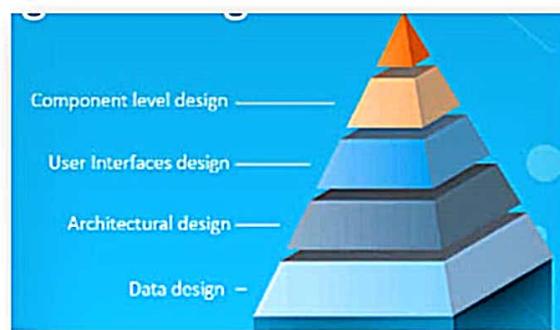


# Design Models

- The design phase of software development, transforming the customer requirements as described in the SRS documents into design forms.
- Designing a model is an important phase and is a multi-process that represent the data structure, program structure, interface characteristic and procedural details.

➤ There are four types of design elements / models

1. Data Design Element / Model
2. User Interface Design Element / Model
3. Architectural Design Element / Model
4. Component Level Design Element / Model



# **Architectural Design Model**

- Architecture Design Model serves as a blueprint for a system.
- The architecture focuses on the early design decisions.

**Architectural Design Styles describe:**

- Set of hardware & software components that will perform a function required by the system.  
Eg. Database, Modules, Frameworks etc.
- Set of connectors will help in coordination & communication between the components.
- Conditions that how components can be integrated to form the system.
- Semantic models that help the designer to understand the overall properties of the system.

## **Importance of Software Architecture**

- 1. Security:** The system is secured against malicious users by encryption or any other security measures due to layered software architecture.
- 2. Performance:** It handle request and response of the page in minimum time.
- 3. Maintainability:** Architectural design process uses easily modifiable and replaceable components which is easy to change them over time according to the new requirements.
- 4. Safety:** Avoid critical functionalities in small components & improve communication of the system.
- 5. Availability:** Architectural design process includes corresponding components, functionalities for handling the occurrence of any type of errors.

## **Decisions of Architectural Design**

- The architectural design process differs as the system differs depending upon the type of system being developed.

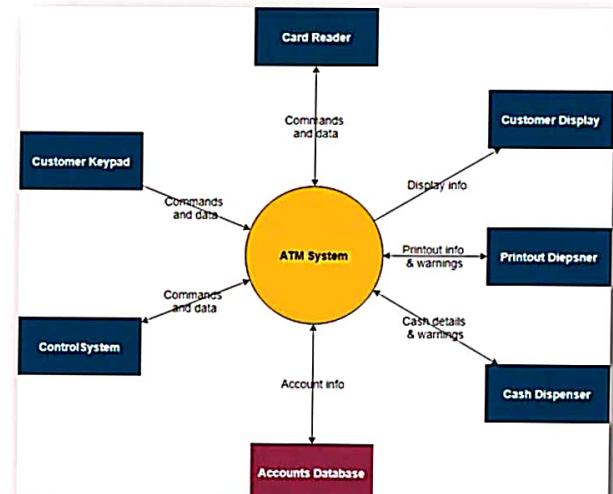
➤ **There are some common decisions that should be taken care of in any design process.**

- ❑ How can the system be distributed across the network?
- ❑ Which approach can be used to structure the system?
- ❑ Which architectural styles are suitable for the proposed system?
- ❑ How can software architecture be documented?
- ❑ How can the system be decomposed into modules?
- ❑ What control strategy must be used to control the operation of the components in the system?
- ❑ How can architectural design be analyzed?

## Taxonomy of Architectural styles

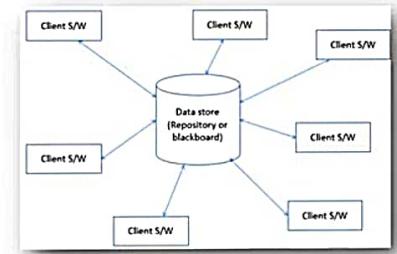
➤ Architectural styles is establish a complete structure & components of all over the system.

1. Data Centered Architecture
2. Data Flow Architecture
3. Object Oriented Architecture
4. Layered Architecture



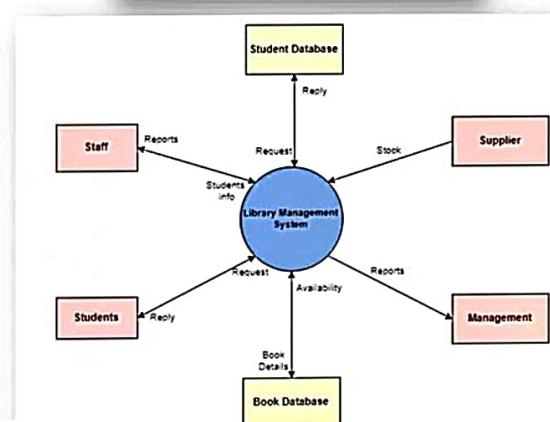
# Data Centered Architecture

- Data store at the center of this architecture and is accessed frequently by everyone.
- Update, add, delete or modify the data present within the store.
- It is widely used in DBMS, Library Information System etc.



## Advantages:

1. Repository of data is independent of clients
2. It may be simple to add additional clients.
3. Modification can be very easy.

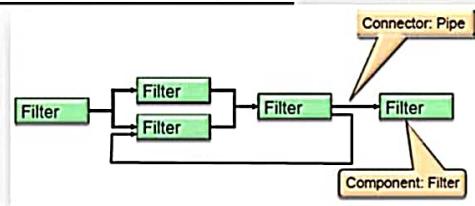


## Disadvantages:

1. Data replication or duplication is possible.
2. Changes in data structure highly affect the clients.

# Data Flow Architecture

- This architecture is used when input data to be transformed into output data through a series of computational manipulative components.
- **Pipe** is a connector which passes the data directionally from one filter to the next.
- **Filter** is a component reads the data from its input pipes and performs its function.
- This data and places the result on all output pipes.

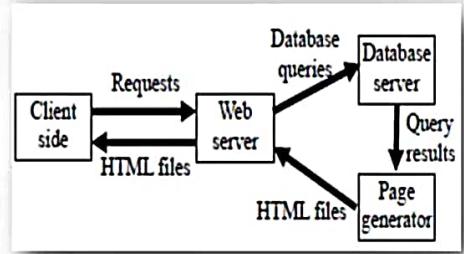


## Advantages:

1. With this design, concurrent execution is supported.

## Disadvantages:

1. Does not allow greater user engagement.



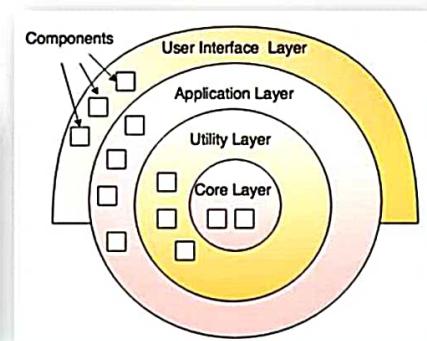
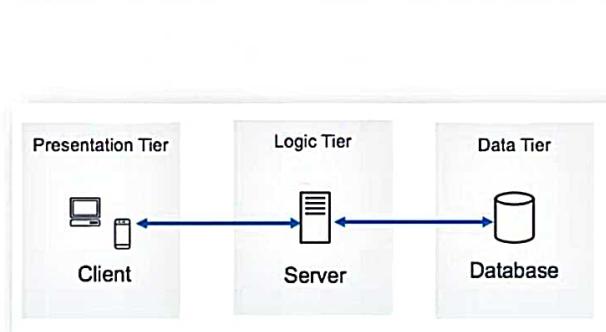
# Object Oriented Architecture

- Objects are the foundational building blocks for all kinds of software applications.
1. **Object:** Object is an instance of a class. Example Student S, Person;
  2. **Class:** It defines all attributes, methods, which represents the functionality of the object.
  3. **Encapsulation:** It is the process of binding similar types of elements of an abstraction.
  4. **Abstraction:** It is the removal of irrelevant essentials from users.
  5. **Inheritance:** It deriving a new class from existing one. It increases code reusability.
  6. **Polymorphism:** It has multiple forms. Ex: draw graphic objects circle, rectangle, triangle
  7. **Message Passing:** Sending and receiving data among objects through function parameters.



# Layered Architecture

- Data moves from one level to another level for processing is called layered architecture.
- Number of different layers are defined every layer performing well-defined set of operations.
- Outer layer components manage the user interface operations.
- Inner layer components will perform the operating system interfacing.
- Intermediate layers provide utility services and application software functions.
- Example: E-commerce web applications development like Amazon.



## Architectural Views

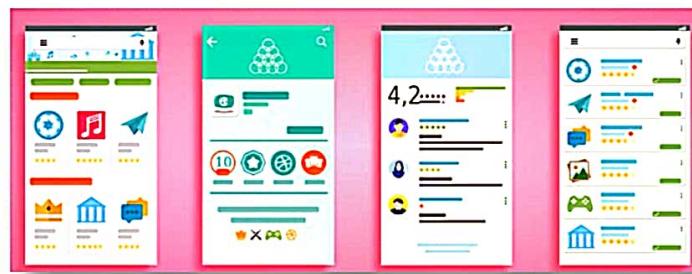
- It is generally used to represent entire architecture that is useful and meaningful to one or more stakeholders in system.
1. **Use case View:** Users view to handling product as per requirement. Use case diagram used.
  2. **Design View:** Organize design information, significant features, entities and attributes.
  3. **Process View:** Describe communication, behavior & synchronization aspects of the design.
  4. **Implementation View:** It address source code integrators and developers of project.
  5. **Deployment View:** It describes & explains environment within system runner & executed.

# User Interface Design Model

- User interface is the front-end application view to which user interacts with the software.
- It determines how commands are given to the computer or the program and how data is displayed on the screen.

➤ The software becomes more popular if its user interface is:

1. Attractive
2. Simple to use
3. Responsive in short time
4. Clear to understand
5. Consistent on all interfacing screens



# Types of User Interface

## Type 1: Text-Based User Interface OR Command Line Interface

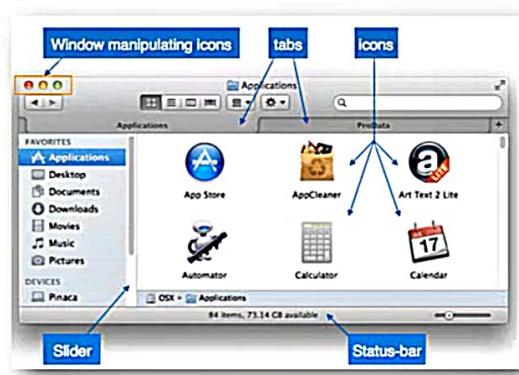
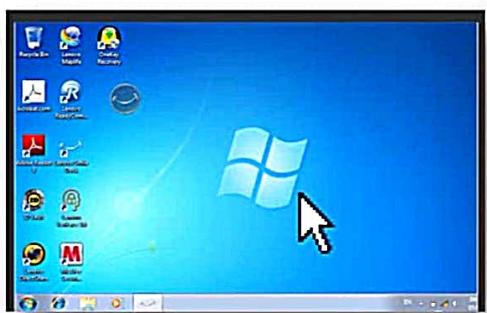
- Text based interface primarily used keyboard handling data.
- Command Line Interface provide command prompt or coding tools where the user types the command towards the system.
- The user needs to remember the syntax of command and its use.
- Command Line Interface used by Technical people or Programmers.



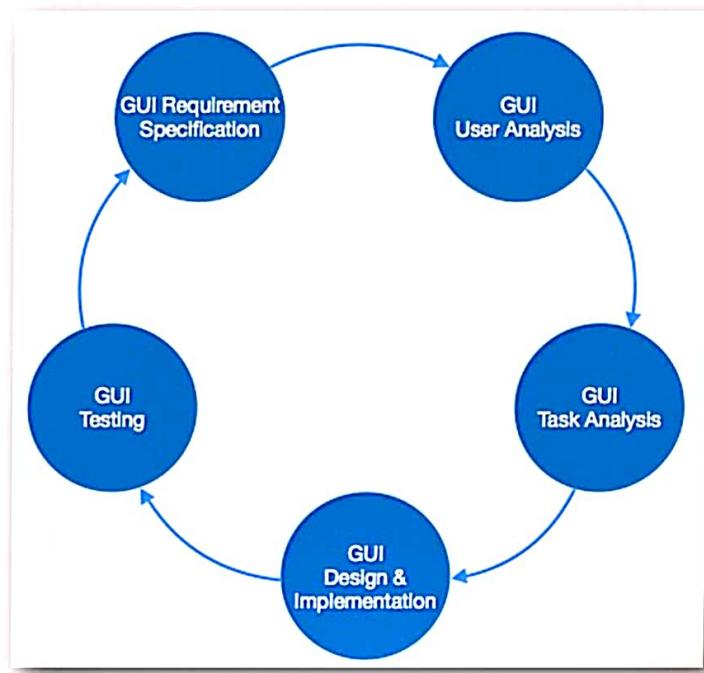
# Types of User Interface

## Type 2: Graphical User Interface

- Graphical User Interface provides the simple interactive interface to interact with the system.
- GUI can be a combination of both hardware and software.
- Graphical user interfaces are easy to learn as compared to the command-line interface.
- GUI provides multiple windows to the user simultaneously to interact with the system.



# User Interface Design Process



# User Interface Design Process

1. **GUI Requirement Gathering:** Designer analyze all functional & non functional requirements mentioned & discussed with customers.
2. **User Analysis:** Designer studies who is going to use the software GUI. The target audience matters as the design details change according to the knowledge level of the user.
3. **Task Analysis:** Designer analyze tasks, sub tasks & flow of the system.
4. **GUI Design & Implementation:** Designer generate actual GUI design by using different implementation tools like Wavemaker, Visual Studio etc.
5. **Testing:** GUI Testing done by Inner designer team, Organization & different stockholders in project.

# User Interface Design Principles

**1. User Familiarity:** The interface should be based of user oriented terms & concepts.

**Example:** In Windows OS, Terms like Desktop, Document, Folder, Rename, Power etc.

**2. Consistency:** The system command and menus should have the same format & parameter.

**Example:** Group of fonts, sizes, colors etc.

**3. Minimal Surprise:** Users never like to see the system working in an unexpected manner.

They get frustrated in such cases. Develop as per the user tendency.

**Example:** Group of similar type of features , In MS Words, File -> New, Open, Save, Print, Close , Restart system

## User Interface Design Principles

4. **Recoverability:** Users most often make mistakes while working with the system. These mistakes can be reduced but cannot be completely eliminated.

**Example:** Undo, Rename

5. **User Diversity:** The interface should provide appropriate interaction facilities for the various types of the system user.

**Example:** Larger Font, Privacy Settings etc.

6. **User Guidance:** The user interface should provide meaningful feedback when errors occur and provide different levels of help and advice.

**Example:** User Manual, Forgot Password etc.

# User Interface Design Golden Rules

## 1. Place the user in control.

- ✓ The user should be able to easily enter and exit the mode.
- ✓ Provide flexible interaction using keyboard, mouse or touch screen etc.
- ✓ User must be able to interrupt the sequence to do some other work.
- ✓ Display descriptive messages and text.
- ✓ Allow users to customize the interface like short cut keys etc.
- ✓ Hide technical internal details from users.
- ✓ Allow users to directly manipulate interface objects.



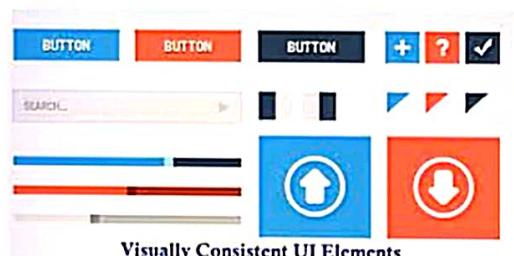
# User Interface Design Golden Rules

## 2. Reduce Users' Memory Load

- ✓ Reduce demand on short-term memory.
- ✓ New features updated in system.
- ✓ Define shortcuts that are intuitive.
- ✓ The visual layout of the interface should be based on a real-world metaphor.
- ✓ Disclose information in a hierarchical & progressive fashion.

## 3. Make the Interface Consistent

- ✓ Meaningful user tasks performed.
- ✓ Maintain consistency across a family of applications.
- ✓ Keep interaction results the same as per user expectations.



## User Interface Design Issues

1. **Response Time:** Time between request and response of the system. If the response time is too long, then the user becomes frustrated.
2. **Error Handling:** Poor error message may result in rejecting the product.
3. **Help Facilities:** User requires help when he needs some information and he cannot find it then the user is in trouble.
4. **Application Accessibility:** It states whether the application is simple to interact with or not. Special guidelines are given to user while interacting with software.

## The Management Spectrum

- In software engineering, the management spectrum describes the management of a software project.
- For properly building a product, there's a very important concept handle by Project Manager.
- These components play a very important role in your project that can help your team meet its goals and objective

➤ Effective software project management focuses on the four P's:

1. People
2. Product
3. Process
4. Project.



# 1. People

- The most important contribution in software project is not made by system or tool, It is done by people. i.e. Human Resources.
- The success of project depend on selecting right kind of people with right talent.

➤ Depending on there roles & responsibilities, following are the main categories:

1. **Senior Manager:** Define the business issues & have significant influence on the project.
2. **Project Manager:** Plan, Motivate, Organize & Control the project work. Has problem solving skills & manageable team abilities.
3. **Software Engineer:** Who deliver technical skills which are necessary in project.
4. **Customer:** Who specify the requirement for the software project.
5. **End Users:** Who interact with the software product.



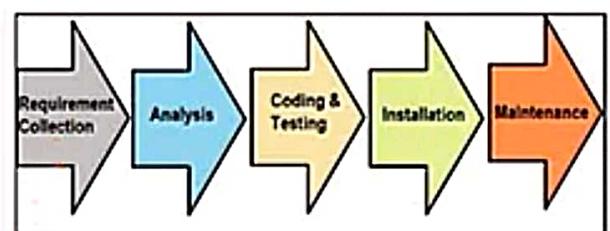
## 2. Product

- The product is the ultimate goal of the project.
- This is any types of software product that has to be developed or deliverable.
- Before a product can be planned, its objectives and scope should be established, alternate solutions should also be considered & technical & management constraints should be identified.
- Lack of these information, it is impossible to define reasonable and accurate estimation of the cost, identify possible risk & define manageable project schedule.



### 3. Process

- Project managers and team members should have a methodology and plan that complete project as per customer requirements.
- Without a clearly defined process, team members will not know what to do and when to carry out project activities.
- Using the right process will increase the project execution success rate that meets its original goals and objectives.
- The Process has several steps involved like, Documentation phase, Designing phase, Implementation phase, software configuration management, Deployment phase and Interaction phase.



## 4. Project

- The project is the complete software project that includes requirement analysis, development, delivery, maintenance and updates.
- The project manager plays a critical role for completing project.
- They are responsible to guide the team members to achieve the project's target and objectives.
- They helping & assisting them with issues, checking on cost and budget and making sure that the project stays on track with the given deadlines.
- They manage complete project activities to avoid project failure.



## About Project Planning

- Before starting a software project, it is essential to decide which tasks to be performed and how to manage all tasks involved in the software development.
- ✓ Project planning is an organized process from requirement gathering to testing and support.
- ✓ Focuses on all activities required for successful completion of the project.
- ✓ Prevents obstacles that arise in the project such as changes in projects, organization's objectives, non-availability of resources and so on.
- ✓ Helps in better utilization of resources and optimal usage of the allotted time for a project.
- ✓ Defines roles and responsibilities of the project management team members.



# Software Project Manager

## ➤ Managing People

- Act as Project Leader.
- Contact with all stakeholders.
- Managing human resources.



## ➤ Managing Project

- Defining and setting up project scope.
- Managing project management activities.
- Monitoring progress and performance.
- Risk analysis at every phase.
- Take necessary step to avoid or come out of problems.



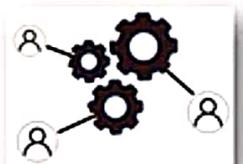
# Project Planning Process



# Project Planning Process

## 1. Identify Stakeholders Need:

- A project must meet the expectations of stakeholders & meet all of their needs to be effective.
- To consider their needs a project planner might identify the characteristics and qualities of the project at hand.



## 2. Identify Project Objectives:

- Project objectives must be specific, measurable, achievable, realistic and time-bound in order to be used to validate project success.
- All team members must contribute ideas and compromise on collective goals for the project.



# **Project Planning Process**

### **3. Identify Deliverables & Due Date:**

- A due date is a scheduled fixed date and time that an objective is due within a project.
- Deliverables are defined as the products, services or results that project quality, size, length, or any other standard that proves to be important in the success of project.



### **4. Identify Project Schedules:**

- This stage is created through different tasks that label a project's start and end date.
- Schedules are used to make sure projects stay on task and are completed in a timely manner.



## Project Planning Process

### 5. Provide Roles & Responsibilities:

- This stage allows for effective communication between everyone involved in the project.
- This segment includes identifying who is involved and what their individual tasks are.
- Everyone understand expected objectives, goals & output of project.



### 6. Identify Project Budget:

- Project budget are calculating the anticipated cost, changing budget cost and monitoring the budget.
- Therefore, it is crucial to prepare for unexpected alterations of project costs.



# Project Planning Process

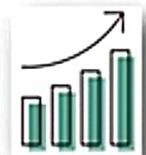
## 7. Identify Communication Plan:

- A communication plan is a tool that able to effectively communicate about a project to your client, team and other stakeholders.
- Communication plan has clear guidelines about how information will be shared & who is responsible within the project.



## 8. Provide Tracking & Management:

- It is an effective way to deliver projects on time and organize tasks.
- Features of this element include planning/scheduling, collaboration, documentation and evaluation.
- Management tools are track productivity and growth of project.



# **Recourses Used in Project Development**

- Project resources simply mean resources that are required for successful development and completion of project on time and on budget.

## **Type 1: Human Recourses**

- In software industry, manager, software developer, software testing and so on.
- These positions are according to their skills and specialty.



## **Type 2: Reusable Components**

- Managing budget for project is one of most important tasks that all project managers.
- The reusable resources are very helpful as they help in reducing overall cost of development.

## **Type 3: Hardware and Software tools**

- It should be planned before starting development of project otherwise it may causes problems.
- These are actually material resources that are part of project.

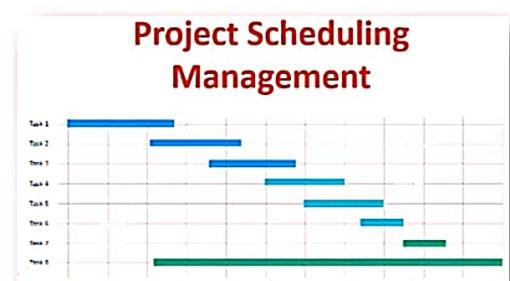
## **Issues in Project Management**

- Project is not completed on schedule.
- Changing customer requirements affect on schedule.
- Technical difficulties are generate.
- Miscommunication among project management.
- Essential software & hardware may be delivered late.
- In large project, Software engineer perform multiple tasks parallel.
- Tasks interdependencies are in project.
- Risk is not considered at beginning of project.



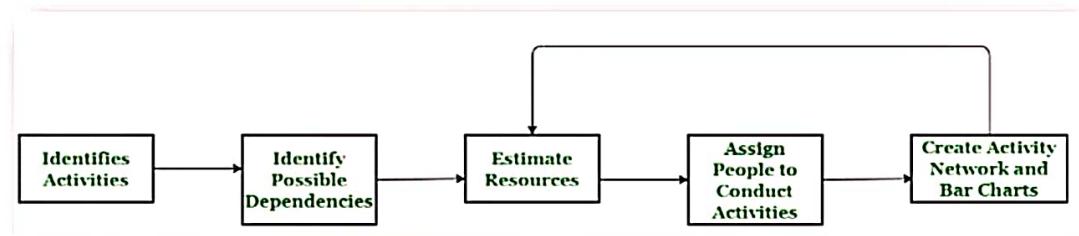
## About Project Scheduling

- Project scheduling is responsible activity of project manager.
- Project schedule is a mechanism that is used to communicate and know about that tasks are needed and has to be performed in project.
- Project manager separate total work task in project into different activities. i.e. WBS
- Project Manager estimate time & resources required to complete activities & organize them into coherent sequence.
- Effective project scheduling leads to success of project, reduced cost and increased customer satisfaction.



# Project Scheduling Process

1. Identify all the functions / modules required to complete the project.
2. Break down large functions into small activities i.e. Develop WBS structure.
3. Determine the dependency among various activities.
4. Allocate resources to activities.
5. Assign people to conduct different activities.
6. Plan the beginning and ending dates for different activities.
7. Create Activity Network & Bar or Gantt charts.



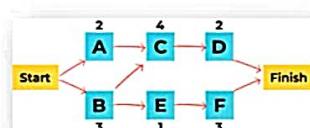
# Project Scheduling Techniques

## 1. Critical Path Method:

- It helps you determine both longest and shortest possible time take to complete a project.
- There are three essential elements: The tasks required to complete the project, Which tasks depend on the completion of others, A time estimate for each activity

### Example:

- There are four tasks in the project – A, B, C, and D.
- Task B and D can only begin after task A completes, whereas task C has no such restriction.
- Task A will be time-sensitive as any delay in its completion can delay in task B & D.
- Called as Critical Task.
- This helps in identifying and separating the independent variables.
- Finally, it adds milestones to the project.



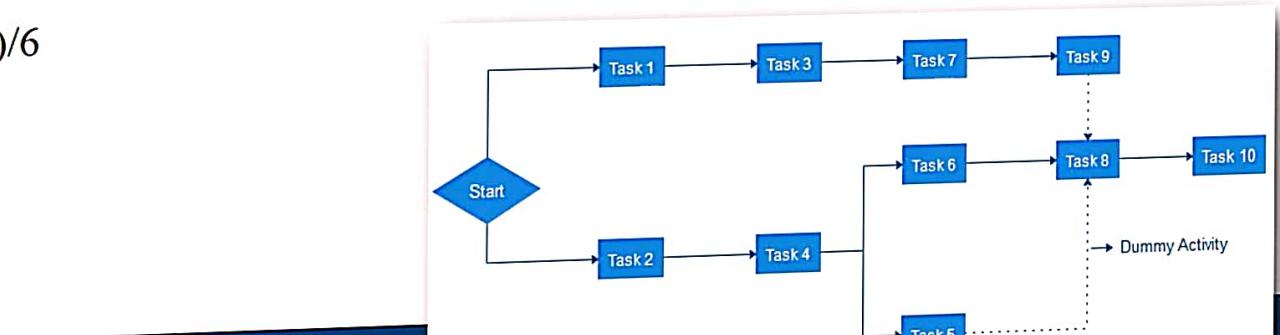
# Project Scheduling Techniques

## 2. Program Evaluation and Review Technique (PERT):

- It is a way to schedule flow of tasks in a project and estimate total time taken to complete it.
- PERT charts offer a visual representation of the major activities (and dependencies) in a project

It calculate:

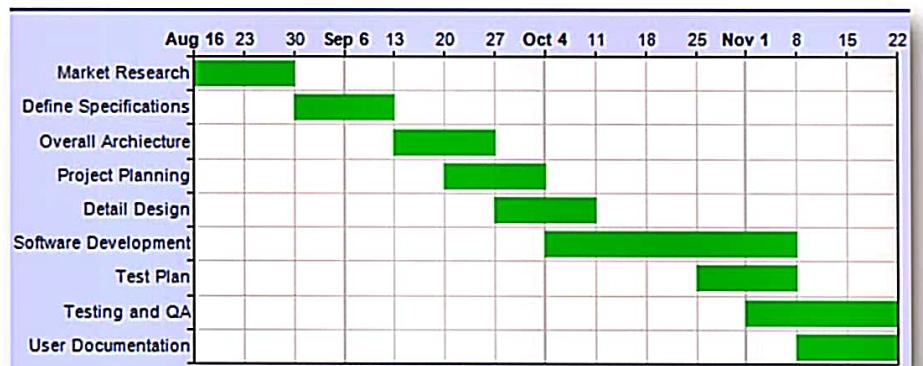
- **Optimistic time (O):** Quickest time you can complete a project
- **Pessimistic time (P):** Longest time it'll take to complete your project
- **Most likely time (M):** How long it'll take to finish your project if there are no problems.
- $(O + 4M + P)/6$



# **Project Scheduling Techniques**

### **3. Gantt Chart:**

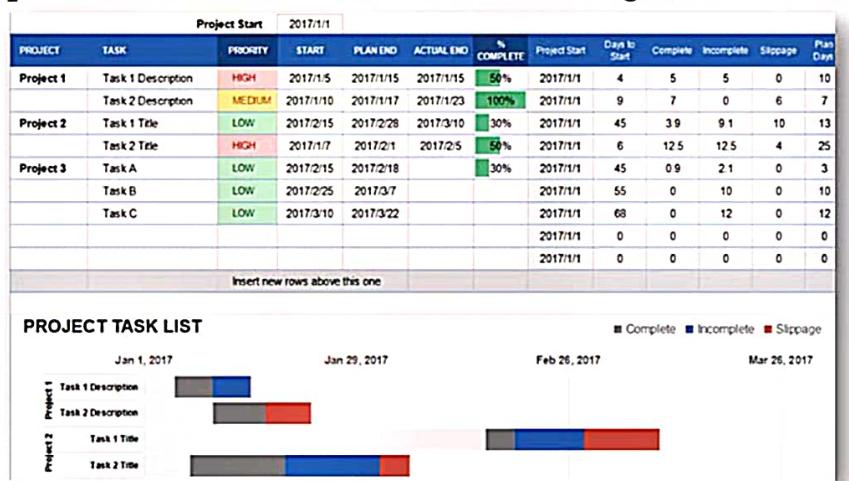
- A Gantt chart is a type of bar graph that project managers use for planning and scheduling in complex project.
  - It represent each task horizontally on a bar chart, which shows the start and end dates & they frequently include deadlines & dependencies of tasks.
  - It easier to visualize the progress of a project and see how different tasks interact with one another.



# Project Scheduling Techniques

## 4. Task List:

- One of the simplest project scheduling techniques is the creation of a task list.
- Create task list using a word processor or spreadsheet software.
- It creates a list of tasks and include important information like the task manager, start date, deadline & completion status.



# Project Scheduling Techniques

## 5. Fast Tracking :

- In Fast Tracking, Project is being implemented by either simultaneously executing many tasks or by overlapping many tasks to each other.
- **Example:** In software development project, designing and development can be taken up in parallel. Once design of essential features is ready and approved, development team can work on it. Meanwhile, the designing team will work on the remaining elements and functions.

## 6. Crashing:

- Crashing deals with involving more resources to finish the project on time.
- **Example:** Add more developer in project, Paying overtime to employee,
- Crashing can only be applied when it fits your project budget.

