

Project Management Technologies PMG 312

Prepared by Dr. Majid Saad
PMP, PMI-ACP, PMI –ACP



PMG 312- L8

S4 - Tools and techniques for developing the project plan
Ch. 9 Creating the project schedule

Project Management Technologies PMG 312

PMG 312- L8

S4 - Tools and techniques for
developing the project plan
Ch. 9 - Creating the project
schedule

Prepared by

Dr. Prepared by Dr. Majid Saad

Section IV: Tools and techniques for developing the project plan

Chapter 9 Creating the project schedule

Work
activities

Sequencing
activities

Software
modeling

Predecessor
relationship
codes

Work unit
resources

Estimating
activity
durations

Developing
the project
schedule

MSP
schedule
generation
mechanics
(stage one)

Section IV: Tools and techniques for developing the project plan

Chapter 9 Creating the project schedule

Sample first cut schedule problem

Results for the first cut plan

Critical path

Supplementary technical reference

Plan status mechanics

Forward path calculations

Backward pass calculation

MSP equivalent output

Tools and techniques for developing the project plan

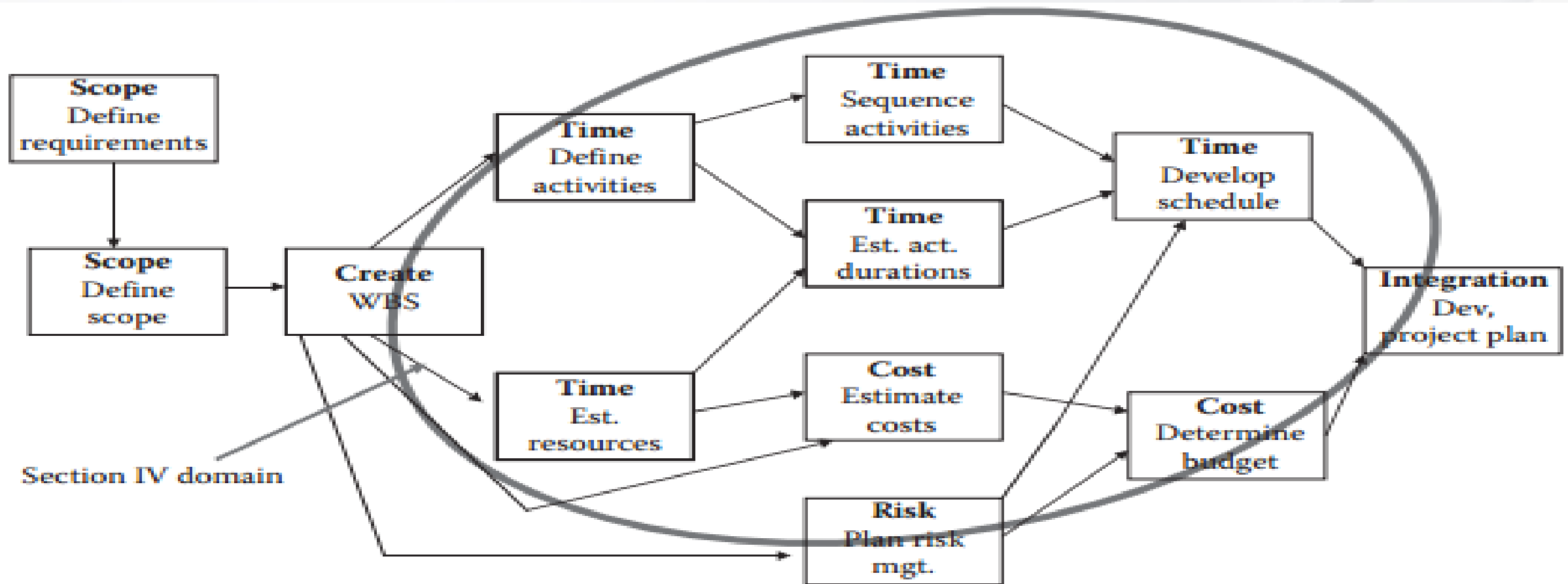


Figure IV.1 Project time and cost plan model processes. (Adapted from Project Management Institute (PMI). 2008. PMBOK® Guide, 4th ed. Newtown Square, PA: Project Management Institute.)

Tools and techniques for developing the project plan

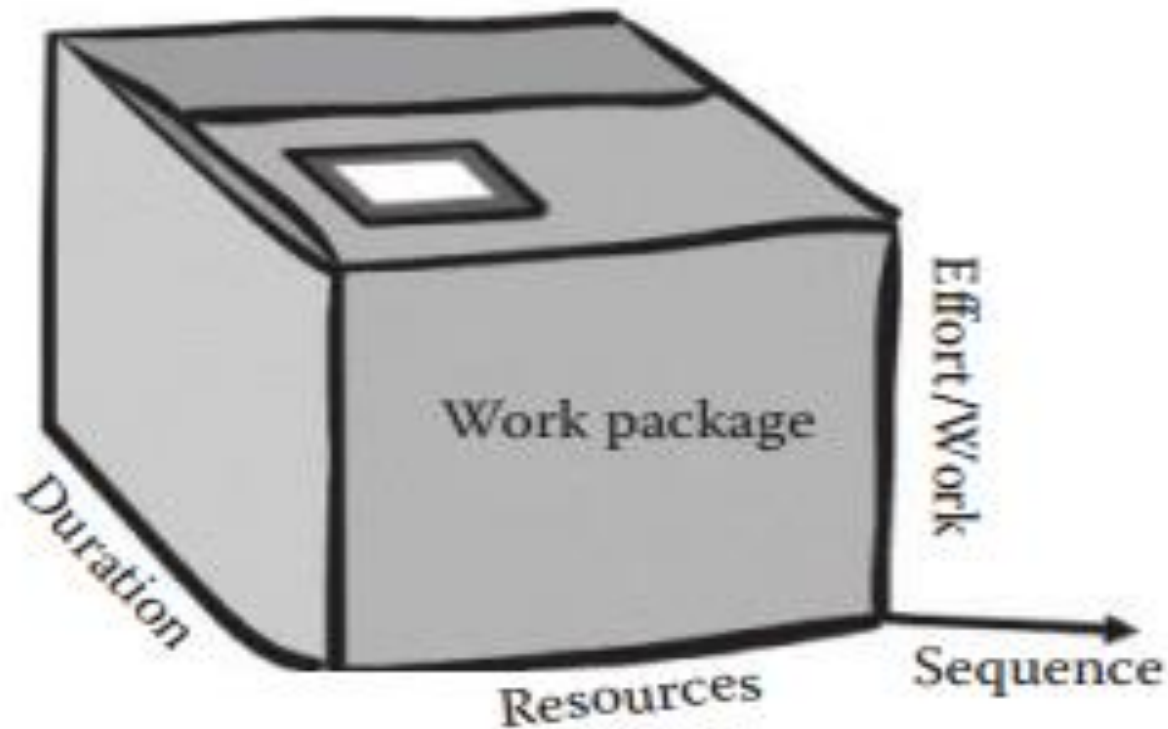


Figure IV.2 Fundamental project work unit.

Tools and techniques for developing the project plan

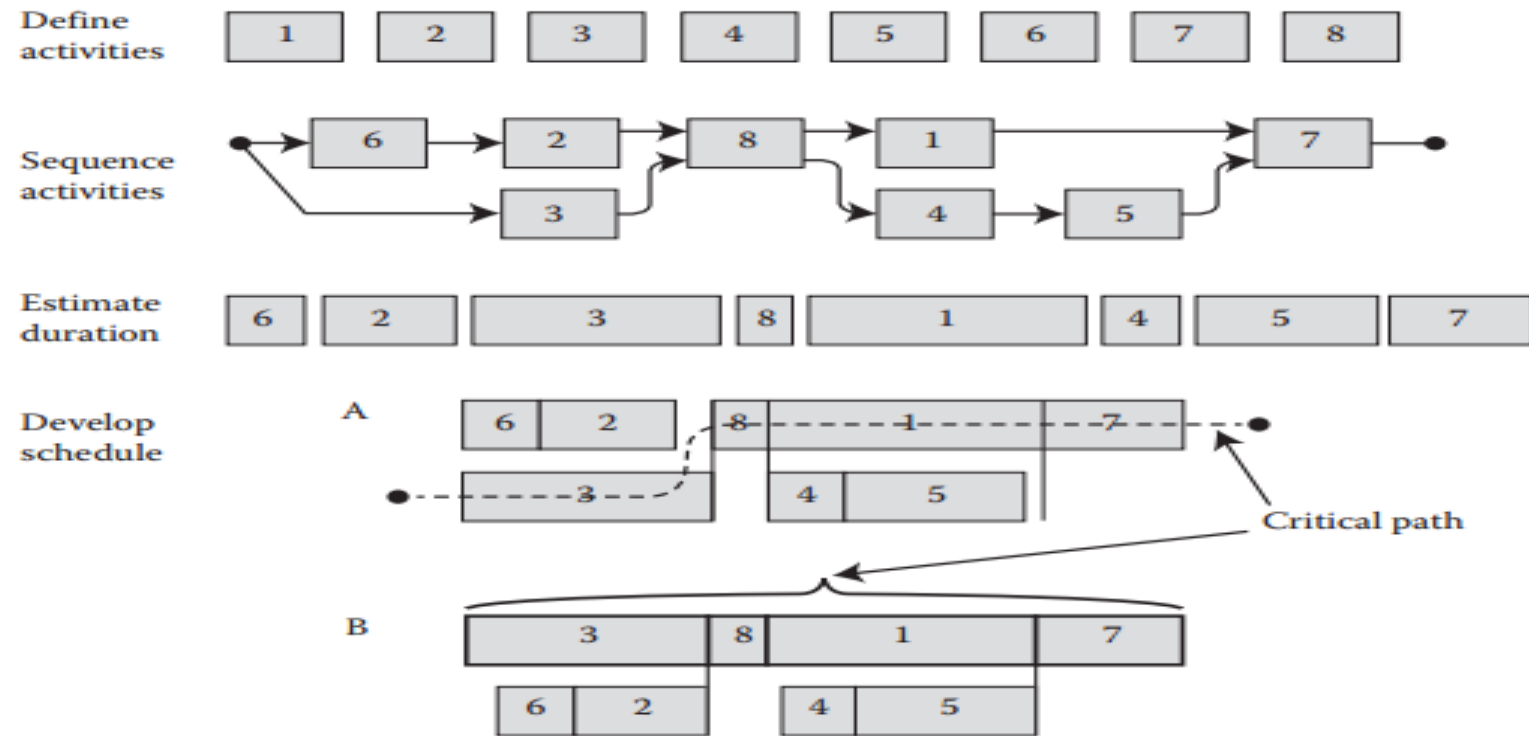


Figure IV.3 Project schedule process sequence.

Work activities and Sequencing activities

- ❑ The view shown in Figure 9.1 is called a Predecessor Diagram Model (PDM).
- ❑ Figure 9.1 illustrates a sample PDM network.
- ❑ This format is useful for examining a sequence plan and evaluating it both technically and managerially.
- ❑ However, as the project grows in size, the graphical method becomes harder to maintain and other computer-based methods then become preferable.

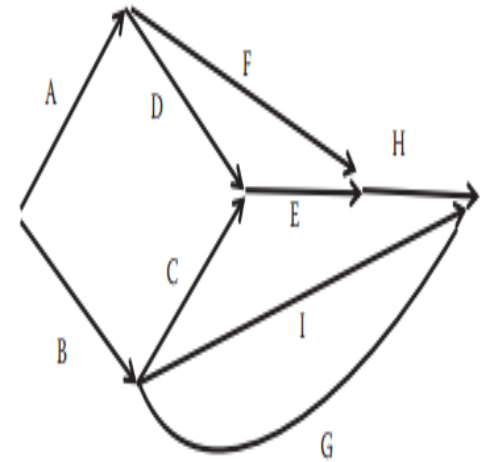


Figure 9.1 PDM network schedule model.

Work activities and Sequencing activities

- ❑ In this case, a graphical WBS view has to be “flattened” into a list containing WBS ID, task, duration, and predecessor.
- ❑ At some high level of detail, it is still easier to visualize sequence graphically.
- ❑ The rule of thumb says that a graphical network becomes unwieldy as the size grows beyond 50 or so work units.
- ❑ Maturation of supporting desktop PC tools has greatly aided this planning activity over the last few years.

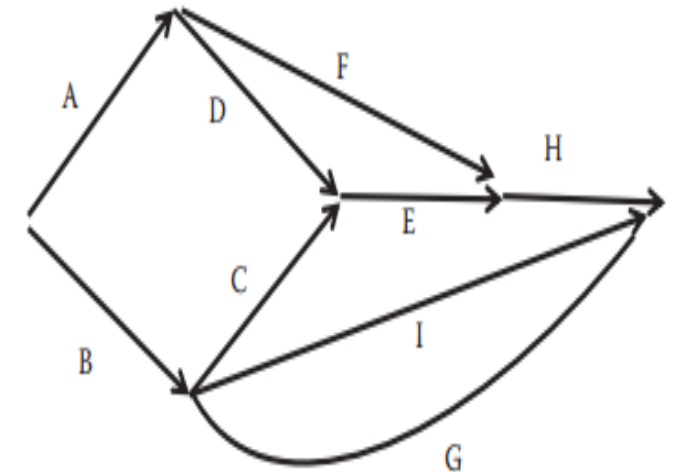


Figure 9.1 PDM network schedule model.

Software modeling

- ❑ Computer software does not handle arrows or graphics well, so in order to move the problem into a computer-centric model, it is necessary to translate this schematic view into a more data-oriented format.
- ❑ The data shown in Table 9.1 mirror the network model shown in Figure 9.1 and this is much closer to a required computer data input format.
- ❑ Note that this shorthand form for sequence coding defines how the various activities are linked to each other.
- ❑ In this way, the defined task sequencing can be correctly modeled by the software.

Table 9.1 Predecessor Coding

Activity	Predecessor
A	— —
B	— —
C	B
D	A
E	D,C
F	A
G	B
H	E,F
I	B

❑ In order to test your understanding of these concepts, the list below contains a few sample interpretations of the activity relationships in Table 9.1.

- 1) Activity A has no predecessor constraint, therefore it start immediately.
- 2) Activity C can start after B is finished.
- 3) A more complex coding example occurs with activity H. In this case, activities E and F have to be finished before H can start.

Table 9.1 Predecessor Coding

Activity	Predecessor
A	— —
B	— —
C	B
D	A
E	D,C
F	A
G	B
H	E,F
I	B

Software modeling

- ❑ For example, if a WBS task defines that a room is to be painted and an adjacent room also needs to be painted, the scheduling sequence could be changed to specify painting the two rooms in sequence rather than simply linking the task to construction of the wall or room.
- ❑ This in effect changes the technical linkage.
- ❑ Logically, once the painters are working in an area, it would be more efficient to just have them move next door and continue working rather than pack up their equipment and come back another time.
- ❑ These points illustrate that the predecessor coding is not just a technical work linkage.
- ❑ Predecessor coding represents the creative part of the schedule development exercise and as illustrated is both a technical and art-oriented process.

Table 9.1 Predecessor Coding

Activity	Predecessor
A	— —
B	— —
C	B
D	A
E	D,C
F	A
G	B
H	E,F
I	B

Predecessor relationship codes

- ❖ The linking process described thus far is called FS, meaning finish–start (FS).
- ❖ This is the end-to-end activity (arrow) relationship specifying that the predecessor work unit has to be finished before its successor can start.
- ❖ However, there are other task relationship options that can be coded to better model the desired work sequences.

Predecessor relationship codes

□ The other sequencing options are:

- **Start/Start (SS)**—this indicates that the two linked activities would start at the same time. So, a code of SS6 would define that the activity would start at the same time as the activity on line 6.
- **Finish/Finish (FF)**—this code indicates that the named activity would need to finish at the same time as its reference activity. So, a code of FF6 would define that the activity should finish at the same time as the activity on line 6.
- **Start/Finish (SF)**—this is a little used relationship for most projects. Its interpretation is that the successor task cannot be completed until the predecessor begins. This format basically fits the Just-in-Time manufacturing approach designed to “pull” the predecessor activity rather than a “push” relationship like the other options reflect.

Predecessor relationship codes

- **Lag**—this is a supplementary relationship to each of the four basic codes above. The term lag means that the coded task obeys one of the basic relationships, but then must wait for some further specified time.
- One easy way to remember this logic is to recall again the concrete pouring example.
- In this case, the required technical sequence for these two events is to pour the concrete slab (task A) and then paint the surface of the slab (task B). There is an obvious required technical sequence between these two tasks; however, in this case the concrete must also be allowed to dry first and this drying time represents a required lag. If we assume the concrete must dry for 5 days, then the predecessor code for task B would be $A + 5$.

Predecessor relationship codes

- **Lead**—a lead relationship is just the opposite of a lag in that the linked task can start before the predecessor is finished. So, the code A-5 would say that B could start 5 days prior to completion of A.
- ❑ It should also be obvious that the predecessor coding activity has the potential to stretch or compress a schedule depending upon the lead, lag, or other predecessor codes.
- ❑ Think about the impact of changing a relationship from FS to SS, which in effect shortens the total sequence for the two activities involved.
- ❑ The project team must understand the implications of these options and their cumulative impact on the resulting schedule.

Work unit resources

- ❑ Our model theory has defined the project scope as being represented by a collection of work elements consisting of Work Packages (target sized at 2 weeks duration and 80 work hours), planning packages (not decomposed yet), and summary groups of lower level boxes.

Work unit resources

- ❑ Review a WBS layout and verify that you understand how these three box types would be represented in the structure. Once these work units are defined, the following steps are required to allocate the specific resources necessary to produce the defined output for each unit:
 - 1) The initial step in the process involves the decision for allocating various human resource skills required to execute each defined work unit in the WBS. As an example, a work package estimate would specify the total work hours required for a carpenter, plumber, and an engineer.
 - 2) Material resource estimates are defined for each work unit based on the technical specifications.
 - 3) Other supplementary costs are estimated for each work unit. One common example for this category is travel costs or common equipment needed by the work unit.

Estimating activity durations

- ❑ Duration estimates are driven by decisions related to work effort by various labor skills.
- ❑ To translate work into activity duration, it is necessary to define the number of actual bodies by skill type to be allocated into that work unit.
- ❑ This in turn yields the planned work duration of the activity.
- ❑ To illustrate this point, assume that the work package activity is defined to be 160 hours of work.
- ❑ Assuming the skill required is defined, the next step in the sequence involves the actual work schedule of the resource.

Estimating activity durations

- ❑ A typical work schedule assumption is 40 hours per week (standard 8×5 work week)
- ❑ So, if one person was allocated to this task, the activity duration would be 20 days or 4 work weeks.
- ❑ However, the Project Manager may decide that he wants this done in 5 days, if possible.
- ❑ In concert with the work unit supervisor, it is decided to allocate four full-time resources and they believe this will accomplish the desired 5-day goal.
- ❑ This second scenario shows how the resource allocation decision impacts the duration estimate and resulting project schedule.

Estimating activity durations

- ❑ From this discussion, it should be obvious that the complexity of the resource allocation process is significant.
- ❑ Finding the balance among available resources, optimum task times, and compressed schedules is one of the most stressful parts of planning and executing the project.
- ❑ It is important to note at this point that the planned resource allocations are really ideal scenarios.
- ❑ There is no assurance that the defined skills and quantities are available since the actual time period for the work is not yet defined.
- ❑ This issue will need to be dealt with before the plan is approved, but that will be deferred for now.

Developing the project schedule

- ❑ There are essentially four decision variables necessary to create a first cut schedule.
- ❑ These are:
 - 1) WBS code ID
 - 2) Work unit name
 - 3) Estimated duration (work days)
 - 4) Predecessor code
- ❑ To develop a simple example, assume that we have only four activities in our sample project and each activity is estimated at 5 days duration.
- ❑ The data setup for translating this into MS Project (MSP) is outlined in Figure 9.2.

	0	WBS	Task Name	Duration	Predecessors
1		1	Total Project	20 days	
2		1.1	A	5 days	
3		1.2	B	5 days	2
4		1.3	C	5 days	3
5		1.4	D	5 days	4

Figure 9.2 Project schedule parameters.

Developing the project schedule

- ❑ Notice that the project schedule is essentially derived from the raw data described.
- ❑ **However, there are two additional less obvious scheduling parameters embedded in the computer version.**
 - ❑ First, the resource work schedule has to be defined since the software uses this to translate how a particular work unit can be executed within the resource's work schedule.
 - ❑ Second, the start date for a project must be set in order for the sequence to properly be shown on a calendar view

Developing the project schedule

- ❑ Observe in Figure 9.3 how the software recognizes nonwork time (gray columns in the Gantt bar area).
- ❑ These nonworking times affect the completion date.
- ❑ For the schedule calculation to be accurate, the calendar must reflect nonworking times such as holidays and vacations for the assigned workers.
- ❑ These two internal components allow a calculation of calendar dates to be added to the raw work duration values that a simple Predecessor Diagram Model PDM-type network would show.
- ❑ This is another advantage of using a computer model over manual processes.

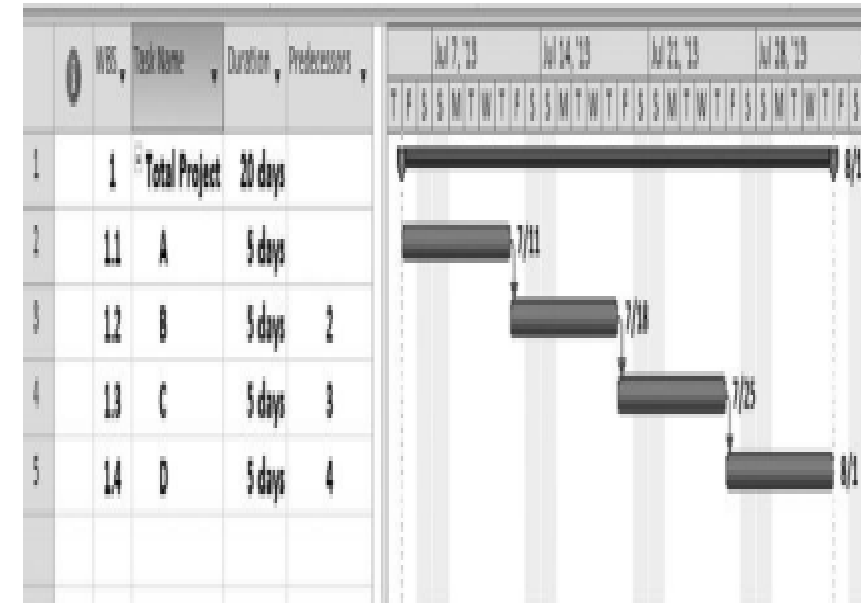


Figure 9.3 First cut project schedule.

Developing the project schedule

- ❑ This first cut version of the plan is quite easy to derive, but it would not be fair to call this the final project schedule.
- ❑ However, it does represent a significant starting place for the project planning model. Because of various other project dynamics and stakeholder pressure, we should expect to go through more decision iterations of this plan before finally getting an approved package of tasks, schedule, and budget that is acceptable to the various decision-makers.
- ❑ It should be anticipated that some stakeholder groups will not like the schedule produced by this initial calculation.
- ❑ Some will think the cost is excessive, whereas others will want more or less functionality (scope).

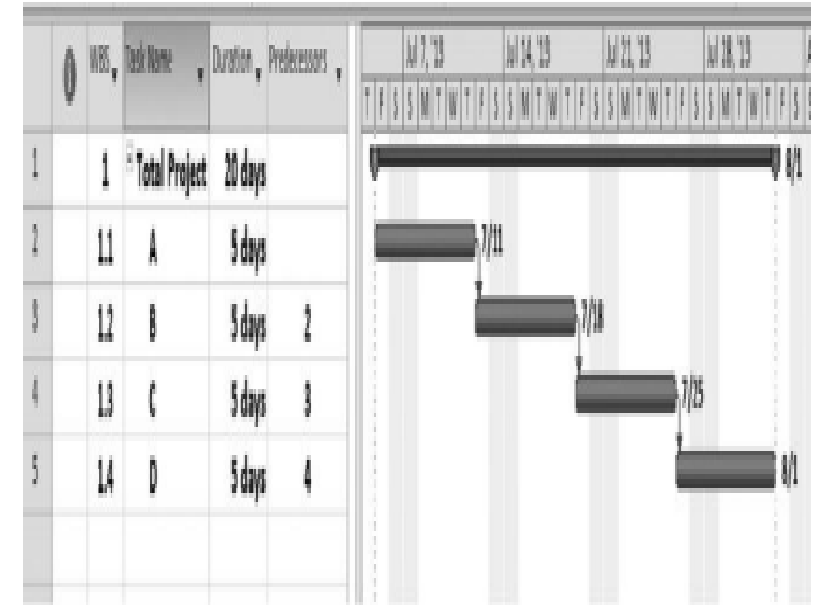


Figure 9.3 First cut project schedule.

Developing the project schedule

- ❑ In essence, all of the triple constraints will be challenged in conflicting ways. This will necessitate manipulation of activities, durations, and resources until a suitable combination is achieved.
- ❑ For now, we will just deal with the basic mechanics to better illustrate the model schedule generation process.
- ❑ The good news in all this is that any needed future changes to this view can be made quickly by changing the respective driver data and the results will be automatically translated from that.

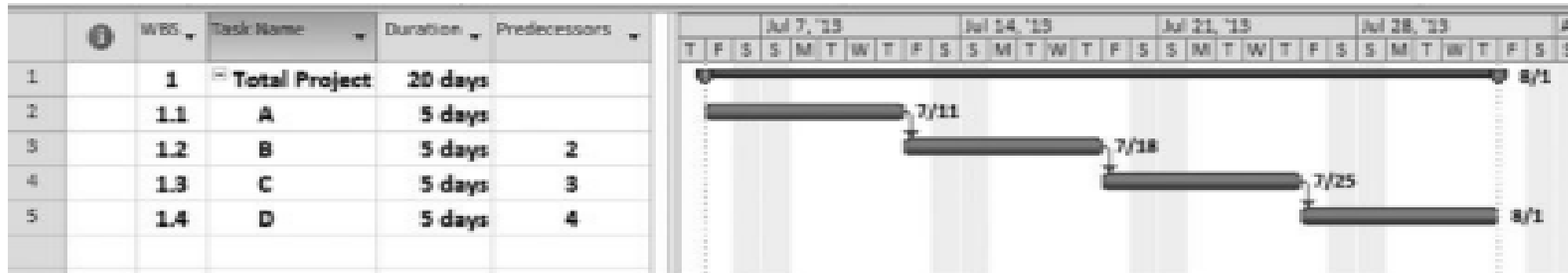


Figure 9.3 First cut project schedule.

Sample first cut schedule problem

- ❑ This example problem is designed to illustrate basic mechanics for translating scope-oriented data into a first cut project schedule.
- ❑ A spreadsheet with this table data are stored on the publisher's website for your use under the file name Table 9.2 First cut plan parameters.
- ❑ Table 9.2 represents that file.
- ❑ Note that the data is in WBS flattened format.
- ❑ The goal of this example is to manipulate this data using the keystroke checklist.
- ❑ The following note list will help guide you through this process:
 - 1) The top of the WBS is line #1
 - 2) Bolded line numbers are summary tasks (i.e., 2, 6, 11, etc.)
 - 3) Line 19 is a summary task for #20 and #21 (they must be indented)
 - 4) Task #33 is a milestone (zero duration)
 - 5) Load the scope data into MSP and set the start date to April 1, 2014
 - 6) Indent the data to mirror the WBS structure defined
 - 7) Once the activity data are loaded and the first cut schedule produced, insert a recurring task after #30 defined as "Project Management" (see the MSP checklist)
- ❑ Load the table data below into MSP and manipulate these data using the "Stage One Keystroke" checklist described earlier.
- ❑ Also, review the companion video tutorials that cover these specific commands.
- ❑ Both of these sources are helpful in learning these important steps. Once accomplished, this represents basic MSP scheduling competency.

Sample first cut schedule problem

Table 9.2 First Cut Project Plan Parameters

WBS	No.	Task name	Predecessors	Duration
1	1	Desktop refresh project		
1.1	2	Hardware selection		
1.1.1	3	Determine technical specifications for hardware		4 days
1.1.2	4	Selection of hardware vendor	3	1 day
1.1.3	5	Test hardware that is selected	4	8 days
1.1.4	6	Procurement of hardware	5	10 days
1.2	7	Software selection		
1.2.1	8	Licensing negotiations with software vendors		5 days
1.2.2	9	Test software from each vendor	8	8 days
1.2.3	10	Procurement of software	9	10 days
1.3	11	Integration of hardware and software		
1.3.1	12	Implement new security model	18	5 days
1.3.2	13	Customization for each organization	12	7 days
1.3.3	14	Testing of the integrated desktop	13	10 days
1.4	15	Verifying existing infrastructure		
1.4.1	16	Test server capacity	5,9	5 days
1.4.2	17	Test the network hardware and software	16	5 days
1.4.3	18	Procure necessary infrastructure upgrades	17	10 days
1.5	19	Training for employees		
1.5.1	20	Create documentation materials	12	15 days
1.5.2	21	Technical architecture documentation	12	8 days
1.5.3	22	User manuals	12	20 days
1.5.4	23	Review documentation materials	20	2 days
1.5.5	24	Conduct user training	23	12 days
1.6	25	Deployment by organizational area		
1.6.1	26	Marketing deployment	24	8 days
1.6.2	27	Engineering deployment	26	8 days
1.6.3	28	Finance deployment	27	8 days
1.6.4	29	Executives deployment	28	8 days
1.6.5	30	Legal deployment	29	8 days
1.6.6	31	IT deployment	30	8 days
1.6.7	32	Administration deployment	31	8 days
1.7	33	Project complete	32	0 days

Results for the first cut plan

- The WBS resulting from Table 9.2 manipulation is shown in Figure 9.4 (this example is created by WBS Chart Pro).

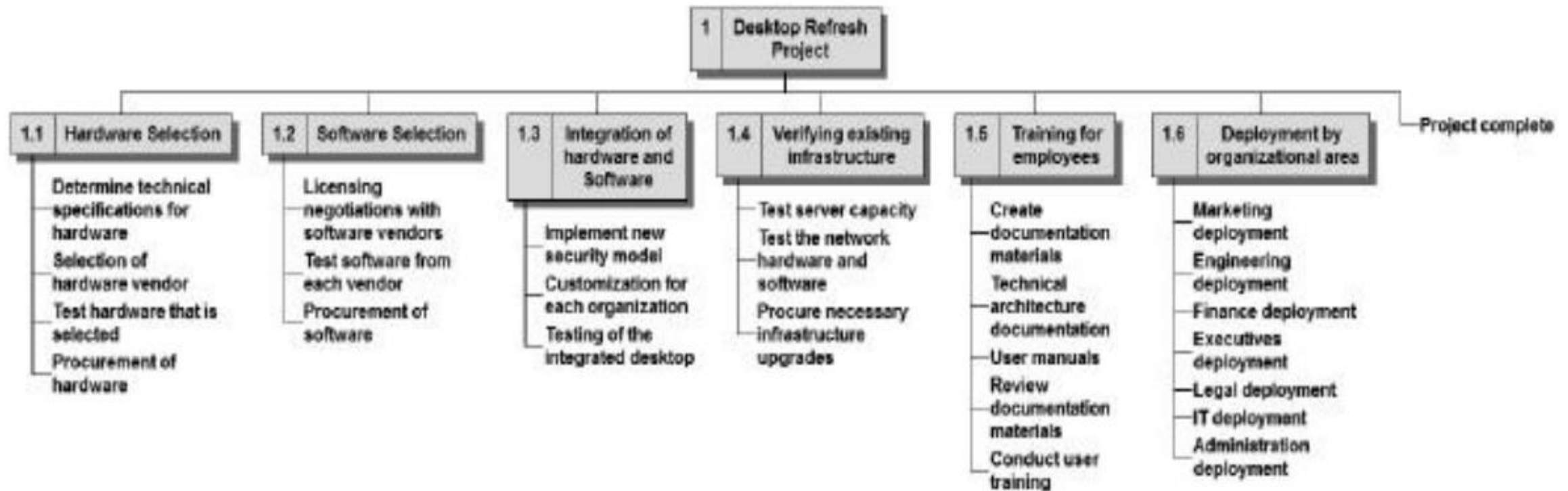


Figure 9.4 WBS for first cut plan.

Critical path

- ❑ Understanding both critical path and slack concepts is vital for effective time management and requires that the PM keep these parameter values in mind as the project unfolds.
- ❑ Also, recognize that errors in duration estimates will cause the slack values to change during project execution, so this is not a static view by any stretch.

Critical path

- ❑ This also helps to facilitate prioritization of resource allocations across the project plan.
- ❑ For example, if there are two activities occurring within the same time range, one critical and one noncritical that are competing for the same resource, the Project Manager can allocate the resource to the critical activity first, then use the slack or float time on the other to delay the start of the noncritical activity.

❑ Float or Slack in Project Management

- ❑ A concept related to, and crucial for using the Critical Path Method is float or slack.
- ❑ In project management, “float” defines the amount of time a task can be delayed without causing a delay in:
 - ❑ Any subsequent, dependent tasks - called “free float”
 - ❑ Any delay in the overall project - called “total float”
 - ❑ Any activity or task on the critical path has zero float. That is, you can’t delay it at all without causing a delay in the project or dependent tasks.
- ❑ However, there are plenty of other activities in the project that can be delayed. The quantification of this delay is called the “float”.

Critical path- Problem

□ The following details are available regarding a project:

- Determine the critical path, the critical activities and the project completion time.

<http://arts.brainkart.com/>

Activity	Predecessor Activity	Duration (Weeks)
A	-	3
B	A	5
C	A	7
D	B	10
E	C	5
F	D,E	4

Supplementary technical reference

- ❑ Some readers may not wish to go through this upcoming set of more technical schedule calculations since MSP performs similar calculations.
- ❑ However, the manual graphical view offers more insights into the actual mechanics involved, so this section is defined as an optional technical reference.
- ❑ Manipulation of the project network sequence diagram involves a fundamental set of calculations used to produce both the critical path and noncritical path (slack) calculations embedded in the project network plan

Plan status mechanics

- ❑ After supplying the basic activity, duration, and predecessor parameters for defining the network plan, there are six management-oriented parameters that can be calculated from these data.
- ❑ These are:
 - 1) ES or T(E)—early start; earliest time the activity can commence.
 - 2) LS or T(L)—late start; latest time the activity can start and not affect project schedule.
 - 3) EF—earliest time that an activity can finish.
 - 4) LF—latest time the activity can finish and not affect project schedule.
 - 5) Total slack (float)—the difference between the earliest date that the activity can start and the latest date the activity can start before delaying the completion date (LS–ES).
 - 6) Free float—the amount of time an activity can be delayed before any successor activity will be delayed (EF–ES).
- ❑ A schematic view of this type of relationship is shown in Figure 9.5.

Plan status mechanics

- ❑ Table 9.3 defines a sample project activity list that will be used to show the manual network mechanics.
- ❑ From this definition, the following arrow-on-arrow network shown in Figure 9.6 can be produced.
- ❑ Review the network setup to be sure that this setup step makes sense as it is the base for the remainder of the mechanics.

Table 9.3 Sample Project Definition

Activity	Duration	Predecessor
A	3	—
B	4	—
C	4	A
D	6	B
E	5	C
F	2	C
G	4	D,E
H	3	F
I	5	G,H

Plan status mechanics

Table 9.3 Sample Project Definition

Activity	Duration	Predecessor
A	3	—
B	4	—
C	4	A
D	6	B
E	5	C
F	2	C
G	4	D,E
H	3	F
I	5	G,H

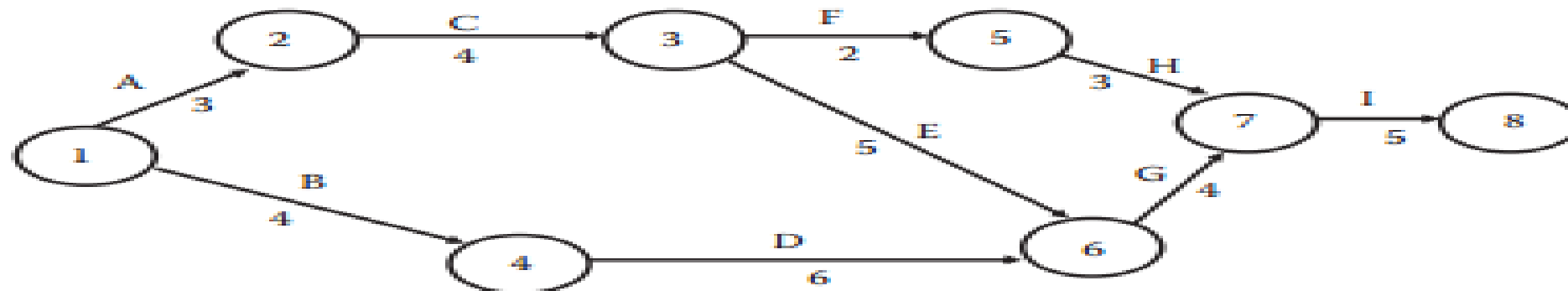


Figure 9.6 Sample activity-on-arrow (AOA) network.

Forward path calculations

- ❑ Step 2 of the mechanical plan calculation is to perform what is defined as a forward pass.
- ❑ This process basically adds the arrow lengths to calculate the total duration time moving from left to right through the network, while obeying the predecessors.
- ❑ Early times to reach each node (end of arrow) are recorded by the format of “T(E),” which is the notation for earliest time to reach that point in the network.
- ❑ Figure 9.7 shows the calculated values for each of the nodes.
- ❑ These example calculations are quite straightforward except for nodes 6 and 7.

Table 9.3 Sample Project Definition

Activity	Duration	Predecessor
A	3	—
B	4	—
C	4	A
D	6	B
E	5	C
F	2	C
G	4	D,E
H	3	F
I	5	G,H

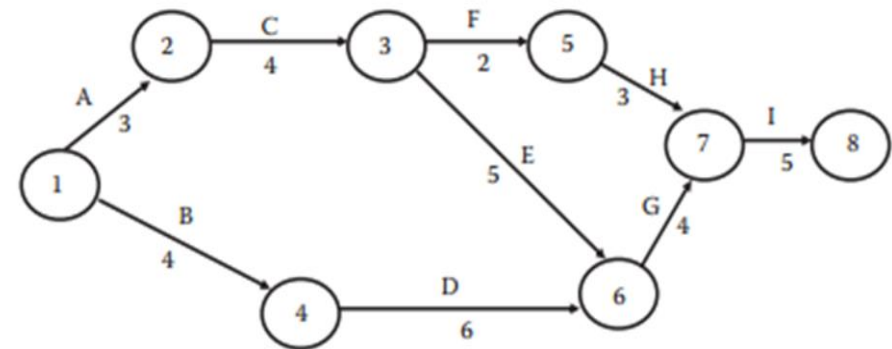


Figure 9.6 Sample activity-on-arrow (AOA) network.

Forward path calculations

- ❑ For node 6, two incoming activity paths have to be considered for the activities D and E.
- ❑ Node 4 has a T(E) value of 4 and activity D has a duration time of 6, therefore, the resulting value for node 6 would be 10.
- ❑ However, the other parallel path coming into node 6 starts at node 3 with a value of 7 and an activity E time of 5, so that path value is 12.
- ❑ For forward path calculations, the highest value among multiple incoming paths is selected for such parallel paths, thus we assign the value 12 for T(E) at node 6.

Table 9.3 Sample Project Definition

Activity	Duration	Predecessor
A	3	—
B	4	—
C	4	A
D	6	B
E	5	C
F	2	C
G	4	D,E
H	3	F
I	5	G,H

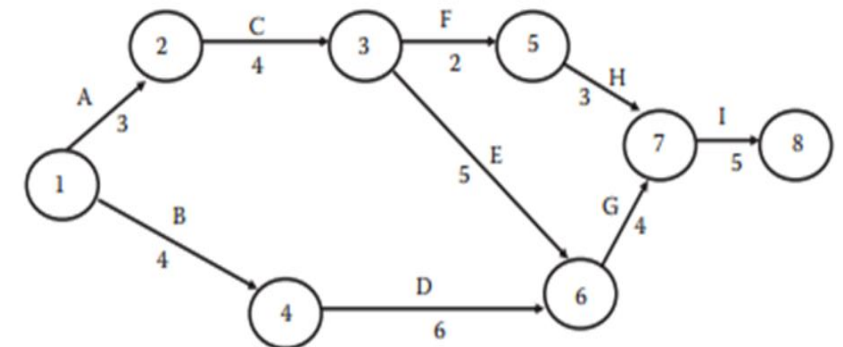


Figure 9.6 Sample activity-on-arrow (AOA) network.

Forward path calculations

- ❑ This calculation indicates that the earliest we can claim completion of these project activities at this point is 12.
- ❑ The same type logic applies to node 7 where a value for T(E) of 16 is calculated.
- ❑ The calculation rule for the forward pass is to remember to take the highest value for multiple input paths at the node.
- ❑ T(E) values at each node represent the earliest time that these points in the project can be reached from a schedule viewpoint.
- ❑ This calculation process also shows that the final node (8) can be reached at time period 21.

Table 9.3 Sample Project Definition

Activity	Duration	Predecessor
A	3	—
B	4	—
C	4	A
D	6	B
E	5	C
F	2	C
G	4	D,E
H	3	F
I	5	G,H

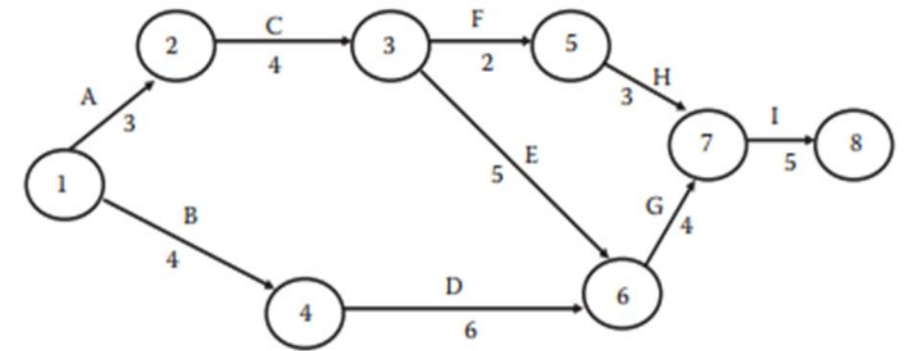


Figure 9.6 Sample activity-on-arrow (AOA) network.

Forward path calculations

- ❑ The final node value defines that the project duration is 21 given this set of parameters.
- ❑ The forward pass then defines a crude project time schedule (Figure 9.7).
- ❑ However, even though this is an interesting value, there are other more worthwhile parameters that need to be defined.
- ❑ We will see these in the next step.

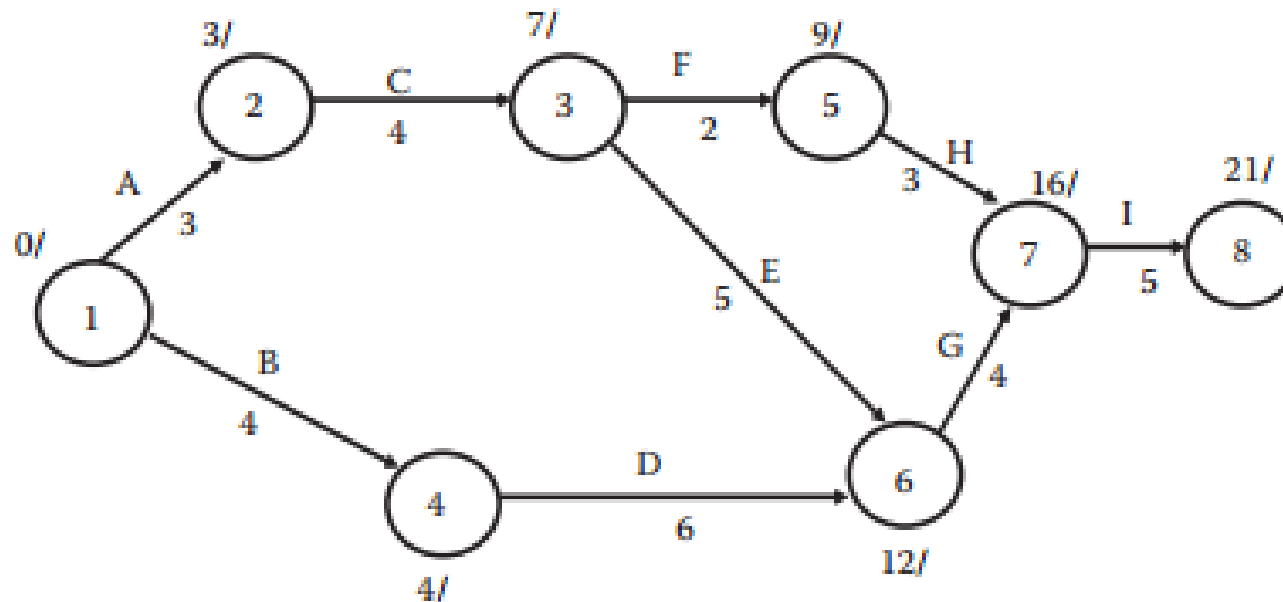


Figure 9.7 Forward pass calculations.

Backward pass calculation

- ❑ Step 2 of the calculation process involves performing a backward pass on the network model.
- ❑ The rationale for needing this is not clear as yet, but let us first describe the mechanics of the calculations before attempting to describe the function of these values.
- ❑ The backward pass generates a variable defined as $T(L)$ and it is formatted on the network node as " $T(E)/T(L)$."
- ❑ Notice that the nodes in Figure 9.8 now show two such numeric values above each node.
- ❑ For this basic network, the assumption is made that the planned time at termination is equal to the forward pass calculated time shown (i.e., 21 time units for the example).
- ❑ Stated another way, we are saying that the forward plan specified that the project would take 21 time units and we are accepting that value for the backward pass.

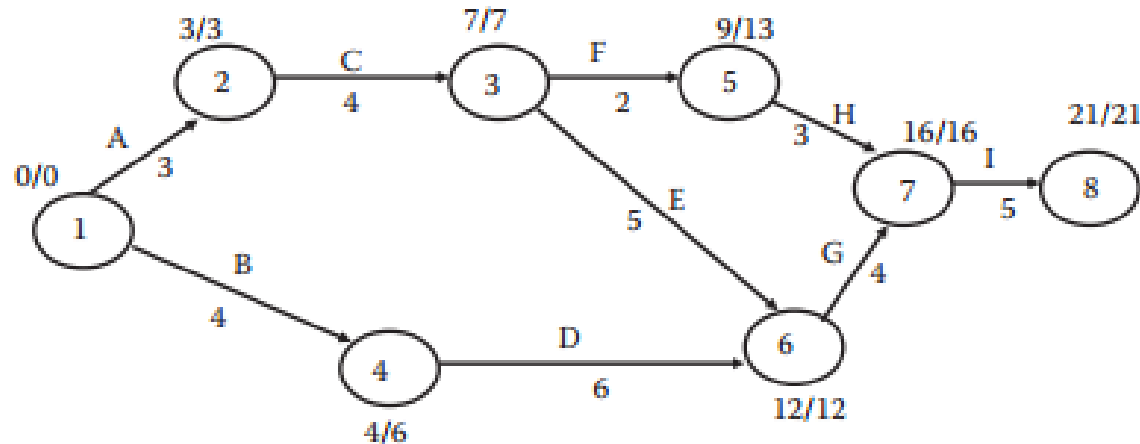


Figure 9.8 Backward pass calculations.

Backward pass calculation

- ❑ To calculate T(L) values, we start at the completion node (8) and insert the same time as calculated for the forward pass—which is 21 in the sample case.
- ❑ From this starting value, work backwards to the front of the network one node at a time. For node 7, the T(L) value would be 21 minus 5 (for activity I), or 16.
- ❑ Node 6 would be calculated as 16 minus 4 (for activity G), or 12. As described in the forward pass mechanics, the T(L) calculations are straightforward until a node that has multiple paths coming back into it such as node 3 is encountered.
- ❑ In this case, the T(L) calculations would compare the two paths for activities F and E yielding a 13 minus 2, or 11 for activity F, versus 12 minus 5, or 7 for activity E.
- ❑ The calculation rule for the backward pass is to take the lower value of the two reverse paths and record that as the nodal T(L) time.

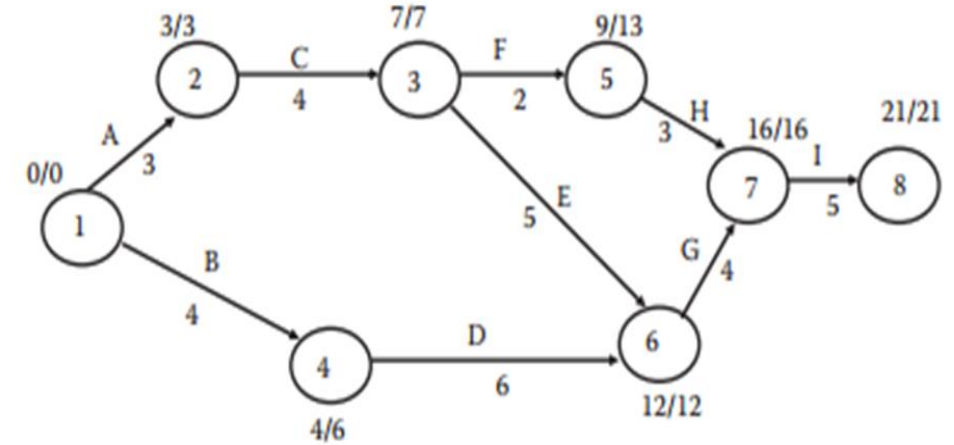


Figure 9.8 Backward pass calculations.

Backward pass calculation

- ❑ This same process is required for node 1 and at that final node the value should be zero since we started with 21/21 at node 8.
- ❑ Failure to return with a zero value indicates that an error was made in the return path.
- ❑ Review the values shown in Figure 9.8 to be sure that you understand the basic idea of T(E) and T(L) mechanical values.
- ❑ Once the values for T(E) and T(L) are calculated, it is possible to analyze two additional important time management factors.
- ❑ These are the longest path through the network (critical path) and slack time details.
- ❑ The simplest item to calculate of these two is nodal slack time.
- ❑ This is simply the difference in T(L) and T(E).

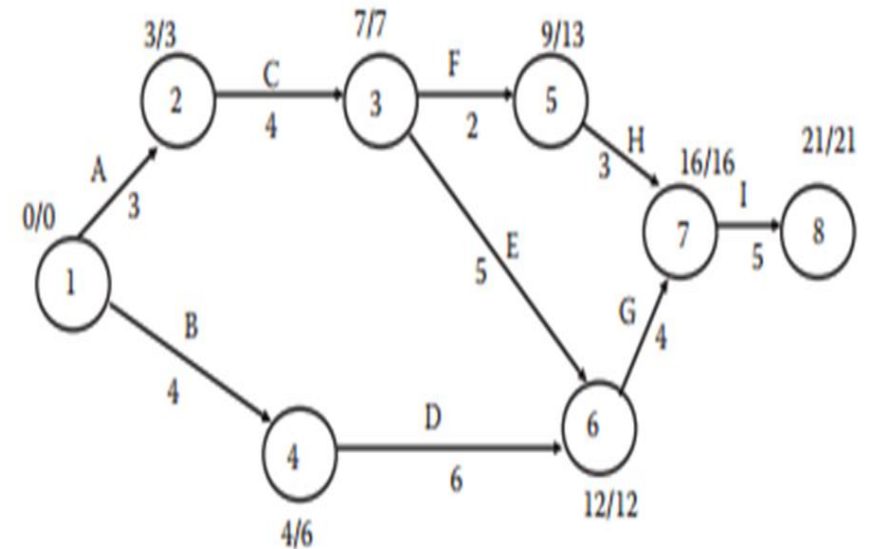


Figure 9.8 Backward pass calculations.

Backward pass calculation

- ❑ So, node 3 would have a slack value of zero (7 minus 7), while node 5 has a slack value of 4 time units (13 minus 9).
- ❑ Each of these values defines the amount of time that this node can set idle without affecting the planned project completion date.
- ❑ Also, note that the computed slack for the start and finish of the network will be zero from using the assumption of 21/21, that is, this says that we are happy with the original computed project duration.
- ❑ You might be thinking about what you would do if some other value for $T(L)$ was desired.

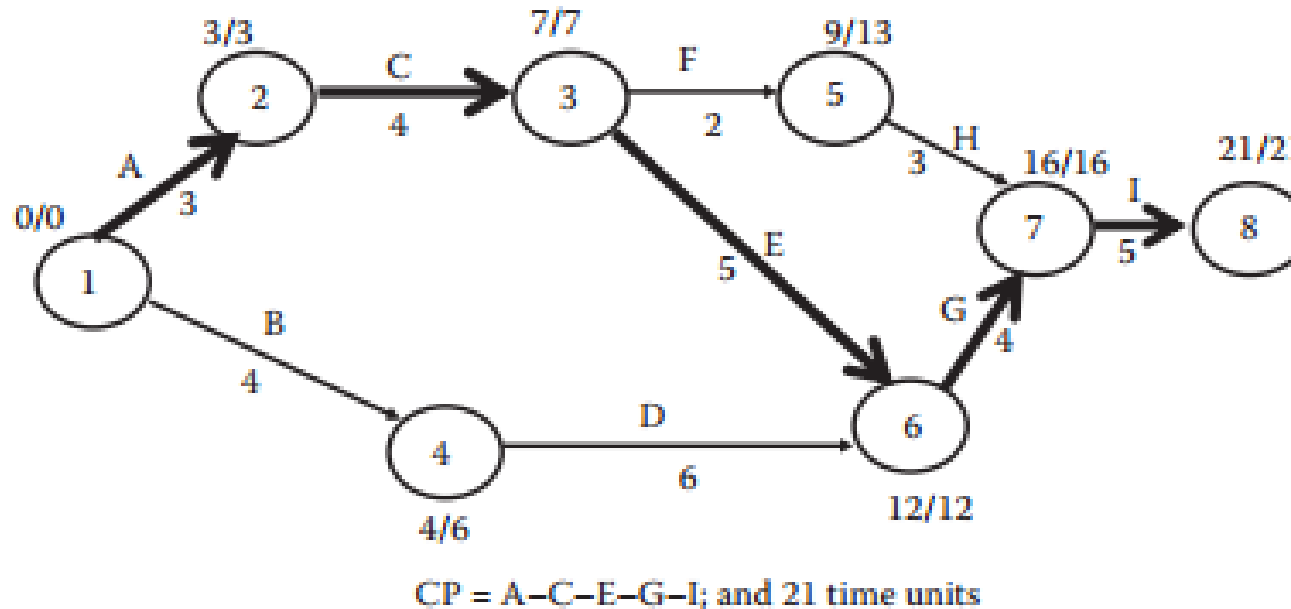


Figure 9.9 Critical path.

Backward pass calculation

- ❑ MSP can also generate this type of information with its internal calculations.
- ❑ To find the network critical path activities, the key mechanic involves checking any activity bounded between zero slack nodes to see if in fact that path is a critical activity.
- ❑ In some more complex networks, parallel paths can both appear to be on the critical path, whereas only one actually is.
- ❑ In this sample case, the critical path definition is straightforward and is outlined as bold arrows in Figure 9.9.
- ❑ Note that the critical path passes through nodes 1-2-3-6-7-8 and the critical path activities are A-C-E-G-I.
- ❑ Specifically the critical path should be thought of as a vector containing both the activity list and the total duration value, so it would be more proper to state the critical path as A-C-E-G-I, and 21 time units.
- ❑ To reiterate, this is the longest path through the network and represents a critical management issue for the Project Manager (Brown 2002).

Critical path- Example

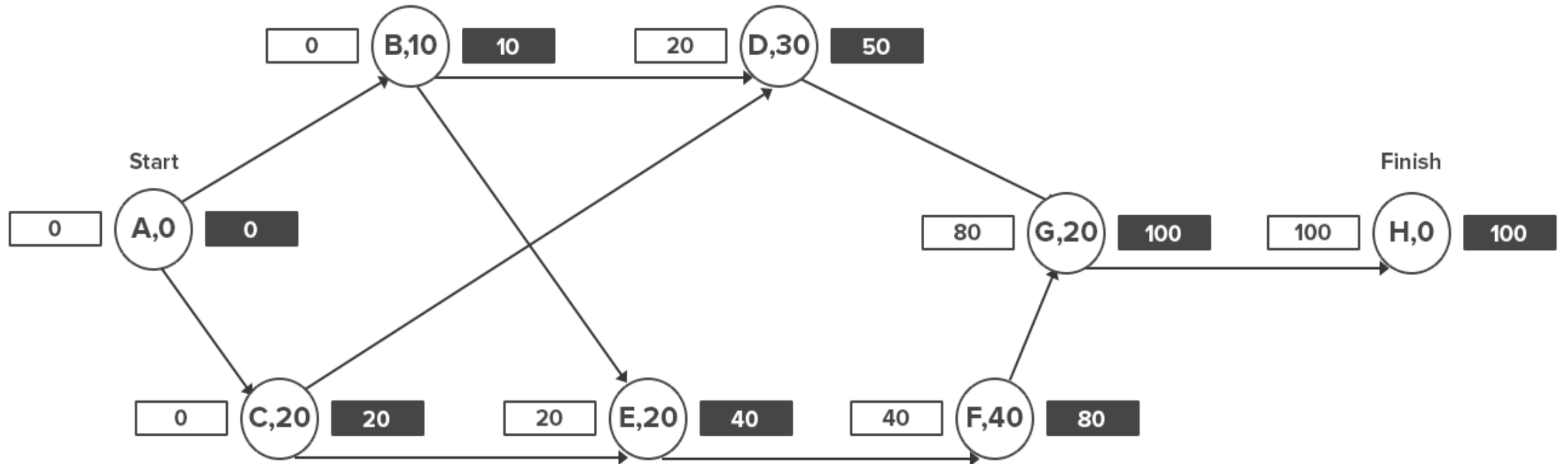
- Suppose you have a list of tasks as shown below. Columns C and D list the tasks that must be accomplished for the activity to begin, and the duration of the activity, respectively.

Task ID	Task Description	Task Predecessors	Task Duration (hours)
A	Project start		0
B	Buy materials for A	A	10
C	Buy materials for B	A	20
D	Build A	B, C	30
E	Build B	B, C	20
F	Polish and finish B	E	40
G	Join A and B	D, F	20
H	Project finish	G	0

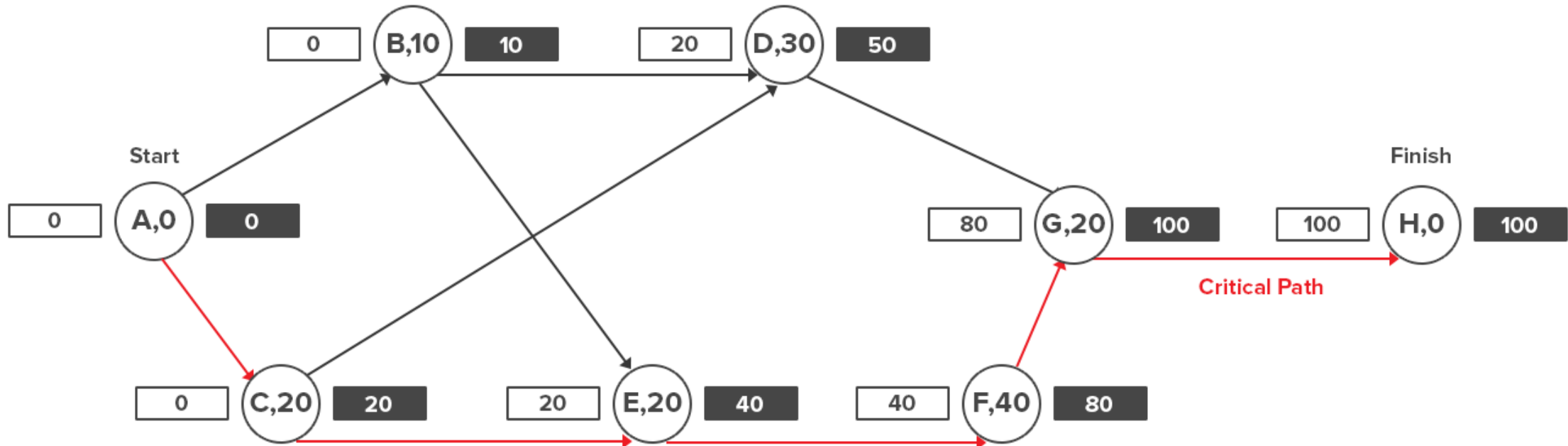
Critical path- Example

- Since the project manager's goal is to complete the project as quickly as possible (without compromising on quality, of course), we'll try to find the earliest finish time for each activity.
- To do this, organize all tasks into a flowchart and note their durations next to the task ID. The arrows indicate the sequence of activities. We'll mark the Earliest Start (ES) time to the left of the activity, and the Earliest Finish (EF) time to the right.:

Critical path- Example



Critical path- Example



Summary

- ❑ This chapter has explored the basic mechanics required to create a first cut project schedule.
- ❑ Also, important core vocabulary has been introduced.
- ❑ Both of these items represent basic skills needed in various other project scenarios.
- ❑ Remember, as good as the first cut plan looks it is important to recognize that there are other steps required before it can be considered a viable plan ready to receive management approval and move into execution.
- ❑ Our final goal is to produce a plan with this basic structure that is agreed upon by key stakeholders and can then be used as a road map through its lifecycle and serve as a baseline for status analysis.
- ❑ The mechanics are now in place, but now the approval remains a forward goal.



Thank You