Activity List Overview

An **Activity List** is a key project management document that identifies the specific tasks required to produce project deliverables. It is detailed enough to determine resource needs and estimate scheduling. The list serves as a roadmap for the project team, ensuring clear task execution and progress tracking.

Key Functions of an Activity List:

- **Task Identification**: Each task involved in the project is clearly outlined, allowing for better management and organization.
- **Resource and Time Estimation**: By detailing each activity, the project team can estimate the resources and time needed to complete the project successfully.
- **Dependency Management**: The list provides insight into task dependencies, helping manage the sequence of activities and preventing bottlenecks.

Common Elements of an Activity List:

1. **Activity Name**: A concise name describing the task.
2. **Activity Identifier/Number**: A unique identifier that facilitates tracking.
3. **Brief Description**: A short explanation of the task's purpose and scope.
4. **Predecessors and Successors**: Defines which tasks must be completed before or after the current activity.
5. **Logical Relationships**: Describes how tasks are interconnected, ensuring proper sequencing.
6. **Leads and Lags**: Indicates timing adjustments needed between dependent tasks.
7. **Resource Requirements**: Specifies the personnel, materials, and equipment needed for the activity.
8. **Constraints**: Any limitations that may affect the task's execution, such as deadlines or resource availability.
9. **Imposed Dates**: Specific start or end dates that are fixed and cannot be changed.
10. **Assumptions**: Hypotheses or conditions that are assumed to be true when planning the activity.

Milestones Overview

A **Milestone** is a key project event that represents a significant stage or achievement in the project lifecycle. Unlike tasks, milestones do not have a duration and are typically used to mark the completion of critical activities.

Key Functions of Milestones:

- **Progress Monitoring**: Milestones act as checkpoints to measure the project's advancement.
- **Goal Setting**: They help define project targets and align team efforts toward common objectives.

Examples:

- Completion of key deliverables.
- Customer approval of major project phases.

Gantt Chart Overview

The **Gantt Chart** is a visual tool that outlines project activities as bars along a timeline, allowing project managers to track progress and dependencies efficiently.

Key Functions of a Gantt Chart:

- **Activity Tracking**: Visualizes project tasks, their duration, and dependencies.
- **Grouping Activities**: Tasks can be grouped into categories or phases to provide a high-level view of the project.
- **Milestone Representation**: Milestones are depicted as diamonds, making them easy to spot.

Key Elements:

- Textual outlines combined with a timeline.
- Bars representing activities.
- Diamonds representing milestones.
- Grouped tasks that provide hierarchical organization.

Hard and Soft Dependencies

In project management, **Dependencies** define the relationships between tasks. They dictate the sequence in which activities are performed.

Hard Dependencies:

- **Definition**: Non-negotiable task relationships dictated by the project logic (e.g., coding must occur before testing) or external factors (e.g., contract sign-off).
- **Impact**: Hard dependencies are critical to the project's structure, and eliminating them may increase risk or cause rework.

Soft Dependencies:

- **Definition**: Flexible task relationships that can be adjusted to optimize workflow. Often inserted by planning tools.
- **Impact**: Soft dependencies provide greater scheduling flexibility but are less rigidly enforced than hard dependencies.

By defining activities, tracking milestones, and understanding dependencies, project managers can ensure the smooth execution of project tasks while maintaining control over resources and timelines.

Duration, Effort, and Manpower in Project Management

In project management, **Duration (D)** is derived from **Effort (E)** and **Manpower (M)**, providing a simple equation to help estimate the time required to complete tasks:

D=EMD = \frac{E}{M}D=ME

Key Concepts:

- **Effort (E)**: The total amount of work required for a task, usually measured in man-hours.
- **Manpower (M)**: The number of resources (people) available to work on a task and their availability percentage.

Formula Example:

- **1 Week = 40 hours**
    - **Effort**: 40 man-hours
    - **Resources**: 1 person @ 100%
    - **Duration (D)**: 40 man-hours / 1 person = 40 hours = 1 week
- **Example with Multiple Resources**:
    - **Effort**: 80 man-hours
    - **Resources**: 2 people @ 100%
    - **Duration (D)**: 80 man-hours / 2 people = 40 hours = 1 week
- **Example with Reduced Availability**:
    - **Effort**: 80 man-hours
    - **Resources**: 1 person @ 50% availability

- **Duration (D)**: 80 man-hours / 0.5 = 160 hours = 4 weeks

Important Considerations:

- **Simplified Model**: The equation is a basic estimation tool, useful for most cases, but it doesn't capture all complexities (e.g., dependencies between variables).
- **Effort Estimation**: Estimating the effort required for tasks is challenging, and precise estimation techniques are necessary for more accurate planning.

Managing Uncertainty in Project Planning

Uncertainty is a natural part of project planning, and project managers use various strategies to manage this:

1. **Implicit Padding**:
   - Each activity includes extra time to account for potential delays.
   - **Challenge**: Estimations become inaccurate, leading to difficulties in managing resources and schedules. Being overly pessimistic may result in early delivery, but the plan remains flawed.
2. **Explicit Padding**:
   - Extra activities or slack time are explicitly added to the project plan to manage delays.
   - **Benefit**: This approach ensures transparency, making data more accurate and easier to monitor.
   - **Challenge**: It may be difficult to get stakeholder buy-in, as customers may view padding as unnecessary or inefficient.
3. **React and Re-plan**:
   - When delays occur, immediate action is taken to address them, and the project plan is revised.
   - **Benefit**: This approach is flexible, allowing project managers to adapt by adding resources, changing dependencies, or reallocating tasks.
   - **Challenge**: This is not a proactive planning technique, but rather a reactive strategy used during project execution.

Recommendations for Managing Uncertainty:

- **Evaluate Delay Costs**: Always assess the financial and operational impact of delays.
- **Clear Strategy**: Choose a contingency strategy (implicit or explicit padding, or react and re-plan) and communicate it clearly to stakeholders.
- **Balanced Approach**: Avoid excessive pessimism or optimism in padding to maintain a realistic and actionable project plan.

Critical Path Method (CPM) Overview

The **Critical Path Method (CPM)** is a vital project management technique used to identify the most important tasks in a project schedule, known as the **critical path**. The critical path determines the project's overall duration, as any delay in these tasks will directly impact the project's finish date.

Key Concepts:

- **Critical Path**: The sequence of tasks that dictates the project's completion time. A delay in any of these tasks will delay the entire project.
- **Non-Critical Activities**: Activities that have some flexibility (float or slack) and won't delay the project if delayed.

Critical Path Determination:

1. **Identify Activities**: List all project tasks.
2. **Network Diagram**: Visualize the sequence and relationships of tasks in a network diagram (graph).
3. **Calculate Durations**: Determine how long each task will take.
4. **Find the Longest Path**: The longest sequence of dependent tasks is the critical path.

Network Diagrams

Network Diagrams are graphical tools used to represent project tasks and their dependencies. They help in visualizing task sequences and relationships. Two classic formats are commonly used:

1. **Activity on Arc (AOA)** or **Activity Diagramming Method (ADM)**:
    - **Arrows** represent tasks, and **circles** represent events (start or end of a task).
    - Example:
        - Task A (arrow) from Event 1 to Event 2.
    - Requires "dummy" tasks to show dependencies without direct activity.
2. **Activity on Node (AON)** or **Precedence Diagramming Method (PDM)**:
    - **Nodes** represent tasks, and **arrows** show dependencies.
    - Example:
        - Task A (node) leads to Task B (node) via an arrow.

AOA vs. AON Comparison

- **AOA (Activity on Arc)**:

    - Historically used for **PERT (Program Evaluation and Review Technique)**.
    - Simpler for certain algorithmic processes but often requires "dummy" tasks to represent complex dependencies.
- **AON (Activity on Node)**:
    - More flexible, easier to visualize, and draw.
    - Better suited for modern project management software and applications.

Example of Task Dependencies (AOA/AON):

Consider a project with five tasks:

Activity	Predecessors	Duration
A	None	3 months
B	None	4 months
C	A	3 months
D	A, B	1 month
E	B	2 months

In an **AOA diagram**, we may need to introduce dummy nodes to account for complex dependencies, while an **AON diagram** can directly show the relationships between tasks using arrows and nodes.

Visual Example:

- **AOA**: Circles represent events, with arrows connecting tasks.
- **AON**: Tasks are inside nodes, and arrows represent the task flow from one to another.

Key Takeaways:

- **Critical Path**: Focus on the tasks on the critical path to ensure no delays.
- **Network Diagrams**: Use AON diagrams for flexibility and ease of understanding.
- **AOA vs. AON**: AOA requires more complexity in handling dependencies, while AON provides a clearer representation of task flows.

Control Schedule, Resource Allocation, and Resource Leveling

Managing resources and controlling schedules are critical components of project management, ensuring that tasks are completed within constraints while optimizing the use of available resources. This process involves careful planning, monitoring, and adjustment to ensure efficient task execution and avoid over-allocation of resources.

Inputs:

1. **Project Plan**: Includes activities, constraints, and effort required for each task.
2. **Project Team**: Number, type, and availability of resources.
3. **Delivery Dates**: Fixed deadlines that may act as constraints.

Resource Allocation

Resource allocation is the process of assigning available resources (e.g., personnel, equipment) to specific tasks, ensuring that each task is carried out according to its planned effort and timeframe.

- **Key Considerations**:
    - Allocation is based on resource availability, skillsets, and task requirements.
    - **No Over-Allocation**: Resources should not be assigned more tasks than their maximum availability (e.g., a person working at 100% capacity should not be overburdened).

If conflicts or issues arise, such as resource over-allocation, adjustments can be made:

- Increase team size.
- Relax certain constraints.
- Reallocate resources from other tasks.

Example of Resource Allocation:

Task	Duration	Resource Allocation
Task 1	1 week	R1 at 50%
Task 2	1 week	R2 at 100%
Task 3	2 weeks	R1 & R2 at 50% each

In this example:

- R1 works 20 hours per week on Task 1 and Task 3.
- R2 works full-time (40 hours) on Task 2 and 50% on Task 3.

Resource Usage and Over-Allocation

Resource usage tracks how much effort or material is required at any given time, ensuring that resources are neither underutilized nor over-allocated.

- **Manpower**: Hours each team member spends on tasks.
- **Equipment**: Number of tools or machines needed.
- **Material**: Quantities of materials consumed.

Over-Allocation occurs when a resource is scheduled to work beyond their capacity. This can lead to inefficiency, delays, and burnout.

Example of Over-Allocation:

Week	Task	Resource	Hours Allocated
Week 1	Task 1	R1	20 hours
Week 2	Task 2	R1	60 hours
Week 3	Task 3	R1	60 hours

In this case, **R1** is over-allocated in Week 2 and Week 3, working 60 hours per week instead of the standard 40.

Resource Leveling

Resource leveling is a technique used to adjust the project schedule and prevent over-allocation of resources. This is done by rescheduling tasks to ensure that no resource exceeds their maximum capacity.

- **Example Solution**: If R1 is over-allocated, delay the start of Task 3 until Task 1 is complete or assign additional resources to share the workload.

If resource leveling is not feasible, other techniques, such as project compression, can be explored to meet deadlines without over-allocating resources.