**Chapter 5**

**Scope**

* **Scope** refers to *all* the work involved in creating the products of the project and the processes used to create them
* A **deliverable** is a product produced as part of a project, such as hardware or software, planning documents, or meeting minutes

**Project scope management processes**

* Project scope management includes the processes involved in defining and controlling what is or is not included in a project
* **Planning scope:** determining how the project’s scope
* and requirements will be managed
* **Collecting requirements:** defining and documenting the features and functions of the products produced during the project as well as the processes used for creating them
* **Defining scope:** reviewing the project charter, requirements documents, and organizational process assets to create a scope statement
* **Creating the WBS:** subdividing the major project deliverables into smaller, more manageable components
* **Validating scope**: formalizing acceptance of the project deliverables
* **Controlling scope:** controlling changes to project scope throughout the life of the project

**Scope creep**

**Variance**

* Variance is the difference between planned and actual performance

**Scope validation**

* **Validating scope**: formalizing acceptance of the project deliverables

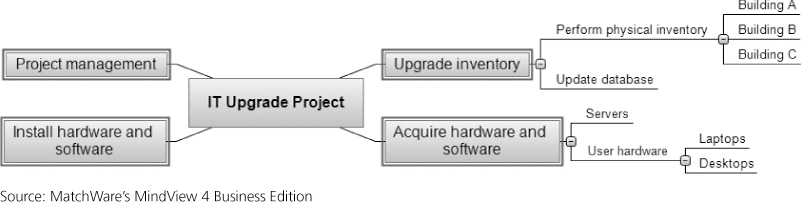
**Techniques used to create WBS**

* A WBS is a deliverable-oriented grouping of the work involved in a project that defines the total scope of the project
* WBS is a foundation document that provides the basis for planning and managing project schedules, costs, resources, and changes
* Decomposition is subdividing project deliverables into smaller pieces
* A work package is a task at the lowest level of the WBS
* The scope baseline includes the approved project scope statement and its associated WBS and WBS dictionary

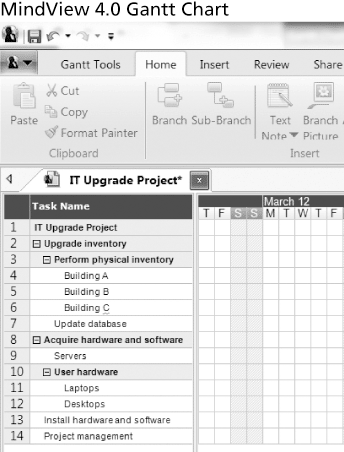
**Approaches to Developing WBSs**

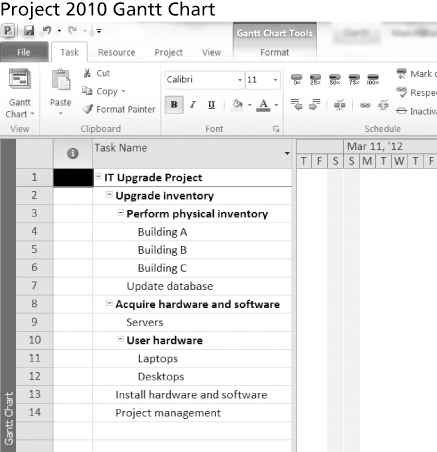
* Using guidelines: Some organizations, like the DOD, provide guidelines for preparing WBSs
* The analogy approach: Review WBSs of similar projects and tailor to your project
* The top-down approach: Start with the largest items of the project and break them down
* The bottom-up approach: Start with the specific tasks and roll them up
* Mind-mapping approach: Mind mapping is a technique that uses branches radiating out from a core idea to structure thoughts and ideas

**Figure 5-7. Sample Mind-Mapping Approach for Creating a WBS**



**Gantt Charts With WBS Generated From a Mind Map**





**Advice for Creating a WBS and WBS Dictionary**

* A unit of work should appear at only one place in the WBS.
* The work content of a WBS item is the sum of the WBS items below it
* A WBS item is the responsibility of only one individual, even though many people may be working on it
* The WBS must be consistent with the way in which work is actually going to be performed; it should serve the project team first, and other purposes only if practical
* Project team members should be involved in developing the WBS to ensure consistency and buy-in
* Each WBS item must be documented in a WBS dictionary to ensure accurate understanding of the scope of work included and not included in that item
* The WBS must be a flexible tool to accommodate inevitable changes while properly maintaining control of the work content in the project according to the scope statement

**Methods used to gather project requirements**

* For some IT projects, it is helpful to divide requirements development into categories called elicitation, analysis, specification, and validation
* It is important to use an iterative approach to defining requirements since they are often unclear early in a project

**Methods for Collecting Requirements**

* Interviewing
* Focus groups and facilitated workshops
* Using group creativity and decision-making techniques
* Questionnaires and surveys
* Observation
* Prototyping

**Chapter 6**

**Milestone**

* A milestone is a significant event that normally has no duration
* It often takes several activities and a lot of work to complete a milestone
* They’re useful tools for setting schedule goals and monitoring progress
* Examples include obtaining customer sign-off on key documents or completion of specific products

**Predecessors/successors**

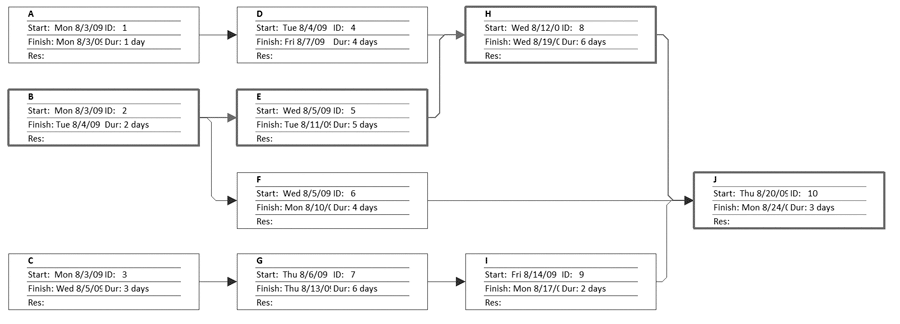
**Precedence diagramming method**

**Forward pass = why?**

**Backward pass = why?**

* Activities are represented by boxes
* Arrows show relationships between activities
* More popular than ADM method and used by project management software
* Better at showing different types of dependencies

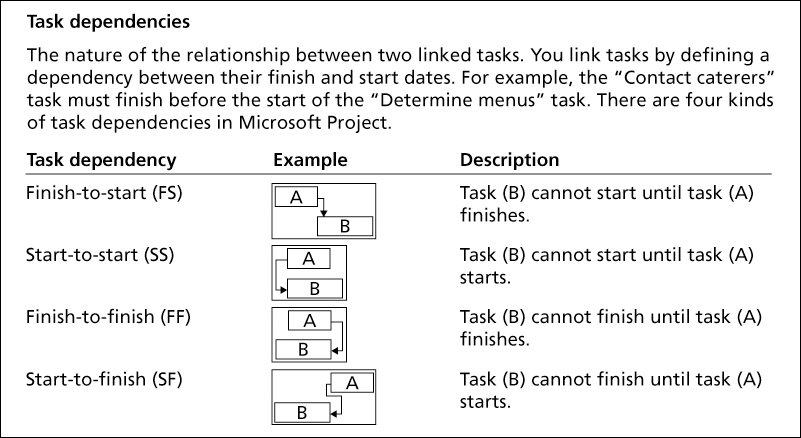
**Sample PDM Network Diagram**

****

* A forward pass through the network diagram determines the early start and finish dates
* A backward pass determines the late start and finish dates

**Task dependencies**

**SS, FF, SF, FS**

****

**Critical path = how to determine**

* CPM is a network diagramming technique used to predict total project duration
* A critical path for a project is the series of activities that determines the earliest time by which the project can be completed
* The critical path is the longest path through the network diagram and has the least amount of slack or float
* Slack or float is the amount of time an activity may be delayed without delaying a succeeding activity or the project finish date

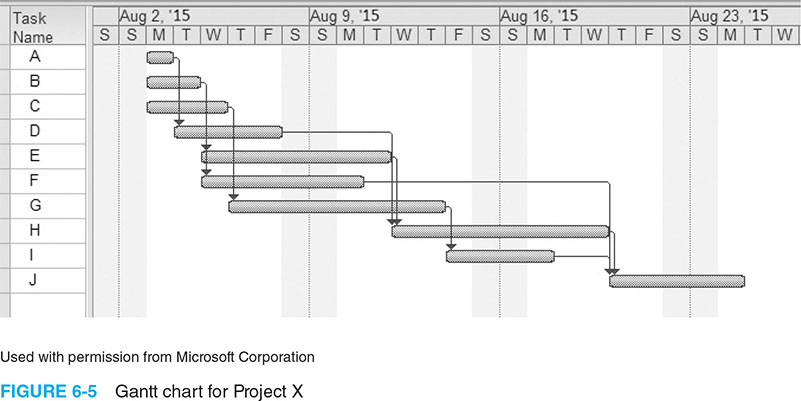
**Calculating the Critical Path**

* First develop a good network diagram
* Add the duration estimates for all activities on each path through the network diagram
* The longest path is the critical path
* If one or more of the activities on the critical path takes longer than planned, the whole project schedule will slip *unless* the project manager takes corrective action
* A project team at Apple computer put a stuffed gorilla on the top of the cubicle of the person currently managing critical task
* The critical path is *not* the one with all the critical activities; it only accounts for time
  + Remember the example of ***growing grass*** being on the critical path for Disney’s Animal Kingdom
* There can be more than one critical path if the lengths of two or more paths are the same
* The critical path can change as the project progresses

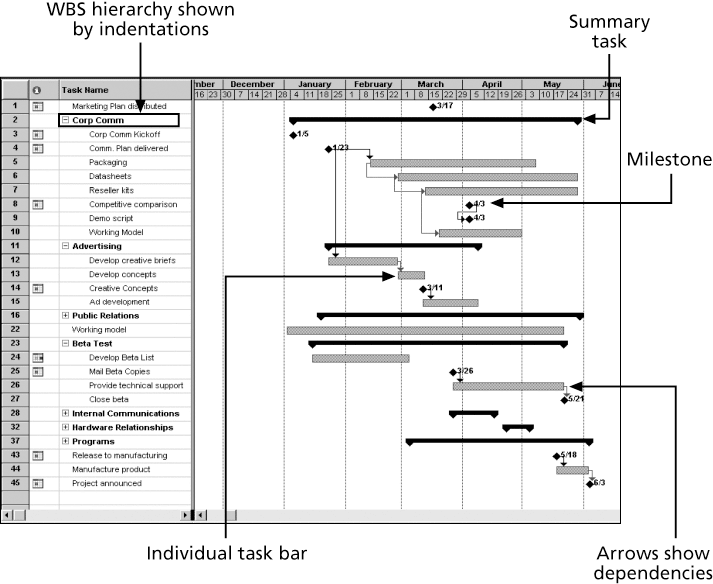
**Gantt charts = information depicted**

* + Gantt charts provide a standard format for displaying project schedule information by listing project activities and their corresponding start and finish dates in a calendar format
  + Symbols include:
  + A black diamond: a milestones
  + Thick black bars: summary tasks
  + Lighter horizontal bars: durations of tasks

Arrows: dependencies between tasks

* + 

**Gantt Chart for Software Launch Project**

****

**Adding Milestones to Gantt Charts**

* Many people like to focus on meeting milestones, especially for large projects
* Milestones emphasize important events or accomplishments on projects
* Normally create milestone by entering tasks with a zero duration, or you can mark any task as a milestone

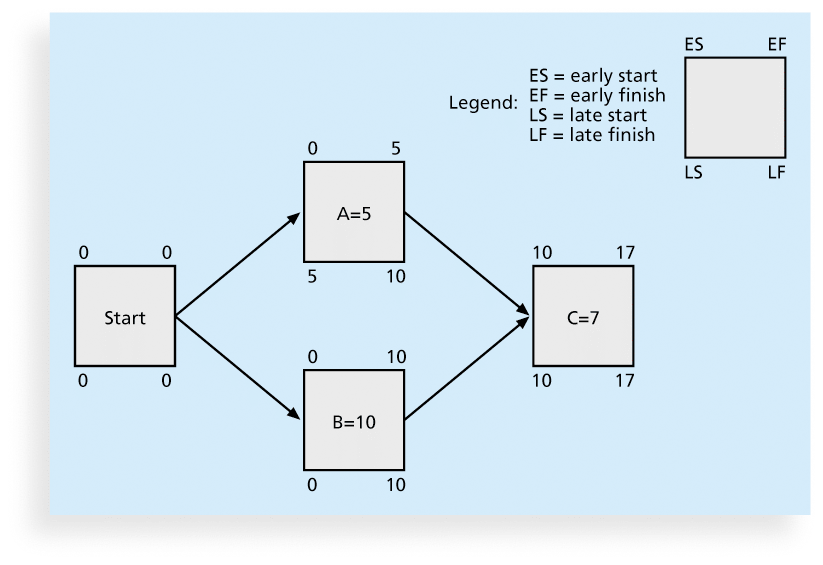
**Slack**

**Float**

**Early start, early finish, late start, late finish**

* Free slack or free float is the amount of time an activity can be delayed without delaying the early start of any immediately following activities
* Total slack or total float is the amount of time an activity may be delayed from its early start without delaying the planned project finish date
* A forward pass through the network diagram determines the early start and finish dates
* A backward pass determines the late start and finish dates

**Calculating Early and Late Start and Finish Dates**



Calculating Stack



Slack = LF – ES - duration

**Slack Time**

* Total Slack
  + Time that a task can be delayed without impacting the early schedule of the project.
  + Free Slack
  + Time a task can be delayed without impacting the early schedule of its successor tasks.

**Compressing the schedule**

**Crashing**

**Fast tracking**

* Three main techniques for shortening schedules
  + Shortening durations of critical activities/tasks by adding more resources or changing their scope
  + Crashing activities by obtaining the greatest amount of schedule compression for the least incremental cost
  + Fast tracking activities by doing them in parallel or overlapping them