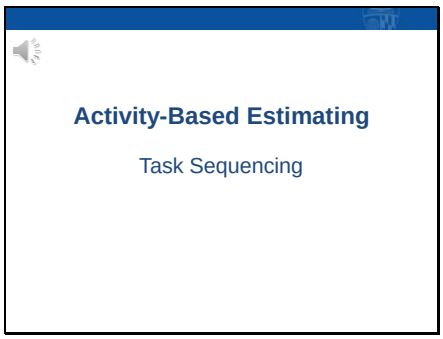
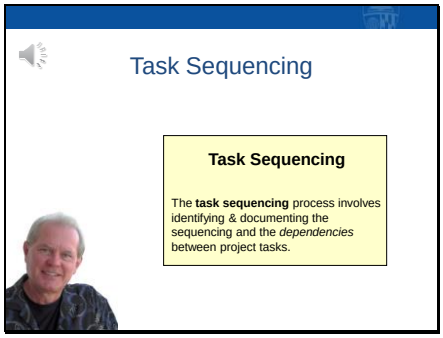
1



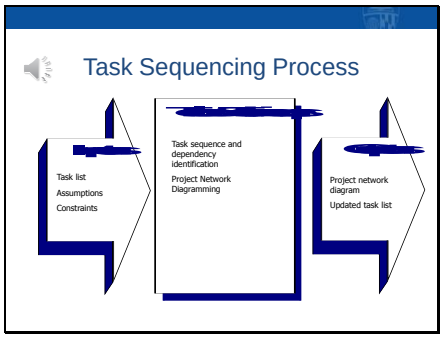
In this lecture we'll discuss the task sequencing process that is part of the activity-based estimating technique.

2



In activity-based estimating, task sequencing must be performed prior to estimating the project schedule. In task sequencing, we identify the dependencies between project tasks. Some tasks may be able to be done in parallel…meaning there's no dependency between them. And some tasks need to be performed in a particular order. For example, requirements need to be done before design, design needs to be done before coding commences, a test plan needs to be developed before tests can be executed…and so forth. Task dependencies have a significant impact on a project's schedule…so it is important to get them right.

3



Here's an input/output model for the task sequencing process.

The primary inputs are a list of the defined tasks, and any assumptions and constraints. Assumptions might be some of the same assumptions that went into the task definition process. An example of constraints might be that certain tasks have mandatory due dates.

The tools and techniques involved in the task sequencing process are primarily the identification of task dependencies and the construction of what is called a project network diagram. Task identification and construction of a project network diagram are very

important because they help form the foundation for estimating a project's schedule.

Some tasks will have what's called mandatory dependencies. These are dependencies that are inherent in the nature of the work that needs to be performed. For example, a design task must precede a coding task, a test plan must be developed before tests can be executed, and so forth.

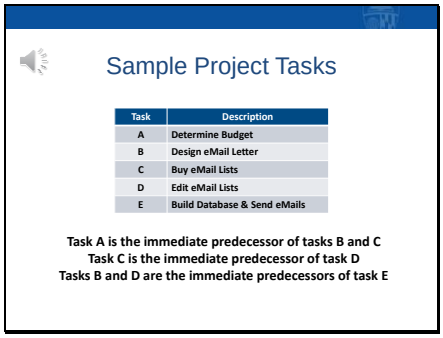
Some tasks may have discretionary dependencies. Discretionary dependencies are those that are defined by the project manager or whoever is responsible for planning the project parts. For example, as a manager, I may decide to break a programming task into sub-tasks and have each subtask assigned to separate programmers so that they can be worked on in parallel.

Another type of dependency is an external dependency. External dependencies arise due to activities that have to be completed, but are outside of the direct sphere of influence of the project manager. For example, pieces of a project may be contracted out to external suppliers. Some of our other project tasks may be dependent upon those that were contracted out, but the day-to-day management of those tasks is outside of our organization.

Project network diagramming involves building a graphical model of the task dependencies. Many project management tools have features that enable building this diagram, and there are different notational formats that might be used. One type of project network diagram is called a PERT chart. A Gantt chart is another common tool for illustrating task dependencies…but the dependencies can be harder to see, depending upon how the Gantt chart is constructed and how much detail is included.

The outputs to the task sequencing process include the project network diagram and possibly an updated task list. As an example, a previously defined task may have been divided into sub-tasks during the task sequencing process.

4



Let's take an example to see how task sequencing and dependency identification are done. Let's assume I am planning a project for a marketing campaign that will advertise my project planning seminars.

I've identified five tasks for this project. Task A will involve determining my overall strategy and budget. For example, I'll need to decide whether to market in-person seminars or virtual seminars. If I decide on in-person seminars, I'll need to determine how many cities I'll hold the seminar in and when the seminars will be held. If I decide on virtual seminars, I'll need to determine how many I will hold and when I will hold them. These decisions will then go into establishing my overall budget and who I will advertise to. Task B consists of designing my email marketing letter that will be sent to potential attendees. Task C involves buying one or more email lists. Task D involves reviewing and refining the email lists to possibly exclude some recipients. And task E involves building the email database on my bulk mail service and scheduling the marketing pieces to be sent. That's pretty much it.

Now, some of these tasks have to logically be done in a particular sequence, and some may be done in parallel.

Task A must be done first, since it determines the contents of my email marketing letter and also determines the number and types of email lists I will buy.

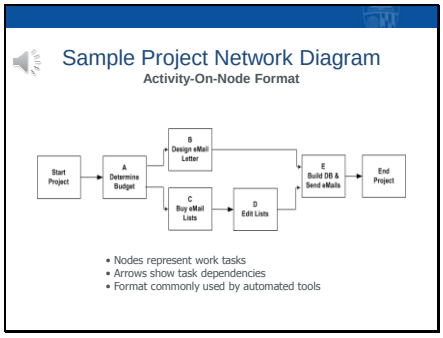
Tasks B and C can be done in parallel, so they are not dependent upon each other, but task A must complete before they can begin. So…tasks B and C are dependent upon task A. In network diagramming terminology we say that task A is the immediate predecessor of tasks B and C.

Task D can also be done in parallel with task B, but can't be started until task C is completed…so there is a dependency between task C and task D. Task C is the immediate predecessor of task A.

The final task, task E, can't be started until both task B (designing the email letter) and task D (editing the mailing list) are completed…so tasks B and D are immediate predecessors of task E.

Now…let's see how we can use this information to construct a project network diagram.

5



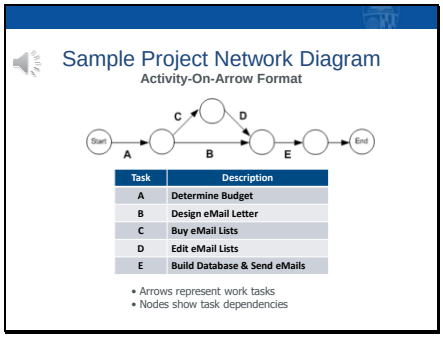
Here's an example of a project network diagram, shown in what is called activity-on-node format. In this diagram, each rectangle is called a node, and is associated with one of the project tasks. The arrows show the dependencies between the tasks, and the left-to-right layout indicates the sequence of tasks. Some project network diagrams also show a start node and an end node, as I've illustrated here.

The project network diagram is very easy to interpret. This diagram models the five tasks associated with the email marketing campaign for my project management seminar series. The diagram indicates that task A must be done first. Once task A is completed, tasks B and C can start and be done in parallel. Task D, culling the mailing lists, can start as soon as task C is finished, and task D can also be done in parallel to task B. The final task, task E, can start only when both task B and task D have completed.

Given that I have time estimates for each task, which, for

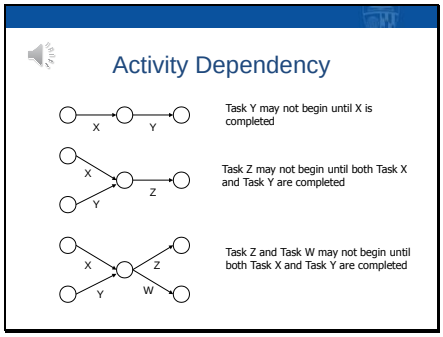
a project like this would be made in terms of elapsed time…or duration…the project network diagram can be used to calculate the estimated project schedule.

6



Here's a network diagram for the same project using a different format, called an activity-on-arrow format. In this format the arrows represent work tasks, and the nodes show the task dependencies. A node with multiple outgoing arrows indicates that the tasks associated with those arrows can be done in parallel…and a node that has one or more incoming arrows indicates that the task to its right can't start until all the incoming tasks have completed.

7



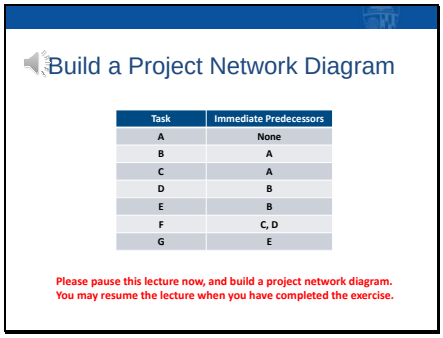
Here's a few more dependencies that commonly occur in project network diagrams. It's pretty straightforward, so I'll leave it for your reference.

It's really important in project planning to identify and analyze correct task dependencies, because the dependencies will be a key driver in determining the actual project schedule.

If I have time estimates for each task, which, for a project like this would be made in terms of elapsed time…or duration…the project network diagram can be used to calculate the estimated project schedule very quickly. If I use a project planning tool, I can build the project network diagram right inside the tool and it will automatically calculate the project schedule for me.

We'll see how to do that in the next lecture.

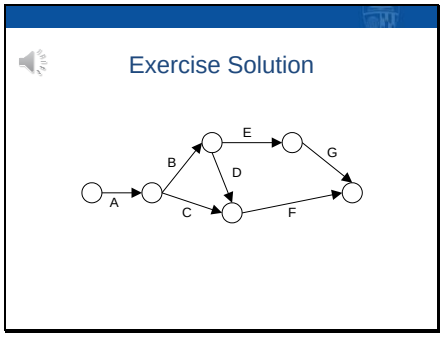
8



At this point, I'd like you to try your hand at building a project diagram. The table consists of a task list and associated dependencies. You may use either the activity-on-node or the activity-on-arrow diagram notation.

Please pause this lecture now, and resume it once you have completed the exercise.

9



Here's a solution to the exercise. In this solution, I used the activity-on-arrow diagramming notation. If you used the same notation your solution should look like this. If you used the activity-on-node notation, then convert your solution to the activity-on-arrow notation to verify that it is the same.