Bachelor of Information Technology(BIT)

Course Title: BIT 253

System Analysis and Design

SHIB KUMAR MAHATO

Department of Computer Science and Information Technology

Masters of Science in Computer Science and Information Technology(M.Sc. CSIT)

in 2006 from CDCSIT- TU, Kirtipur

Faculty: Patan Multiple Campus

Email: shibmahto@gmail.com,Mobile: 977-9841420985

Systems Analysis and Design

UNIT 1

FOUNDATIONS FOR SYSTEMS DEVELOPMENT

Course Contents

Our Recommended books:

- Jeffery A. Hoffer, Joseph S. Valacich and Joey F. George, Modern Systems Analysis and Design,8th Edition, Pearson education Asia.

Course Contents

Unit 1: Foundations for Systems Development(12 Hrs.)

- System Concepts
- Introduction to Information System Analysis and Design
- Modern Approach of System Analysis and Design
- Information System and its Type
- Developing Information Systems and the Systems Development life Cycle(SDLC)
- The Heart of the Systems Development Process

Course Contents

- The Traditional Waterfall SDLC
- Approaches for Improving Development,Rapid Application Development (RAD), prototyping,Joint Application Development (JAD) and Computer Aided Software Engineering(CASE),agile methodologies,extreme programming,Service–Oriented Architecture and Object Oriented Analysis and Design.

Foundations for Systems Development

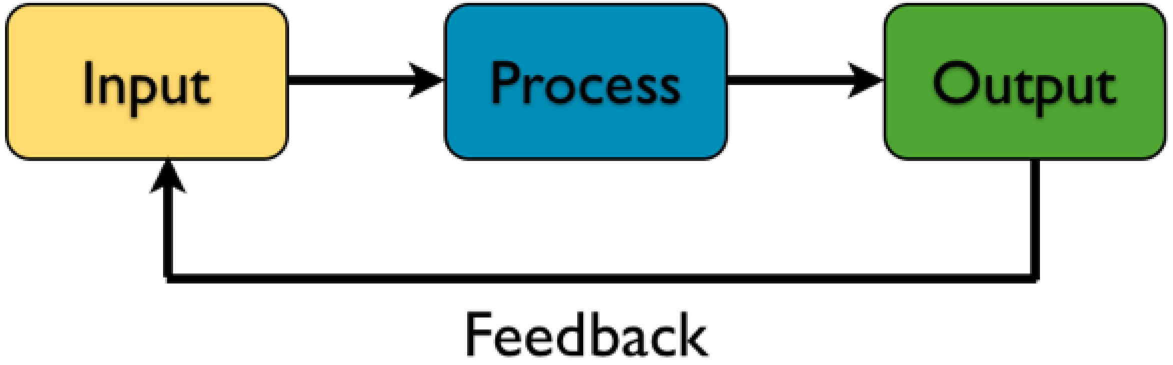
A. The systems development Environment

System concepts

A system is a collection of interrelated components towards some common goal to achieving input and producing output in an organized transformation process. This process is called dynamic process and it is also known as a system. A system consists of the following functions.

The systems development Environment

- Input
- Process
- Output



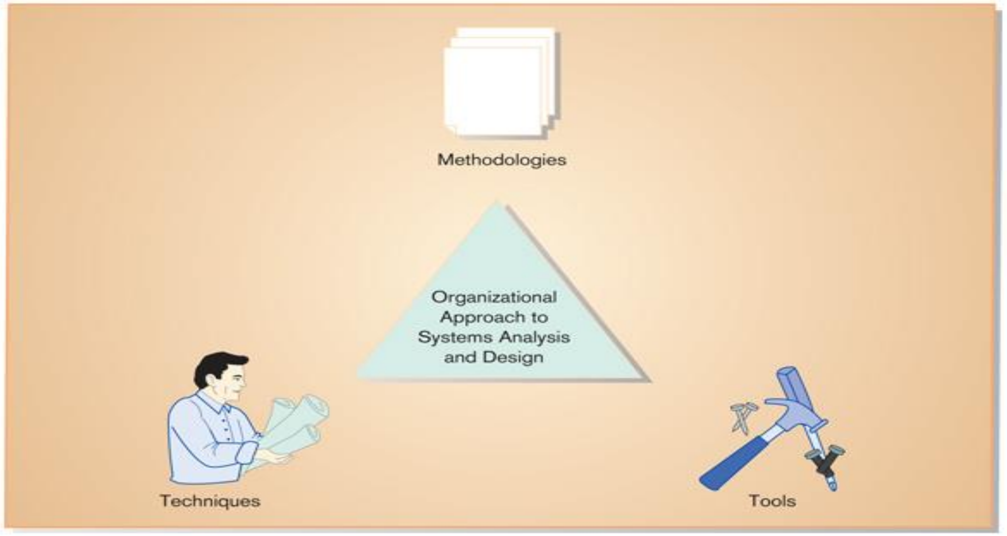
The system is doing correct work is called +ve feedback. The system is not doing correct work is called –ve feedback. Feedback is generally performance of the system.

Introduction

Information Systems Analysis and Design

- Complex organizational process.
- Used to develop and maintain computer-based information systems.
- Used by a team of business and systems professionals.

Introduction



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

A Modern Approach to Systems Analysis and Design

- 1950s: focus on efficient automation of existing processes
- 1960s: advent of 3GL, faster and more reliable computers
- 1970s: system development becomes more like an engineering discipline
- 1980s: major breakthrough with 4GL, CASE tools, object oriented methods
- 1990s:focus on system integration, GUI applications, client/server platforms, Internet
- The new century:Web application development,wireless PDAs,component-based applications

A Modern Approach to Systems Analysis and Design

Application Software

- Computer software designed to support organizational functions or processes.

Systems Analyst

- System Analyst is a person who is responsible for analyzing, designing and implementation of a system.
- Organizational role most responsible for analysis and design of information systems.

Information System

Introduction

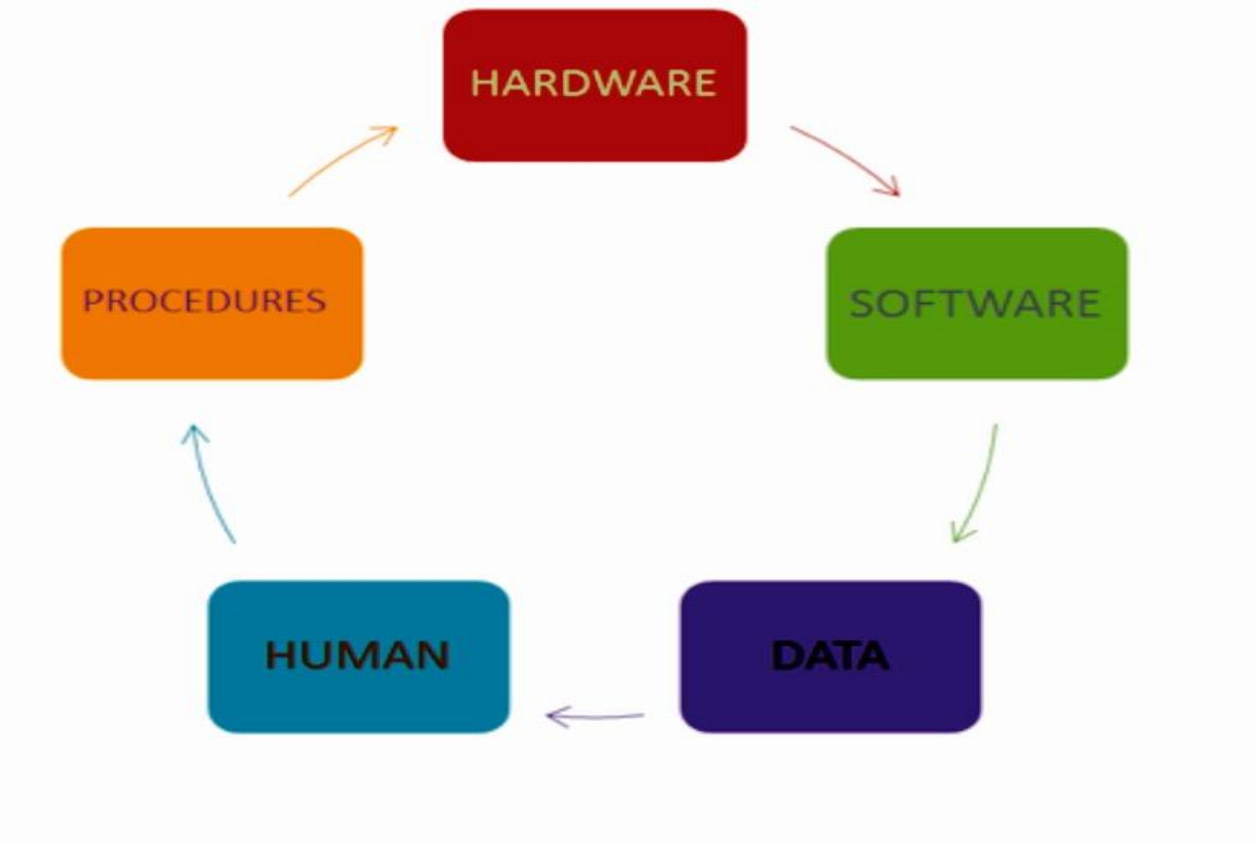
An information system that is the subsystem of the business. Specially, it is an arrangement of interdependent of human & machine components that interact to support the operation, managerial & decision-making information needs.

Resources or Components of Information System

The components of Information systems are

- Human or People
- Hardware
- Software
- Procedures or Methods
- Data &
- Networks

Resources or Components of Information System



Types of Information System and System Development

There are three types of Information System

1. Strategic Information System
2. Operational Information System
3. Tactical or Management Information System(MIS)

Types of Information System

1.Strategic Information System

This is the information needed for long range planning &directing the course the business should take. Strategic information system make decisions which can manage by senior managers. System which facilitates decisions for the strategic level of the organization requiring the use of knowledge, experience & expertise are called intelligent support system.

Types of Information System

Strategic Information System

- Decision support system(DSS)
- Artificial Intelligence(AI)
- Expert system(ES)

Types of Information System

Decision Support System(DSS)

DSS are designed to help organizational decision makers make decisions. A DSS provides an interactive environment in which decision makers can quickly manipulate data & models of business operations. It involves executive information system.

Types of Information System

Features of DSS

- Adaptability and flexibility
- High levels of Interactivity
- Efficiency and effectiveness
- Complete control by decision-makers

Types of Information System

Artificial Intelligence(AI)

AI is a branch of Computer Science whose goal is to design & develop machines with human intelligence.

Four goals of AI

- System that think like humans.
- System that act like humans.
- System that think rationally.(good sense or agreeable)
- System that act rationally.

Types of Information System

Expert System(ES)

Expert system is also known as knowledge based systems. An expert system is a programmed decision-making IS that captures & reproduces the knowledge & expert problem solving skills.

Types of Information System

Executive Information System

An **executive information system** (**EIS**) also known as an **executive support system** (**ESS**) is a type of management support system that facilitates and supports senior executive information and decision-making needs. It provides easy access to internal and external information relevant to organizational goals. It is commonly considered a specialized form of decision support system(DSS).

Strategic Information System

Some key features of an EIS are:

- Easy-to-use graphical interfaces
- Advanced data analysis tools
- Real-time access to internal and external data sources
- Customizable reporting
- Alerts and notifications and
- Support for mobile devices

Types of Information System

2.Operational Information System

Information system that exists on the bottom level of management is called operational information system. It helps to support day by day activities of organization.

- Transaction Processing System(TPS)
- Office Automation System(OAS)

Types of Information System

Transaction Processing System(TPS)

- A TPS is a software system that supports transaction processing.
- Automate handling of data about business activities (transactions)
- Process orientation

Types of Information System

Features of Transaction Processing System

- **Controlled access:** As a critical component of any business's information processing system, a robust TPS should provide secure controlled access for only authorized users and administrators.
- **Connection with external environments:** By definition, a TPS is designed to connect seamlessly with various external systems to distribute and receive information between customers, merchants, suppliers and where applicable, banks and creditors.

Types of Information System

Features of Transaction Processing System

- **Expedited response times:** For real-time TPS, fast response times are considered table stakes for businesses seeking to provide quick and easy transactions for their customers.

Types of Information System

Office Automation System(OAS)

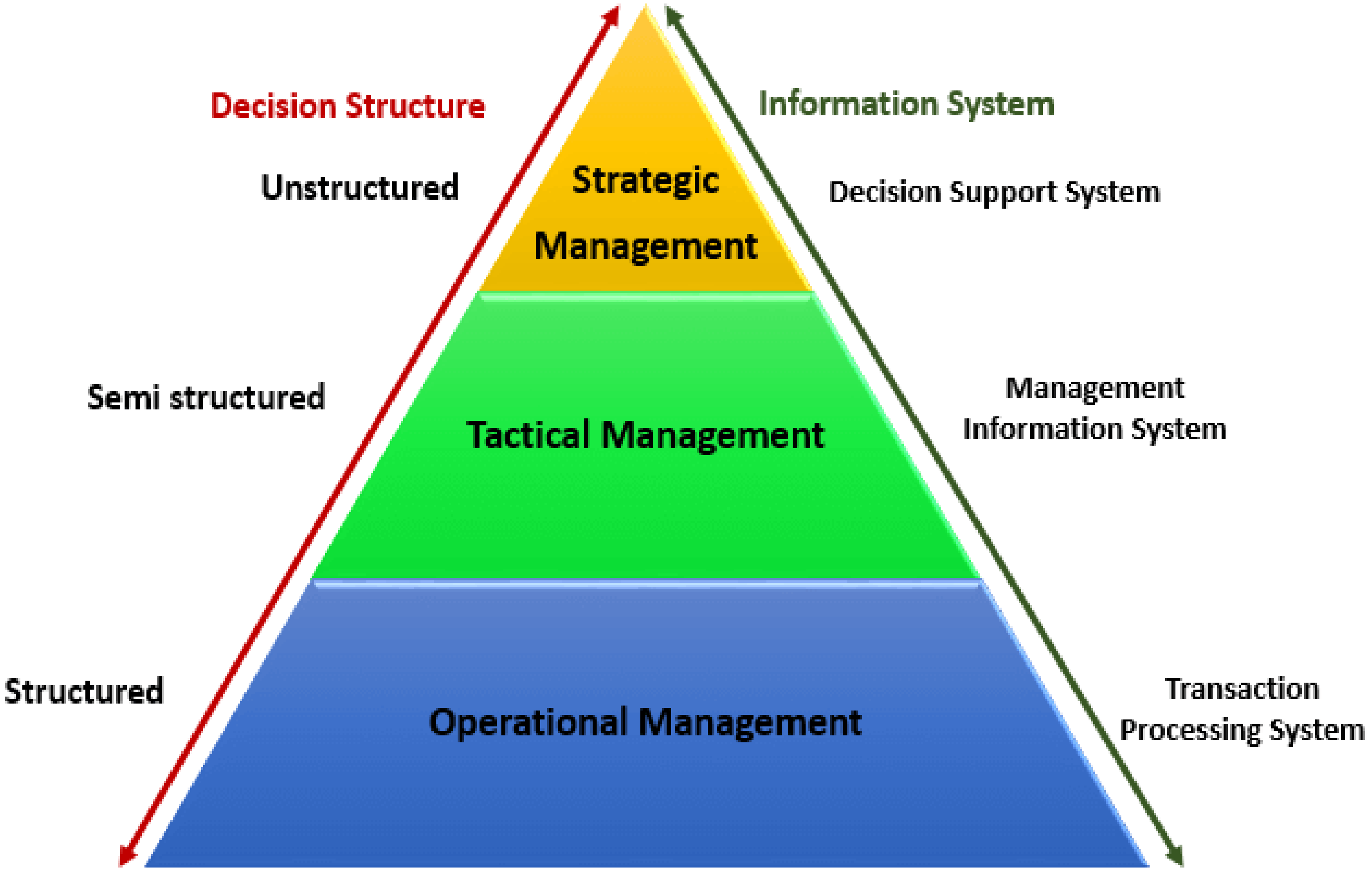
refers to mechanical, electrical and electronic devices to enhance communication in the workplace & increase the efficiency & productivity of knowledge workers.

Types of Information System

3.Tactical or Management Information System(MIS)

MIS assist managers in planning,organizing,staffing,co-ordinating, directing & controlling the operations of organization. It can be used short term business.

Pyramid Structure of IS



Developing Information Systems

System Development Methodology is a standard process followed in an organization to conduct all the steps necessary to analyze, design, implement, and maintain information systems.

Systems Development Life Cycle (SDLC)

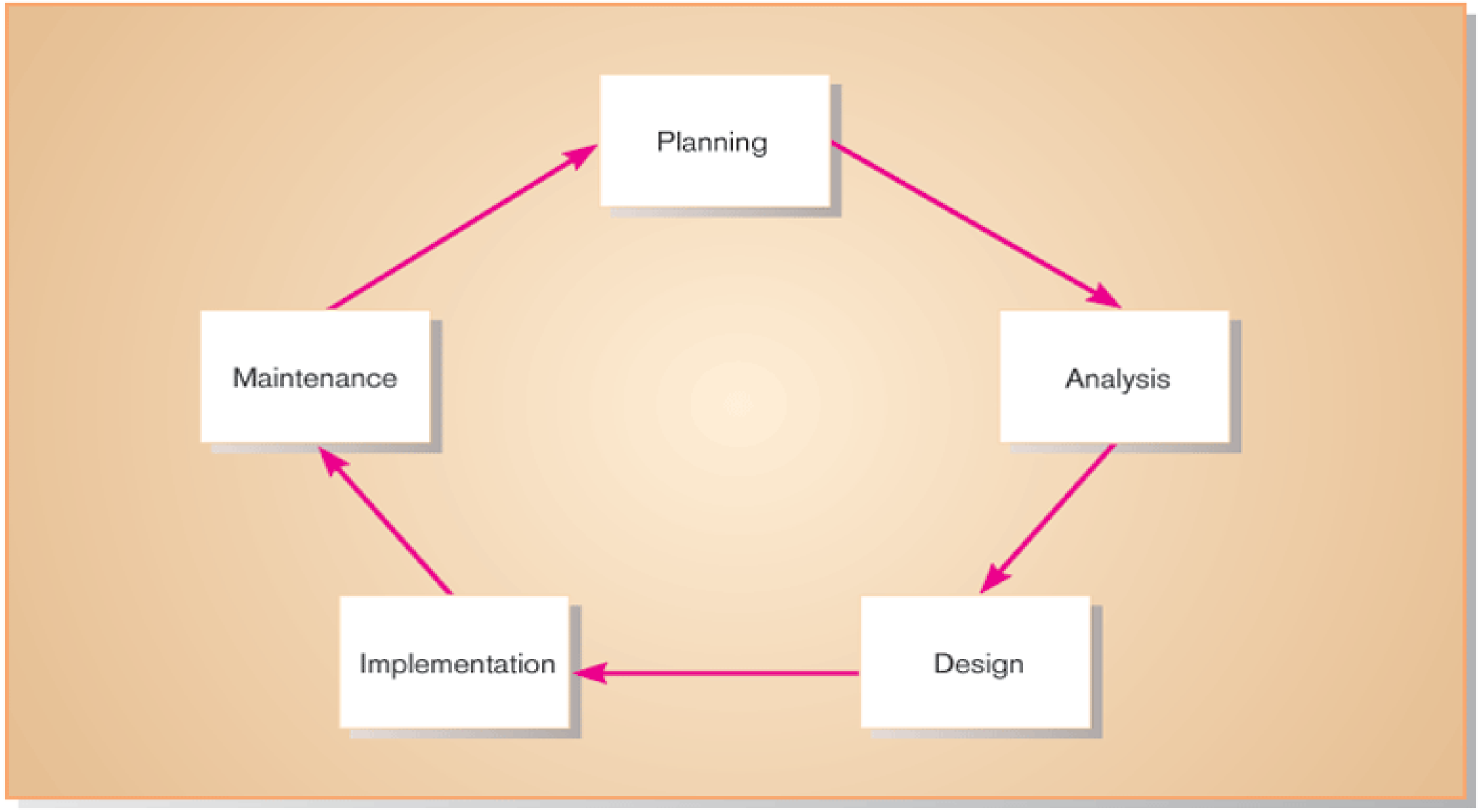
Traditional methodology used to develop, maintain and replace information systems.

Phases in SDLC:

1. Planning
2. Analysis
3. Design
4. Implementation
5. Maintenance

Systems Development Life Cycle (SDLC)

Figure 1-3 The systems development life cycle



Planning

Analysis

Design

Implementation

Maintenance

Systems Development Life Cycle (SDLC)

1. Planning – an organization’s total information system needs are identified, analyzed, prioritized and arranged.

- Define the problem and scope of existing system.
- Overview the new system and determine its objectives.
- Confirm project feasibility and produce the project schedule.
- During this phase threats, constraints, integration and security of system are also considered. A feasibility report for the entire project is created at the end of this phase.

Systems Development Life Cycle (SDLC)

2. Analysis and specification – system requirements are studied and structured.

- Gather, analyze and validate the information.
- Define the requirements and prototypes for new system.
- Evaluate the alternatives and prioritize the requirements.
- Examine the information needs of end-user and enhances the system goal.

Systems Development Life Cycle (SDLC)

3. Design

Includes the design of application, network, databases, user interfaces, and system interfaces. System design is basically concerned with the development of a good system. It provides mathematical foundation of a system.

- Create a training, maintenance, and operation plan.
- Review the proposed design.
- Ensure that the final design must meet the requirements.
- Finally, prepare a design document which will be used during next phases.

Systems Development Life Cycle (SDLC)

- **Logical design** – all functional features of the system chosen for development in analysis are described independently of any computer platform.
- **Physical design** – the logical specifications of the system from logical design are transformed into the technology-specific details from which all programming and system construction can be accomplished.

Systems Development Life Cycle (SDLC)

4.Implementation-Implementation means to develop a new system. Implementation includes coding, testing and installation.

- Implement the design into source code through coding.
- Combine all the modules together into training environment that detects errors and defects. A test report which contains errors is prepared through test plan that includes test related tasks such as test case generation, testing criteria and resource allocation for testing. Integrate the information system into its environment and install the new system.

Systems Development Life Cycle (SDLC)

5.Maintenance – an information system is systematically repaired and improved.

- It also includes handling the residual errors and resolve any issues that may exist in the system even after the testing phase.
- Maintenance and support may be needed for a longer time for large systems and for a short time for smaller systems. Maintenance is expensive.

The Heart of the Systems Development Process

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

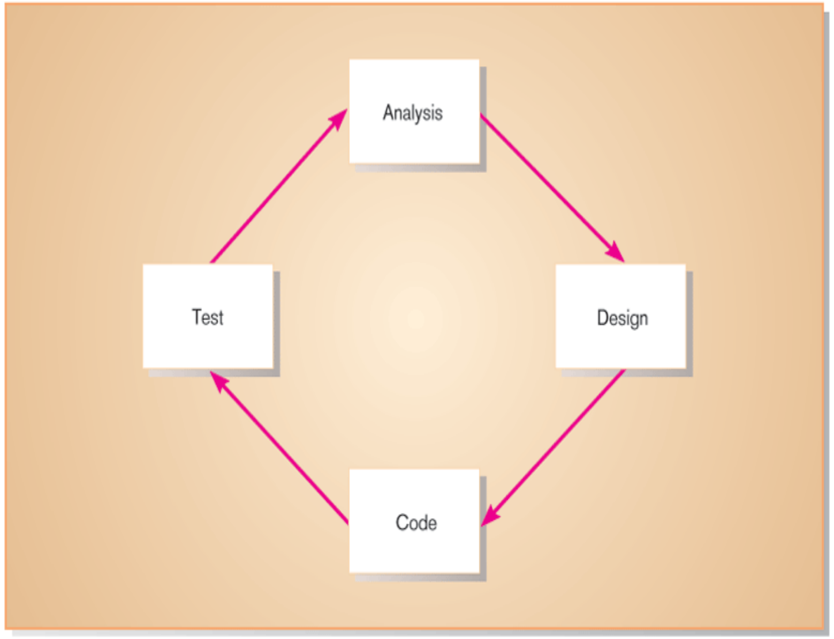
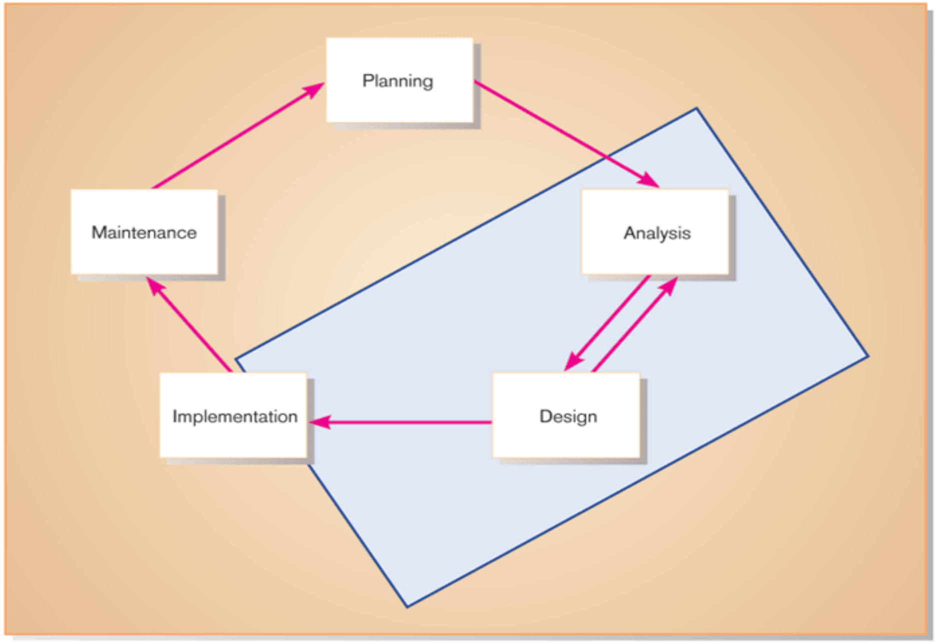


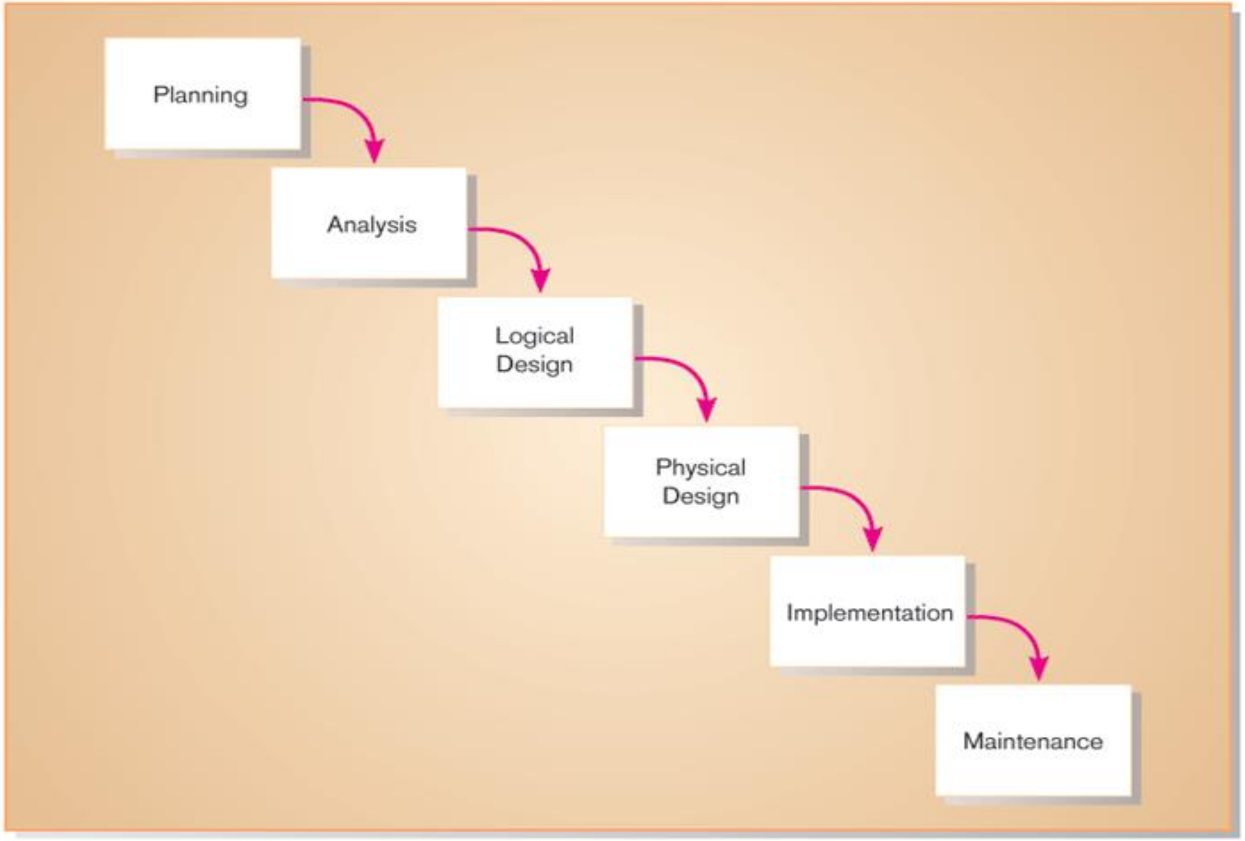
Figure 1-9 The heart of systems development



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

Traditional Waterfall SDLC

Figure 1-10 A traditional waterfall SDLC



One phase begins when another completes, little backtracking and looping

Waterfall Model

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development.

Waterfall Model

Some of the major advantages of the Waterfall Model are as follows −

- 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.

Waterfall Model

- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

Waterfall Model

The major disadvantages of the Waterfall Model are as follows −

- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.

Waterfall Model

- It is difficult to measure progress within stages.
- Cannot accommodate changing requirements.
- Adjusting scope during the life cycle can end a project.

Different Approaches to Improving Development

- Prototyping
- Computer-Aided Software Engineering (CASE) Tools
- Joint Application Design (JAD)

Different Approaches to Improving Development

- Rapid Application Development (RAD)
- Agile Methodologies
- Extreme Programming

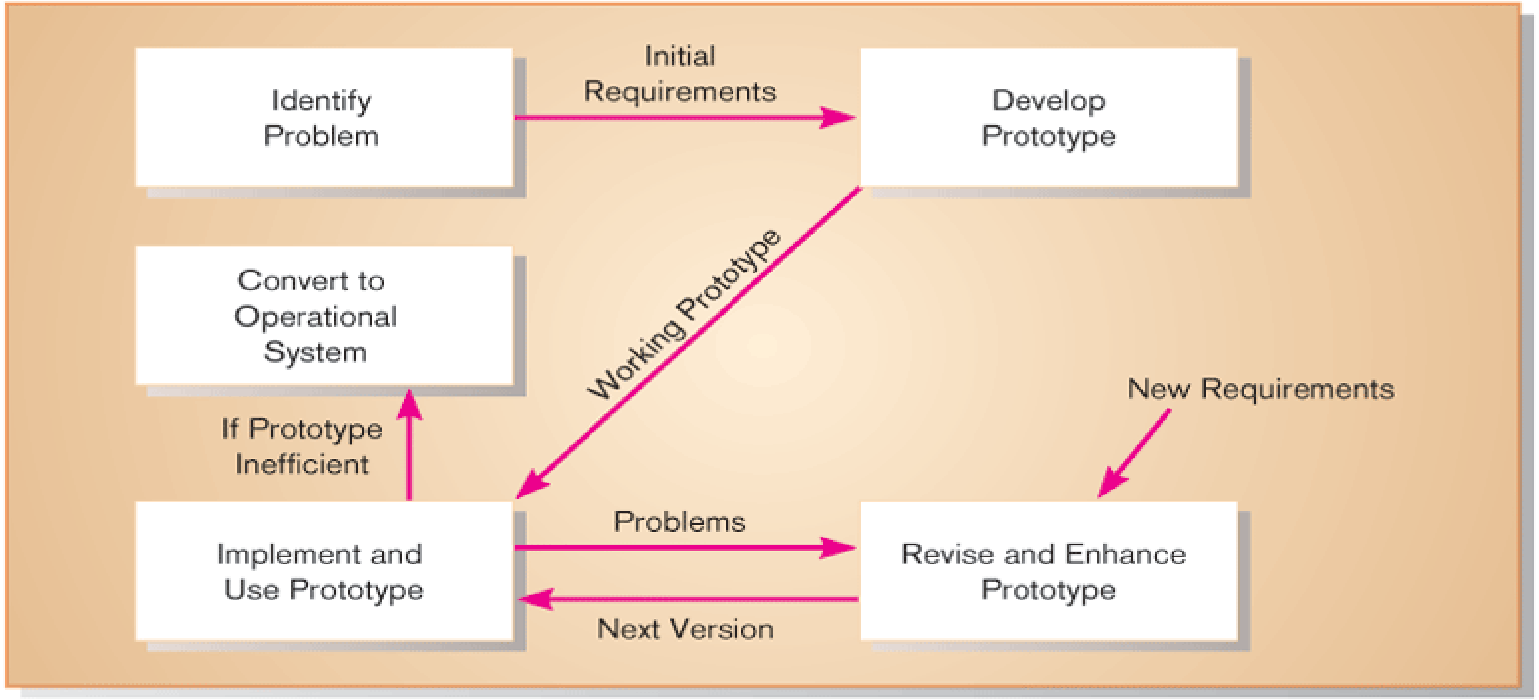
Prototyping

An iterative process of systems development in which requirements are converted to a working system that is continually revised through close work between an analyst & users. You can build a prototype with any language or development tool but special prototyping tools have been developed to simplify the process.

Prototyping is a form of rapid application development(RAD).

Prototyping (Cont.)

Figure 1-11 The prototyping methodology



Source: Adapted from "Prototyping: The New Paradigm for Systems Development," by J. D. Naumann and A. M. Jenkins, *MIS Quarterly* 6 (3): 29–44.

Computer-Aided Software Engineering (CASE) Tools

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. Its objective is increasing productivity & improving the overall quality of systems. We use tools to build Internet & Electronic Commerce Applications.

Computer-Aided Software Engineering (CASE) Tools

The general types of CASE tools are

1.Diagramming tools enable graphical representation.

2.Computer displays and report generators help prototype how systems “look and feel”.

3.Analysis tools automatically check for consistency in diagrams, forms and reports.

Computer-Aided Software Engineering (CASE) Tools (Cont.)

4. Central repository for integrated storage of diagrams, reports and project management specifications.
5. Documentation generators standardize technical and user documentation.
6. Code generators enable automatic generation of programs and database code directly from design documents, diagrams, forms and reports.

Joint Application Design (JAD)

In the late 1970’s system development at IBM developed a new process for collecting information system requirements & reviewing system designs. The process is called JAD.The basic idea behind JAD is to bring structure to the requirements determination phase of analysis & to the reviews that occurs as part of design.

- Structured process involving users, analysts and managers.
- Several-day intensive workgroup sessions.
- Purpose: to specify or review system requirements.

Rapid Application Development (RAD)

Methodology to radically decrease design and implementation time.

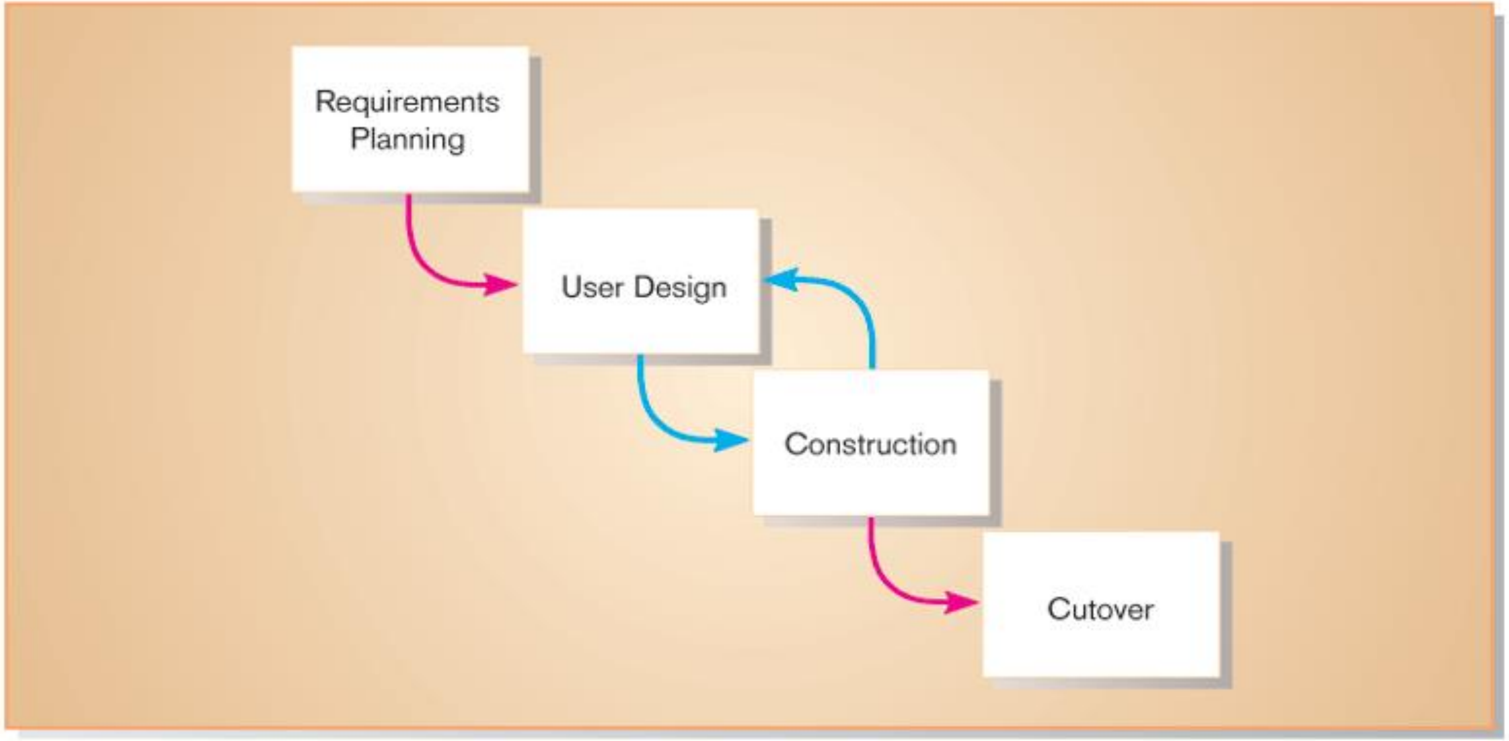
Involves: extensive user involvement, prototyping, JAD sessions, integrated CASE tools and code generators. **Rapid application development**(**RAD**) also called **rapid application building**(**RAB**),is both a general term for adaptive software development approaches and the name for James Martin's method of rapid development. In general, RAD approaches to software development put less emphasis on planning and more emphasis on an adaptive process.

Rapid Application Development (RAD) Contd.

RAD is especially well suited for (although not limited to) developing software that is driven by user interface requirements.

Rapid Application Development (RAD) (Cont.)

Figure 1-12 RAD life cycle



Agile Methodologies

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.

Agile Methodology

In software development, agile(sometimes written Agile) practices include requirements discovery and solutions improvement through the collaborative effort of self-organizing and cross-functional teams with their customers/end users, adaptive planning, evolutionary development, early delivery, continual improvement, and flexible responses to changes in requirements, capacity and understanding of the problems to be solved. Popularized in the 2001 Manifesto for Agile Software Development, these values and principles were derived from and underpin a broad range of software development frameworks.

Service-oriented architecture (SOA)

Service-oriented architecture (SOA) is a flexible set of design principles used during the phases of systems development and integration in computing. A system based on a SOA will package functionality as a suite of interoperable services that can be used within multiple separate systems from several business domains. A service-oriented architecture is essentially a collection of services. These services communicate with each other. The communication can involve either simple data passing or it could involve two or more services

Extreme Programming

- Short, incremental development cycles.
- Automated tests.
- Two-person programming teams.

Extreme Programming (Cont.)

□ Coding and testing operate together.

Advantages:

- Communication between developers.
- High level of productivity.
- High-quality code.

Object-Oriented Analysis and Design (OOAD)

Based on objects rather than data or processes.

- **Object**: a structure encapsulating attributes and behaviors of a real-world entity.
- **Object class**: a logical grouping of objects sharing the same attributes and behaviors.
- **Inheritance**: hierarchical arrangement of classes enable subclasses to inherit properties of super classes.

B. THE ORIGINS OF SOFTWARE

Course Contents

- 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

- There are various sources of software for organizations.
- There are criteria to evaluate software from different sources.
- The impact of reuse on software development.

Systems Acquisition: Outsourcing (Cont.)

- **Outsourcing**: Turning over responsibility of some or all of an organization's information systems applications and operations to an outside firm.

Systems Acquisition: Outsourcing (Cont.)

❑ Outsourcing Examples

- A company that runs payroll applications for clients.
- A company that runs your applications at your site.

Outsourcing (Cont.)

❑Reasons to outsource

- Cost-effective.
- Take advantage of economies of scale.
- Free up internal resources.

Outsourcing (Cont.)

- Reduce time to market.
- Increase process efficiencies.
- System development is a non-core activity for the organization.

Sources of Software

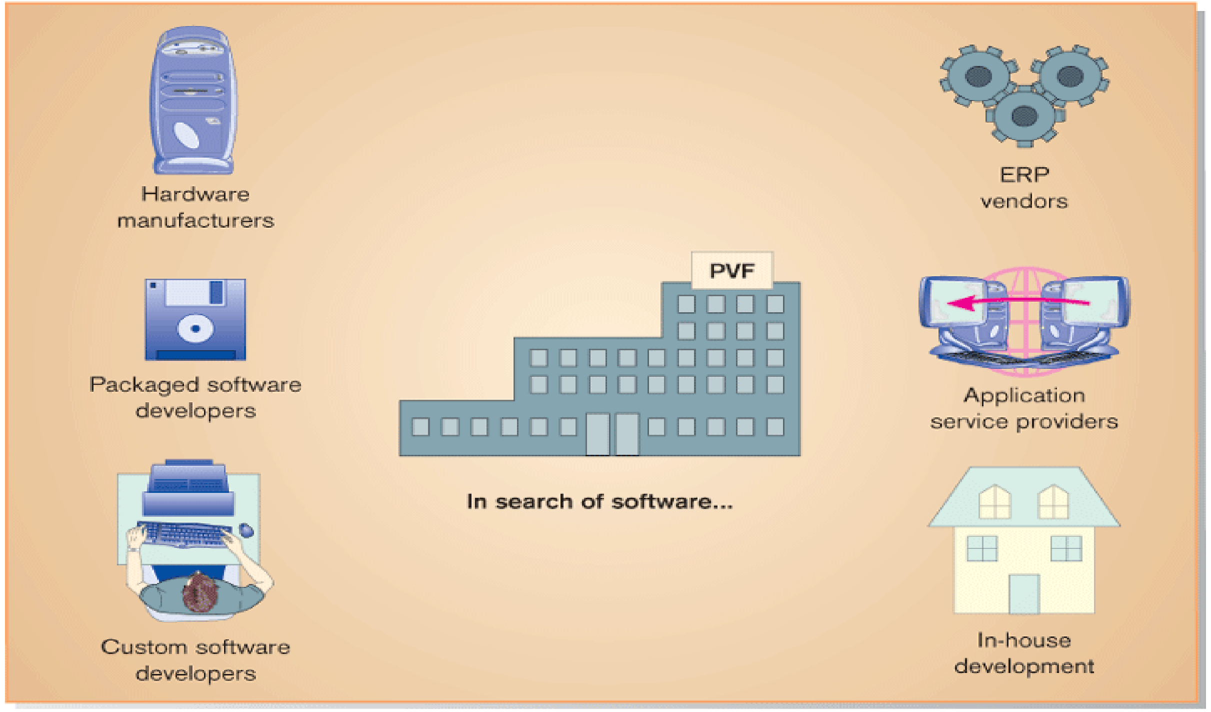
- ❑Information technology services firm.
- ❑Packaged software producers.
- ❑Enterprise-wide solutions.

Sources of Software (Cont.)

- ❑Application service providers (ASPs)
- ❑Open source software.
- ❑In-house developers.

Sources of Software (Cont.)

Figure 2-1 Sources of application software


Hardware manufacturers
ERP vendors
Packaged software developers
PVF
In search of software...
Application service providers
Custom software developers
In-house development

Information Technology (IT) Services Firms

- Help companies develop custom information systems for internal use.
- Develop, host and run applications for customers.
- Provide other services.

Packaged Software Producers

- Serve many market segments.
- Software ranges from broad-based packages (i.e. general ledger) to niche packages (i.e. day care management).

Packaged Software Producers

- Prepackaged software is off-the-shelf software.
- Software runs on microcomputers to large mainframes.

Packaged Software Producers (Cont.)

- Prepackaged software is turnkey software (i.e. not customizable).
- Off-the-shelf software at best meets 70 percent of organization’s needs.

Packaged Software Producers (Cont.)

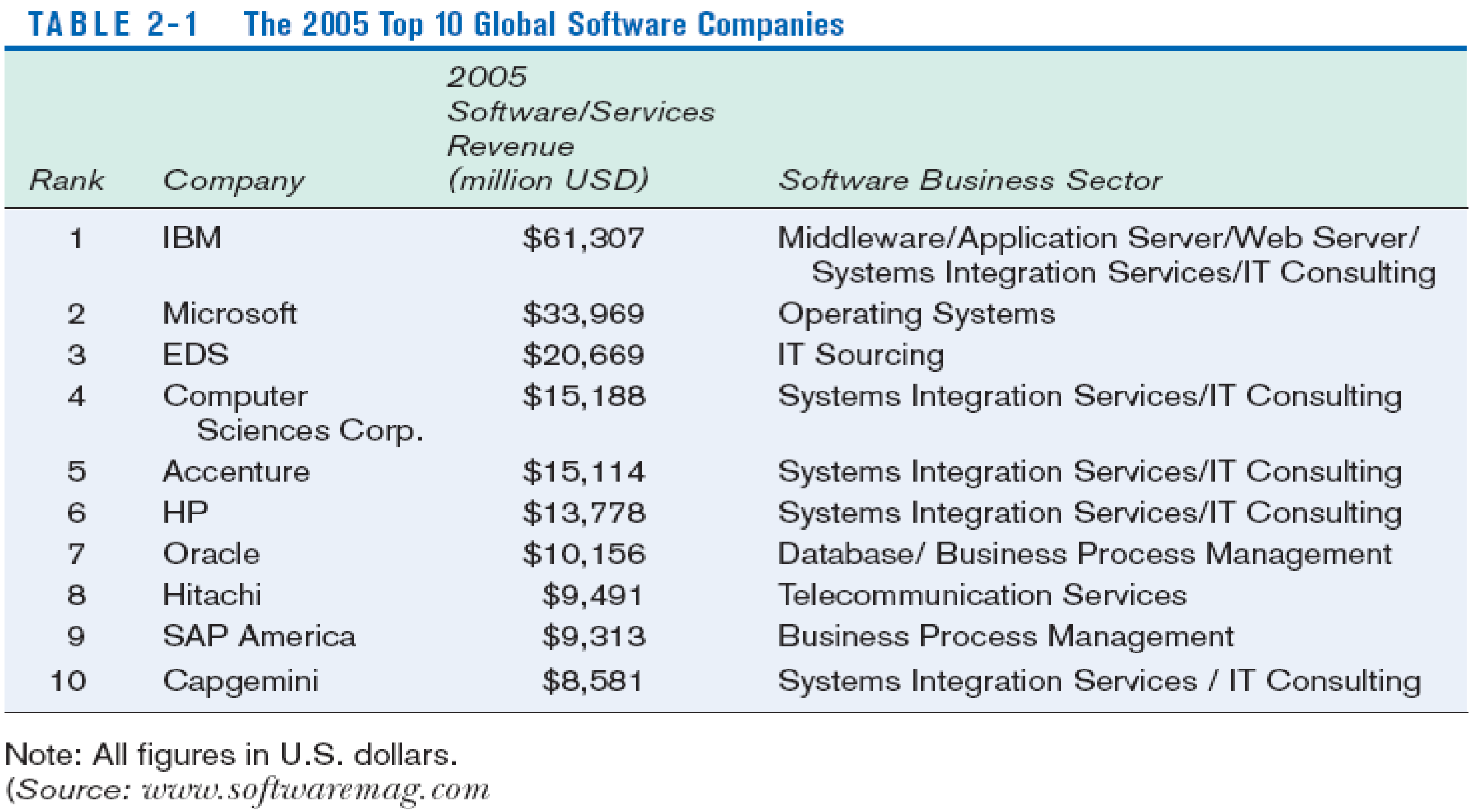


TABLE 2-1 The 2005 Top 10 Global Software Companies

Rank	Company	2005 Software/Services Revenue (million USD)	Software Business Sector
1	IBM	$61,307	Middleware/Application Server/Web Server/ Systems Integration Services/IT Consulting
2	Microsoft	$33,969	Operating Systems
3	EDS	$20,669	IT Sourcing
4	Computer Sciences Corp.	$15,188	Systems Integration Services/IT Consulting
5	Accenture	$15,114	Systems Integration Services/IT Consulting
6	HP	$13,778	Systems Integration Services/IT Consulting
7	Oracle	$10,156	Database/ Business Process Management
8	Hitachi	$9,491	Telecommunication Services
9	SAP America	$9,313	Business Process Management
10	Capgemini	$8,581	Systems Integration Services / IT Consulting

Note: All figures in U.S. dollars.
(*Source: www.softwaremag.com*

Open Source Software

- Freely available including source code.
- Developed by a community of interested people.
- Performs the same functions as commercial software.
- Examples: Linux, MySQL, Firefox.

In-House Development

- If sufficient system development expertise with the chosen platform exists in-house, then some or all of the system can be developed by the organization’s own staff.
- Hybrid solutions involving some purchased and some in-house components are common.

Off-the-Shelf Software

Most common criteria for selecting:

- **Cost**: comparing the cost of developing the same system in-house with the cost of purchasing or licensing the software package.

Off-the-Shelf Software

- **Functionality**: the tasks that the software can perform and the mandatory, essential and desired system features.

Off-the-Shelf Software (Cont.)

- **Vendor support**: whether or how much support the vendor can provide and at what cost.

Off-the-Shelf Software (Cont.)

- **Viability of vendor**: can the software adapt to changes in systems software and hardware.

Off-the-Shelf Software (Cont.)

- **Flexibility**: how easy it is to customize the software.
- **Documentation: is** the user’s manual and technical documentation understandable and up-to-date.

Reuse

The use of previously written software resources, especially objects and components in new applications.

Reuse (Cont.)

Commonly applied to two different development technologies:

- Object-oriented development
- Component-based development

Reuse (Cont.)

Object-oriented development

- Object, class, encapsulates data and behavior of common organizational entities (e.g. employees).

Component-based development

- Components can be as small as objects or as large as pieces of software that handle single business functions.

Reuse (Cont.)

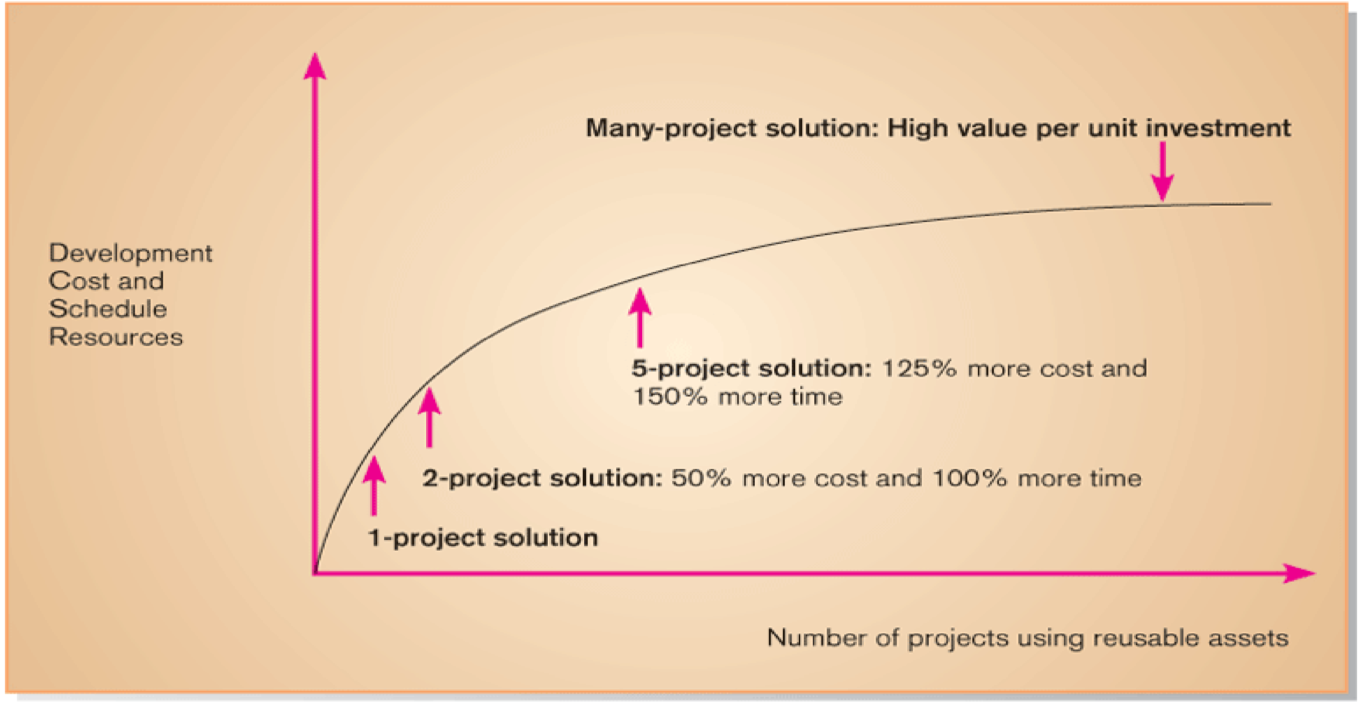
Object-oriented development reuse is using object classes in more than one application (e.g. Employee).

Reuse (Cont.)

Component-based development reuse is the assembly of an application from many different components at many different levels of complexity and size (e.g. Currency conversion).

Costs and Benefits of Reuse

Figure 2-4 Investments necessary to achieve reusable components



Source: Royce, 1998, used by permission.

Approaches to Reuse

- **Ad-hoc**: individuals are free to find or develop reusable assets on their own.
- **Facilitated**: developers are encouraged to practice reuse.

Approaches to Reuse (Cont.)

- **Managed**: the development, sharing, and adoption of reusable assets is mandated.
- **Designed**: mandating assets be designed for reuse as they are being designed for specific applications.

C. Managing the
Information Systems Project

Course Contents

- Explain the process of managing an information system project.
- Describe the skills required to be an effective project manager.
- List and describe the skills and activities of a project manager during project initiation, project execution and project closedown.

Course Contents

- Explain what is meant by critical path scheduling and describe the process of creating Gantt charts and Network diagrams.
- Explain how commercial project management software packages can be used to assist in representing and managing project schedules.

Managing the Information Systems Project

A **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.

Managing the
Information Systems Project

- The project manager’s environment is one of continual change and problem solving.
- The project manager’s understanding of the project management process is critical.
- Juanita Lopez and Chris Martin are the project managers for Pine Valley Furniture.

Managing the Information Systems Project (Cont.)

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

Deliverable – an end product of an SDLC phase

Managing the Information Systems Project (Cont.)

System Service Request (**SSR**) – a standard form for requesting or proposing systems development work within an organization

- It includes the contact person, a problem statement, a service request statement and liaison contact information.

Managing the
Information Systems Project

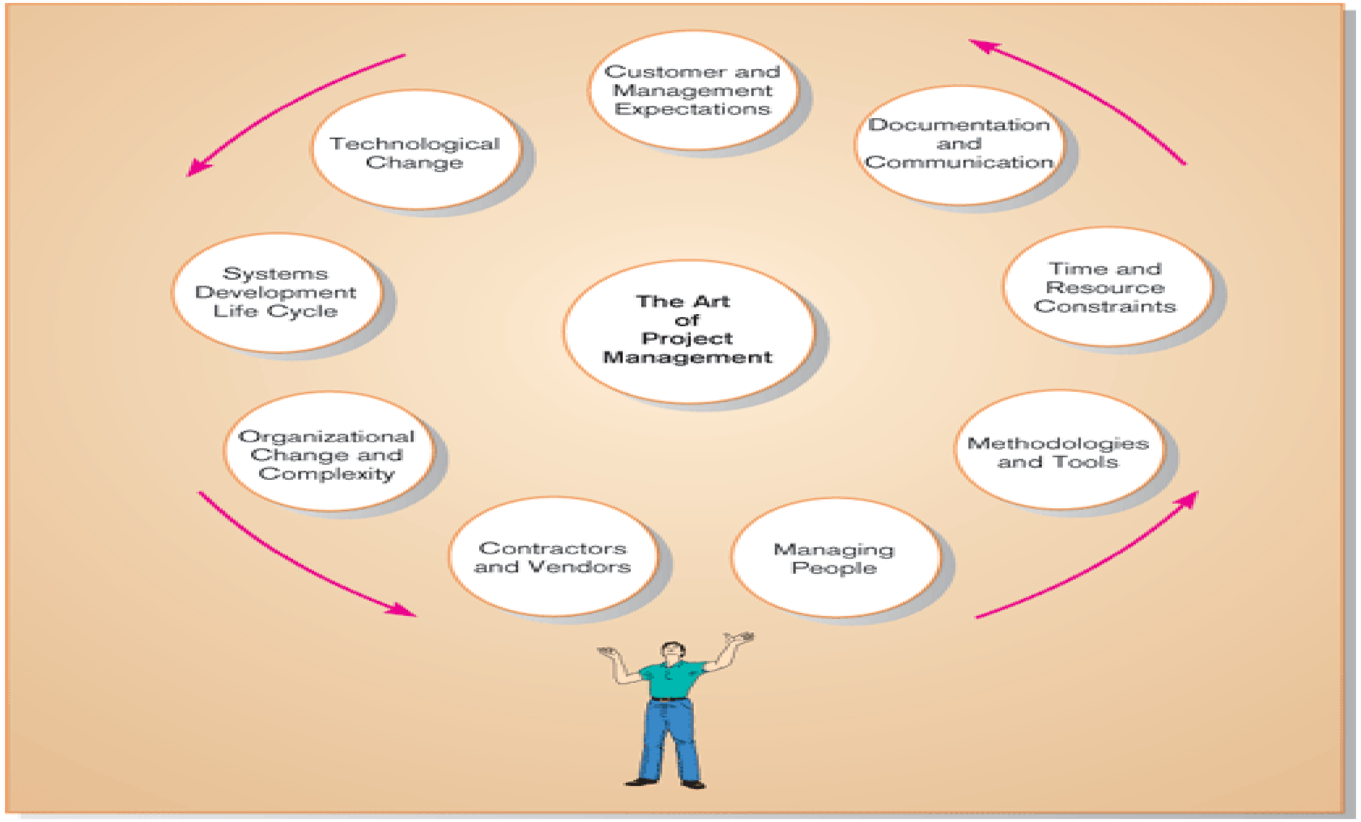
Project management: a controlled process of initiating, planning, executing and closing down a project.

❑ Project Management Process

- Initiating the Project.
- Planning the Project.
- Executing the Project.
- Closing down the Project.

Project Management Activities

Figure 3-4 A project manager juggles numerous activities.


Customer and Management Expectations
Documentation and Communication
Time and Resource Constraints
Methodologies and Tools
Managing People
Contractors and Vendors
Organizational Change and Complexity
Systems Development Life Cycle
Technological Change
The Art of Project Management

Initiating a Project

Project 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

Initiating a Project (Cont.)

The process of project initiation includes, establishing and developing:

- ❑An initiation team
  - Organize an initial core of project team members to assist in accomplishing project initiation
- ❑A relationship with the customer
- ❑A project initiation plan
  - Define activities required to organize team

Initiating a Project (Cont.)

- ❑ Management procedures
  - Develop team communication and reporting procedures.
- ❑ A project management environment
  - Collect and organize tools that will be used to manage project.

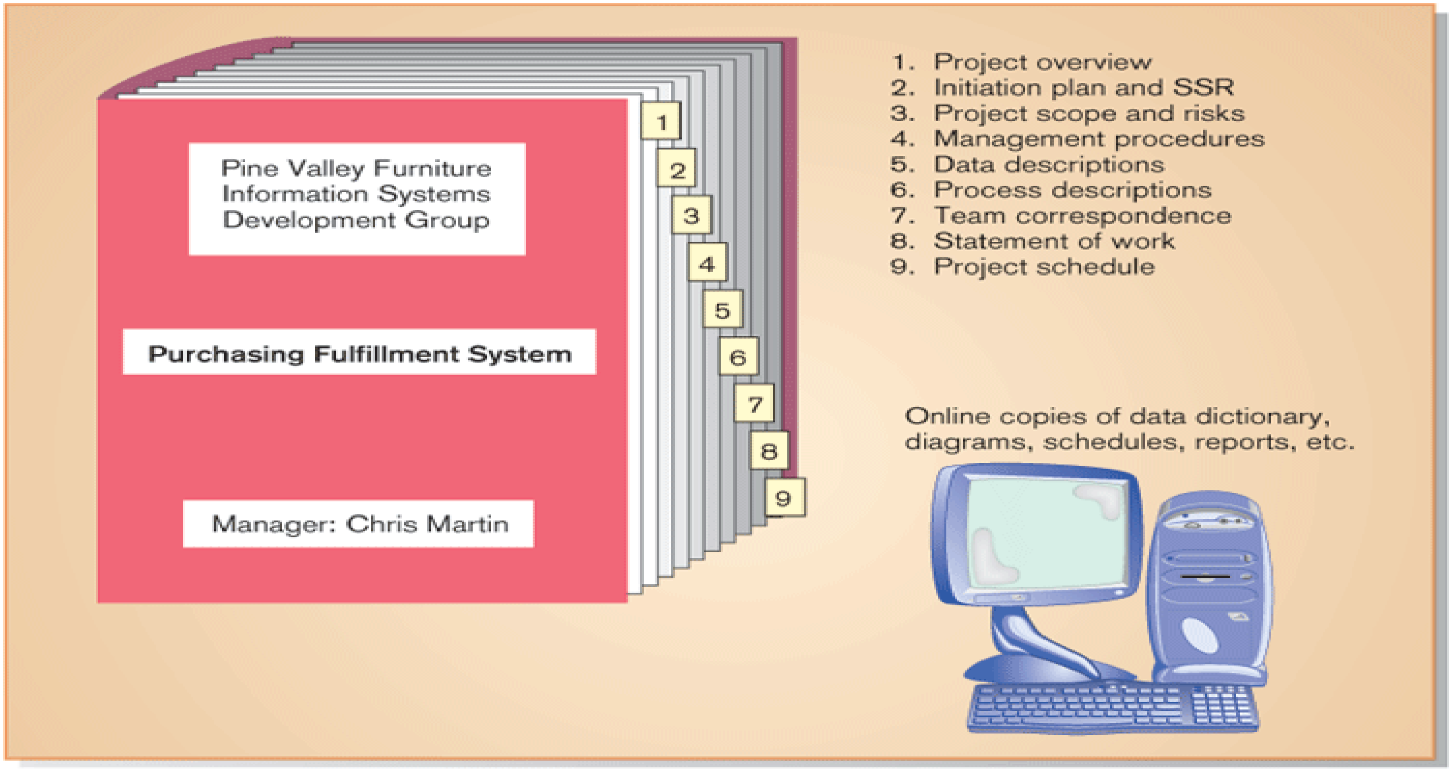
Initiating a Project (Cont.)

□ A project workbook

- **Project workbook** – an online or hard-copy repository for all project correspondence, inputs, outputs, deliverables, procedures and standards that are used.

Project Workbook

Figure 3-6 The project workbook for the Purchase Fulfillment System project contains nine key documents in both hard-copy and electronic form.



Pine Valley Furniture Information Systems Development Group
Purchasing Fulfillment System
Manager: Chris Martin
1
2
3
4
5
6
7
8
9

1. Project overview
2. Initiation plan and SSR
3. Project scope and risks
4. Management procedures
5. Data descriptions
6. Process descriptions
7. Team correspondence
8. Statement of work
9. Project schedule

Online copies of data dictionary, diagrams, schedules, reports, etc.

Initiating a Project (Cont.)

❑A project charter

❑**Project charter** – a short, high-level document prepared for both internal and external stakeholders.

- It formally announces the establishment of the project.
- It briefly describes its objectives, key assumptions and stakeholders.

Planning the Project

Project 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

Planning the Project (Cont.)

- Describe project scope, alternatives and feasibility:
- What problems or opportunities 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?

Planning the Project (Cont.)

Divide the project into manageable tasks:

- **Work Breakdown Structure** (**WBS**) – the process of dividing the project into manageable tasks and logically ordering them.
- **Gantt chart** – a graphical representation of a project that shows each task as a horizontal bar whose length is proportional to its time for completion.

Planning the Project (Cont.)

❑ The characteristics of a *task:*

- Can be done by one person or a well-defined group
- Has a single and identifiable deliverable
- Has a known method or technique
- Has well-accepted predecessor and successor steps
- Is measurable so that percent completed can be determined

Planning the Project (Cont.)

❑ Estimate resources and create a resource plan:

- **Constructive Cost Model** (**COCOMO**) – a widely used method which uses parameters that are derived from prior projects of differing complexity.
- **COCOMO** uses these different parameters to predict human resource requirements for basic, intermediate and very complex systems.

Planning the Project (Cont.)

❑ Develop a preliminary schedule:

- **Network diagram** – depicts project tasks and their interrelationships.

Planning the Project (Cont.)

Develop a communication plan:

- 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?

Planning the Project (Cont.)

❑ Determine project standards and procedures:

- During this activity, you will specify how various deliverables are produced and tested by you and your project team.

Planning the Project (Cont.)

❑Create a preliminary budget:

- A preliminary budget outlines the planned expenses and revenues associated with your project.

Planning the Project (Cont.)

❑ Develop a Project Scope Statement:

- Developed primarily for the customer
- Outlines work that will be done and clearly describes what the project will deliver.
- Provides a clear understanding of project size, duration and outcomes.

Planning the Project (Cont.)

❑Setting a Baseline Project Plan (BPP):

- Provides an estimate of the project’s tasks and resource requirements and is used to guide the next project phase – execution.

Executing the Project

Project 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.

Executing the Project (Cont.)

❑Executing the Baseline Project:

- 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.

Executing the Project (Cont.)

❑Monitor project progress against the Baseline Project Plan (BPP)

Manage changes to the BPP:

- A slipped completion date for an activity.
- A bungled 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.

Executing the Project (Cont.)

❑ Maintain the Project Workbook

❑ Communicate the project status

- Meetings, status reports, meeting minutes, seminars and workshops, bulletin boards, memos, specification documents, brown bag lunches, hallway discussions, newsletters and project workbook.

Closing Down the Project

Project Closedown – the final phase of the project management process that focuses on bringing a project to an end.

- Closing down the project
- Conducting post project reviews
- Closing the customer contract

Representing and Scheduling Project Plans

❑ Key differences between Gantt Charts and Network Diagrams:

❑ **Gantt charts**

- Show task durations.
- Show time overlap.
- Show slack time in duration.

□ **Network diagrams**

- Show task dependencies.
- Do not show time overlap
- but show parallelism.
- Show slack time in boxes.

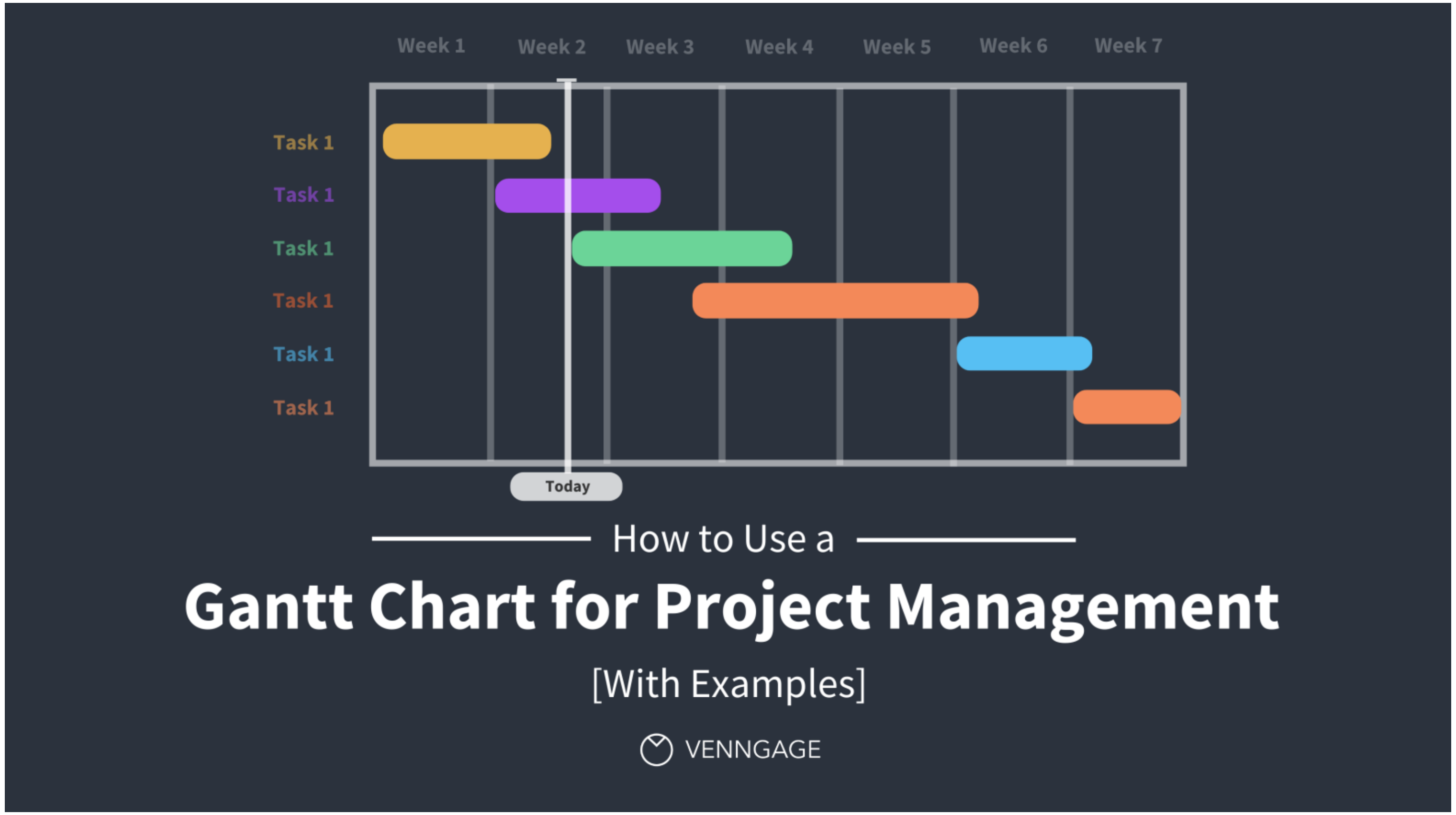
Representing Project Plans

❑**Resources** – any person, group of people, piece of equipment, or material used in accomplishing an activity.

❑**Critical Path Scheduling** – a scheduling technique whose order and duration of a sequence of task activities directly affect the completion date of a project.

❑**Critical Path** – a sequence of task activities whose order and durations directly affect the completion date of a project.

Gantt Chart Diagram



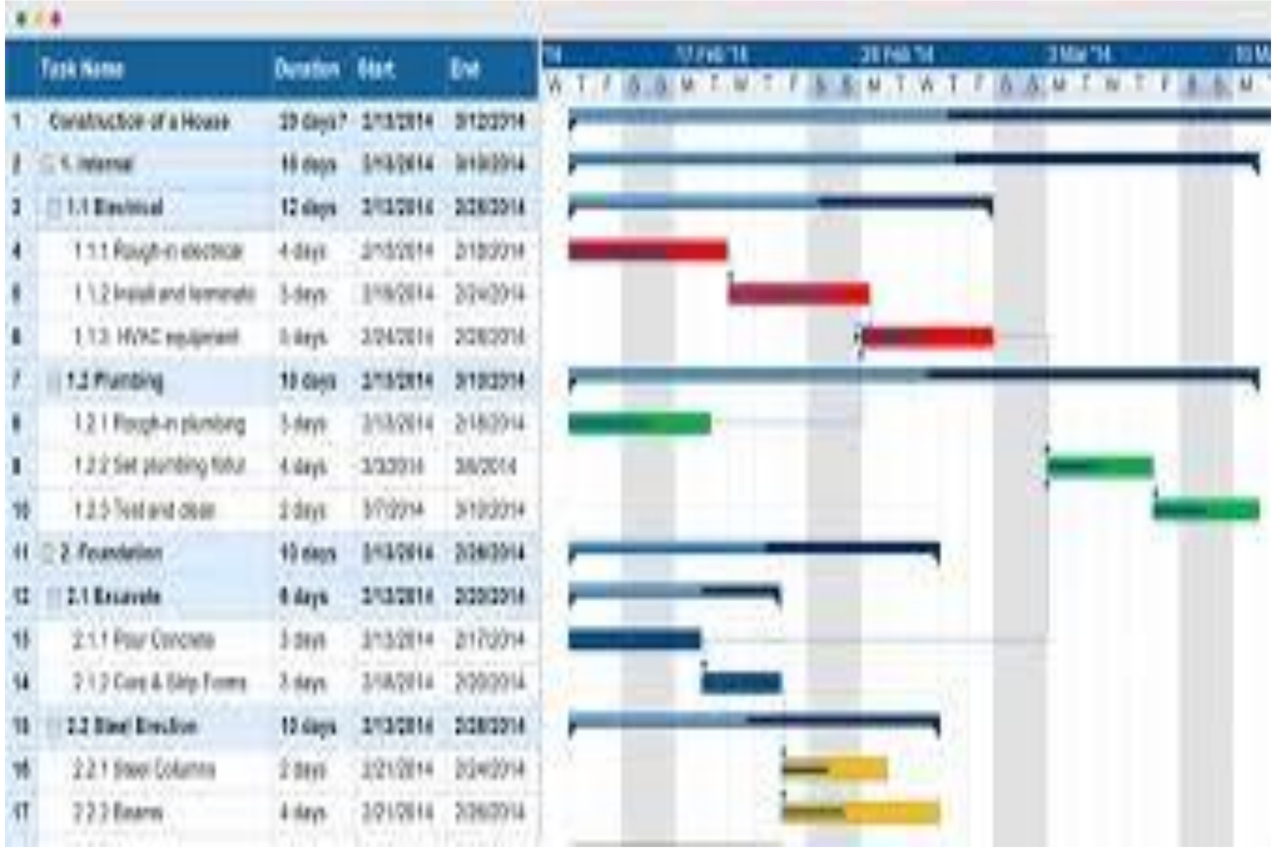
How to Use a

Gantt Chart for Project Management

[With Examples]

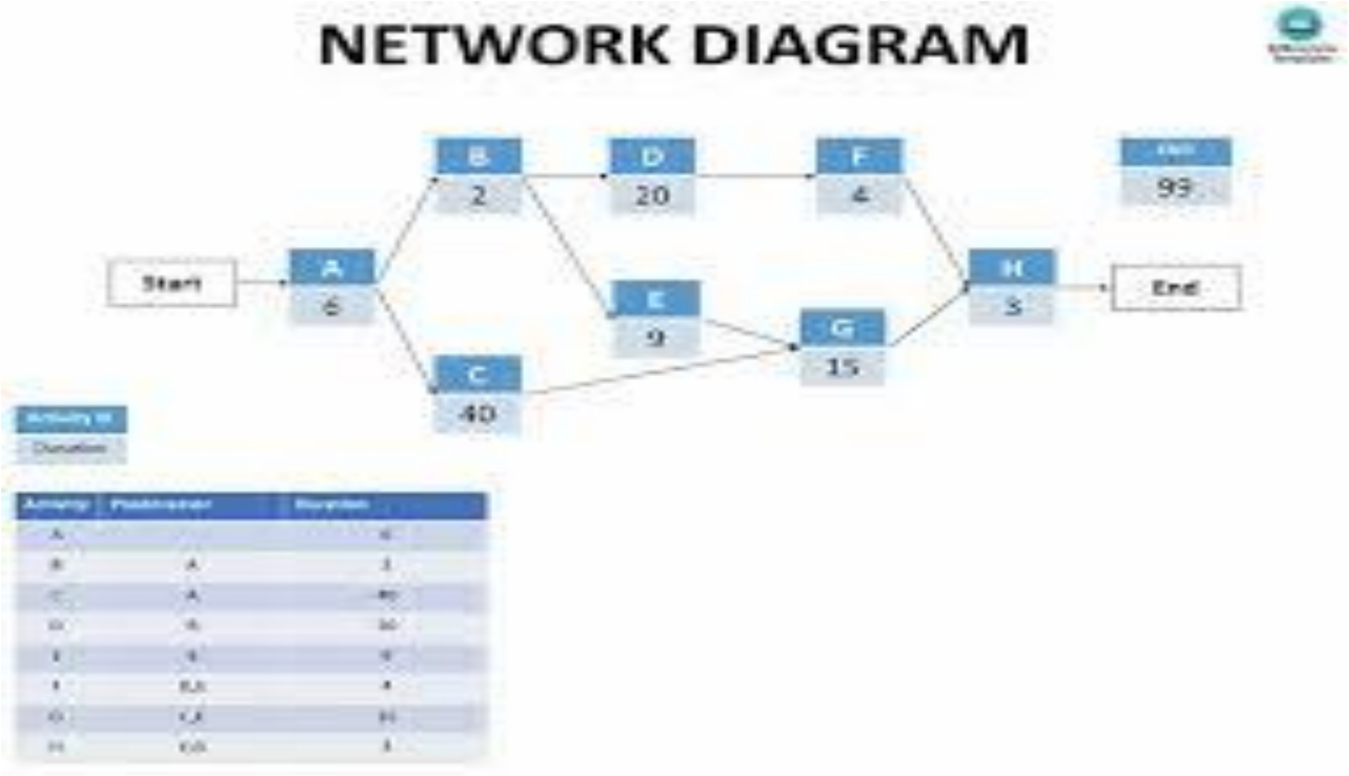
VENNGAGE

Gantt Chart Activity



	Task Name	Duration	Start	End
1	Construction of a House	20 days?	2/13/2014	3/12/2014
2	1. Internal	18 days	2/13/2014	3/10/2014
3	1.1 Electrical	12 days	2/13/2014	2/28/2014
4	1.1.1 Rough-in electrical	4 days	2/13/2014	2/18/2014
5	1.1.2 Install and terminate	5 days	2/18/2014	2/24/2014
6	1.1.3 HVAC equipment	5 days	2/24/2014	2/28/2014
7	1.2 Plumbing	18 days	2/13/2014	3/10/2014
8	1.2.1 Rough-in plumbing	5 days	2/13/2014	2/18/2014
9	1.2.2 Set plumbing fixtur	4 days	3/3/2014	3/6/2014
10	1.2.3 Test and clean	2 days	3/7/2014	3/10/2014
11	2. Foundation	10 days	2/13/2014	2/26/2014
12	2.1 Excavate	6 days	2/13/2014	2/20/2014
13	2.1.1 Pour Concrete	3 days	2/13/2014	2/17/2014
14	2.1.2 Cure & Strip Forms	3 days	2/18/2014	2/20/2014
15	2.2 Steel Erection	10 days	2/13/2014	2/26/2014
16	2.2.1 Steel Columns	2 days	2/21/2014	2/24/2014
17	2.2.2 Beams	4 days	2/21/2014	2/26/2014

Network Diagram



Activity	Predecessor	Duration
A		6
B	A	2
C	A	40
D	B	20
E	B	9
F	D,E	4
G	C,E	15
H	F,G	5

Representing Project Plans (Cont.)

❑Networking diagramming is a critical path scheduling technique and used when tasks:

- Are well-defined and have a clear beginning and end point.
- Can be worked on independently of other tasks.
- Are ordered.
- Serve the purpose of the project.

Calculating Expected Time Durations using PERT

PERT (**Program Evaluation Review Technique**) – a technique that uses optimistic, pessimistic and realistic time estimates to calculate the expected time for a particular task.

Calculating Expected Time Durations using PERT (Cont.)

Formula for Estimated Time:

- $ET = (o + 4r + p)/6$

Where

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

Using Project Management Software

❑Many powerful software tools exist for assisting with project management.

❑Special-purpose project management software is also available.

For example, Microsoft Project can help with

- Establishing a project starting or ending date.
- Entering tasks and assigning task relationships.
- Selecting a scheduling method to review project reports.

Using Project Management Software (Cont.)

❑Microsoft Project Gantt Charts:

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

❑Microsoft Project Network Diagrams:

- Hexagon shape indicates a milestone.
- Red boxes and arrows indicate critical path (no slack).