Software Project Management

Chapter Seven

Risk management

Risk management

This lecture will touch upon:

- Definition of ‘risk’ and ‘risk management’
- Some ways of categorizing risk
- Risk management
  - Risk identification – what are the risks to a project?
  - Risk analysis – which ones are really serious?
  - Risk planning – what shall we do?
  - Risk monitoring – has the planning worked?
- We will also look at PERT risk and critical chains

Some definitions of risk

'the chance of exposure to the adverse consequences of future events' PRINCE2

'an uncertain event or condition that, if it occurs, has a positive or negative effect on a project's objectives' PM-BOK

- Risks relate to **possible future** problems, not current ones
- They involve a possible cause and its effect(s) e.g. developer leaves > task delayed

A framework for dealing with risk

The planning for risk includes these steps:

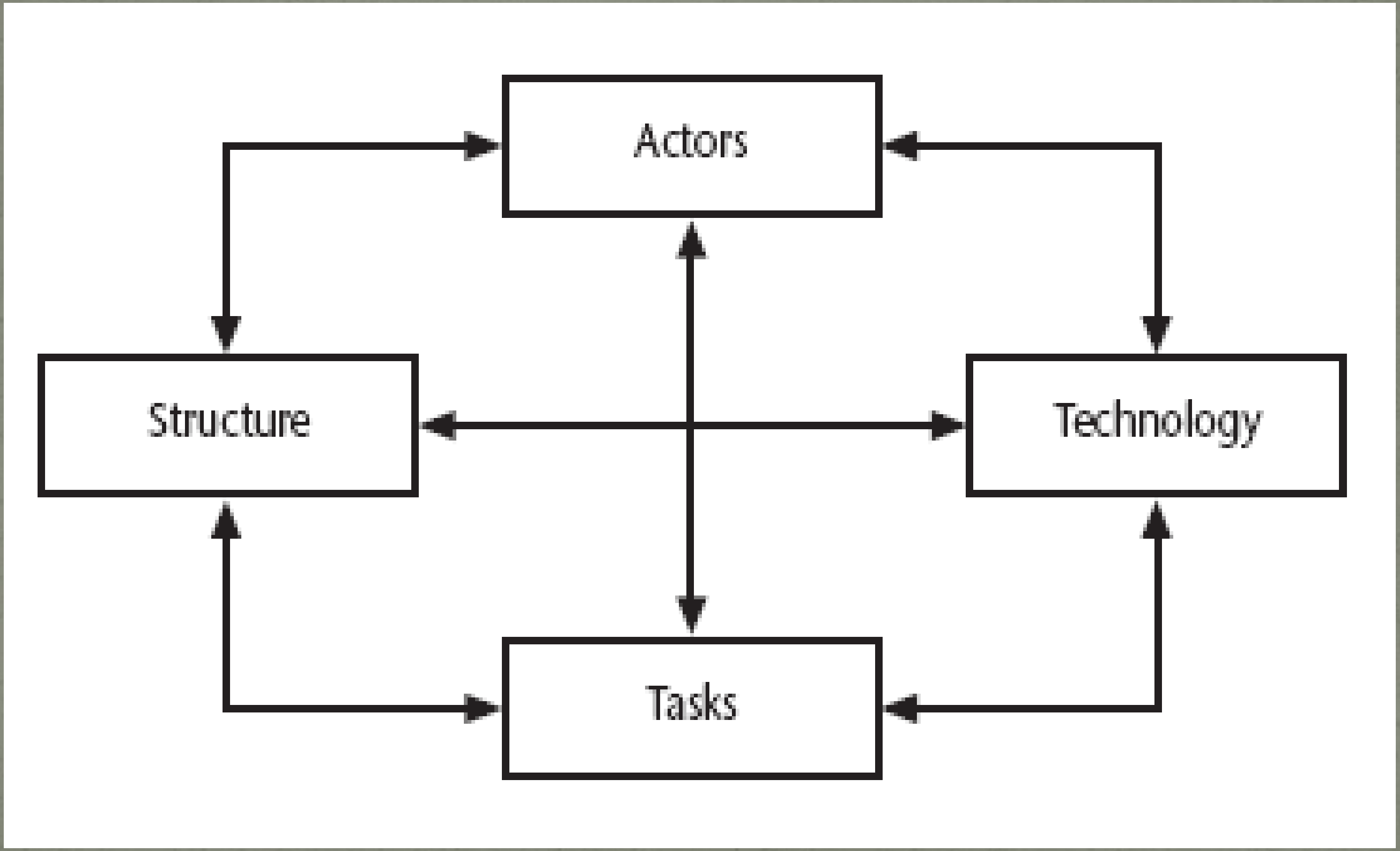
- Risk identification – what risks might there be?
- Risk analysis and prioritization – which are the most serious risks?
- Risk planning – what are we going to do about them?
- Risk monitoring – what is the current state of the risk?

Risk identification

Approaches to identifying risks include:

- Use of checklists – usually based on the experience of past projects
- Brainstorming – getting knowledgeable stakeholders together to pool concerns
- Causal mapping – identifying possible chains of cause and effect

Causal mapping - interventions



Boehm's top 10 development risks

Risk	*Risk reduction techniques*
Personnel shortfalls	Staffing with top talent; job matching; teambuilding; training and career development; early scheduling of key personnel
Unrealistic time and cost estimates	Multiple estimation techniques; design to cost; incremental development; recording and analysis of past projects; standardization of methods
Developing the wrong software functions	Improved software evaluation; formal specification methods; user surveys; prototyping; early user manuals
Developing the wrong user interface	Prototyping; task analysis; user involvement

Boehm's top ten risk - continued

Gold plating	Requirements scrubbing, prototyping, design to cost
Late changes to requirements	Change control, incremental development
Shortfalls in externally supplied components	Benchmarking, inspections, formal specifications, contractual agreements, quality controls
Shortfalls in externally performed tasks	Quality assurance procedures, competitive design etc
Real time performance problems	Simulation, prototyping, tuning
Development technically too difficult	Technical analysis, cost-benefit analysis, prototyping , training

Risk prioritization

Risk exposure (RE)

= (potential damage) x (probability of occurrence)

Ideally

Potential damage: a money value e.g. a flood would cause £0.5 millions of damage

Probability 0.00 (absolutely no chance) to 1.00 (absolutely certain) e.g. 0.01 (one in hundred chance)

RE = £0.5m x 0.01 = £5,000

Crudely analogous to the amount needed for an insurance premium

Risk Reduction Leverage (RRL)

- RRL is used to determine whether it is worthwhile to carry out the risk reduction plan.
- The higher is the RRL value, the more worthwhile is to carry out the risk reduction plan.

$$RRL = \frac{RE_{before} - RE_{after}}{risk\ reduction\ cost}$$

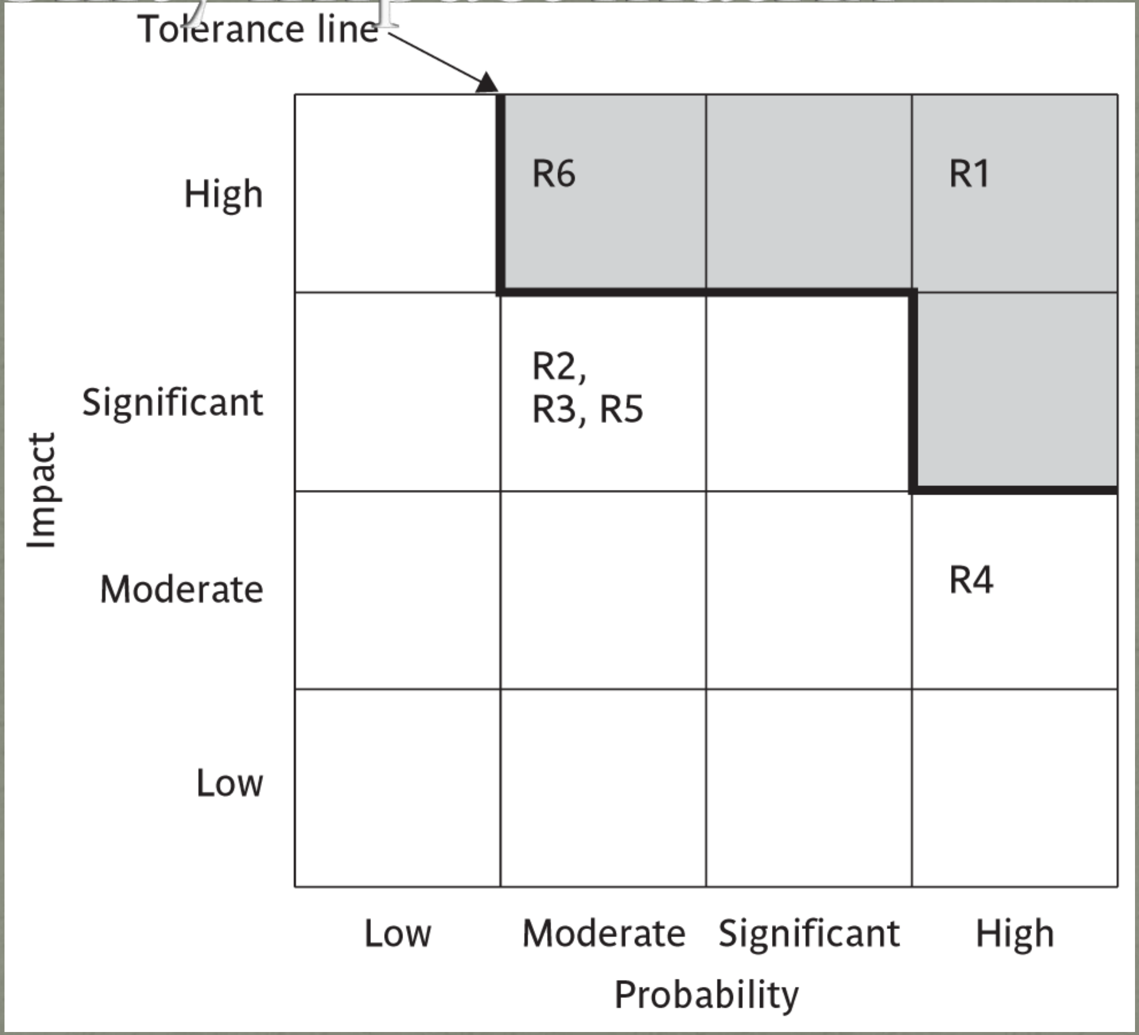
Risk probability: qualitative descriptors

Probability level	*Range*
High	Greater than 50% chance of happening
Significant	30-50% chance of happening
Moderate	10-29% chance of happening
Low	Less than 10% chance of happening

Qualitative descriptors of impact on cost and associated range values

Impact level	*Range*
High	Greater than 30% above budgeted expenditure
Significant	20 to 29% above budgeted expenditure
Moderate	10 to 19% above budgeted expenditure
Low	Within 10% of budgeted expenditure.

Probability impact matrix



	Low	Moderate	Significant	High
High		R6		R1
Significant		R2, R3, R5		
Moderate				R4
Low				

Risk planning

Risks can be dealt with by:

- Risk acceptance
- Risk avoidance
- Risk reduction
- Risk transfer
- Risk mitigation/contingency measures

Risk acceptance – do nothing option. The cost of avoiding the risk may be greater than the actual cost of the damage that might be inflicted

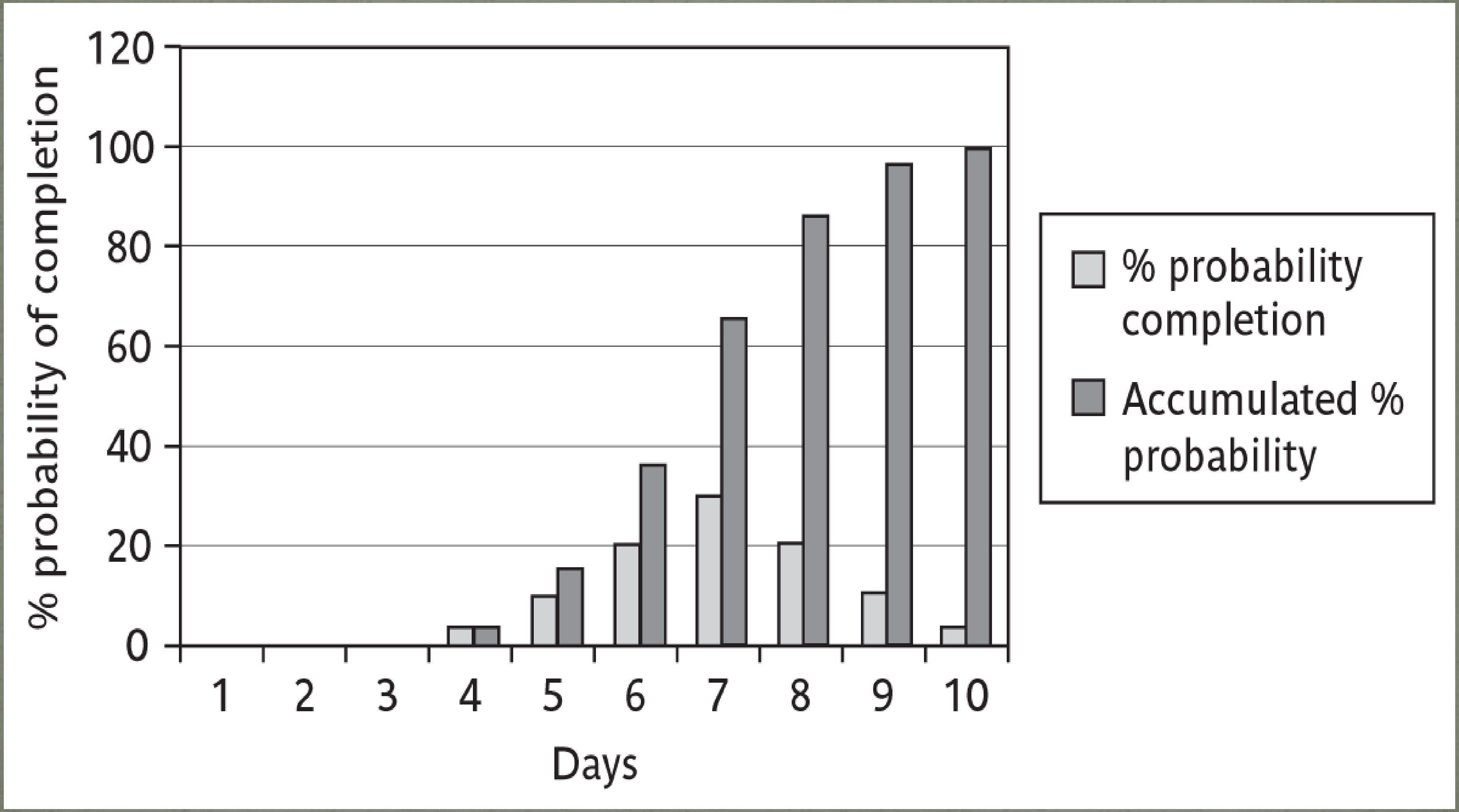
Risk avoidance – avoid the environment in which the risk occurs e.g. buying an OTS application would avoid a lot of the risks associated with software development e.g. poor estimates of effort.

Risk reduction – the risk is accepted but actions are taken to reduce its likelihood e.g. prototypes ought to reduce the risk of incorrect requirements

Risk transfer – the risk is transferred to another person or organization. The risk of incorrect development estimates can be transferred by negotiating a fixed price contract with an outside software supplier.

Risk mitigation – tries to reduce the impact if the risk does occur e.g. taking backups to allow rapid recovery in the case of data corruption

Probability chart

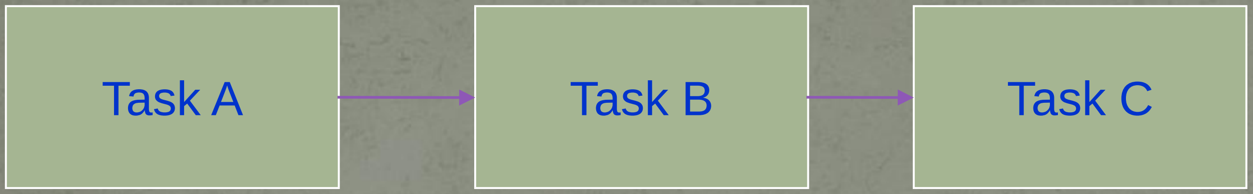


Using PERT to evaluate the effects of uncertainty

Three estimates are produced for each activity

- *Most likely time (m)*
- *Optimistic time (a)*
- *Pessimistic (b)*
- 'expected time' t_e = (a + 4m +b) / 6
- 'activity standard deviation' S = (b-a)/6

A chain of activities



Task A

Task B

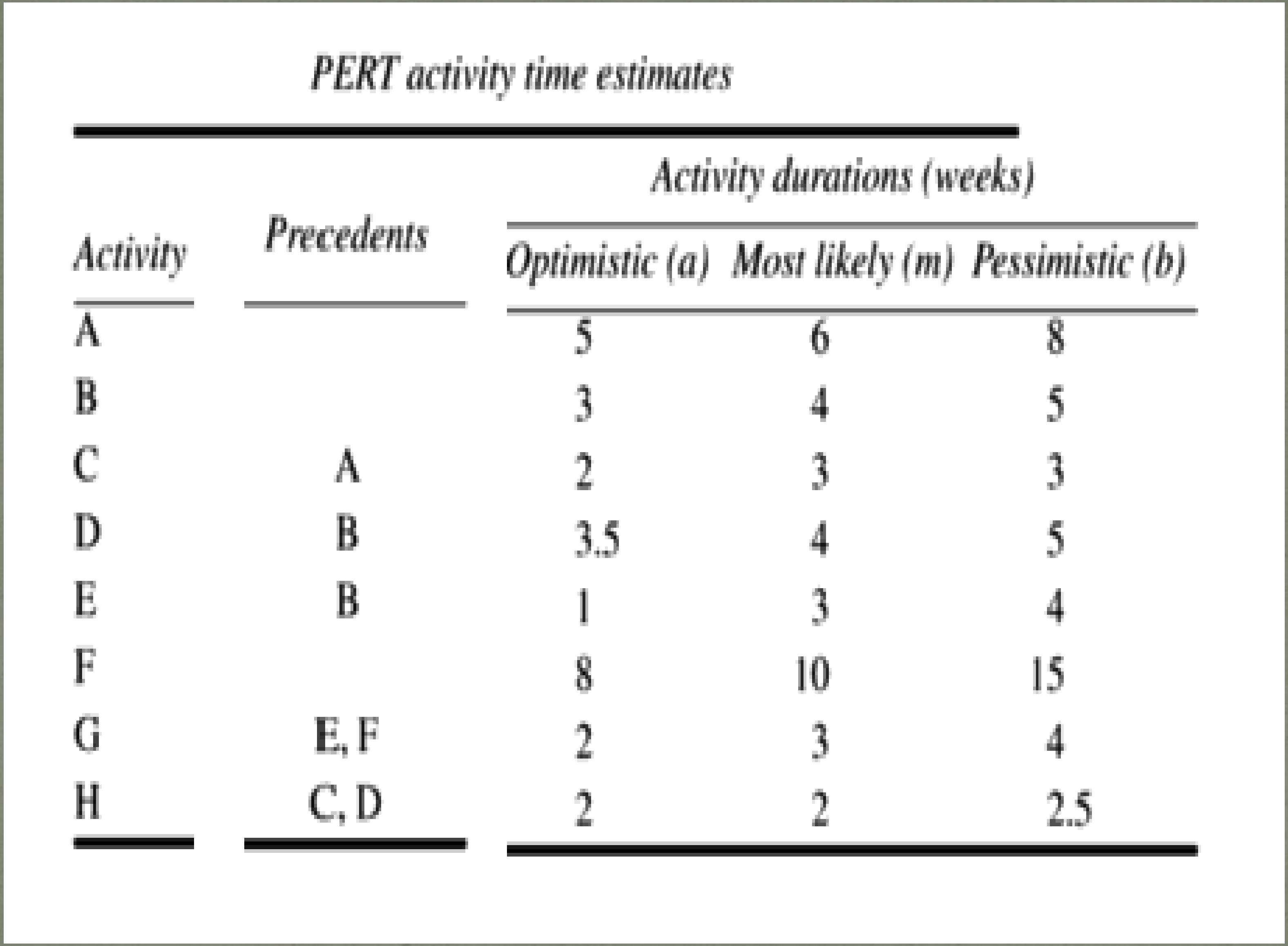
Task C

Task	a	m	b	t_e	s
A	10	12	16	?	?
B	8	10	14	?	?
C	20	24	38	?	?

A chain of activities

- What would be the expected duration of the chain A + B + C?
- Answer: 12.66 + 10.33 + 25.66 i.e. 48.65
- What would be the standard deviation for A + B+ C?
- Answer: square root of ($1^2 + 1^2 + 3^2$) i.e.

3.32



PERT activity time estimates

Activity	Precedents	Activity durations (weeks)		
		Optimistic (a)	Most likely (m)	Pessimistic (b)
A		5	6	8
B		3	4	5
C	A	2	3	3
D	B	3.5	4	5
E	B	1	3	4
F		8	10	15
G	E, F	2	3	4
H	C, D	2	2	2.5

, 2011

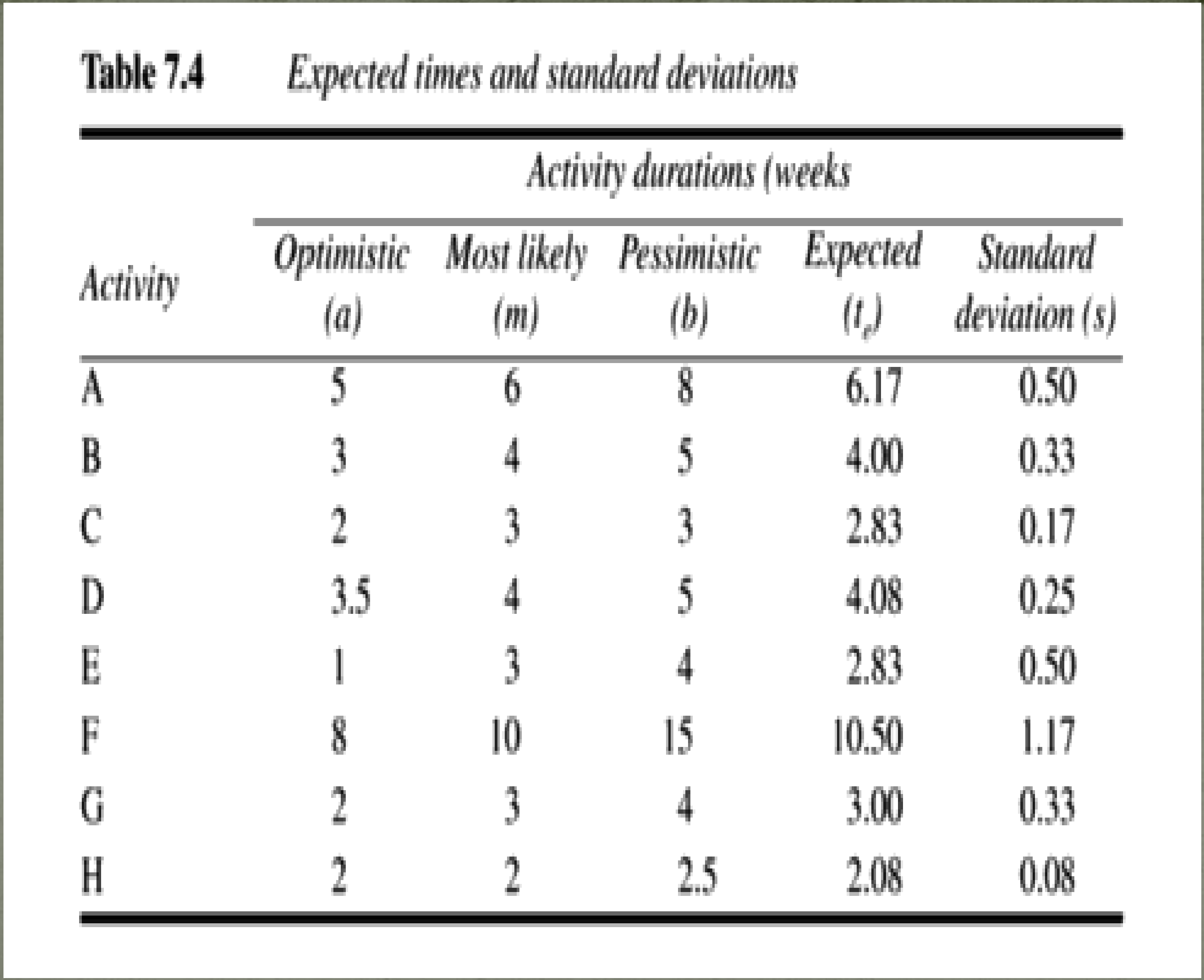
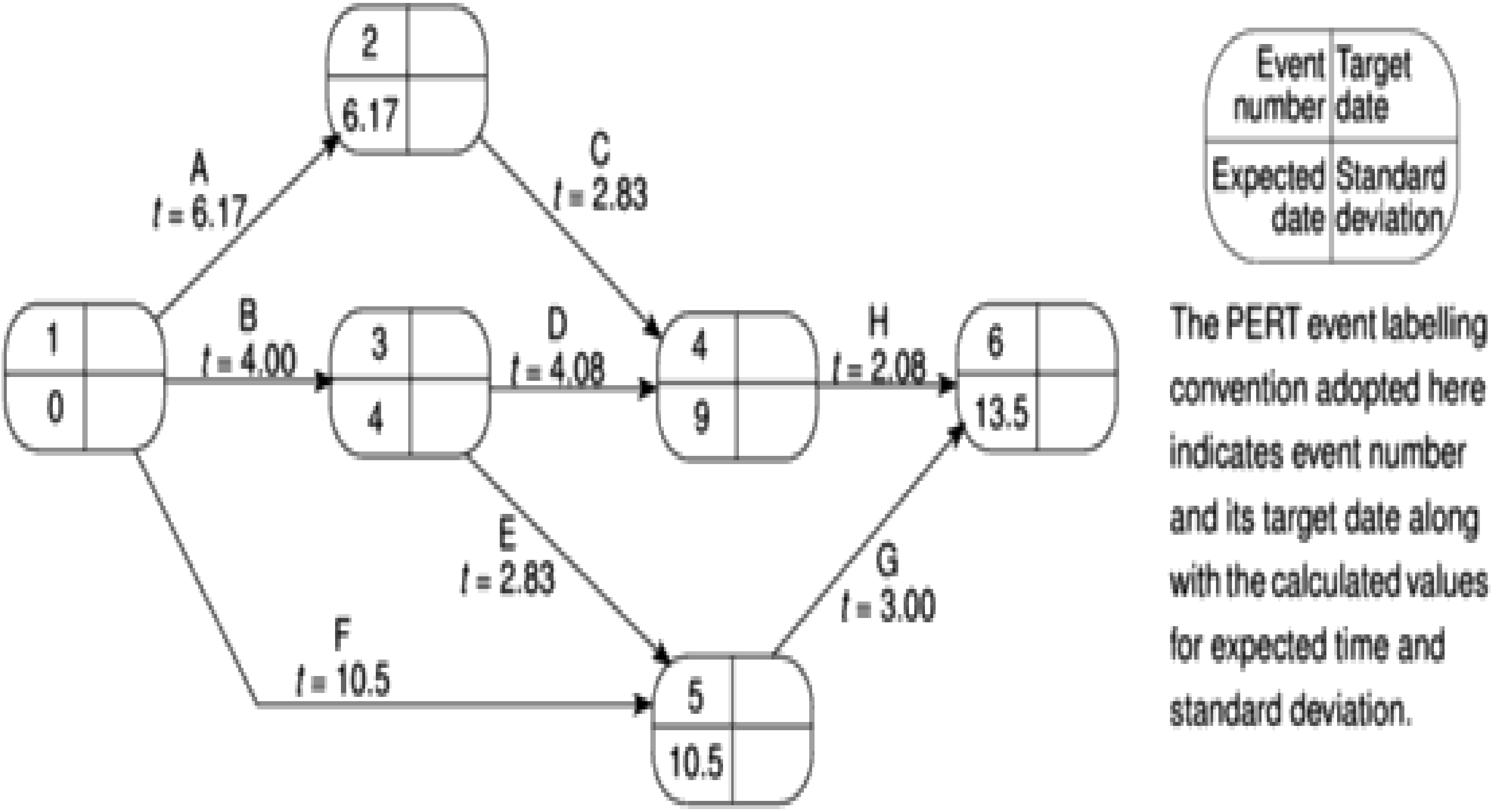


Table 7.4 *Expected times and standard deviations*

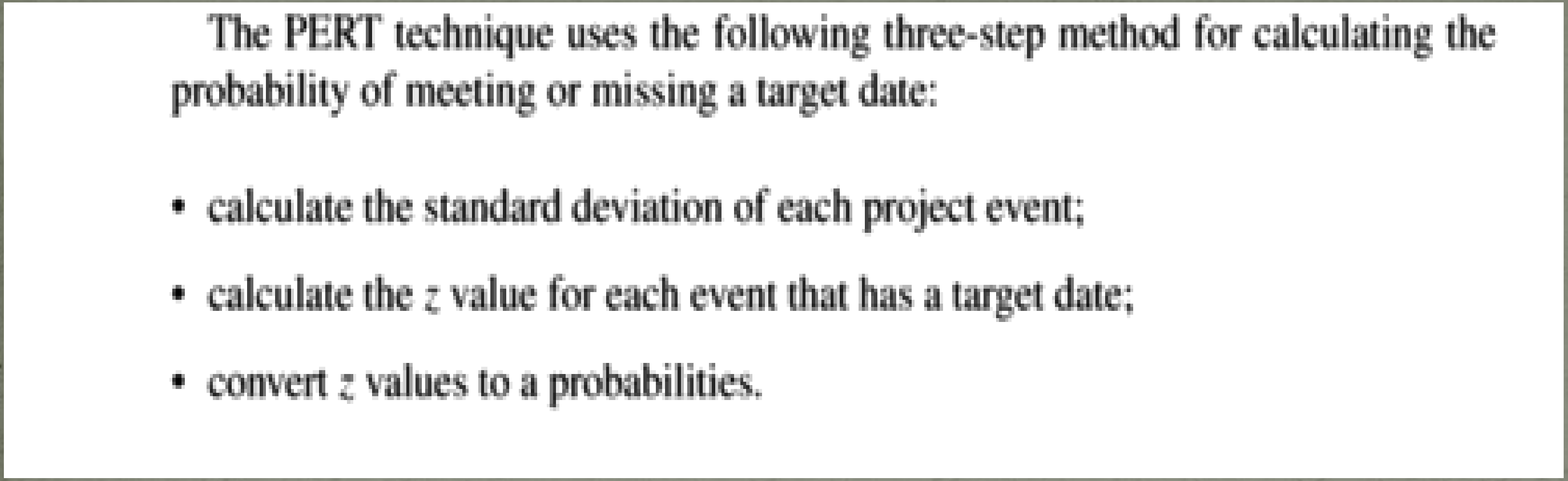
Activity	Activity durations (weeks				
	Optimistic (a)	Most likely (m)	Pessimistic (b)	Expected (t_e)	Standard deviation (s)
A	5	6	8	6.17	0.50
B	3	4	5	4.00	0.33
C	2	3	3	2.83	0.17
D	3.5	4	5	4.08	0.25
E	1	3	4	2.83	0.50
F	8	10	15	10.50	1.17
G	2	3	4	3.00	0.33
H	2	2	2.5	2.08	0.08

, 2011



The PERT event labelling convention adopted here indicates event number and its target date along with the calculated values for expected time and standard deviation.

Figure 7.3 *The PERT network after the forward pass.*



The PERT technique uses the following three-step method for calculating the probability of meeting or missing a target date:

- calculate the standard deviation of each project event;
- calculate the *z* value for each event that has a target date;
- convert *z* values to a probabilities.

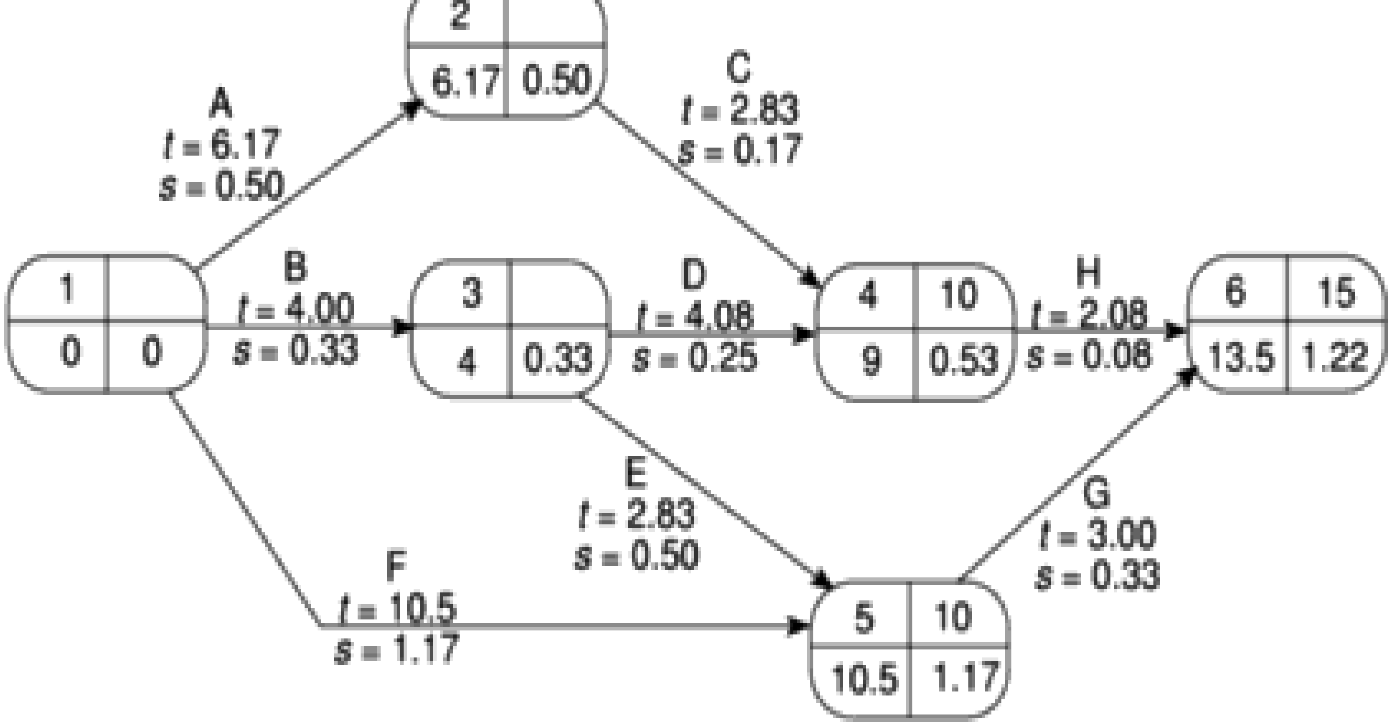


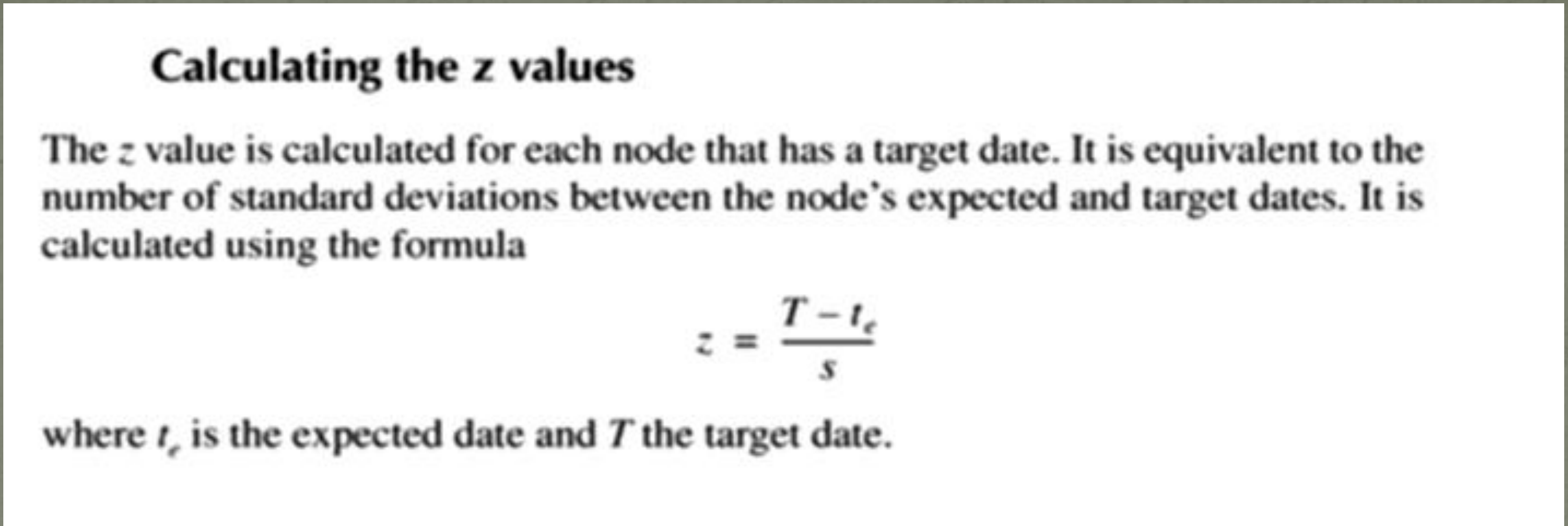
Figure 7.4 *The PERT network with three target dates and calculated event standard deviations.*

Calculating the standard deviation of each project event

Standard deviations for the project events can be calculated by carrying out a forward pass using the activity standard deviations in a manner similar to that used with expected durations. There is, however, one small difference – to add two standard deviations we must add their squares and then find the square root of the sum.

The standard deviation for event 3 depends solely on that of activity B. The standard deviation for event 3 is therefore 0.33.

For event 5 there are two possible paths, B + E or F. The total standard deviation for path B + E is $\sqrt{(0.33^2 + 0.50^2)} = 0.6$ and that for path F is 1.17; the standard deviation for event 5 is therefore the greater of the two, 1.17.

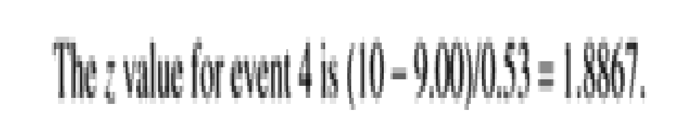


Calculating the z values

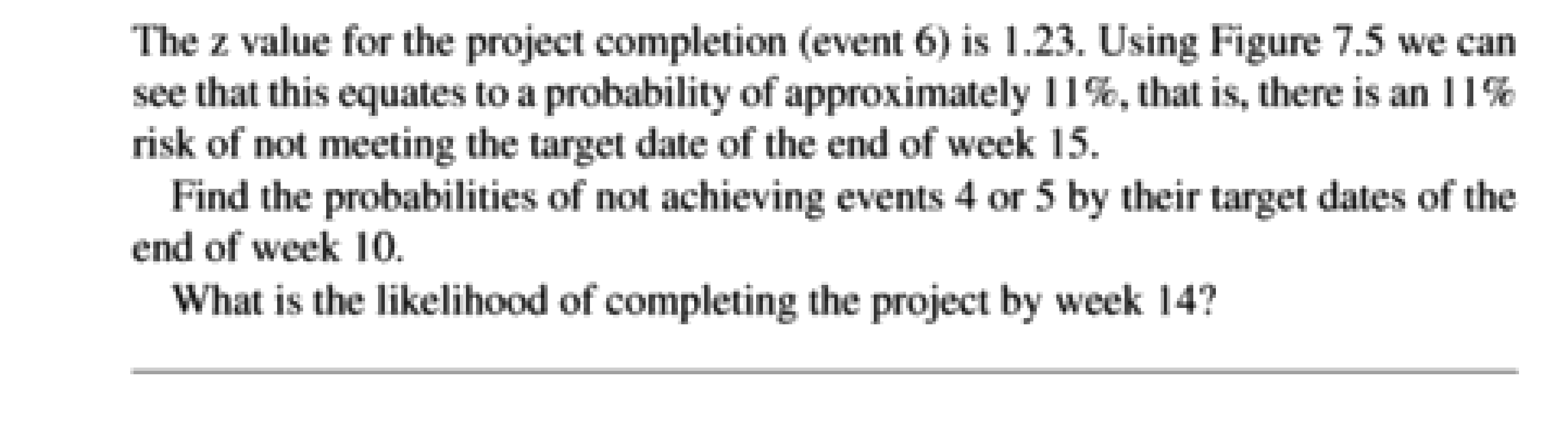
The z value is calculated for each node that has a target date. It is equivalent to the number of standard deviations between the node's expected and target dates. It is calculated using the formula

$$z = \frac{T - t_e}{s}$$

where t_e is the expected date and T the target date.



The z value for event 4 is (10 − 9.00)/0.53 = 1.8867.



The z value for the project completion (event 6) is 1.23. Using Figure 7.5 we can see that this equates to a probability of approximately 11%, that is, there is an 11% risk of not meeting the target date of the end of week 15.

Find the probabilities of not achieving events 4 or 5 by their target dates of the end of week 10.

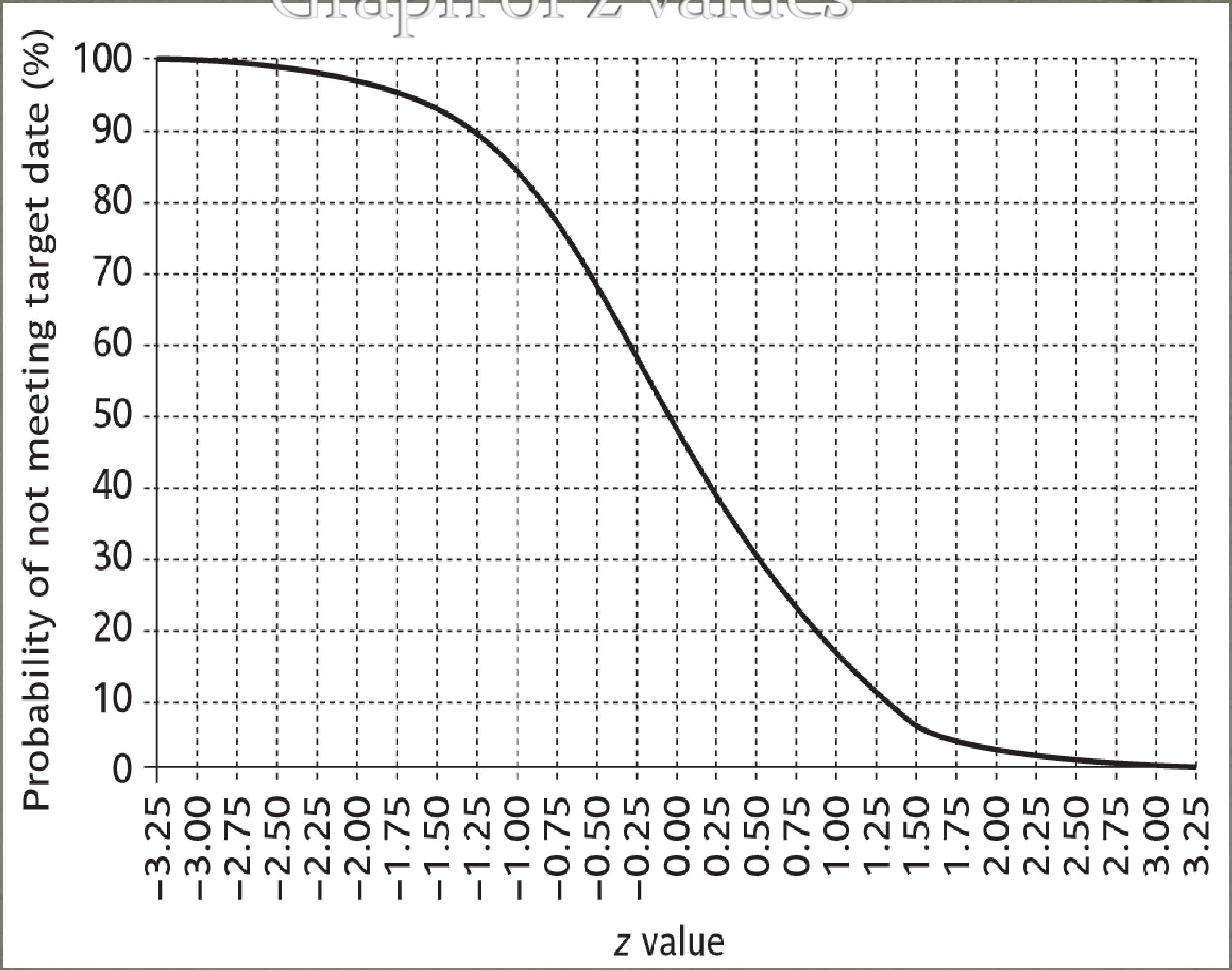
What is the likelihood of completing the project by week 14?

Assessing the likelihood of meeting a target

- Say the target for completing A+B+C was 52 days (T)
- Calculate the z value thus

  $z = (T - t_e)/s$
- In this example z = (52-48.33)/3.32 i.e. 1.01
- Look up in table of z values – see next overhead

Graph of z values



Converting z values to probabilities

A *z* value may be converted to the probability of not meeting the target date by using the graph in Figure 7.5.

This graph is the equivalent of tables of *z* values, also known as standard normal deviates, which may be found in most statistics textbooks

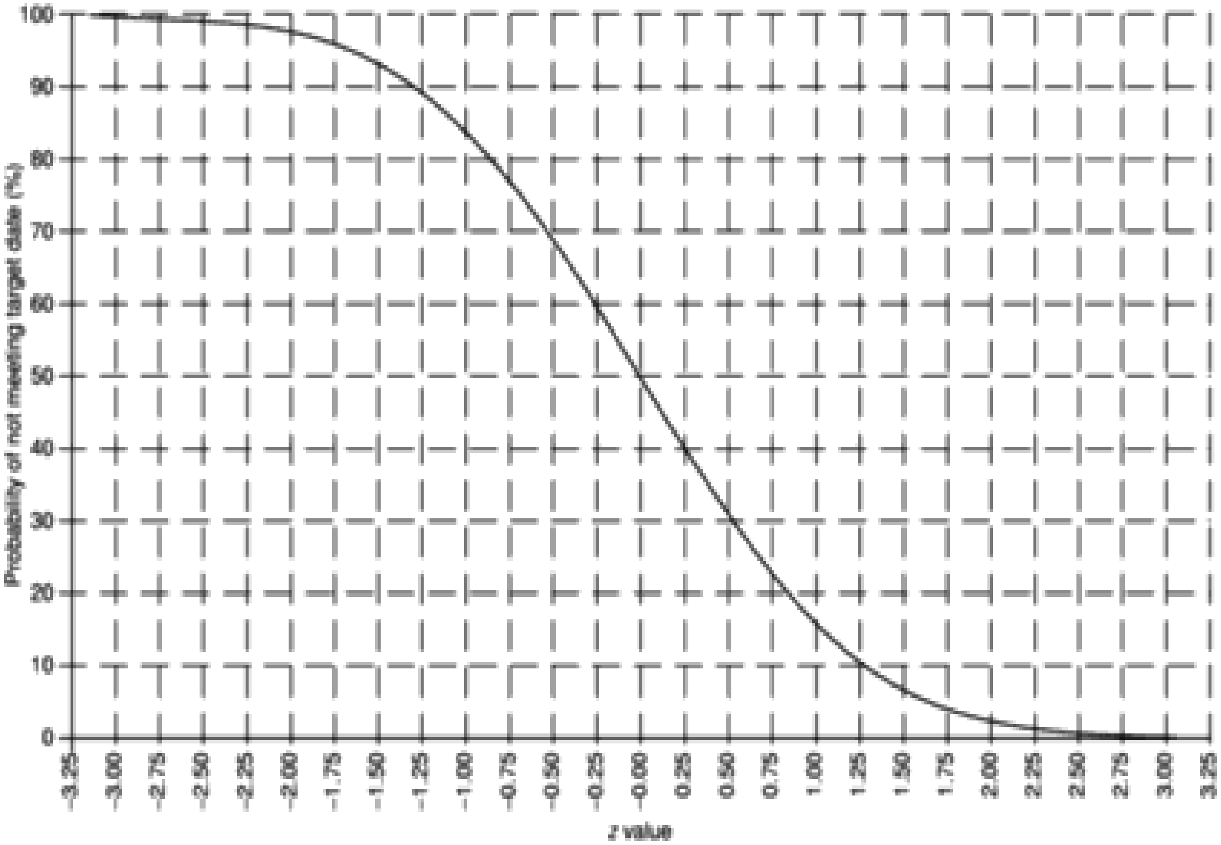


Figure 7.5 *The probability of obtaining a value within z standard deviations of the mean for a normal distribution.*

Monte Carlo Simulation

- An alternative to PERT.
- A class of general analysis techniques:
  - Valuable to solve any problem that is complex, nonlinear, or involves more than just a couple of uncertain parameters.
- Monte Carlo simulations involve repeated random sampling to compute the results.
- Gives more realistic results as compared to manual approaches.

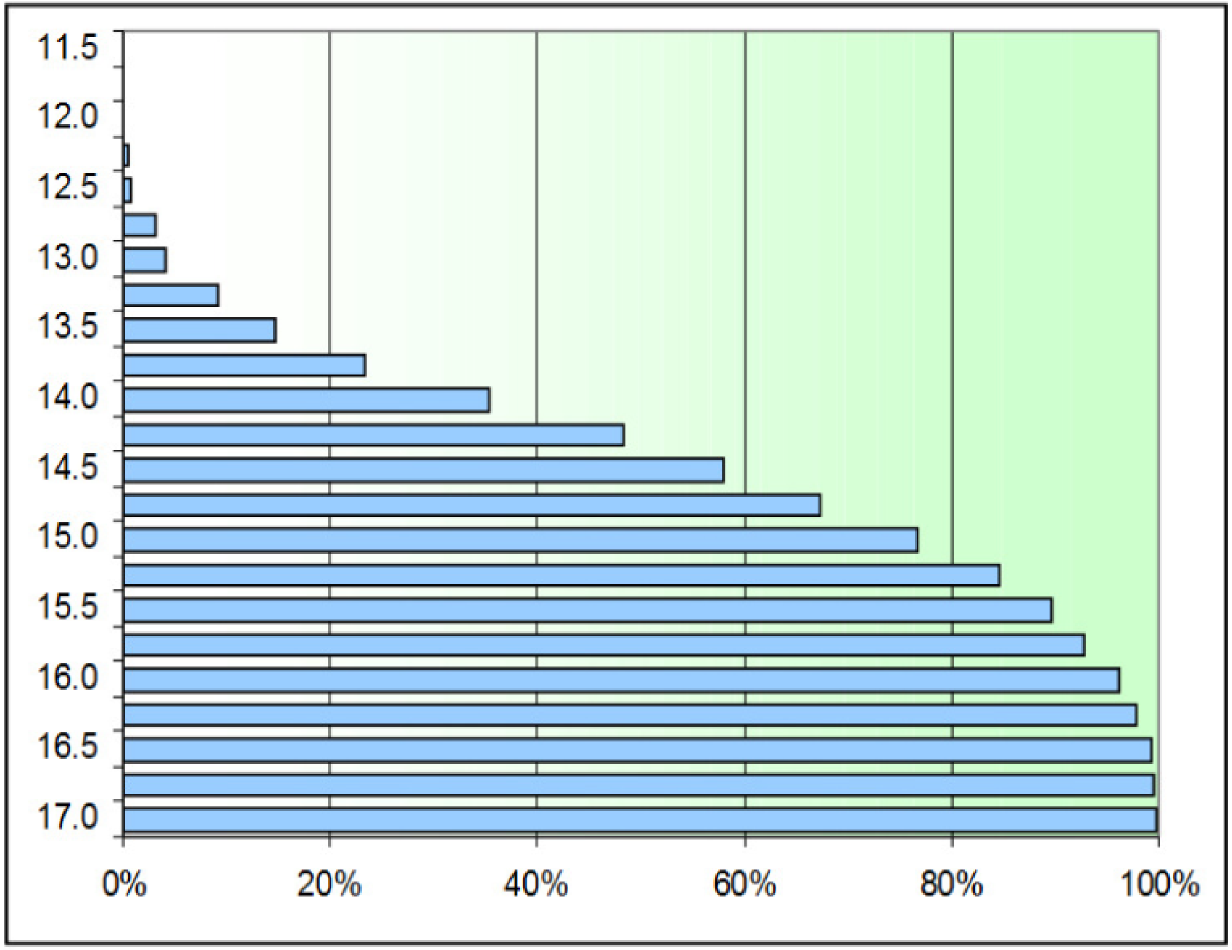


Figure 1: Probability of Completion Within Specified Time (Months)

Critical chain approach

One problem with estimates of task duration:

- Estimators add a safety zone to estimate to take account of possible difficulties
- Developers work to the estimate + safety zone, so time is lost
- No advantage is taken of opportunities where tasks can finish early – and provide a buffer for later activities

Critical chain approach

One answer to this:

1. Ask the estimators for two estimates
   - Most likely duration: 50% chance of meeting this
   - Comfort zone: additional time needed to have 95% chance
2. Schedule all activities suing most likely values and starting all activities on latest start dates

Critical chain approach

Originally this concept is developed by Eliyahu Goldratt.CCPM helps to overcome the following phenomenon

□ - Parkinson’s Law: Work expands to fill the available time.

□ - Student Syndrome: People start to work in full fledge only when deadline is near.

□ - Murphy's Law: What can go wrong will go wrong.

□ - Bad Multi Tasking: Bad multitasking can delay start of the successor tasks

General steps in critical chain concept

- The target date is given to the developer is one where it is estimated that there is a 50% chance of success.
- Working towards from the target completion date, each activity is scheduled to start as late as possible. Among other things, this should reduce the chance of staff being pulled off the project on to other work.

Scheduling Buffers

- In traditional estimates, people often add a buffer to each task and use it if it's needed or not
- Critical chain scheduling removes buffers from individual tasks and instead creates:
  - **Project buffers** or additional time added before the project's due date
  - **Feeding buffers** or additional time added before tasks on the critical path

Project buffer

- The ***project buffer*** protects the project from missing its scheduled end date due to variations along the critical chain. It places a portion of the safety margin time that was removed from each task estimate into a buffer task, thus moving the times of uncertainty from individual tasks to a pooled buffer task

Project buffer

- The project buffer is inserted between the final scheduled task and the scheduled project end date. The critical chain starts at the beginning of the project and ends at the start of the project buffer, not at the end of the project.
- Time is added to or subtracted from the project buffer as the actual time required to complete each task changes.
- The project buffer will be calculated as

  half the sum of the comfort zones of the activities on the critical chain

Feeding buffer

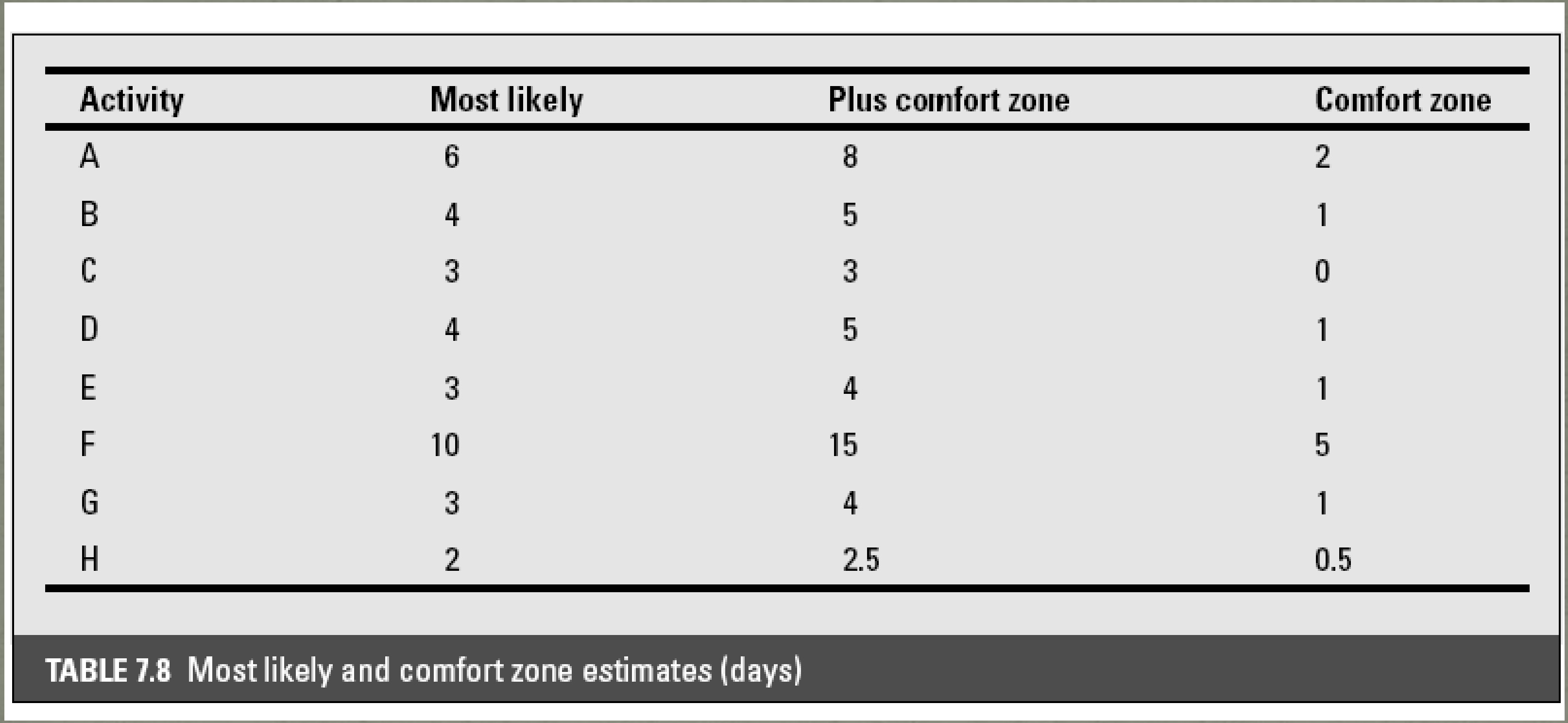
- The ***feeding buffer*** minimizes the risk that late completion of a non-critical chain task will affect the critical chain. The project manager inserts an amount of time at those points in the schedule where inputs from non-critical chain tasks merge with critical chain tasks. The result is very similar to a relay race where the speed of the race, in general, is able to be maintained by the overlap in runners at the hand-off point.
- Insert feeding buffers at points where non-critical chain paths intersect the critical chain

It can be calculated as …half the size of the comfort time taken out of the feeding path.

Comfort zone

- Comfort zone is the difference between the pessimistic and the most likely durations

Most likely and comfort zone estimates

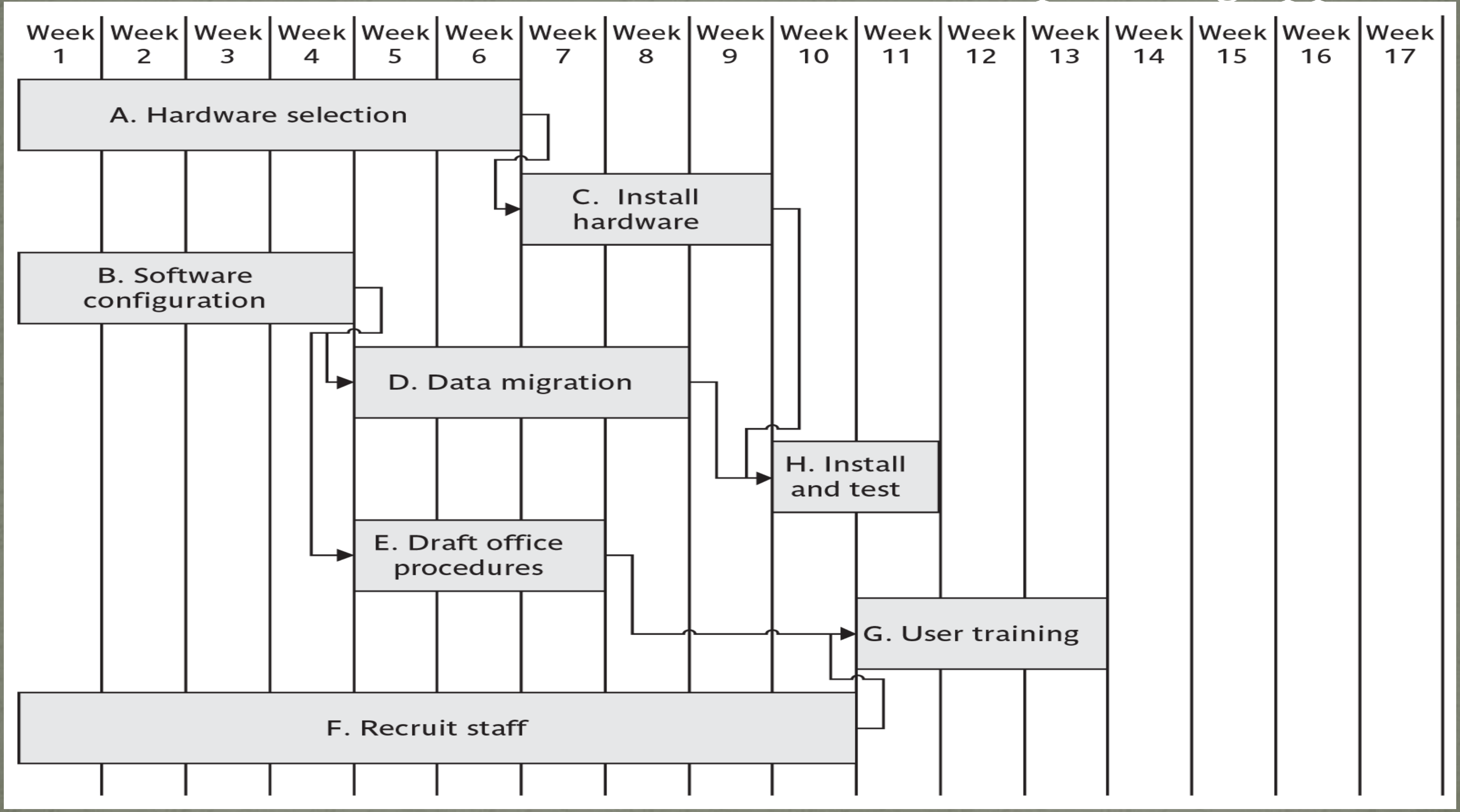


Activity	Most likely	Plus comfort zone	Comfort zone
A	6	8	2
B	4	5	1
C	3	3	0
D	4	5	1
E	3	4	1
F	10	15	5
G	3	4	1
H	2	2.5	0.5

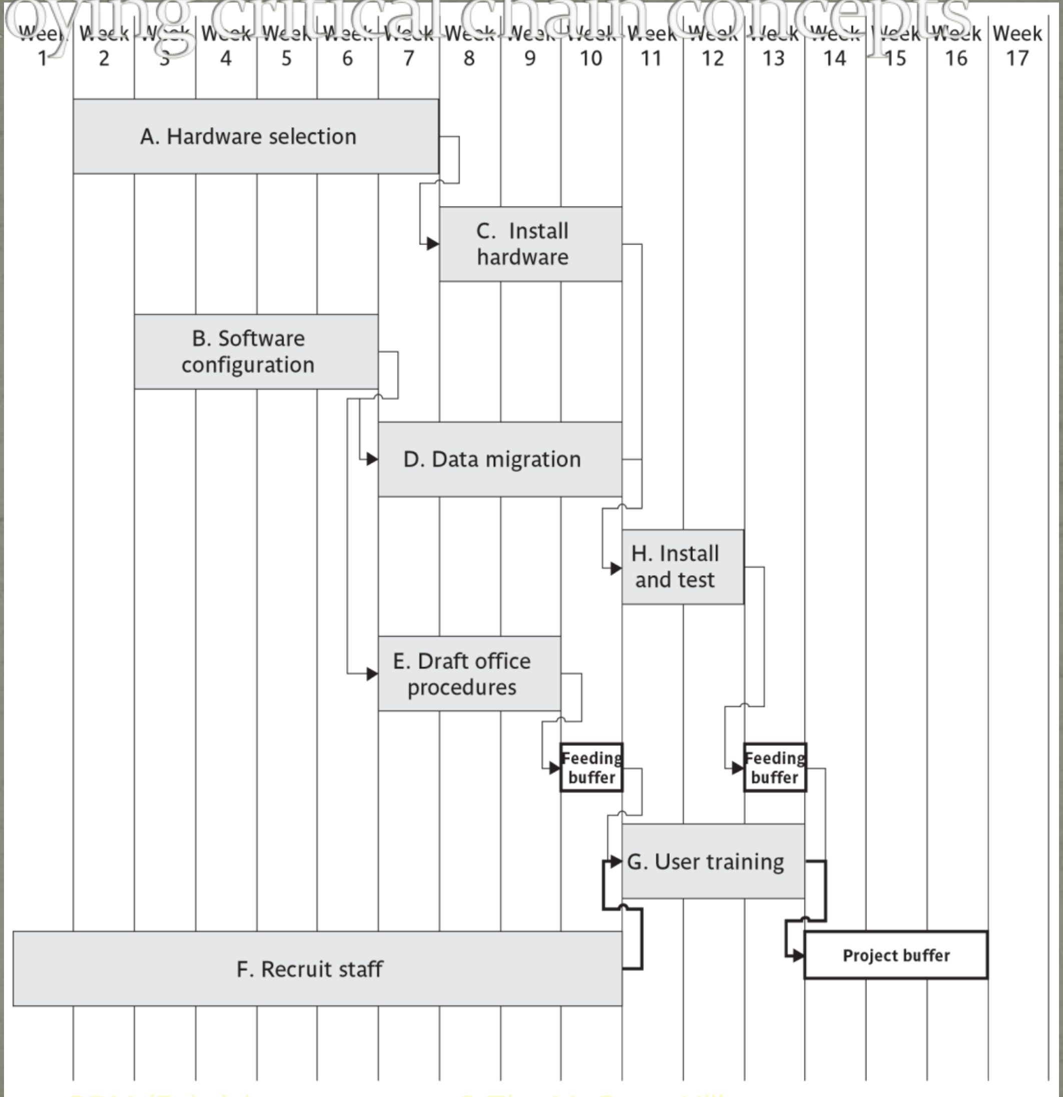
TABLE 7.8 Most likely and comfort zone estimates (days)

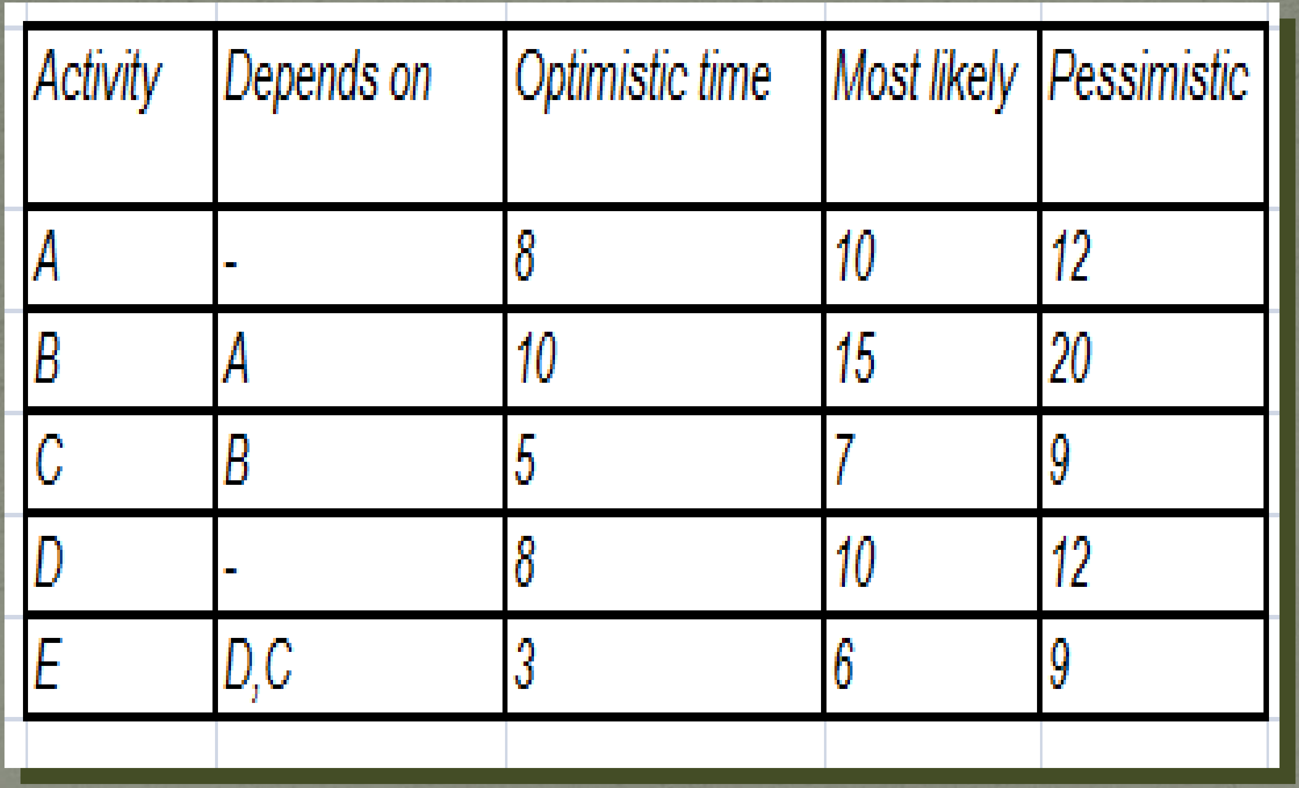
Critical chain concept

Traditional planning approach



Plan employing critical chain concepts

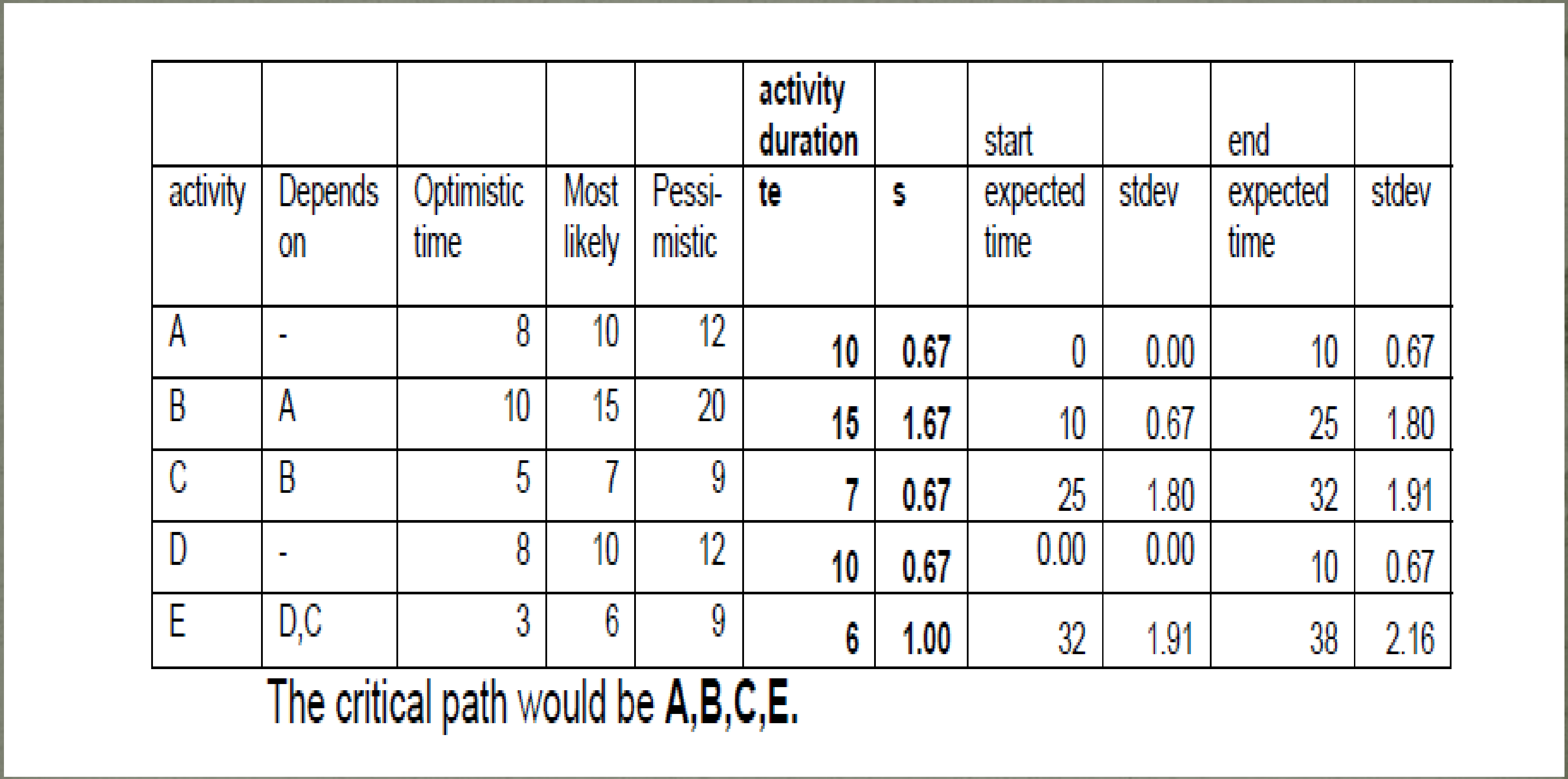




Activity	Depends on	Optimistic time	Most likely	Pessimistic
A	-	8	10	12
B	A	10	15	20
C	B	5	7	9
D	-	8	10	12
E	D,C	3	6	9

Using the activity table

- Calculate the expected duration and standard deviation for each activity
- Identify the critical path
- Draw up an activity diagram applying critical chain principles for this project
- Locate the places where the buffers will need to be located
- Assess the size of the buffers
- Start all activities as late as possible



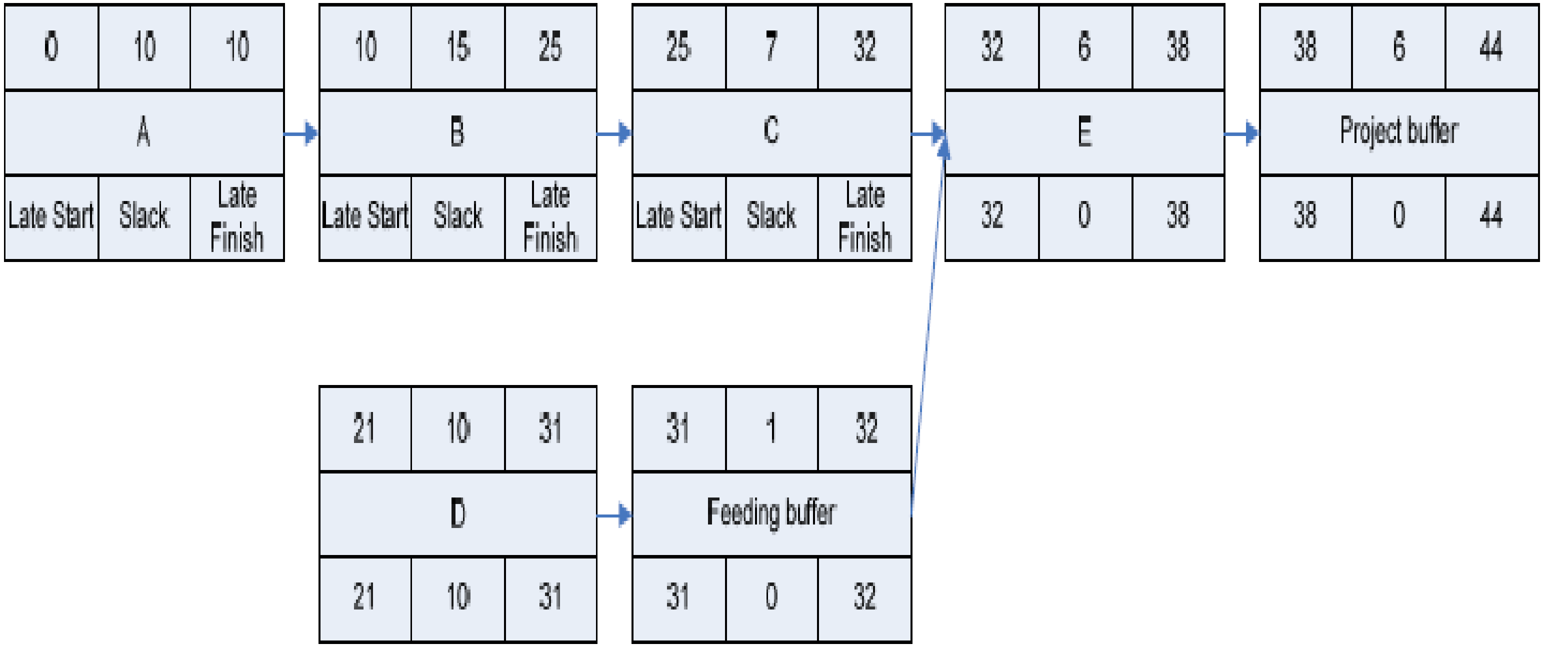
					activity duration		start		end	
activity	Depends on	Optimistic time	Most likely	Pessimistic	**te**	**s**	expected time	stdev	expected time	stdev
A	-	8	10	12	**10**	**0.67**	0	0.00	10	0.67
B	A	10	15	20	**15**	**1.67**	10	0.67	25	1.80
C	B	5	7	9	**7**	**0.67**	25	1.80	32	1.91
D	-	8	10	12	**10**	**0.67**	0.00	0.00	10	0.67
E	D,C	3	6	9	**6**	**1.00**	32	1.91	38	2.16

The critical path would be **A,B,C,E.**

Note: project buffer is half the sum of the differences between the pessimistic and most likely durations of the activities on the critical chain: i.e.

((12-10) + (20 -15) + (9-7) + (9-6))/2 = (2+5+2+3)/2 = 6 days

The feeding buffer is half the difference between the pessimistic and the most likely durations for Activity D



0	10	10
A		
Late Start	Slack	Late Finish

10	15	25
B		
Late Start	Slack	Late Finish

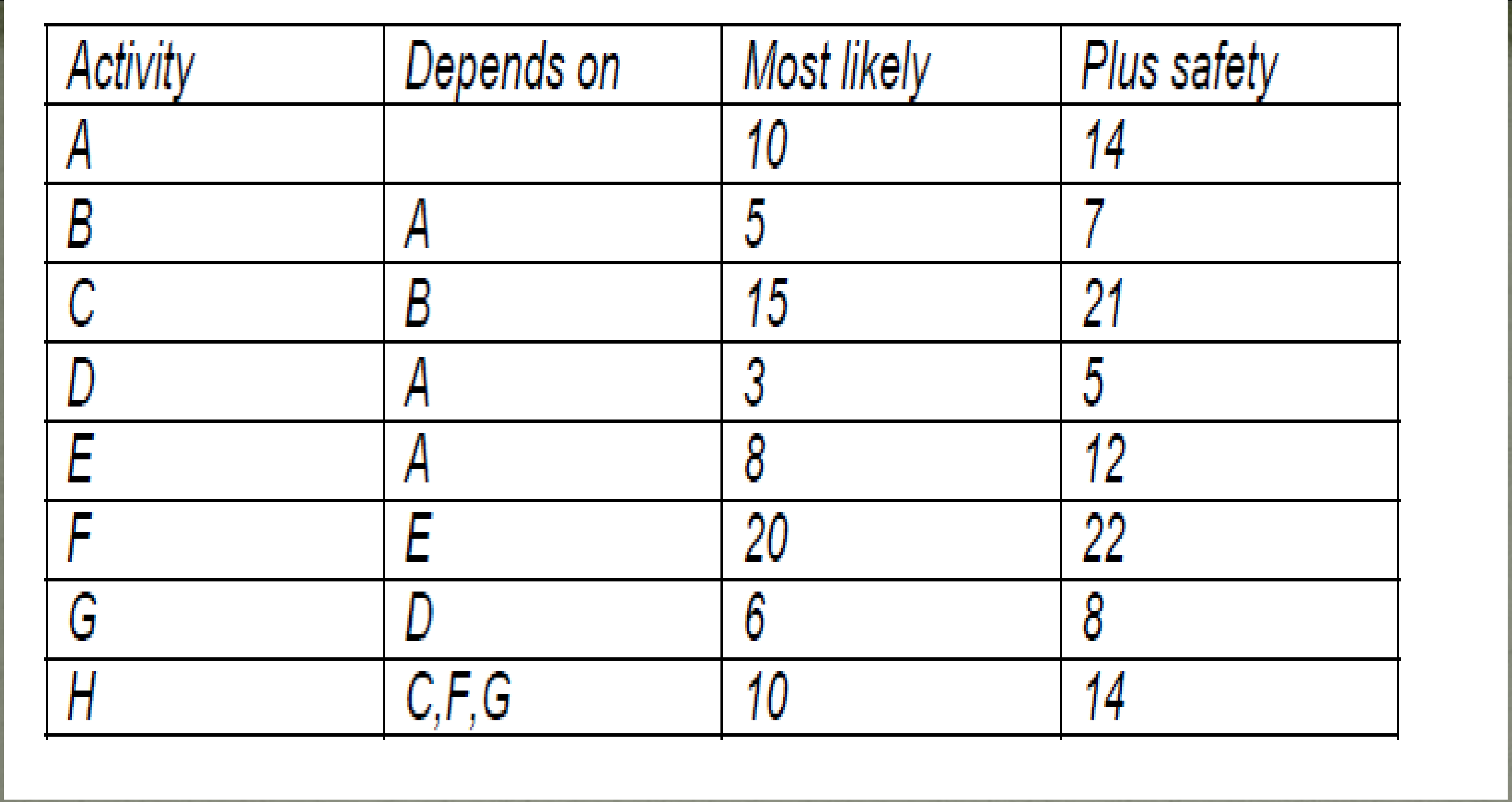
25	7	32
C		
Late Start	Slack	Late Finish

32	6	38
E		
32	0	38

38	6	44
Project buffer		
38	0	44

21	10	31
D		
21	10	31

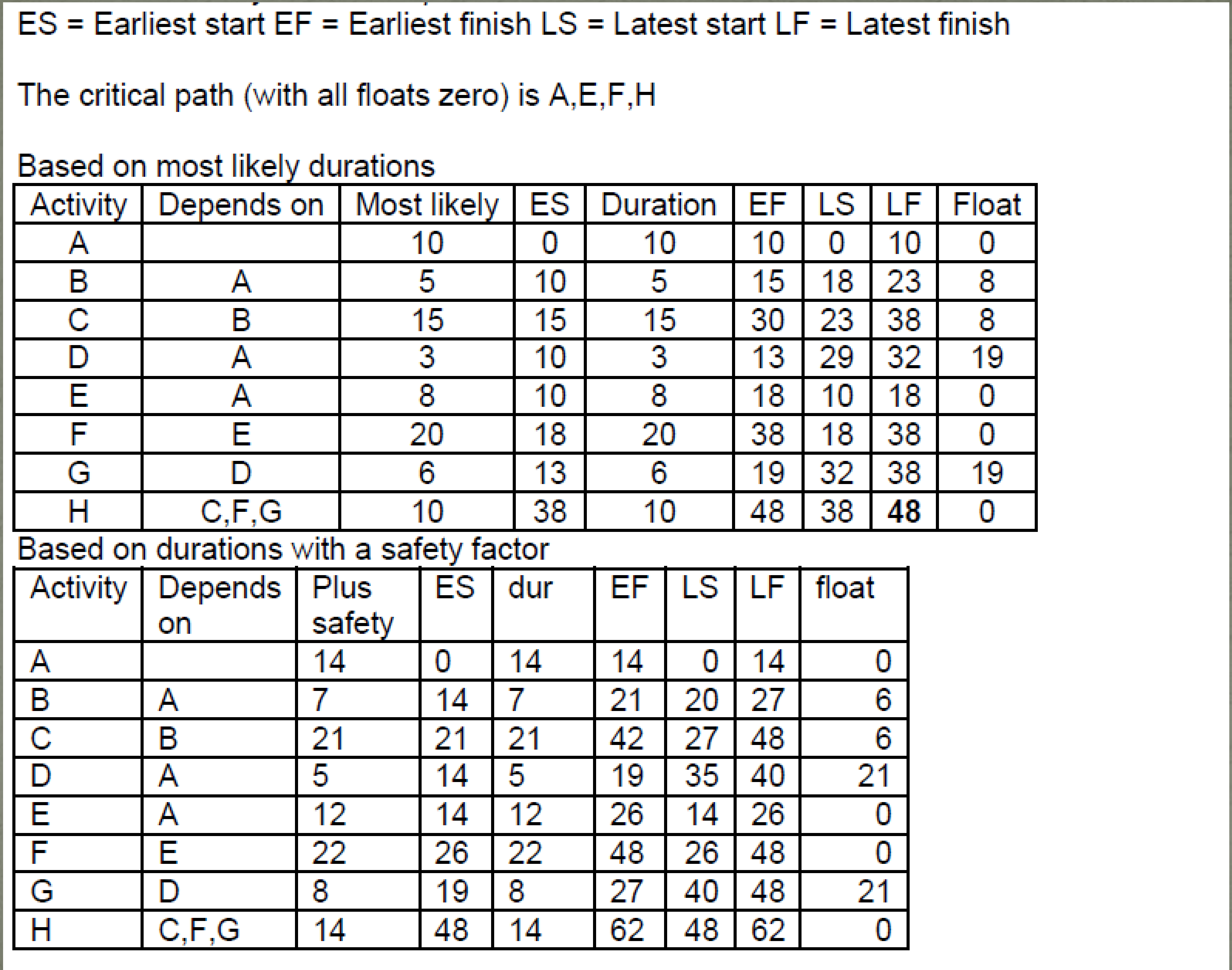
31	1	32
Feeding buffer		
31	0	32



Activity	Depends on	Most likely	Plus safety
A		10	14
B	A	5	7
C	B	15	21
D	A	3	5
E	A	8	12
F	E	20	22
G	D	6	8
H	C,F,G	10	14

Using the activity table

- Calculate the expected duration and standard deviation for each activity
- Identify the critical path
- Draw up an activity diagram applying critical chain principles for this project
- Locate the places where the buffers will need to be located



ES = Earliest start EF = Earliest finish LS = Latest start LF = Latest finish

The critical path (with all floats zero) is A,E,F,H

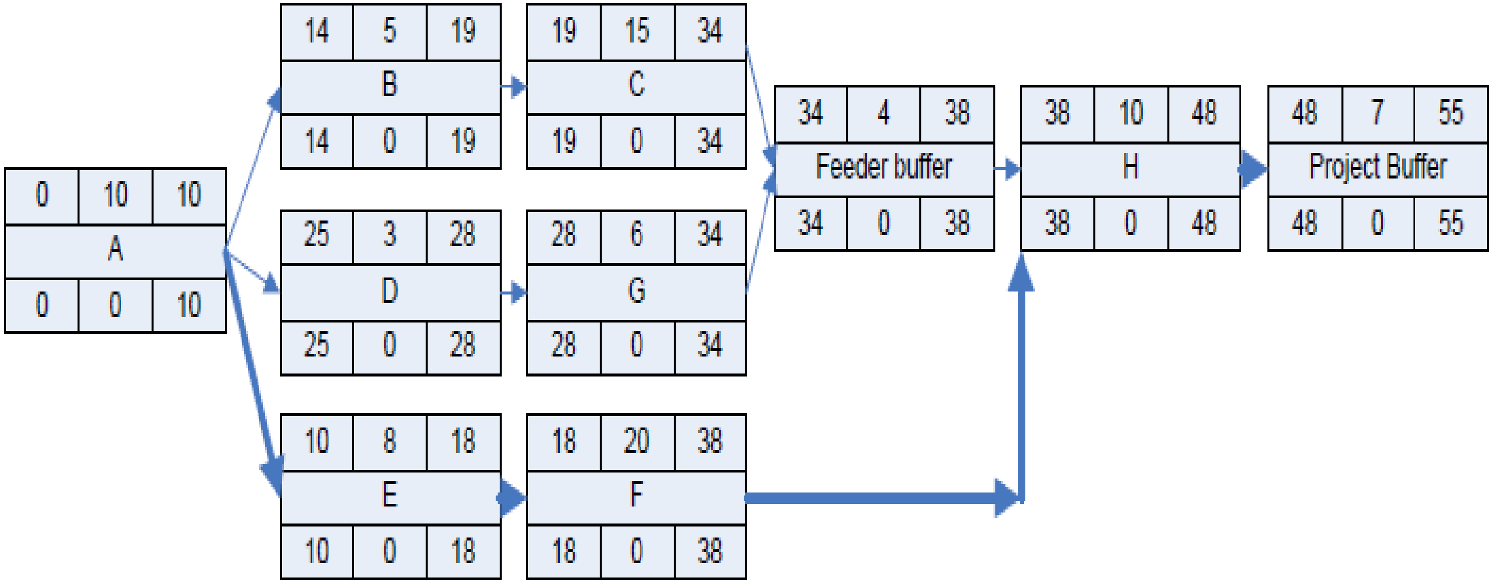
Based on most likely durations

Activity	Depends on	Most likely	ES	Duration	EF	LS	LF	Float
A		10	0	10	10	0	10	0
B	A	5	10	5	15	18	23	8
C	B	15	15	15	30	23	38	8
D	A	3	10	3	13	29	32	19
E	A	8	10	8	18	10	18	0
F	E	20	18	20	38	18	38	0
G	D	6	13	6	19	32	38	19
H	C,F,G	10	38	10	48	38	**48**	0

Based on durations with a safety factor

Activity	Depends on	Plus safety	ES	dur	EF	LS	LF	float
A		14	0	14	14	0	14	0
B	A	7	14	7	21	20	27	6
C	B	21	21	21	42	27	48	6
D	A	5	14	5	19	35	40	21
E	A	12	14	12	26	14	26	0
F	E	22	26	22	48	26	48	0
G	D	8	19	8	27	40	48	21
H	C,F,G	14	48	14	62	48	62	0

Note that there is only one feeder buffer for B+C and D+G, as these are running in parallel. In this case they share a buffer with the size based on whichever feeding buffer would have been bigger. For B+C this would have been 4 days (50% of 2+6 days) as opposed to 2 days for D+G (50% of 2+2 days).



0	10	10
A		
0	0	10

14	5	19
B		
14	0	19

19	15	34
C		
19	0	34

25	3	28
D		
25	0	28

28	6	34
G		
28	0	34

10	8	18
E		
10	0	18

18	20	38
F		
18	0	38

34	4	38
Feeder buffer		
34	0	38

38	10	48
H		
38	0	48

48	7	55
Project Buffer		
48	0	55

Executing the critical chain-based plan

- No **chain** of tasks is started earlier than scheduled, but once it has started is finished as soon as possible
- This means the activity following the current one starts as soon as the current one is completed, even if this is early – the relay race principle

Executing the critical chain-based plan

Buffers are divided into three zones:

- Green: the first 33%. No action required
- Amber : the next 33%. Plan is formulated
- Red : last 33%. Plan is executed.