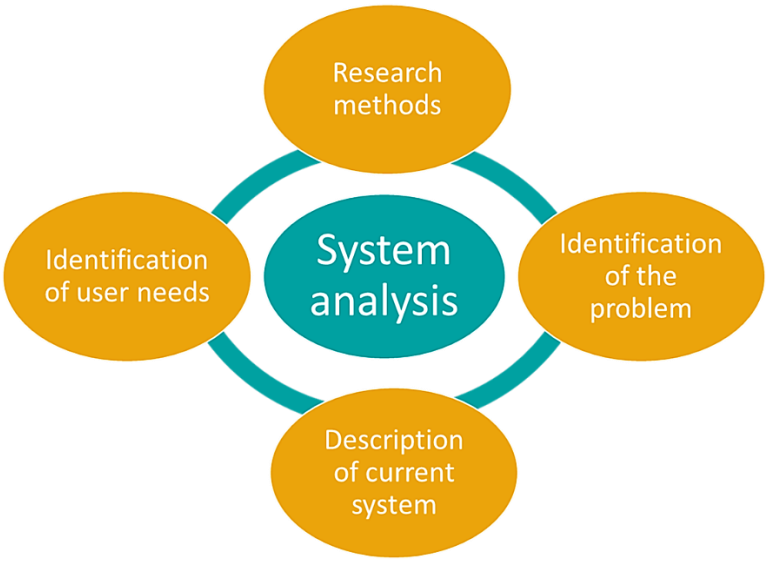
System Analysis and Design - Overview

Chapter-1

- Systems analysis
- Systems design

Systems Analysis

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



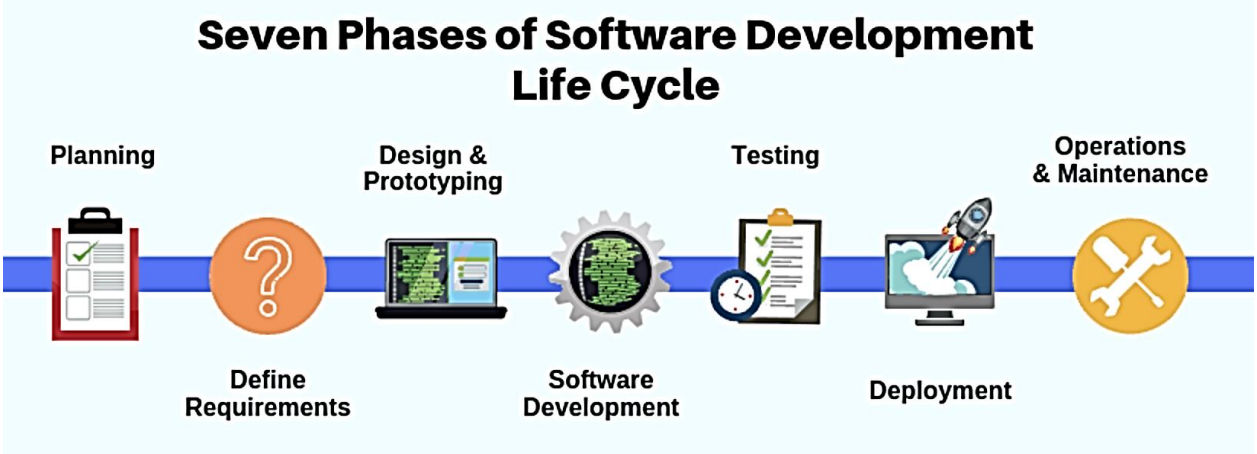
Systems Design

Systems design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements.

Systems Development Life Cycle (SDLC)

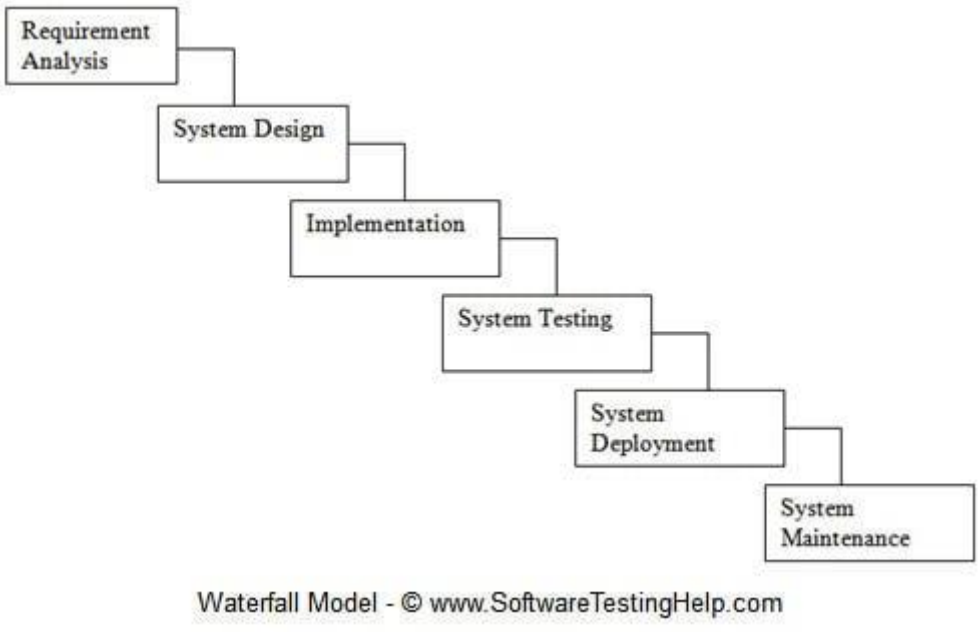
Phases in SDLC:

- Planning
- Analysis
- Design
- Implementation
- Maintenance



1. Planning: This step involves defining the scope of the system, identifying the goals and objectives of the project, and outlining the resources required.
2. Analysis: During this step, the requirements of the system are defined, and the current system is analysed to identify any deficiencies.
3. Design: The design step involves creating a detailed plan for the new system, including the system architecture, data structures, and interfaces.
4. Development: During this step, the actual system is built, and the software code is developed.
5. Testing: The system is tested to ensure that it functions as intended and meets the requirements.
6. Implementation: The new system is installed and made operational.
7. Maintenance: After the system is implemented, it is monitored and maintained to ensure that it continues to function correctly.

Waterfall SDLC



Waterfall Approach

- Feedback ignored, milestones lock in design specs even when conditions change
- Limited user involvement (only in requirements phase)
- Too much focus on milestone deadlines of SDLC phases to the detriment of sound development practices.

Different Approaches to Improving Development

- CASE Tools
- Agile Methodologies
- eXtreme Programming

CASE Tools

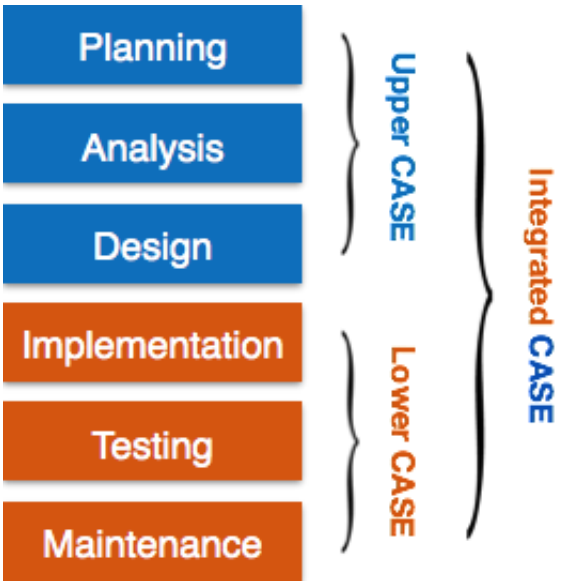
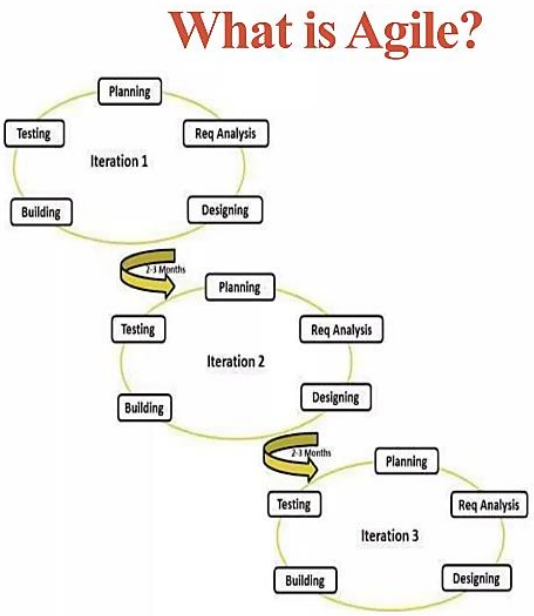


TABLE 1-2 Examples of CASE Usage within the SDLC

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

Agile Methodologies

- Agile model believes,
  - Every project needs to be **handled differently**
  - Existing methods need to be tailored to best suit the project requirements
  - Tasks are **divided to time boxes** (small time frames) to **deliver specific features for a release**
- **Iterative approach** is taken & working software build is delivered after each iteration
- Each **build is incremental** in terms of features
- Final build holds all the features required by the customer



When to use Agile Methodologies

If your project involves:

- Unpredictable or dynamic requirements
- Responsible and motivated developers
- Customers who understand the process and will get involved

- In waterfall
  - development teams only have one chance to get each aspect of a project right
- In an agile paradigm
  - every aspect of development — requirements, design, etc. — is continually revisited throughout the lifecycle. When a team stops and re-evaluates the direction of a project every two weeks, there's always time to steer it in another direction

The Manifesto for Agile Software Development

- **Individuals and interactions**
  - in agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.
- **Working software**
  - Demo working software is considered the best means of communication with the customer to understand their requirement, instead of just depending on documentation.
- **Customer collaboration**
  - As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.
- **Responding to change**
  - agile development is focused on quick responses to change and continuous development.

Sr #	Agile	Traditional
1	Incremental Value & Risk Management	Phased approach with an attempt to know everything at the start
2	Embracing Change	Change Prevention
3	Delivery Early, Fail Early	Delivers value at the end, fails at the end
4	Transparency	Detailed planning, stagnant control
5	Inspect and Adapt	Meta Solution w/tightly controlled procedures & final answers
6	Self Managed	Command and Control
7	Continual Learning	Learning is secondary to the pressure of delivery

eXtreme Programming

- Short, incremental development cycles
- Automated tests
- Two-person programming teams
- Coding, testing, listening, designing

Advantages:

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

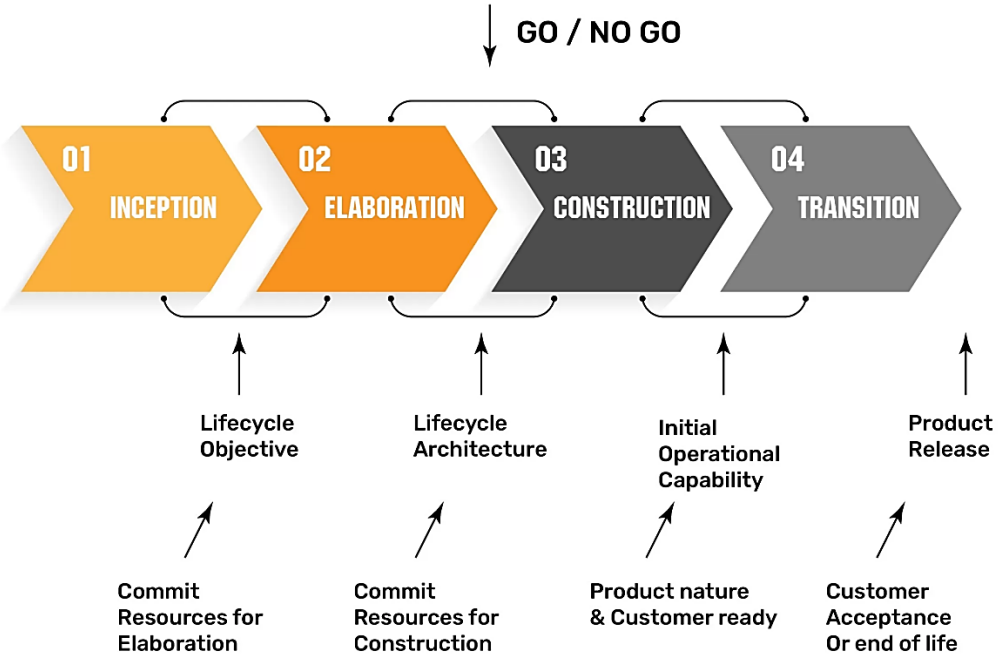
Object-Oriented Analysis and Design (OOAD)

- **Object-oriented analysis and design** (OOAD) is a software engineering approach that models a system as a group of interacting objects .
- **Analysis** — understanding, finding and describing concepts in the problem domain.
- **Design** — understanding and defining software solution/objects that *represent* the analysis concepts and will eventually be implemented in code.
- **OOAD** — Analysis is object-oriented and design is object-oriented. A software development approach that emphasizes a logical solution based on objects.

What is RUP?

- RUP was originally developed by Rational Software (now part of IBM).
  - It is a Software engineering process
  - It is a process product
  - It enhances team productivity
  - It creates and maintains models
  - It is a guide to effectively use the Unified Modeling Language
- Its goal is to delivery a high quality product that the customer actually wants.

PHASES OF RUP



Chapter-2

Describe six different sources of software.

1. **Information technology service firms:** These are firms that provide customized software solutions for specific business needs. They work with clients to understand their requirements and then design, develop, and maintain software solutions that meet those requirements.
2. **Packaged software providers:** These are companies that develop and sell software packages that can be used by a wide range of customers. Examples of packaged software include Microsoft Office, Adobe Creative Suite, and QuickBooks.
3. **Vendors of enterprise-wide solution software:** These are companies that develop software solutions that are designed to meet the needs of large organizations. Examples of enterprise-wide software solutions include SAP, Oracle, and Salesforce.
4. **Cloud computing:** Cloud computing refers to the delivery of computing resources over the internet. Cloud computing providers offer a range of software services that can be accessed and used by customers via the internet. Examples of cloud computing services include Amazon Web Services, Microsoft Azure, and Google Cloud Platform.
5. **Open-source software:** Open-source software is software that is developed by a community of developers and made available to the public for free. Examples of open-source software include the Linux operating system, the Apache web server, and the MySQL database.
6. **In-house development:** In-house development refers to the development of software solutions by an organization's own team of developers. This can be a cost-effective way of developing software solutions that are customized to an organization's specific needs.

Systems Acquisition: Outsourcing

Outsourcing is a common approach to system acquisition that involves hiring external vendors or service providers to develop, maintain, or support an organization's information systems. Outsourcing can be a cost-effective way to obtain IT services or systems, as it allows organizations to access specialized skills and expertise without having to invest in expensive infrastructure and personnel.

Outsourcing can bring several benefits to an organization, including cost savings, access to specialized skills and expertise, flexibility to scale up or down IT services as needed, and reduced risk of technology obsolescence. However, outsourcing also has potential drawbacks, such as loss of control over the IT function, security and confidentiality risks, and the need to manage the vendor relationship effectively.

Selecting Off-the-Shelf Software

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

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

Vendor support: whether and how much support the vendor can provide and at what cost

Viability of vendor: can vendor continue to adapt/update software to changes in systems software and hardware.

Flexibility: the ease with which software is customized

Documentation: understandable and up-to-date user's manual and technical documentation.

Response time: how long it takes the software package to respond to the user's requests in an interactive session.

Ease of installation: a measure of the difficulty of loading the software and making it operational.

Validating Purchased Software Information

Send a request for proposal (RFP): This is a formal document that outlines your requirements for the software product you intend to purchase. It can be sent to multiple vendors to solicit their proposals, which should include information on the product's features, pricing, support, and implementation timeline. By reviewing and comparing the proposals received, you can validate the information provided by the vendors and ensure that it aligns with your needs.

Use a variety of information sources: You can also validate purchased software information by using a variety of information sources, such as the software vendor's website, documentation, license agreement, reviews, and feedback from other users. By checking these sources, you can verify the accuracy and reliability of the information provided by the vendor.

Reuse

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

Commonly applied to two different development technologies:

1. Object-oriented development
2. Component-based development

Object-oriented development

Object class encapsulates data and behaviour 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.

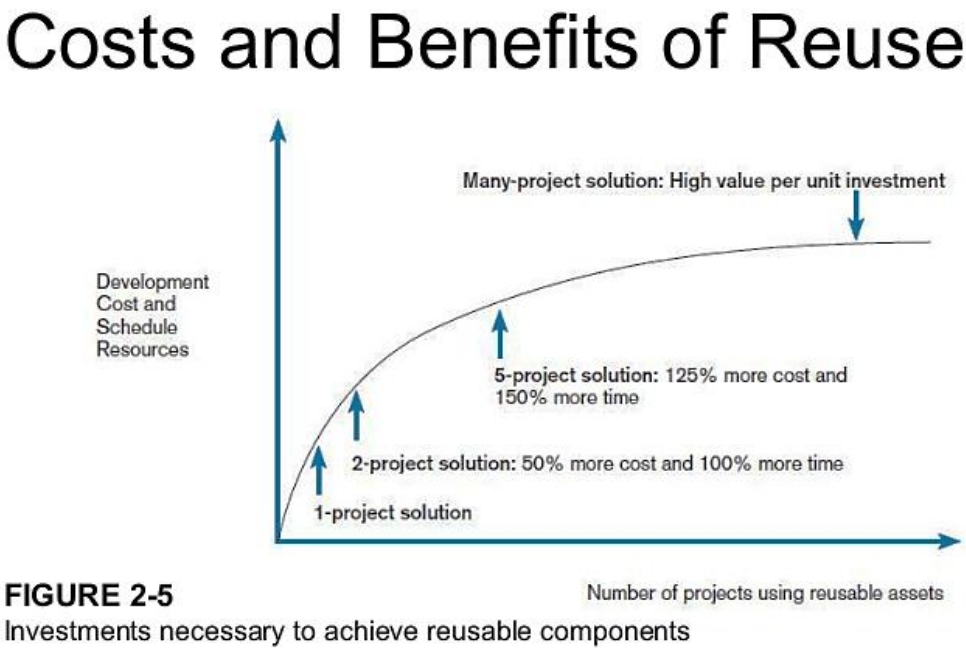


FIGURE 2-5
Investments necessary to achieve reusable components

3 Steps of Software Reuse

Abstraction – design of reusable piece of software

Storage – making software assets available for others

Recontextualization – making the software understandable to developers

Approaches to Reuse

Ad-hoc: individuals are free to find or develop reusable assets on their own

Facilitated: developers are encouraged to practice reuse

Managed: the development, sharing, and adoption of reusable assets is mandated

Designed: assets mandated for reuse as they are being designed for specific applications.

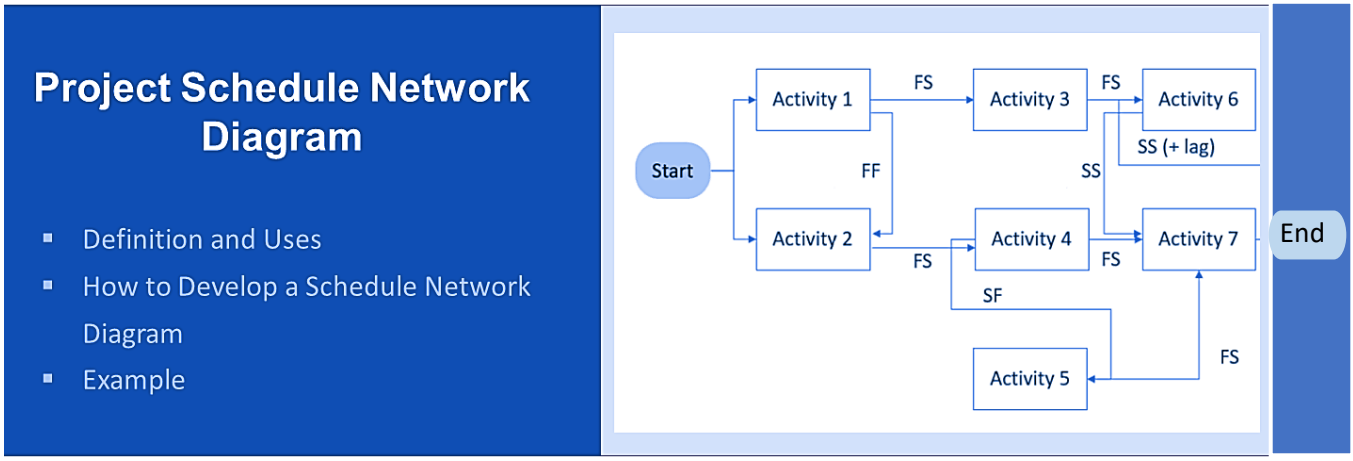
Chapter-3

Project management

Project management of systems development involves planning, organizing, and managing resources to bring a system development project to completion. This includes defining the project scope, identifying stakeholders, establishing project goals, creating a project schedule, and assigning tasks to team members.

Phases of Project Management Process.

1. **Initiation**: This includes identifying the project's goals and objectives, determining the project scope, identifying stakeholders, and conducting a feasibility study.
2. **Planning**: In this phase, a detailed project plan is created. This includes defining the project schedule, identifying project resources, creating a budget, and determining the tasks and activities required to achieve project goals.



Determining Project Standards and Procedures

- Type of SDLC methodology
- Documentation styles
- Status updates
- Terminology

Identifying and Assessing Risk

- Sources of risk
- Consequences of risk
- Possible sources: new technology, user resistance, critical resource availability, competitive reactions, regulatory changes, team member experience

3. **Execution**: In this phase, the project plan is put into action. Project team members work to complete the tasks and activities outlined in the project plan. This phase involves managing resources, coordinating tasks, and monitoring project progress.
4. **Closedown**: In this phase, the project is completed, and the deliverables are handed over to the customer or end-user. This includes conducting a project review to evaluate project performance, document lessons learned, and close out the project.

Gantt Charts vs. 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.

Estimating Task Duration

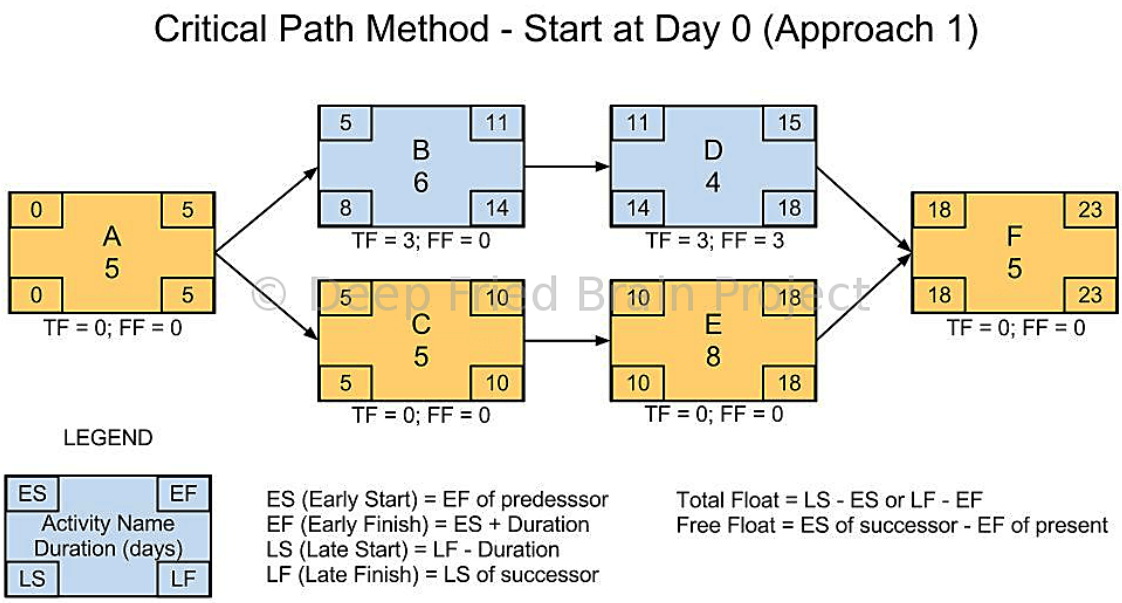
- Formula for Estimated Time:
  - $ET = \frac{o + 4r + p}{6}$

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:

ACTIVITY	TIME ESTIMATE (in weeks) o	r	p	EXPECTED TIME (ET) $\frac{o + 4r + p}{6}$
1. Requirements Collection	1	5	9	5
2. Screen Design	5	6	7	6
3. Report Design	3	6	9	6
4. Database Design	1	2	3	2
5. User Documentation	2	6	7	5.5
6. Programming	4	5	6	5
7. Testing	1	3	5	3
8. Installation	1	1	1	1

Critical Path Calculation



Project Evaluation Criteria

In general terms, a project evaluation process goes over the project constraints including **time, cost, scope, resources, risk and quality.** In addition, organizations may add their own business goals, strategic objectives and other metrics

Chapter-4

Identifying and Selecting Systems Development Projects

The Process of Identifying and Selecting IS Development Projects

Project identification and selection consists of three primary activities:

1. **Identifying potential development projects**
   The first step in identifying and selecting IS development projects is to generate a list of potential projects. This can be done through various means such as brainstorming sessions, customer feedback, market research, competitor analysis, and strategic planning. The goal is to identify potential projects that align with the organization's goals, mission, and vision.
2. **Classifying and ranking IS development projects**
   Once potential projects are identified, they must be classified and ranked to determine which projects are worth pursuing. Projects can be classified based on factors such as their strategic alignment, feasibility, complexity, impact, and cost. The classification and ranking process should involve input from key stakeholders such as executives, business owners, and IT personnel.
3. **Selecting IS development projects**
   The final step is to select the IS development projects that will be pursued. This decision should be based on the results of the classification and ranking process, as well as the organization's priorities and resource constraints. It is important to consider the potential risks, benefits, and trade-offs of each project before making a final decision. The selection process should involve input from all relevant stakeholders, and the final decision should be communicated clearly to all parties involved.

Organizations benefit from a planning-based approach in several ways, including:

1. Clarity of purpose: By setting clear and specific goals, organizations can focus their resources and efforts on achieving the desired outcomes.
2. Improved decision-making: A well-planned approach helps organizations to make informed decisions based on a thorough analysis of available data and potential outcomes.
3. Resource optimization: Effective planning enables organizations to allocate resources more efficiently, minimizing waste and maximizing returns.
4. Risk management: Planning helps organizations to identify potential risks and develop strategies to mitigate or manage them effectively.

5. Increased accountability: With a well-defined plan in place, organizations can hold individuals and teams accountable for meeting specific targets and deadlines.

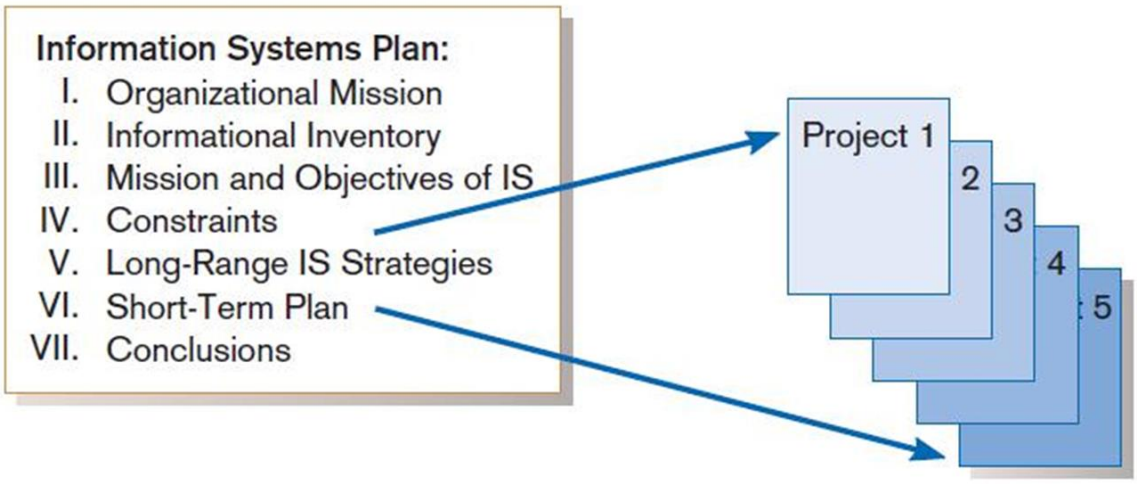
Corporate strategy involves:

- **Mission statement**: This statement defines the purpose and reason for the organization's existence.
- **Objective statements :**The objectives can be financial, operational, customer-focused, or employee-focused, depending on the organization's priorities and strategic goals.
- **Description of competitive strategy:** The competitive strategy is informed by the organization's strengths, weaknesses, opportunities, and threats, as well as market trends and customer preferences.

Difference between top-down and bottom-up planning

	TOP DOWN	BOTTOM UP
	Top down approach: upper level management makes a purchase and forces its use down the organization.	**Bottom up approach**: lower and more technical levels begin using the solution, adoption grows organically. Starts with user. Land and expand model.
Level of Entry	High	Low
Value Justification	Priority at the top, with support from key stakeholders below.	Proven use makes teams more efficient and effective, garnering stakeholder support at all levels.
Sales Strategy	Sales people with experience with face to face sales.	Viral, web based
Sales Cycle	Long	Short. Viral.
Avg Sale Price	High	Low
Revenue	A few sales may be all you need.	Requires high volumes of customers paying small amounts each.

Information Systems Planning



- **Organizational Mission, Objectives, and Strategy**
    - Brief description of mission, objectives, and strategy of the organization. The current and future views of the company are also briefly presented
- **Information Inventory**
    - Summary of processes, functions, data entities, and information needs of the enterprise
- **Mission and Objectives of IS**
    - Primary role IS will play in the organization to transform enterprise from current to future state
- **Constraints on IS Development**
    - Limitations imposed by technology and current levels of financial, technical, and personnel resources
- **Systems Needs and IS Strategy**
    - Summarize overall information systems needs in the company and set long-term (2-5 year) strategies for filling the needs
- **Short Term Plan**
    - Detailed inventory of present projects and systems and detailed plan for the current year
- **Conclusions**
    - Unknown but likely events that can affect the plan, presently known business change elements and their impact on the plan.

- **value chain analysis:** Conducting a value chain analysis prompts you to consider how each step adds or subtracts value from your final product or service. This, in turn, can

help you realize some form of competitive advantage, such as: Cost reduction, by making each activity in the value chain more efficient and, therefore, less expensive.

❖ Suppose you are considering alternative designs for a given system in order to make a project decision. There are four alternative designs that could be pursued (A, B, C, and D). The total score for requirements and constraints for each alternative as follows: (total score for requirements: A = 220, B = 150, C = 200, D = 220) and (total score for constraints: A = 200, B = 150, C = 230, and D = 150). Using weighted multicriteria analysis, which alternative is the best? Explain your answer and show your work.

Ans:

To determine the best alternative using weighted multicriteria analysis, we need to assign weights to the criteria. Let's assume that the requirements criterion has a weight of 60% and the constraints criterion has a weight of 40%.

Weighted score = (Score for requirements x Weight for requirements) + (Score for constraints x Weight for constraints)

Alternative A: (220 x 0.60) + (200 x 0.40) = 132 + 80 = 212
Alternative B: (150 x 0.60) + (150 x 0.40) = 90 + 60 = 150
Alternative C: (200 x 0.60) + (230 x 0.40) = 120 + 92 = 212
Alternative D: (220 x 0.60) + (150 x 0.40) = 132 + 60 = 192

we can see that both Alternatives A and C have the highest score of 212, which means they are equally good alternatives.

Electronic Commerce:

Identifying and Selecting Projects