google Apps, where users can share and create documents, spreadsheets, and presentations

Open-Source Software

- 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 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.

In-House Development

- In-house development has become a progressively smaller piece of all systems development work that takes place in and for organizations.
- In-house development can lead to a larger maintenance burden than other development methods, such as packaged applications.
- Of course, in-house development need not entail 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 2-2 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

- **The use of previously written software resources, especially objects and components, in new applications.**
- Reuse is the use of previously written software resources in new applications.
- 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 anew 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.

The advantages of Reuse

- Increase software productivity
- Shorten software development time
- Improve software system interoperability
- Develop software with fewer people
- Move personnel more easily from project to project
- Reduce software development and maintenance costs
- Produce more standardized software
- Produce better quality software and provide a powerful competitive advantage

Managing the Information System Project

After studying this chapter, you should be able to

- Explain the process of managing an information systems project, including project initiation, project planning, project execution, and project closedown,
- Describe how to represent and schedule project plans using Gantt charts and network diagrams, and
- Explain how commercial project management software packages can be used to assist in representing and managing project schedules

Introduction to chapter

- In this chapter, we focus on the systems analyst's role in managing information systems projects and will refer to this role as the **project manager**.
- 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 chapter 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.

MANAGING THE INFORMATION SYSTEMS PROJECT

- 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.
- A project is a planned undertaking of a series of related activities to reach an objective that has a beginning and an end.
- Project management is a controlled process of initiating, planning, executing, and closing down a project.

FIGURE 3-4
A project manager juggles numerous activities

Source: ra2 studio/Fotolia

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

- The remainder of this chapter will focus on the project management process, which involves four phases:
- 1. Initiating the project
- 2. Planning the project
- 3. Executing the project
- 4. Closing down the project

INITIATING A PROJECT

- 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.
- 2. Establishing a relationship with the customer
- 3. Establishing the project initiation plan
- 4. Establishing Management Procedures
- 5. Establishing the Project Management Environment and Project Workbook
- 6. Developing the Project Charter

PLANNING THE PROJECT

- 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.

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

EXECUTING THE PROJECT PROJECT

- 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.

Activities During Project Execution

- 1. Executing the Baseline Project Plan
- 2. Monitoring Project Progress against the Baseline Project Plan
- 3. Managing Changes to the Baseline Project Plan
- 4. Maintaining the Project Workbook
- 5. Communicating the Project Status

CLOSING DOWN THE PROJECT

- 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 Project Closing

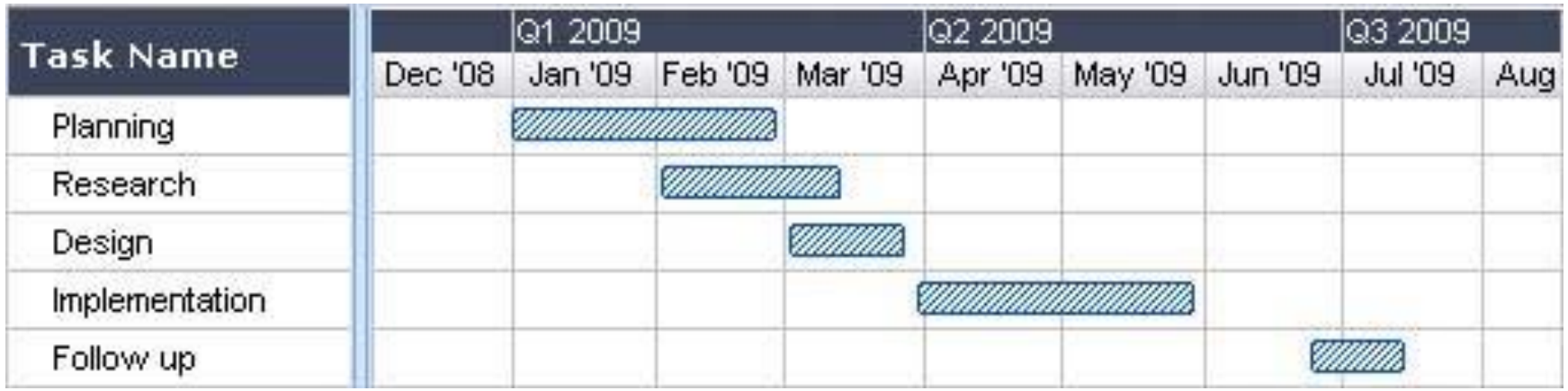
- 1. Closing down the project.
- 2. Conducting post project reviews.
- 3. Closing the customer contract.

REPRESENTING AND SCHEDULING PROJECT PLANS

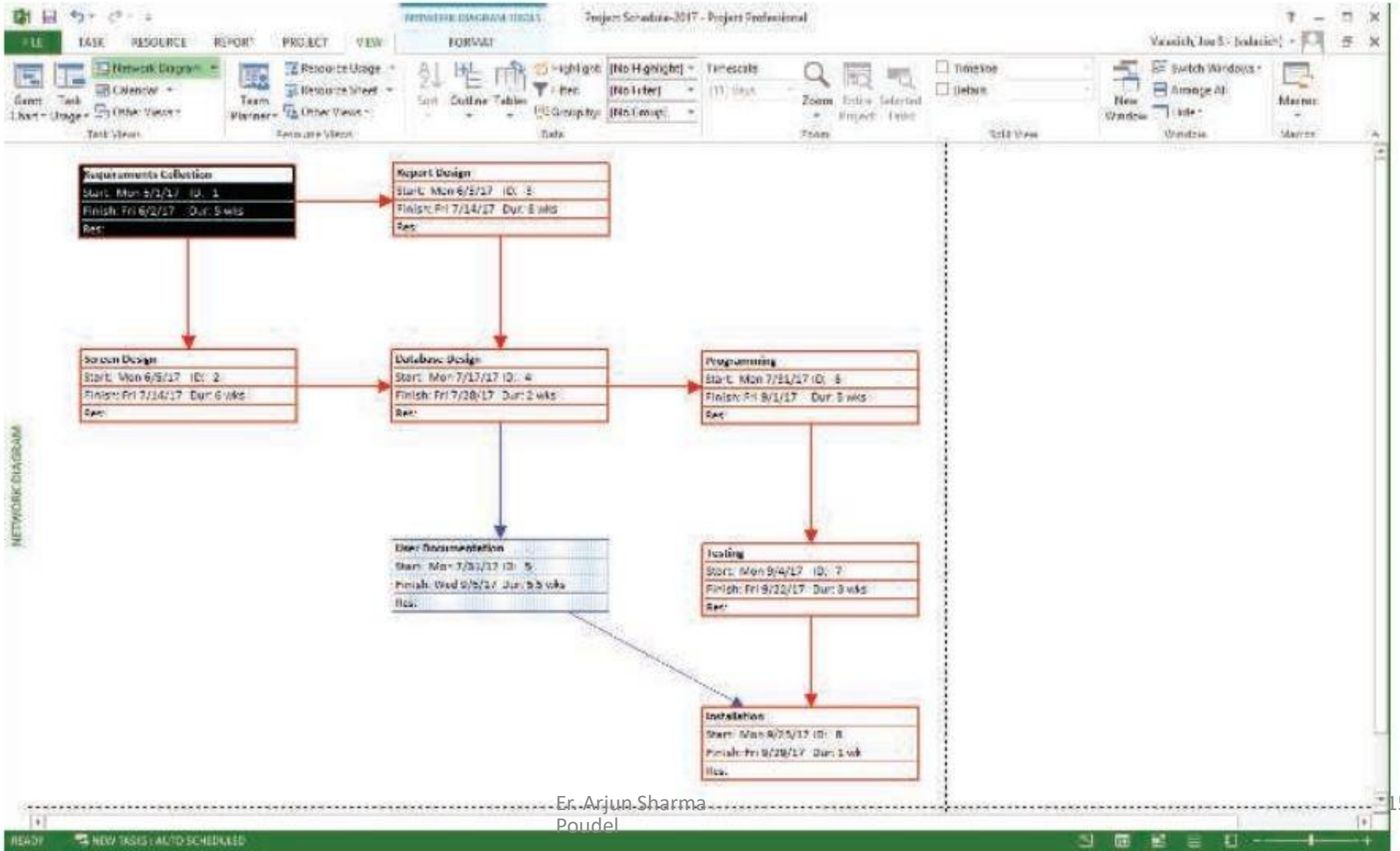
- A project manager has a wide variety of techniques available for depicting and documenting project plans.
- These planning documents can take the form of graphical or textual reports, although graphical reports have become most popular for depicting project plans.
- The most commonly used methods are 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.

Gantt Chart


Task Name
Q1 2009
Q2 2009
Q3 2009
Dec '08
Jan '09
Feb '09
Mar '09
Apr '09
May '09
Jun '09
Jul '09
Aug
Planning
Research
Design
Implementation
Follow up

Network Diagram



Key Differences Between Two Charts

- 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.

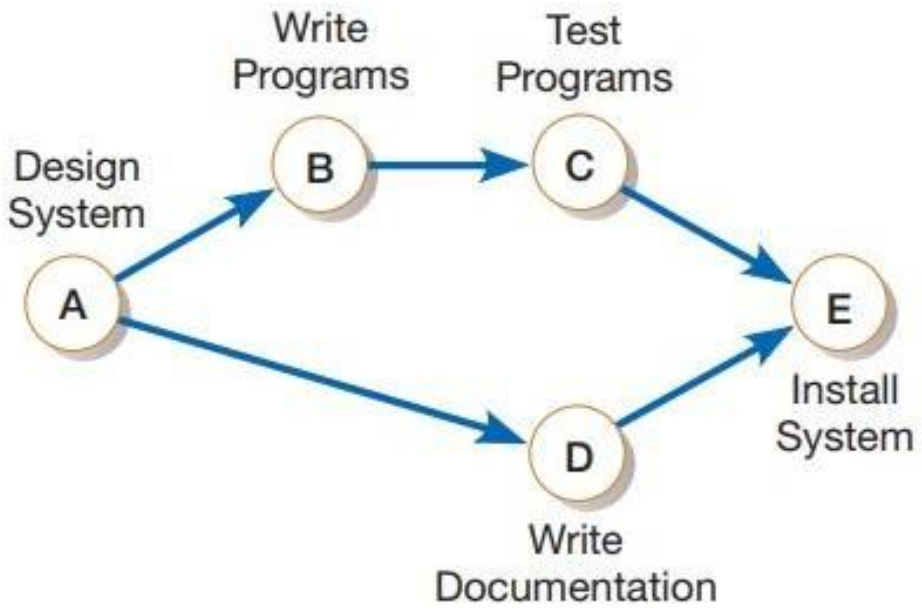


FIGURE 3-20
A network diagram showing activities (represented by circles) and sequence of those activities (represented by arrows)

CALCULATING EXPECTED TIME DURATIONS USING PERT

- 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 pessimistic (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.

End of Chapter One