Definition of an SP.

Software project is the complete procedure of software development from requirement gathering to testing & maintenance, carried out according to execution methodologies, in a specified period of time to achieve intended software product.

SP v/s other types of projects.

1. Invisibility ⇒ SP's are not immediately visible.
2. Complexity ⇒ SP's contain more complexity than other engineering artefacts.
3. Conformity ⇒ compliance with standards, rules or laws.
4. Flexibility ⇒ s/w change very frequently. must be able to accomodate other components.

Activities covered by SPM

Feasibility study

Planning

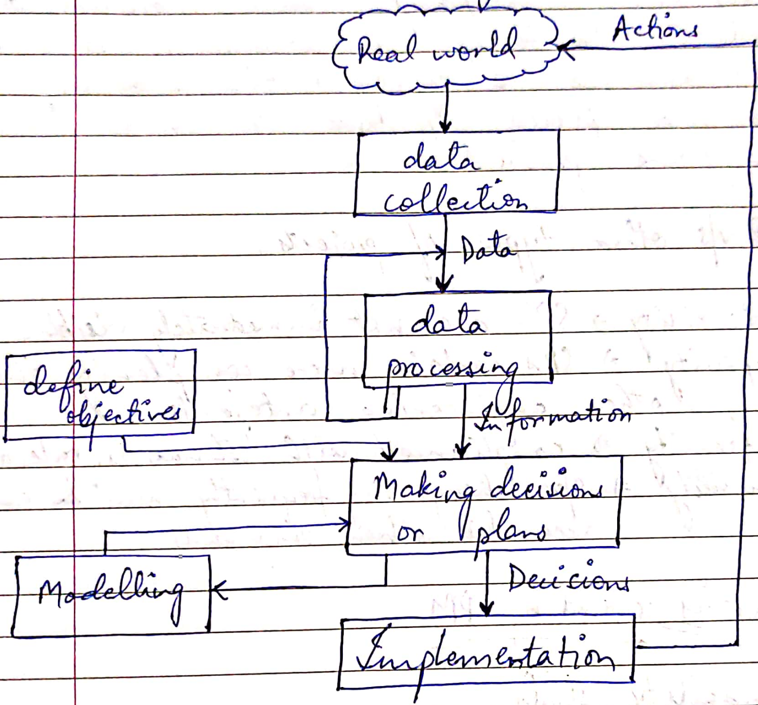
Project Execution

SP categories.

Compulsory v/s Voluntary users
Info system v/s embedded systems
Objectives v/s products.

Management Control

↳ involves setting objectives for a system & then monitoring ~~its~~ the performance.



Project Scheduling: → what, when & who?

↳ Project planning activity.
↳ Decides which funcs. to be performed & when.

1. Identify all the fncs. reqd to complete the project.
2. Break down large funcs.
3. Project scope, design & develop task breakdown structure.
4. Project-related tasks.
5. Identify HR & material requisites.
6. ~~Eo~~ Evaluate approx time reqd.
7. Allocation of resources.
8. Analyze schedule
9. Monitor & govern the schedule.

Work Breakdown Structure

↳ deliverable oriented hierarchical decomposition of the work to be executed by project team to accomplish the project objectives & create reqd deliverables.
↳ identified, estimated, scheduled & budgeted.

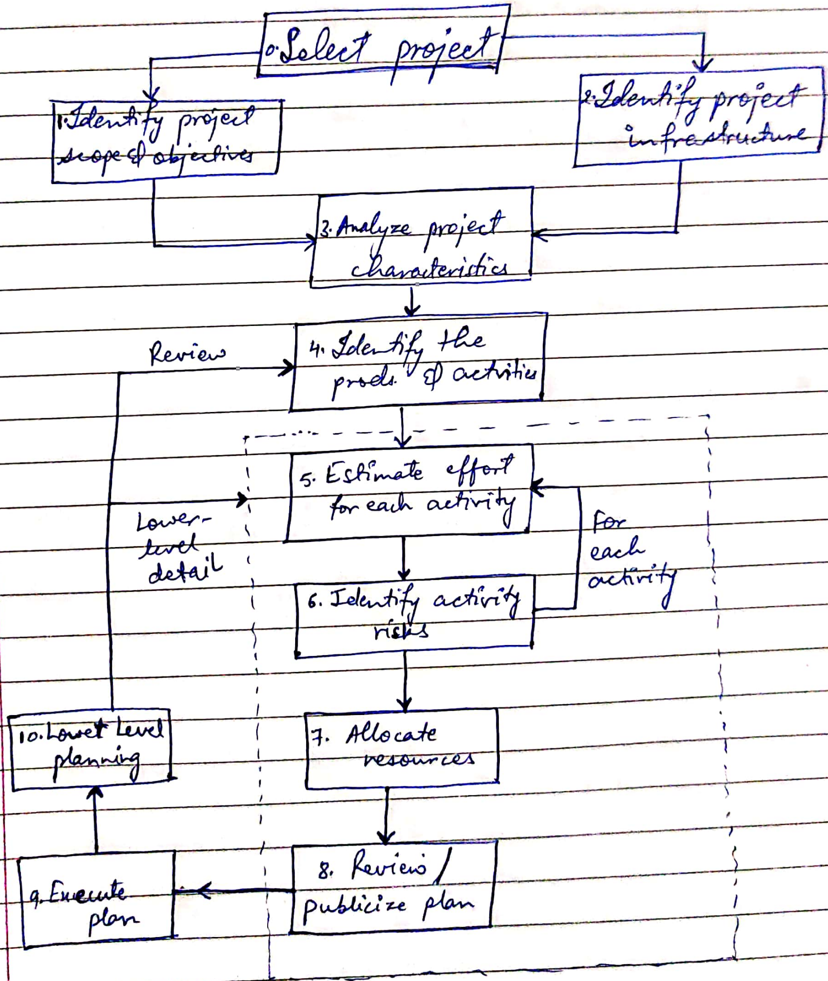
Characteristics: 1. Definable 2. Manageable
3. Estimateable 4. Independant
5. Integratable 6. Measureable
7. Adaptable

~~(1) Selecting a project~~
~~(2) Identify project scope & objectives~~

Project Planning

0 Select Project
1. Identify project scope & objectives.
   1.1. Measures & aim of the effectiveness
   1.2. Establish project authority.
   1.3. Stakeholders
   1.4 Modification of objs. due to Stakeholders
   1.5 Communicate.
2. Identify project infrastructure
   2.1. Reln/ship b/w project & strategic planning
   2.2. Install stds & procedures.
   2.3. Project Team Orgns.
3. Analyze project characteristics
   3.1. objective or product-driven
   3.2. Addnal project charac.
   3.3. High level risks.
   3.4. Review resources.
4. Identify project product & activities.
   4.1. Describe project products (quality criteria)
   4.2 Document product flow
   4.3. Recognise product instances
   4.4. Produce ideal activity n/w.
   4.5. Modify stages & checkpoints
5. Estimate effort for each activity
   5.1. Bottom up estimates.
6. Identify activity risks
   6.1 Quantify risks

6.2. Plan risk reduction
6.3. Adjust plans & estimations (money)
7. Allocate resources
7.1. Identify & revise
8. Review/publicize plan
9./10. Execute plan / lower level planning.



Unit II

Project Estimation & Evaluation

Estimation is a ~~technique~~ process of finding an estimate or approximation, which is a value that can be used for some purpose even if input data may be incomplete, uncertain or unstable. It determines money, effort, resource & time.

Steps:
- ↳ Size of the development prod.
- ↳ effort in person-months or hours.
- ↳ schedule in calendar months
- ↳ cost.

Decomposition Technique → [Divide & Conquer]

1. Understand the scope
2. Software size
   - ↳ start with scope
     - ↳ build funcs. & calculate size of funcs.
     - ↳ derive effort & cost.
     - ↳ combine func. estimates & apply to entire project (estimate)
3. Break-down the SDLC activities
   - ↳ identify sequence of activities & then estimate cost for each (as in step 2).
4. Compare & consider both estimates & vast divergence must not occur. Check scope and/or data.
5. Lessen the divergence & reconcile estimates again.

from past experience or obsvs.

Empirical Estimation Models

↳ It is a formula derived from data collected from past s/w projects, that uses software size to estimate effort.

Regression Analysis on data collected from past s/w projects.

$$E = A + B(e_v)^C$$

effort

LOC or FP

line of code or func. Points

A, B, C ⟹ derived constants.

★ COCOMO II Model

↳ require sizing info

① LOC ② FP ③ Object Points.

$$NOP = (\text{Object Points}) \times \left(\frac{100 - \%\text{reuse}}{100}\right)$$

new OP

$$Prod = \frac{NOP}{\text{person-month}}$$

productivity rate

$$Effort = \frac{NOP}{Prod}$$

* Software Equation
  ↳ dynamic model that considers many variables (effort, duration, skills, productivity) that assumes a specific distribution of effort over life of a s/w project.

Estimation for object Oriented Projects

Approaches
↳ effort decomposition, FP analysis.
↳ Use cases developed
↳ Classes which are imp.
↳ Interfaces, support classes multiplier.

Agile development projects

Estimation for web Engineering projects

↳ mostly based on prediction.
↳ short schedules & highly fluidic scope.

Prediction process

↳ identification of Web pages, images etc.
↳ static or dynamic pages.
↳ capture historical data.
↳ project characteristics.

Cost Benefit Analysis

2 steps

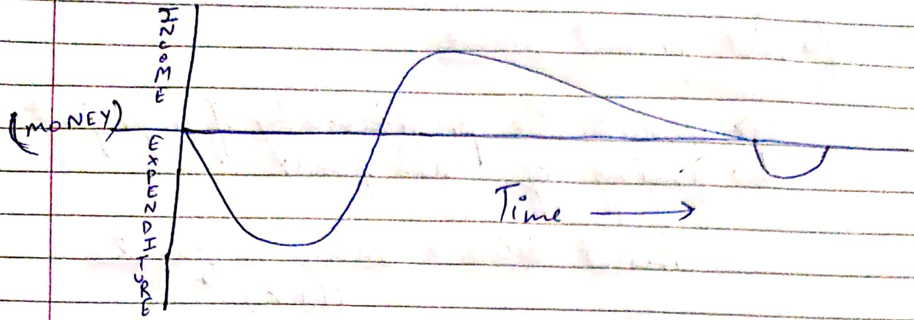
(1) Identifying all of the costs and benefits of carrying out the project & operating the delivered app.
↳ development, operating & benefits
(1) cost (2) cost (3) expected.

(2) Expressing these costs & benefits in common units (money).

* development cost (staff & team)
* setup cost (new hardware ~~&~~ or training)
* operational cost (operate the system after install)

Cash flow forecasting.

↳ when expenditure and income will take place.



A forecast is required to know when the cash (money) will come and go.

- Salary, system, setup, staff training, returns.
- Negative cash flow during development & positive cash flow during its operation.

Cost-benefit evaluation techniques

(1) **Net profit** = Total income − Total costs

(2) **Payback Period**
↳ project is broken even to give back the initial investment
↳ orgn wants to minimise the time that a project is 'in debt'
↳ it is like paying in EMI's.

(3) **Return on Investment (ROI)** $= \dfrac{\text{average annual profit}}{\text{total investment}} \times 100$

↳ no account of the timing of cash flow.

(4) Net present value

↳ account the profitability of a project
↳ timing of cash flow!

$$\text{Present value} = \frac{\text{value in year } t}{(1+r)^t}$$

r → discount rate
t → no. of years into future

Discount rate decision is main difficulty.

(5) Internal rate of return

↳ attempts to provide a profitibility measure as a %age return that is directly comparable with interest rate.
↳ cannot tell absolute size of return.

Risk Evaluation

(1) Risk identification & ranking
↳ give high, medium and low rank to any risk you identify & prioritize.

(2) Risk & net present value.
↳ if project is risky, use a higher disc. rate to calculate NPV.

(3) Cost-benefit analysis
↳ consider each possible outcome
↳ estimate probability of corresponding value of outcome

(4) Risk profile analysis
↳ using sensitivity analysis.
↳ check each parameter with project's profit & you'll know the sensitive profiles.

(5) Use decision trees

Select appropriate project

↳ objectivedriven or product-driven
↳ analyse other project characteristics.
↳ high level project risks
↳ requirements of implementation
↳ select general life cycle approach.

Choice of process models

↳ A no. of interrelated activities have to be undertaken to create a final product.
↳ These activities can be organised in diff. ways.

Structured methods

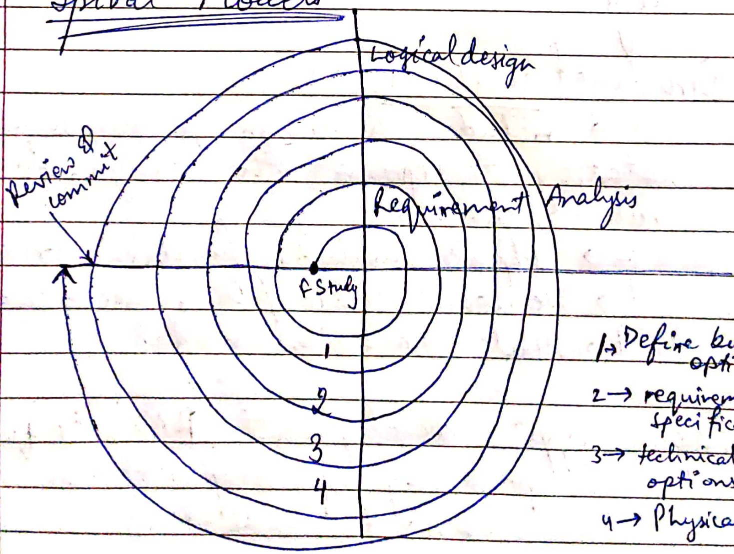
↳ include a design process model, notations to represent the design, formats, rules of design.
↳ object models, function oriented
↳ sequence, data flow, use-case
↳ state transition model.

★ Rapid app^n development (RAD) ⇒ quickly producing prototypes of the s/w for users to evaluate.

Water fall model

Feasibility Study
↓
User requirements
↓
Analysis
↓
System Design
↓
Program Design
↓
Coding
↓
Testing
↓
Operation

Spiral Models



1→ Define business option
2→ requirement specification
3→ technical system options
4→ Physical design

Prototyping delivery

1. Identify basic requirements
2. Develop initial prototype
3. Review
4. Revise & enhance.

Types → 1. Throwaway
2. Evolutionary
3. Incremental → many prototypes & joined them at the end
4. Extreme → depending fully on contract.

Incremental delivery breaks into small components which are implemented & delivered in a sequence.

Albercht func. point Analysis

↳ Top-down approach
↳ Based on 5 function points
1. External input types
2. External output types
3. External inquiry types
4. Logical internal file types
5. External interface file types

1. input transactions which update internal files.
2. output transn. where data is o/p to the user. (enquiry)
3. only for inquiry. Do not update internal files
4. data items accessed together (relational tables)
5. allow for o/p or input that may pass to and from comp. appns. (files shared).

Unit III

Activity planning

↳ pinpoint the activities required to achieve deliverables.

★ Objectives:

1. Feasibility assessment.
   ↳ time constraints & resource & staff.
2. Resource Allocation
   ↳ time scale & resource availability & desire to spend additionally.
3. Detailed costing
   ↳ cost & expenditure (detailed)
4. Motivation
   ↳ provide targets & monitor achievements
5. Coordination

Minimum time, acceptable cost, meeting a set target.

Project schedules

1. Ideal activity plan
2. Risk analysis
3. Resource allocation
4. Schedule production

(2–4) } alterations to ideal activity plan. Constraints are considered.

Project & activities

↳ Defining activities

- A project is composed of no. of interrelated activities.

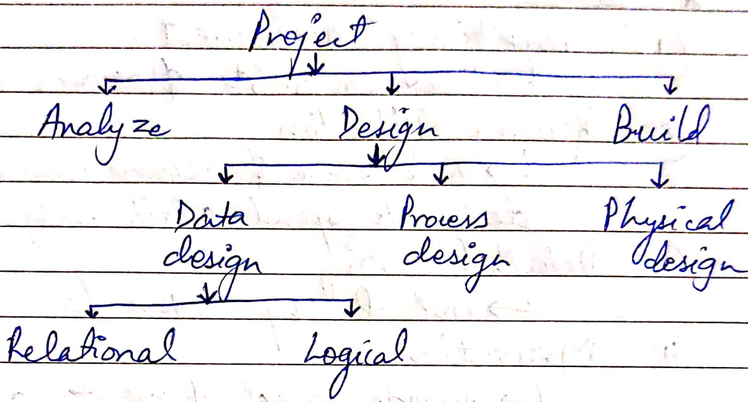
↳ Identifying activities

① The activity based approach.

↳ creating a list of all the activities that the project is thought to involve.

↳ subdivide the project

↳ create a WBS!



WBS

Work Breakdown Structure

↳ non-overlapping activities.

② Product-based approach

↳ consists of producing a Product Breakdown Structure and a product Flow Diagram.

↳ The products form a hierarchy.

↳ Progress reports can be added during the course of the project.

Date

- Project Products
  - System prod.
    - Overall specification
    - Integration test cases
    - Tested integrated s/w
  - Module prod.
    - module design docs
    - Module Code
  - Mngmt prod.
    - Progress report

③ Hybrid approach

↳ structuring of activities

↳ not all products after activity structuring will be final products. Some will be further refined in subsequent steps.

↳ Project

↳ Deliverables ⇒ s/w, manual etc.

↳ Components ⇒ modules, tests.

↳ Work-packages ⇒ data, tasks.

↳ Tasks ⇒ assign to each person.

Sequencing & Scheduling activities

↳ A bar chart is used which takes up account for the nature of the development process and the resources that are available.

↳ Sequence the tasks & schedule them.

↳ Sequence acc. to logical relationships.

↳ Schedule them considering resources, staff etc.

N/w planning models

↳ Time flows from left to right
↳ CPM (Critical Path Method)
↳ PERT (Program Evaluation Review Technique)

Activity on Arrow	Activity on Node
→ arrows are activities which join nodes & nodes tell start/completion of the activity.	→ activities are nodes & link b/w nodes represent precedence.

CPM (Critical Path Method)

↳ algo for scheduling a set of project activities

↳ Steps

1. Activity specification (WBS)
2. Activity sequence establishment
3. Network diagram (AOA or AON)
4. Estimates for each activity (COCOMO II FP's estimation)
5. Identification of Critical Path

↳ Earliest start time (ES)

The earliest time an activity can start once the previous dependent activities are over.

↳ Earliest finish time (EF)

ES + activity duration is EF.

↳ Latest finish time (LF)

The latest time activity can finish without delay

↳ Latest start time (LS) activity duration

6. Critical path diagram to show project progress.

PERT

Program Evaluation Review Technique.

↳ all about management probabilities.

3 estimates are required:

1. Most likely time (m)
   ↳ to take the task under normal circumstances.
2. Optimistic time (a)
   ↳ shortest time in which we could expect to complete the activity.
3. Pessimistic time (b)
   ↳ worst possible time

$$t_e = \left(\frac{a+4m+b}{6}\right)$$

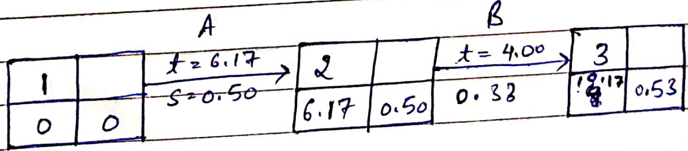
pg 177 – 180

t_e → expected time duration

Using expected durations (forward pass)

Event no.	Target date
Expected date	Standard deviation

Act.	a	m	b	t_e	s
A	5	6	8	6.17	0.50
B	3	4	5	4.00	0.33



Standard deviation

$$s = \frac{b-a}{6}$$

quick to exploit the opportunities to start subsequent activities earlier than scheduled. Critical chain management will be explored as a way of tackling this problem.

7.10 Applying the PERT technique

Using PERT to evaluate the effects of uncertainty

PERT (Program Evaluation and Review Technique) was published in the same year as CPM. Developed for the Fleet Ballistic Missiles Program, it is said to have saved considerable time in development of the Polaris missile.

PERT was developed to take account of the uncertainty surrounding estimates of task durations. It was developed in an environment of expensive, high-risk and state-of-the-art projects – not that dissimilar to many of today's large software projects.

The method is very similar to the CPM technique (indeed many practitioners use the terms PERT and CPM interchangeably) but, instead of using a single estimate for the duration of each task, PERT requires three estimates.

- *Most likely time*: the time we would expect the task to take under normal circumstances. We shall identify this by the letter *m*.
- *Optimistic time*: the shortest time in which we could expect to complete the activity, barring outright miracles. We shall use the letter *a* for this.
- *Pessimistic time*: the worst possible time, allowing for all reasonable eventualities but excluding 'acts of God and warfare' (as they say in most insurance exclusion clauses). We shall call this *b*.

PERT then combines these three estimates to form a single expected duration, t_e, using the formula

$$\boxed{t_e = \frac{a + 4m + b}{6}}$$

EXERCISE 7.5

Table 7.6 provides additional activity duration estimates for the network shown in Figure 6.29. There are new estimates for a and b and the original activity duration estimates have been used as the most likely times, m. Calculate the expected duration, t_e, for each activity.

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

TABLE 7.6 PERT activity time estimates

Using expected durations

The expected durations are used to carry out a forward pass through a network, using the same method as the CPM technique. In this case, however, the calculated event dates are not the earliest possible dates but the dates by which we expect to achieve those events.

EXERCISE 7.6

Before reading further, use your calculated expected activity durations to carry out a forward pass through the network (Figure 6.29) and verify that the project duration is 13.5 weeks. What does an expected duration of 13.5 weeks mean in terms of the completion date for the project?

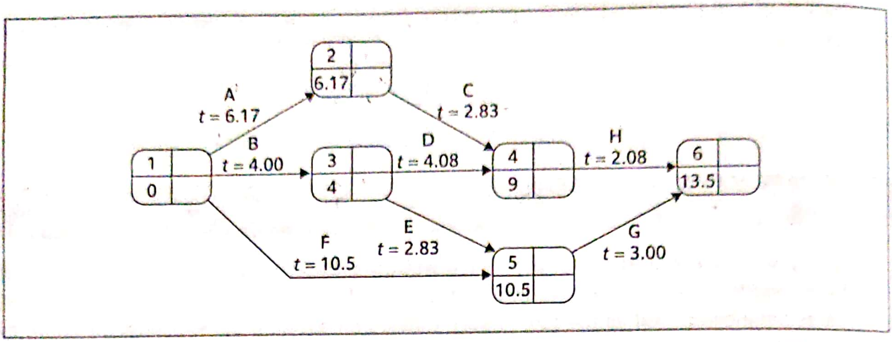
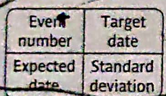


FIGURE 7.6 The PERT network after the forward pass

The PERT network illustrated in Figure 7.6 indicates that we expect the project to take 13.5 weeks. In Figure 7.6 we have used an activity-on-arrow network as this form of presentation makes it easier to separate visually the estimated activity data (expected durations and, later, their standard deviations) from the calculated data (expected completion dates and target completion dates). The method can, of course, be equally well supported by activity-on-node diagrams.



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.

Unlike the CPM approach, the PERT method does not indicate the earliest date by which we could complete the project but the expected (or most likely) date. An advantage of this approach is that it places an emphasis on the uncertainty of the real world. Rather than being tempted to say '*the completion date for the project is . . .*' we are led to say '*we expect to complete the project by . . .*'.

It also focuses attention on the uncertainty of the estimation of activity durations. Requesting three estimates for each activity emphasizes the fact that we are not certain what will happen – we are forced to take into account the fact that estimates are approximate.

Activity standard deviations

A quantitative measure of the degree of uncertainty of an activity duration estimate may be obtained by calculating the standard deviation s of an activity time, using the formula

$$s = \frac{b - a}{6}$$

This standard deviation formula is based on the rationale that there are approximately six standard deviations between the extreme tails of many statistical distributions.

The activity standard deviation is proportional to the difference between the optimistic and pessimistic estimates, and can be used as a ranking measure of the degree of uncertainty or risk for each activity. The activity expected durations and standard deviations for our sample project are shown in Table 7.7.

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

TABLE 7.7 Expected times and standard deviations

The likelihood of meeting targets

The main advantage of the PERT technique is that it provides a method for estimating the probability of meeting or missing target dates. There might be only a single target date – the project completion – but we might wish to set additional intermediate targets.

Suppose that we must complete the project within 15 weeks at the outside. We expect it will take 13.5 weeks but it could take more or, perhaps, less. In addition, suppose that activity C must be completed by week 10, as it is to be carried out by a member of staff who is scheduled to be working on another project, and that event 5 represents the delivery of intermediate products to the customer, which must take place by week 10. These three target dates are shown on the PERT network in Figure 7.7.

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

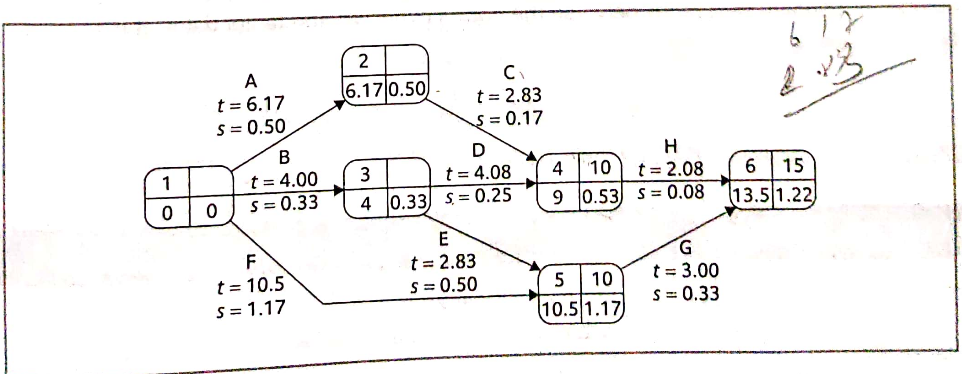


FIGURE 7.7 The PERT network with three target dates and calculated event standard deviations

6.17
2.83
9.00

- 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.

Calculating the standard deviation of each project event

The square of the standard deviation is known as the variance. Standard deviations may not be added together but variances may.

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. Exercise 7.7 illustrates the technique. One practical outcome of this is that the contingency time to be allocated to a sequence of activities as a whole would be less than the sum of the contingency allowances for each of the component activities. This has implications that can be exploited in critical chain project management, which are discussed in the next section.

EXERCISE 7.7

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.

Verify that the standard deviations for each of the other events in the project are as shown in Figure 7.7.

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

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

$$\frac{10 - 2.83}{0.17}$$

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

EXERCISE 7.8

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

Calculate the z values for the other events with target dates in the network shown in Figure 7.7.

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 7.8.

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

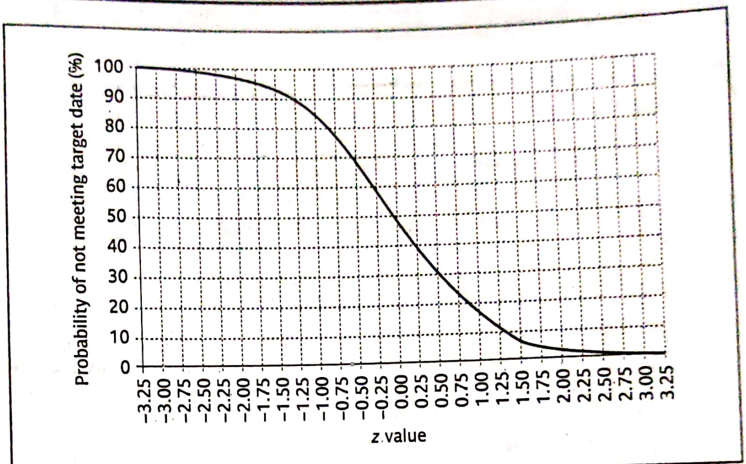


FIGURE 7.8 The probability of obtaining a value within *z* standard deviations of the mean for a normal distribution

EXERCISE 7.9

The *z* value for the project completion (event 6) is 1.23. Using Figure 7.8 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?

The advantages of PERT

We have seen that by requesting multi-valued activity duration estimates and calculating expected dates, PERT focuses attention on the uncertainty of forecasting. We can use the technique to calculate the standard deviation for each task and use this to rank them according to their degree of risk. Using this ranking, we can see, for example, that activity F is the one regarding which we have greatest uncertainty, whereas activity C should, in principle, give us relatively little cause for concern.

6.10 The forward pass

The forward pass is carried out to calculate the earliest dates on which each activity may be started and completed.

Where an actual start date is known, the calculations may be carried out using actual dates. Alternatively we can use day or week numbers and that is the approach we shall adopt here. By convention, dates indicate the end of a period and the project is therefore shown as starting at the end of week zero (or the beginning of week 1).

During the forward pass, earliest dates are recorded as they are calculated.

The forward pass and the calculation of earliest start dates are carried out according to the following reasoning.

- Activities A, B and F may start immediately, so the earliest date for their start is zero.
- Activity A will take 6 weeks, so the earliest it can finish is week 6.
- Activity B will take 4 weeks, so the earliest it can finish is week 4.
- Activity F will take 10 weeks, so the earliest it can finish is week 10.
- Activity C can start as soon as A has finished so its earliest start date is week 6. It will take 3 weeks so the earliest it can finish is week 9.
- Activities D and E can start as soon as B is complete so the earliest they can each start is week 4. Activity D, which will take 4 weeks, can therefore finish by week 8 and activity E, which will take 3 weeks, can therefore finish by week 7.
- Activity G cannot start until both E and F have been completed. It cannot therefore start until week 10 – the later of weeks 7 (for activity E) and 10 (for activity F). It takes 3 weeks and finishes in week 13.
- Similarly, Activity H cannot start until week 9 – the later of the two earliest finish dates for the preceding activities C and D.
- The project will be complete when both activities H and G have been completed. Thus the earliest project completion date will be the later of weeks 11 and 13 – that is, week 13.

The forward pass rule: the earliest start date for an activity is the earliest finish date for the preceding activity. Where there is more than one immediately preceding activity we take the *latest* of the *earliest finish dates* for those activities.

The results of the forward pass are shown in Figure 6.15.

6.11 The backward pass

The second stage in the analysis of a critical path network is to carry out a backward pass to calculate the latest date at which each activity may be started and finished without delaying the end date of the project. In calculating the latest dates, we assume that the latest finish date for the project is the same as the earliest finish date – that is, we wish to complete the project as early as possible.

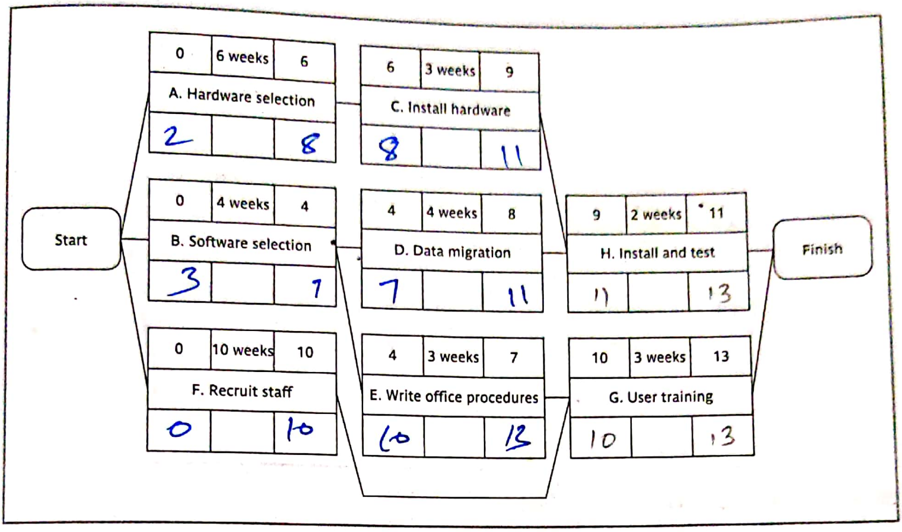


FIGURE 6.15 The network after the forward pass

The backward pass rule: the latest finish date for an activity is the latest start date for the activity that commences immediately that activity is complete. Where more than one activity can commence we take the earliest of the latest start dates for those activities.

Figure 6.16 illustrates our network after carrying out the backward pass. The latest activity dates are calculated as follows.

- The latest completion date for activities G and H is assumed to be week 13.
- Activity H must therefore start at week 11 at the latest (13 – 2) and the latest start date for activity G is week 10 (13 – 3).
- The latest completion date for activities C and D is the latest date at which activity H must start – that is, week 11. They therefore have latest start dates of week 8 (11 – 3) and week 7 (11 – 4) respectively.
- Activities E and F must be completed by week 10 so their earliest start dates are weeks 7 (10 – 3) and 0 (10 – 10) respectively.
- Activity B must be completed by week 7 (the latest start date for both activities D and E) so its latest start is week 3 (7 – 4).
- Activity A must be completed by week 8 (the latest start date for activity C) so its latest start is week 2 (8 – 6).
- The latest start date for the project start is the earliest of the latest start dates for activities A, B and F. This is week zero. This is, of course, not very surprising since it tells us that if the project does not start on time it won't finish on time.

?

Scanned with CamScanner

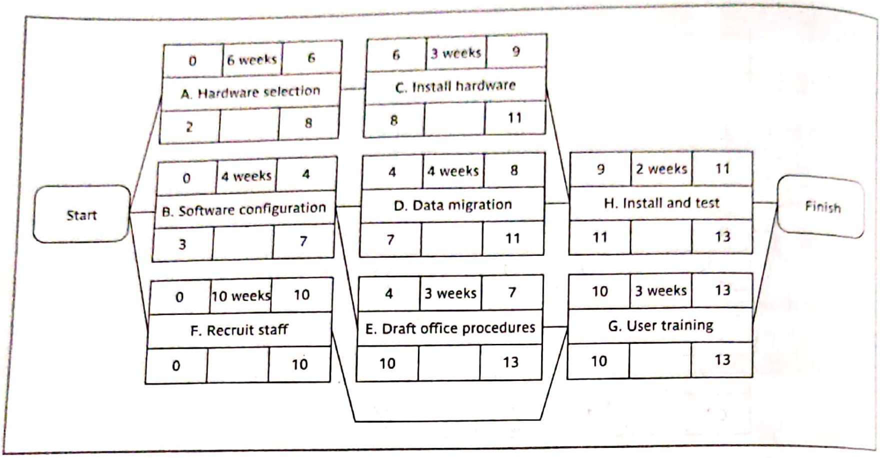


FIGURE 6.16 The network after the backward pass

6.12 Identifying the critical path

> The critical path is the longest path through the network.

> This float is also known as total float to distinguish it from other forms of float – see Section 6.13.

There will be at least one path through the network (that is, one set of successive activities) that defines the duration of the project. This is known as the *critical path*. Any delay to any activity on this critical path will delay the completion of the project.

The difference between an activity's earliest start date and its latest start date (or, equally, the difference between its earliest and latest finish dates) is known as the activity's *float* – it is a measure of how much the start or completion of an activity may be delayed without affecting the end date of the project. Any activity with a float of zero is critical in the sense that any delay in carrying out the activity will delay the completion date of the project as a whole. There will always be at least one path through the network joining those critical activities – this path is known as the critical path and is shown bold in Figure 6.17.

The significance of the critical path is two-fold.

- In managing the project, we must pay particular attention to monitoring activities on the critical path so that the effects of any delay or resource unavailability are detected and corrected at the earliest opportunity.
- In planning the project, it is the critical path that we must shorten if we are to reduce the overall duration of the project.

Figure 6.17 also shows the *activity span*. This is the difference between the earliest start date and the latest finish date and is a measure of the maximum time allowable for the activity. However, it is subject to the same conditions of interpretation as activity float, which is discussed in the next section.

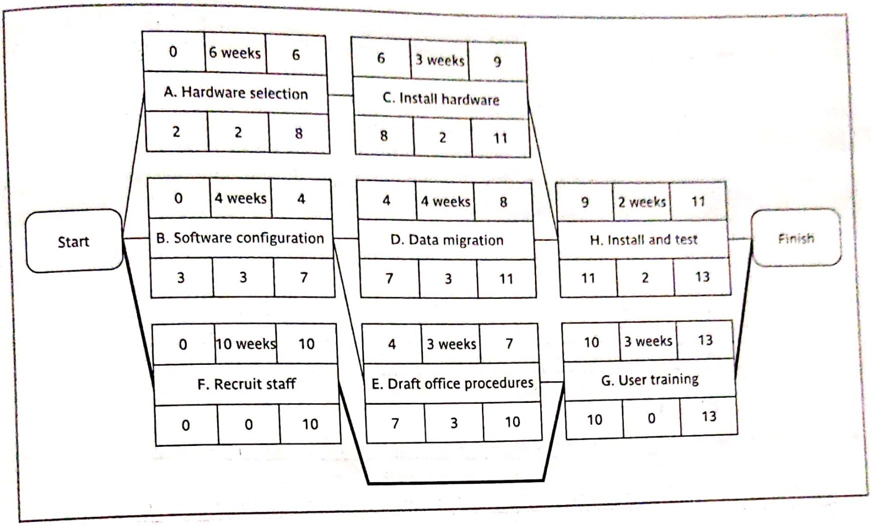


FIGURE 6.17 The critical path

EXERCISE 6.2

Refer back to Amanda's CPM network illustrated in Figure 6.7.

Using the activity durations given in Table 6.2, calculate the earliest completion date for the project and identify the critical path on your network.

Activity	Estimated duration (days)	Activity	Estimated duration (days)
Specify overall system	34	Design module C	4
Specify module A	20	Design module D	4
Specify module B	15	Code/test module A	30
Specify module C	25	Code/test module B	28
Specify module D	15	Code/test module C	15
Check specification	2	Code/test module D	25
Design module A	7	System integration	6
Design module B	6		

TABLE 6.2 Estimated activity durations for Amanda's network

est finish time (LF)

Precedence N/w

- A project n/w should have 1 start node
- Should have 1 end node
- A node has duration.
- Precedent are the immediate preceding activities.
- Time moves from left to right.
- Should not have loops.
- should not contain dangles.

Forward Pass

- calculate the earliest dates on which each activity may be started & completed.
- where an actual start date is known, calculations may be carried out using actual dates.

Rule
- The earliest start date for an activity is the earliest finish date for the preceding activity.
- Where there is more than one immediately preceding activity we take the latest of the earliest finish dates for those activities.

Backward Pass

- calculate the latest data at which each activity may be started and finished without delaying the end date of project.

↳ In calculating the latest dates, we assume that the latest finish date for the project is the same as the earliest final date that is, we wish to complete the project as early as possible.

~~Critical Path~~

↳ The backward pass rule: the latest finish date for an activity is the latest start date for the activity that commences immediately that activity is complete.

↳ Where more than one activity can commence we take the earliest of the latest start dates for those activities.

Critical path

↳ There will be at least one path through the n/w that defines the duration of the project.

↳ Longest path through the n/w.

↳ The different between an activity's earliest start date & its latest start date is activity's float - it is a measure of how much the start or completion of an activity may be delayed without affecting the end date of the project.

↳ Significance:

1. Managing the project,
2. ~~Plagi~~ Planning the projects.