Critical Path Method

Contents

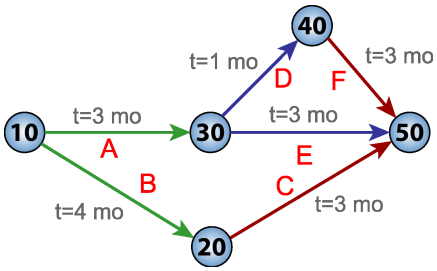
Articles

References

Article Licenses

Critical path method

The **critical path method (CPM)** is an algorithm for scheduling a set of project activities.



PERT chart for a project with five milestones (10 through 50) and six activities (A through F). The project has two critical paths: activities B and C, or A, D, and F – giving a minimum project time of 7 months with fast tracking. Activity E is sub-critical, and has a float of 1 month.

History

The critical path method (CPM) is a project modeling technique developed in the late 1950s by Morgan R. Walker of DuPont and James E. Kelley, Jr. of Remington Rand. Kelley and Walker related their memories of the development of CPM in 1989. Kelley attributed the term "critical path" to the developers of the Program Evaluation and Review Technique which was developed at about the same time by Booz Allen Hamilton and the U.S. Navy. The precursors of what came to be known as Critical Path were developed and put into practice by DuPont between 1940 and 1943 and contributed to the success of the Manhattan Project.

CPM is commonly used with all forms of projects, including construction, aerospace and defense, software development, research projects, product development, engineering, and plant maintenance, among others. Any project with interdependent activities can apply this method of mathematical analysis. Although the original CPM program and approach is no longer used, the term is generally applied to any approach used to analyze a project network logic diagram.

Basic technique

The essential technique for using CPM [1] is to construct a model of the project that includes the following:

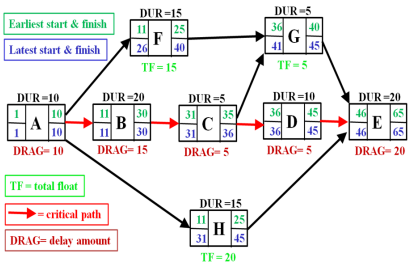
1. A list of all activities required to complete the project (typically categorized within a work breakdown structure),
2. The time (duration) that each activity will take to complete,
3. The dependencies between the activities and,
4. Logical end points such as milestones or deliverable items.

Using these values, CPM calculates the longest path of planned activities to logical end points or to the end of the project, and the earliest and latest that each activity can start and finish without making the project longer. This process determines which activities are "critical" (i.e., on the longest path) and which have "total float" (i.e., can be delayed without making the project longer). In project management, a critical path is the sequence of project network activities which add up to the longest overall duration. This determines the shortest time possible to complete the project. Any delay of an activity on the critical path directly impacts the planned project completion date (i.e. there is no float on the critical path). A project can have several, parallel, near critical paths. An additional parallel path through the network with the total durations shorter than the critical path is called a sub-critical or non-critical path.

CPM analysis tools allow a user to select a logical end point in a project and quickly identify its longest series of dependent activities (its longest path). These tools can display the critical path (and near critical path activities if desired) as a cascading waterfall that flows from the project's start (or current status date) to the selected logical end point.

Although the activity-on-arrow diagram ("PERT Chart") is still used in a few places, it has generally been superseded by the activity-on-node diagram, where each activity is shown as a box or node and the arrows represent the logical relationships going from predecessor to successor as shown here in the "Activity-on-node diagram".

In this diagram, Activities A, B, C, D, and E comprise the critical or longest path, while Activities F, G, and H are off the critical path with floats of 15 days, 5 days, and 20 days respectively. Whereas activities that are off the critical path have float and are therefore not delaying completion of the project, those on the critical path will usually have critical path drag, i.e., they delay project completion. The drag of a critical path activity can be computed using the following formula:



Activity-on-node diagram showing critical path schedule, along with total float and critical path drag computations

1. If a critical path activity has nothing in parallel, its drag is equal to its duration. Thus A and E have drags of 10 days and 20 days respectively.
2. If a critical path activity has another activity in parallel, its drag is equal to whichever is less: its duration or the total float of the parallel activity with the least total float. Thus since B and C are both parallel to F (float of 15) and H (float of 20), B has a duration of 20 and drag of 15 (equal to F's float), while C has a duration of only 5 days and thus drag of only 5. Activity D, with a duration of 10 days, is parallel to G (float of 5) and H (float of 20) and therefore its drag is equal to 5, the float of G.

These results, including the drag computations, allow managers to prioritize activities for the effective management of project completion, and to shorten the planned critical path of a project by pruning critical path activities, by "**fast tracking**" (i.e., performing more activities in parallel), and/or by "**crashing the critical path**" (i.e., shortening the durations of critical path activities by adding resources).

Crash duration

"Crash duration" is a term referring to the shortest possible time for which an activity can be scheduled. It is achieved by shifting more resources towards the completion of that activity, resulting in decreased time spent and often a reduced quality of work, as the premium is set on speed. Crash duration is typically modeled as a linear relationship between cost and activity duration, however in many cases a convex function or a step function is more applicable.

Expansion

Originally, the critical path method considered only logical dependencies between terminal elements. Since then, it has been expanded to allow for the inclusion of resources related to each activity, through processes called activity-based resource assignments and resource leveling. A resource-leveled schedule may include delays due to resource bottlenecks (i.e., unavailability of a resource at the required time), and may cause a previously shorter path to become the longest or most "resource critical" path. A related concept is called the critical chain, which attempts to protect activity and project durations from unforeseen delays due to resource constraints.

Since project schedules change on a regular basis, CPM allows continuous monitoring of the schedule, which allows the project manager to track the critical activities, and alerts the project manager to the possibility that non-critical activities may be delayed beyond their total float, thus creating a new critical path and delaying project completion. In addition, the method can easily incorporate the concepts of stochastic predictions, using the program evaluation and review technique (PERT) and event chain methodology.

Currently, there are several software solutions available in industry that use the CPM method of scheduling, see list of project management software. The method currently used by most project management software is based on a manual calculation approach developed by Fondahl of Stanford University.

Flexibility

A schedule generated using critical path techniques often is not realised precisely, as estimations are used to calculate times: if one mistake is made, the results of the analysis may change. This could cause an upset in the implementation of a project if the estimates are blindly believed, and if changes are not addressed promptly. However, the structure of critical path analysis is such that the variance from the original schedule caused by any change can be measured, and its impact either ameliorated or adjusted for. Indeed, an important element of project postmortem analysis is the As Built Critical Path (ABCP), which analyzes the specific causes and impacts of changes between the planned schedule and eventual schedule as actually implemented.

References

[1] Samuel L. Baker, Ph.D. "Critical Path Method (CPM)" (http://hspm.sph.sc.edu/COURSES/J716/CPM/CPM.html) *University of South Carolina*, Health Services Policy and Management Courses

Further reading

- Project Management Institute (2013). *A Guide To The Project Management Body Of Knowledge* (5th ed.). Project Management Institute. ISBN 978-1-935589-67-9.
- Klastorin, Ted (2003). *Project Management: Tools and Trade-offs* (3rd ed.). Wiley. ISBN 978-0-471-41384-4.
- Heerkens, Gary (2001). *Project Management (The Briefcase Book Series)*. McGraw–Hill. ISBN 0-07-137952-5.
- Harold Kerzner (2003). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling* (8th ed.). ISBN 0-471-22577-0.
- Lewis, James (2002). *Fundamentals of Project Management* (2nd ed.). American Management Association. ISBN 0-8144-7132-3.
- Milosevic, Dragan Z. (2003). *Project Management ToolBox: Tools and Techniques for the Practicing Project Manager*. Wiley. ISBN 978-0-471-20822-8.
- O'Brien, James J.; Plotnick, Fredric L. (2010). *CPM in Construction Management, Seventh Edition*. McGraw Hill. ISBN 978-0-07-163664-3.
- Woolf, Murray B. (2012). *CPM Mechanics: The Critical Path Method of Modeling Project Execution Strategy*. ICS-Publications. ISBN 978-0-9854091-0-4.
- Woolf, Murray B. (2007). *Faster Construction Projects with CPM Scheduling*. McGraw Hill. ISBN 978-0-07-148660-6.
- Trauner, Manginelli, Lowe, Nagata, Furniss (2009). *Construction Delays, 2nd Ed.: Understanding Them Clearly, Analyzing Them Correctly* (http://www.amazon.com/Construction-Delays-Second-Edition-Understanding/dp/1856176770). Burlington, MA: Elsevier. p. 266. ISBN 978-1-85617-677-4.
- Malakooti, B (2013). *Operations and Production Systems with Multiple Objectives*. John Wiley & Sons. ISBN 978-1-118-58537-5.

Float (project management)

In project management, **float** or **slack** is the amount of time that a task in a project network can be delayed without causing a delay to:

- subsequent tasks ("*free float*")
- project completion date ("*total float*")

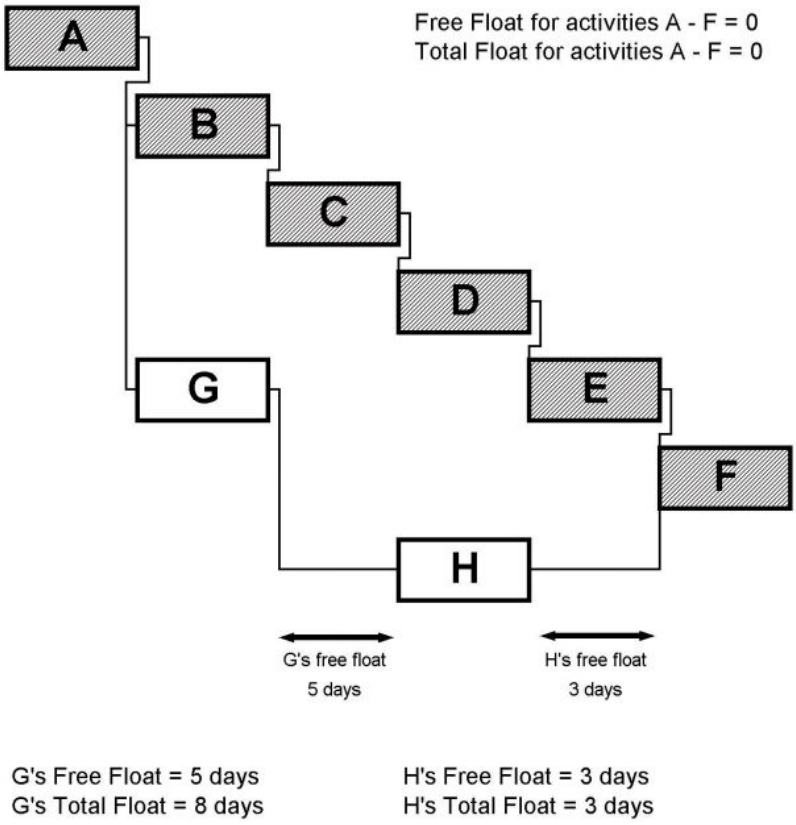
An activity on critical path has "*zero free float*", but activity that has *zero free float* might not be on the critical path. *Total float* is associated with the path. If a project network chart/diagram has 4 non-critical paths then that project would have 4 *total float* values. The *total float* of a path is the combined *free float* values of all activities in a path.

The *total float* represents the schedule flexibility and can also be measured by subtracting early dates from late dates of path completion. Float is core to critical path method, with the total floats of noncritical activities key to computing the critical path drag of an activity, i.e., the amount of time it is adding to the project's duration.

Example

Consider the process of replacing a broken pane of glass in the window of your home. There are various component activities involved in the project as a whole; obtaining the glass and putty, installing the new glass, choosing the paint, obtaining a tin once it has set, wiping the new glass free of finger smears etc.

Some of these activities can run concurrently e.g. obtaining the glass, obtaining the putty, choosing the paint etc., while others are consecutive e.g. the paint cannot be bought until it has been chosen, the new window cannot be painted until the window is installed and the new putty has set. Delaying the acquisition of the glass is likely to delay the entire project - this activity will be on the critical path and have no float, of any sort, attached to it and hence it is a 'critical activity'. A relatively short delay in the purchase of the paint may not automatically hold up the entire project as there is still some waiting time for the new putty to dry before it can be painted anyway - there will be some 'free float' attached to the activity of purchasing the paint and hence it is not a critical activity. However a delay in choosing the paint in turn inevitably delays buying the paint which, although it may not subsequently mean any delay to the entire project, does mean that choosing the paint has no 'free float' attached to it - despite having no free float of its own the choosing of the paint is involved with a path through the network which does have 'total float'.



References

- AACE International (August 2007). "Cost Engineering Terminology, AACE International Recommended Practice No. 10S-90" (http://www.aacei.org/technical/rps/10s-90.pdf) (PDF). www.aacei.org.

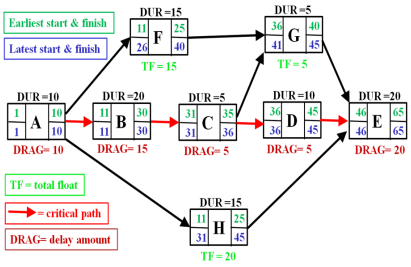
Critical path drag

Critical path drag is a project management metric developed by Stephen Devaux as part of the Total Project Control (TPC) approach to schedule analysis and compression [1] in the critical path method of scheduling. It is the quantified amount of time that an activity or constraint on the critical path is adding to the project duration.

In networks where all dependencies are finish-to-start (FS) relationships (i.e., where a predecessor must finish before a successor starts), the drag of a critical path activity is equal to whichever is less: its remaining duration or (if there is one or more parallel activity) the total float of the parallel activity that has the least total float.[2]

In this diagram, Activities A, B, C, D, and E comprise the critical path, while Activities F, G, and H are off the critical path with floats of 15 days, 5 days, and 20 days respectively. Whereas activities that are off the critical path have float and are therefore not delaying completion of the project, those on the critical path have critical path drag, i.e., they delay project completion.

1. Activities A and E have nothing in parallel and therefore have drags of 10 days and 20 days respectively.
2. Activities B and C are both parallel to F (float of 15) and H (float of 20). B has a duration of 20 and drag of 15 (equal to F's float), while C has a duration of only 5 days and thus drag of only 5.
3. Activity D, with a duration of 10 days, is parallel to G (float of 5) and H (float of 20) and therefore its drag is equal to 5, the float of G.



Activity-on-node diagram showing critical path schedule, along with total float and critical path drag computations

In network schedules that include start-to-start (SS), finish-to-finish (FF) and start-to-finish (SF) relationships and lags, drag computation can be quite complex, often requiring either the decomposition of critical path activities into their components so as to create all relationships as finish-to-start, or the use of scheduling software that computes critical path drag with complex dependencies.

Critical path drag is often combined with an estimate of the increased cost and/or reduced expected value of the project due to each unit of the critical path's duration. This allows such cost to be attributed to individual critical path activities through their respective drag amounts (i.e., the activity's drag cost). If the cost of each unit of time in the diagram above is $10,000, the drag cost of E would be $200,000, B would be $150,000, A would be $100,000, and C and D $50,000 each.

This in turn can allow a project manager to justify those additional resources that will reduce the drag and drag cost of specific critical path activities where the cost of such resources would be less than the value generated by reduction in drag. For example, if the addition of $50,000 worth of resources would reduce the duration of B to ten days, the project would take only 55 days, B's drag would be reduced to five days, and its drag cost would be reduced to $50,000.

Sources

[1] William Duncan and Stephen Devaux "Scheduling Is a Drag" (http://www.projectsatwork.com/content/articles/246653.cfm) Projects@Work on-line magazine

[2] Stephen A. Devaux "The Drag Efficient: The Missing Quantification of Time on the Critical Path" (http://www.dau.mil/pubscats/ATL Docs/Jan_Feb_2012/Devaux.pdf) Defense AT&L magazine of the Defense Acquisition University.

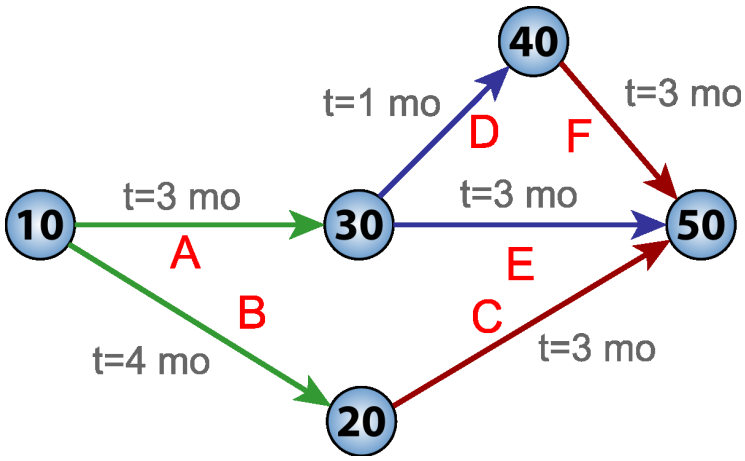
Further reading

- Wideman, R. Max (2004). Total Project Control: A book review (http://www.maxwideman.com/papers/totalcontrol/totalcontrol.pdf)
- Borfitz, Deb (2009). [www.ecliniqua.com/2009/08/07/prochain.html "ProChain Solutions: Diagnosing the Drag in Clinical Development" eCliniqua Magazine]
- Mosaic Projects. Basic CPM Calculations (http://www.mosaicprojects.com.au/PDF/Schedule_Calculations.pdf)

Program evaluation and review technique (PERT)

For other uses, see Pert (disambiguation).

The **program** (or **project**) **evaluation and review technique**, commonly abbreviated **PERT**, is a statistical tool, used in project management, which was designed to analyze and represent the tasks involved in completing a given project. First developed by the United States Navy in the 1950s, it is commonly used in conjunction with the critical path method (**CPM**).



PERT network chart for a seven-month project with five milestones (10 through 50) and six activities (A through F).

History

Program evaluation and review technique

The Navy's Special Projects Office, charged with developing the Polaris-Submarine weapon system and the Fleet Ballistic Missile capability, has developed a statistical technique for measuring and forecasting progress in research and development programs. This program evaluation and review technique (code-named PERT) is applied as a decision-making tool designed to save time in achieving end-objectives, and is of particular interest to those engaged in research and development programs for which time is a critical factor.

The new technique takes recognition of three factors that influence successful achievement of research and development program objectives: time, resources, and technical performance specifications. PERT employs time as the variable that reflects planned resource-applications and performance specifications. With units of time as a common denominator, PERT quantifies knowledge about the uncertainties involved in developmental programs requiring effort at the edge of, or beyond, current knowledge of the subject – effort for which little or no previous experience exists.

Through an electronic computer, the PERT technique processes data representing the major, finite

> accomplishments (events) essential to achieve end-objectives; the inter-dependence of those events; and estimates of time and range of time necessary to complete each activity between two successive events. Such time expectations include estimates of "most likely time", "optimistic time", and "pessimistic time" for each activity. The technique is a management control tool that sizes up the outlook for meeting objectives on time; highlights danger signals requiring management decisions; reveals and defines both methodicalness and slack in the flow plan or the network of sequential activities that must be performed to meet objectives; compares current expectations with scheduled completion dates and computes the probability for meeting scheduled dates; and simulates the effects of options for decision – before decision.
>
> The concept of PERT was developed by an operations research team staffed with representatives from the Operations Research Department of Booz, Allen and Hamilton; the Evaluation Office of the Lockheed Missile Systems Division; and the Program Evaluation Branch, Special Projects Office, of the Department of the Navy.
>
> — Willard Fazar (Head, Program Evaluation Branch, Special Projects Office, U. S. Navy), The American Statistician, April 1959.[1]

Overview

PERT is a method to analyze the involved tasks in completing a given project, especially the time needed to complete each task, and to identify the minimum time needed to complete the total project.

PERT was developed primarily to simplify the planning and scheduling of large and complex projects. It was developed for the U.S. Navy Special Projects Office in 1957 to support the U.S. Navy's Polaris nuclear submarine project.[2] It was able to incorporate uncertainty by making it possible to schedule a project while not knowing precisely the details and durations of all the activities. It is more of an event-oriented technique rather than start- and completion-oriented, and is used more in projects where time is the major factor rather than cost. It is applied to very large-scale, one-time, complex, non-routine infrastructure and Research and Development projects. An example of this was for the 1968 Winter Olympics in Grenoble which applied PERT from 1965 until the opening of the 1968 Games.[3]

This project model was the first of its kind, a revival for scientific management, founded by Frederick Taylor (Taylorism) and later refined by Henry Ford (Fordism). DuPont's critical path method was invented at roughly the same time as PERT.

Terminology

- *PERT event*: a point that marks the start or completion of one or more activities. It consumes no time and uses no resources. When it marks the completion of one or more activities, it is not "reached" (does not occur) until *all* of the activities leading to that event have been completed.
- *predecessor event*: an event that immediately precedes some other event without any other events intervening. An event can have multiple predecessor events and can be the predecessor of multiple events.
- *successor event*: an event that immediately follows some other event without any other intervening events. An event can have multiple successor events and can be the successor of multiple events.
- *PERT activity*: the actual performance of a task which consumes time and requires resources (such as labor, materials, space, machinery). It can be understood as representing the time, effort, and resources required to move from one event to another. A PERT activity cannot be performed until the predecessor event has occurred.
- *PERT sub-activity*: a PERT activity can be further decomposed into a set of sub-activities. For example, activity A1 can be decomposed into A1.1, A1.2 and A1.3 for example. Sub-activities have all the properties of activities, in particular a sub-activity has predecessor or successor events just like an activity. A sub-activity can be decomposed again into finer-grained sub-activities.

- *optimistic time* (O): the minimum possible time required to accomplish a task, assuming everything proceeds better than is normally expected
- *pessimistic time* (P): the maximum possible time required to accomplish a task, assuming everything goes wrong (but excluding major catastrophes).
- *most likely time* (M): the best estimate of the time required to accomplish a task, assuming everything proceeds as normal.
- *expected time* (T_E): the best estimate of the time required to accomplish a task, accounting for the fact that things don't always proceed as normal (the implication being that the expected time is the average time the task would require if the task were repeated on a number of occasions over an extended period of time).

$$\mathbf{T_E = (O + 4M + P) \div 6}$$

- *float or slack* is a measure of the excess time and resources available to complete a task. It is the amount of time that a project task can be delayed without causing a delay in any subsequent tasks (*free float*) or the whole project (*total float*). Positive slack would indicate *ahead of schedule*; negative slack would indicate *behind schedule*; and zero slack would indicate *on schedule*.
- *critical path*: the longest possible continuous pathway taken from the initial event to the terminal event. It determines the total calendar time required for the project; and, therefore, any time delays along the critical path will delay the reaching of the terminal event by at least the same amount.
- *critical activity*: An activity that has total float equal to zero. An activity with zero float is not necessarily on the critical path since its path may not be the longest.
- *Lead* [4] *time*: the time by which a *predecessor event* must be completed in order to allow sufficient time for the activities that must elapse before a specific PERT event reaches completion.
- *lag time*: the earliest time by which a *successor event* can follow a specific PERT event.
- *fast tracking*: performing more critical activities in parallel
- *crashing critical path*: Shortening duration of critical activities

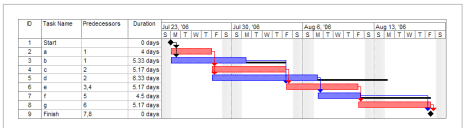
Implementation

The first step to scheduling the project is to determine the tasks that the project requires and the order in which they must be completed. The order may be easy to record for some tasks (*e.g.* When building a house, the land must be graded before the foundation can be laid) while difficult for others (There are two areas that need to be graded, but there are only enough bulldozers to do one). Additionally, the time estimates usually reflect the normal, non-rushed time. Many times, the time required to execute the task can be reduced for an additional cost or a reduction in the quality.

In the following example there are seven tasks, labeled *A* through *G*. Some tasks can be done concurrently (*A* and *B*) while others cannot be done until their predecessor task is complete (*C* cannot begin until *A* is complete). Additionally, each task has three time estimates: the optimistic time estimate (*O*), the most likely or normal time estimate (*M*), and the pessimistic time estimate (*P*). The expected time (T_E) is computed using the formula ($O + 4M + P$) ÷ 6.

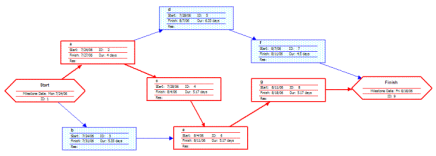
Activity	Predecessor	Time estimates			Expected time
		Opt. (*O*)	Normal (*M*)	Pess. (*P*)	
A	—	2	4	6	4.00
B	—	3	5	9	5.33
C	*A*	4	5	7	5.17
D	*A*	4	6	10	6.33
E	*B*, *C*	4	5	7	5.17
F	*D*	3	4	8	4.50
G	*E*	3	5	8	5.17

Once this step is complete, one can draw a Gantt chart or a network diagram.



A Gantt chart created using Microsoft Project (MSP). Note (1) the critical path is in red, (2) the slack is the black lines connected to non-critical activities, (3) since Saturday and Sunday are not work days and are thus excluded from the schedule, some bars on the Gantt chart are longer if they cut through a weekend.

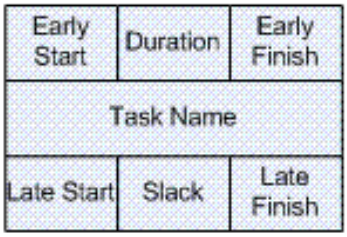
A network diagram can be created by hand or by using diagram software. There are two types of network diagrams, activity on arrow (AOA) and activity on node (AON). Activity on node diagrams are generally easier to create and interpret. To create an AON diagram, it is recommended (but not required) to start with a node named *start*. This "activity" has a duration of zero (0). Then you draw each activity that does not have a predecessor activity (*a* and *b* in this example) and connect them with an arrow from start to each node. Next, since both *c* and *d* list *a* as a predecessor activity, their nodes are drawn with arrows coming from *a*. Activity *e* is listed with *b* and *c* as predecessor activities, so node *e* is drawn with arrows coming from both *b* and *c*, signifying that *e* cannot begin until both *b* and *c* have been completed. Activity *f* has *d* as a predecessor activity, so an arrow is drawn connecting the activities. Likewise, an arrow is drawn from *e* to *g*. Since there are no activities that come after *f* or *g*, it is recommended (but again not required) to connect them to a node labeled *finish*.



A network diagram created using Microsoft Project (MSP). Note the critical path is in red.

By itself, the network diagram pictured above does not give much more information than a Gantt chart; however, it can be expanded to display more information. The most common information shown is:

1. The activity name
2. The normal duration time
3. The early start time (ES)
4. The early finish time (EF)
5. The late start time (LS)
6. The late finish time (LF)
7. The slack



A node like this one (from Microsoft Visio) can be used to display the activity name, duration, ES, EF, LS, LF, and slack.

In order to determine this information it is assumed that the activities and normal duration times are given. The first step is to determine the ES and EF. The ES is defined as the maximum EF of all predecessor activities, unless the activity in question is the first activity, for which the ES is zero (0). The EF is the ES plus the task duration (EF = ES + duration).

- The ES for *start* is zero since it is the first activity. Since the duration is zero, the EF is also zero. This EF is used as the ES for *a* and *b*.
- The ES for *a* is zero. The duration (4 work days) is added to the ES to get an EF of four. This EF is used as the ES for *c* and *d*.
- The ES for *b* is zero. The duration (5.33 work days) is added to the ES to get an EF of 5.33.
- The ES for *c* is four. The duration (5.17 work days) is added to the ES to get an EF of 9.17.
- The ES for *d* is four. The duration (6.33 work days) is added to the ES to get an EF of 10.33. This EF is used as the ES for *f*.
- The ES for *e* is the greatest EF of its predecessor activities (*b* and *c*). Since *b* has an EF of 5.33 and *c* has an EF of 9.17, the ES of *e* is 9.17. The duration (5.17 work days) is added to the ES to get an EF of 14.34. This EF is used as the ES for *g*.
- The ES for *f* is 10.33. The duration (4.5 work days) is added to the ES to get an EF of 14.83.
- The ES for *g* is 14.34. The duration (5.17 work days) is added to the ES to get an EF of 19.51.
- The ES for *finish* is the greatest EF of its predecessor activities (*f* and *g*). Since *f* has an EF of 14.83 and *g* has an EF of 19.51, the ES of *finish* is 19.51. *Finish* is a milestone (and therefore has a duration of zero), so the EF is also 19.51.

Barring any unforeseen events, the project should take 19.51 work days to complete. The next step is to determine the late start (LS) and late finish (LF) of each activity. This will eventually show if there are activities that have slack. The LF is defined as the minimum LS of all successor activities, unless the activity is the last activity, for which the LF equals the EF. The LS is the LF minus the task duration (LS = LF − duration).

- The LF for *finish* is equal to the EF (19.51 work days) since it is the last activity in the project. Since the duration is zero, the LS is also 19.51 work days. This will be used as the LF for *f* and *g*.
- The LF for *g* is 19.51 work days. The duration (5.17 work days) is subtracted from the LF to get an LS of 14.34 work days. This will be used as the LF for *e*.
- The LF for *f* is 19.51 work days. The duration (4.5 work days) is subtracted from the LF to get an LS of 15.01 work days. This will be used as the LF for *d*.
- The LF for *e* is 14.34 work days. The duration (5.17 work days) is subtracted from the LF to get an LS of 9.17 work days. This will be used as the LF for *b* and *c*.
- The LF for *d* is 15.01 work days. The duration (6.33 work days) is subtracted from the LF to get an LS of 8.68 work days.
- The LF for *c* is 9.17 work days. The duration (5.17 work days) is subtracted from the LF to get an LS of 4 work days.

- The LF for *b* is 9.17 work days. The duration (5.33 work days) is subtracted from the LF to get an LS of 3.84 work days.
- The LF for *a* is the minimum LS of its successor activities. Since *c* has an LS of 4 work days and *d* has an LS of 8.68 work days, the LF for *a* is 4 work days. The duration (4 work days) is subtracted from the LF to get an LS of 0 work days.
- The LF for *start* is the minimum LS of its successor activities. Since *a* has an LS of 0 work days and *b* has an LS of 3.84 work days, the LS is 0 work days.

The next step is to determine the critical path and if any activities have slack. The critical path is the path that takes the **longest** to complete. To determine the path times, add the task durations for all available paths. Activities that have slack can be delayed without changing the overall time of the project. Slack is computed in one of two ways, slack = LF − EF *or* slack = LS − ES. Activities that are on the critical path have a slack of zero (0).

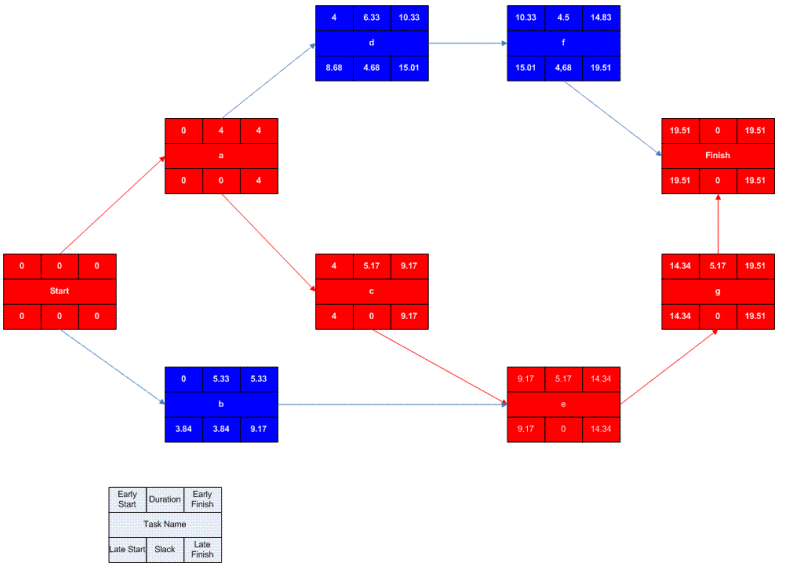
- The duration of path *adf* is 14.83 work days.
- The duration of path *aceg* is 19.51 work days.
- The duration of path *beg* is 15.67 work days.

The critical path is *aceg* and the critical time is 19.51 work days. It is important to note that there can be more than one critical path (in a project more complex than this example) or that the critical path can change. For example, let's say that activities *d* and *f* take their pessimistic (b) times to complete instead of their expected (T_E) times. The critical path is now *adf* and the critical time is 22 work days. On the other hand, if activity *c* can be reduced to one work day, the path time for *aceg* is reduced to 15.34 work days, which is slightly less than the time of the new critical path, *beg* (15.67 work days).

Assuming these scenarios do not happen, the slack for each activity can now be determined.

- *Start* and *finish* are milestones and by definition have no duration, therefore they can have no slack (0 work days).
- The activities on the critical path by definition have a slack of zero; however, it is always a good idea to check the math anyway when drawing by hand.
  - $LF_a - EF_a = 4 - 4 = 0$
  - $LF_c - EF_c = 9.17 - 9.17 = 0$
  - $LF_e - EF_e = 14.34 - 14.34 = 0$
  - $LF_g - EF_g = 19.51 - 19.51 = 0$
- Activity *b* has an LF of 9.17 and an EF of 5.33, so the slack is 3.84 work days.
- Activity *d* has an LF of 15.01 and an EF of 10.33, so the slack is 4.68 work days.
- Activity *f* has an LF of 19.51 and an EF of 14.83, so the slack is 4.68 work days.

Therefore, activity *b* can be delayed almost 4 work days without delaying the project. Likewise, activity *d* **or** activity *f* can be delayed 4.68 work days without delaying the project (alternatively, *d* and *f* can be delayed 2.34 work days each).



A completed network diagram created using Microsoft Visio. Note the critical path is in red.

Advantages

- PERT chart explicitly defines and makes visible dependencies (precedence relationships) between the work breakdown structure (commonly WBS) elements.
- PERT facilitates identification of the critical path and makes this visible.
- PERT facilitates identification of early start, late start, and slack for each activity.
- PERT provides for potentially reduced project duration due to better understanding of dependencies leading to improved overlapping of activities and tasks where feasible.
- The large amount of project data can be organized & presented in diagram for use in decision making.

Disadvantages

- There can be potentially hundreds or thousands of activities and individual dependency relationships.
- PERT is not easily scalable for smaller projects.
- The network charts tend to be large and unwieldy requiring several pages to print and requiring specially sized paper.
- The lack of a timeframe on most PERT/CPM charts makes it harder to show status although colours can help (e.g., specific colour for completed nodes).

Uncertainty in project scheduling

During project execution, however, a real-life project will never execute exactly as it was planned due to uncertainty. This can be due to ambiguity resulting from subjective estimates that are prone to human errors or can be the result of variability arising from unexpected events or risks. The main reason that PERT may provide inaccurate information about the project completion time is due to this schedule uncertainty. This inaccuracy may be large enough to render such estimates as not helpful.

One possible method to maximize solution robustness is to include safety in the baseline schedule in order to absorb the anticipated disruptions. This is called *proactive scheduling*. A pure proactive scheduling is a utopia; incorporating safety in a baseline schedule which allows for every possible disruption would lead to a baseline schedule with a very large make-span. A second approach, termed *reactive scheduling*, consists of defining a procedure to react to disruptions that cannot be absorbed by the baseline schedule.

References

[1] Fazar, W., "Program Evaluation and Review Technique", *The American Statistician*, Vol. 13, No. 2, (April 1959), p.10.

[2] Malcolm, D. G., J. H. Roseboom, C. E. Clark, W. Fazar Application of a Technique for Research and Development Program Evaluation OPERATIONS RESEARCH Vol. 7, No. 5, September–October 1959, pp. 646–669

[3] 1968 Winter Olympics official report. (http://www.la84foundation.org/6oic/OfficialReports/1968/or1968.pdf) p. 49. Accessed 1 November 2010. &

[4] http://en.wiktionary.org/wiki/lead#Verb_2

Further reading

- Project Management Institute (2013). *A Guide to the Project Management Body of Knowledge* (5th ed.). Project Management Institute. ISBN 978-1-935589-67-9.
- Klastorin, Ted (2003). *Project Management: Tools and Trade-offs* (3rd ed. ed.). Wiley. ISBN 978-0-471-41384-4.
- Harold Kerzner (2003). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling* (8th Ed. ed.). Wiley. ISBN 0-471-22577-0.
- Milosevic, Dragan Z. (2003). *Project Management ToolBox: Tools and Techniques for the Practicing Project Manager*. Wiley. ISBN 978-0-471-20822-8.

Wikimedia Commons has media related to ***PERT charts***.

Article Sources and Contributors

Critical path method *Source*: http://en.wikipedia.org/w/index.php?oldid=611713858 *Contributors*: Achalmeena, Alanl, Aleksd, Amarvk, Amgunite, Amientan, Andreas Kaufmann, BBar, Beldridge, Bgwhite, BlinderBomber, Bob1960evens, Breawycker, Bunnyhop11, CPMTutor, CTZMSC3, CallmeDrNo, Carolfrog, Cdh1001, Cgtdk, Chachrist, Cheese Sandwich, Cnbrb, Corpx, Darguz Parsilvan, David Eppstein, Delaszk, Donreed, Dragmas, Dwolbach, Elkinsra, Eubulides, Ferengi, Firien, Fraggle81, G2opdtsl, Garyarnold1970, George100, Graibeard, HangingCurve, Harda, Hazmat2, Hekerui, Hobbes Goodyear, Honacan, Hu12, Inwind, IrishInNY, Ivolution, Jamelan, Jbarmash, Jeltz, Jojalozzo, Julesd, Kaihsu, Kenmckinley, Khoshino, Kuru, Kwhitten, Leszek Jańczuk, Lindsay658, Lordkyl, MER-C, Mausy5043, Mdd, Michael Hardy, Mod.torrentrealm, Moorshed k, Mr. Shoeless, Msea85, NCurse, Nascar fan mx, NcSchu, Newman9997, Niceguyedc, Nixdorf, Nuggetkiwi, Oxymoron83, PatrickWeaver, PigFlu Oink, Pingveno, Pm master, Ppntori, ProjMngt, Quuxplusone, RHaworth, Rich Farmbrough, SNIyer12, Shlomke, Sin-man, SkipHuffman, Sleigh, Sridarshan, Stevenwmccrary58, Thesydneyknowitall, Theuniversalcynic, Tijuana Brass, Timdaman42, Tnanoc, Tony1, Tosek, Vincehk, Vincnet, Vvkvaranasi, Werttyguby, Widr, Winterstein, Woood, Wrduncan3, Wsmith202, 152 anonymous edits

Float (project management) *Source*: http://en.wikipedia.org/w/index.php?oldid=612155566 *Contributors*: Busy Stubber, CPMTutor, Carabinieri, CertSchool PMP, ChrisGualtieri, Dia^, Fti74, Jojalozzo, Krellis, LoganLopez, McGeddon, Mkoval, Mplemmons, Nuggetkiwi, Rjwilmsi, Smjg, SueHay, Tijuana Brass, Timkeys, Topbanana, 24 anonymous edits

Critical path drag *Source*: http://en.wikipedia.org/w/index.php?oldid=604881337 *Contributors*: FreeRangeFrog, Nuggetkiwi

Program evaluation and review technique (PERT) *Source*: http://en.wikipedia.org/w/index.php?oldid=616700617 *Contributors*: A Magician, Achalmeena, Amarvk, Avishek Barua, Benwu, Bhadani, Bhause, Bjdehut, Blueronin, C1776M, Cagnol, CanadianLinuxUser, Catgut, Cgs, Chachrist, Cvanhasselt, Cybersteel8, Dbsheajr, Delaszk, Diego Azeta, Disquetemestizo, Dissolve, Djanke, Druzhnik, E. Ripley, Enderminh, FF2010, Flyer22, Former user, Gavin White, George100, Georgia guy, Gilliam, GoShow, Graham87, Graibeard, Guy Siedes, Harej, Harryboyles, Harsh.humtum.eng, Hazmat2, Hooperbloob, Ironwolf, JEBBAH, Jack Greenmaven, JavierMC, Jeff G., Jemoederiseenhoer1, Jeremykemp, Jojalozzo, JorgeGG, Karada, Kate, Kaushalarchana, Ketiltrout, Klavierspieler, Kstarsinic, Kuru, L Kensington, LaurensvanLieshout, Leszek Jańczuk, Ligulem, Lindsay658, Maclir2001, MagnusA, Manofradio, Markhurd, Marlon.dumas, Materialscientist, Maurreen, MaxPont, Mdd, Miller17CU94, Minesweeper, Mkoval, Mydogategodshat, Niravsdesai, Nixdorf, Olaf2, Oubiwann, Pasternaker, PatrickWeaver, Piano non troppo, Pm master, Pmtoolbox, Recognizance, RedWolf, Rgephart, RichardF, Ronz, Rui Covelo, SFK2, Saurabh13686, Sbugs, Scirocco6, SeanDuggan, Seangies, SkipHuffman, Smjg, Sspecter, Stovl, Swpb, Tga123, Tifoo, Tim Pritlove, Tolly4bolly, Tommy2010, Tony1, Truthraj, TutterMouse, Vrenator, Wayne Slam, Wikisteff, Wonderfl, Yuripobrasil, Zooloo, أحمد مصطفى السيد, 260 anonymous edits

Image Sources, Licenses and Contributors

File:Pert chart colored.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Pert_chart_colored.svg *License*: Public Domain *Contributors*: Pert_chart_colored.gif: Original uploader was Jeremykemp at en.wikipedia derivative work: Hazmat2 (talk)

File:SimpleAONwDrag3.png *Source*: http://en.wikipedia.org/w/index.php?title=File:SimpleAONwDrag3.png *License*: Creative Commons Attribution-Sharealike 3.0 *Contributors*: User:Nuggetkiwi

Image:CPMScheduleFloat.jpg *Source*: http://en.wikipedia.org/w/index.php?title=File:CPMScheduleFloat.jpg *License*: Public Domain *Contributors*: Alvin-cs, Timkeys

Image:Pert chart colored.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Pert_chart_colored.svg *License*: Public Domain *Contributors*: Pert_chart_colored.gif: Original uploader was Jeremykemp at en.wikipedia derivative work: Hazmat2 (talk)

Image:pert example gantt chart.gif *Source*: http://en.wikipedia.org/w/index.php?title=File:Pert_example_gantt_chart.gif *License*: GNU Free Documentation License *Contributors*: Dbsheajr, FSII, Jni, Sagsaw, Sfan00 IMG, Vanished 1850, 8 anonymous edits

Image:pert example network diagram.gif *Source*: http://en.wikipedia.org/w/index.php?title=File:Pert_example_network_diagram.gif *License*: GNU Free Documentation License *Contributors*: Dbsheajr, 2 anonymous edits

Image:pert example node legend.GIF *Source*: http://en.wikipedia.org/w/index.php?title=File:Pert_example_node_legend.GIF *License*: GNU Free Documentation License *Contributors*: Dbsheajr

Image:pert example network diagram visio.gif *Source*: http://en.wikipedia.org/w/index.php?title=File:Pert_example_network_diagram_visio.gif *License*: GNU Free Documentation License *Contributors*: "I created this myself using Microsoft Visio" -

Image:Commons-logo.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Commons-logo.svg *License*: logo *Contributors*: Anomie

License