20/9/18

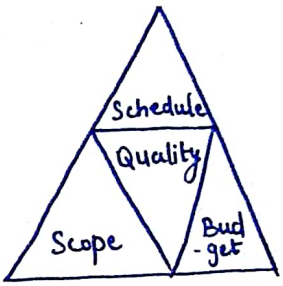
Software Project Management

Project : It is a well defined class which is collection of several operation done in order to achieve goal.

A software project consist of several stages such as req. analysis, development, testing, maintainance in a specified period of time.

SOFTWARE PROJECT CONSTRAINTS :

These affect our project → (SSQB)



1) **Scope :** It creates boundary of project. It defines what project is trying to achieve. It has all the outcomes & processes involved in producing those outcomes. So, basically it is the boundary of the project. Hence project is nothing more & nothing less than its scope.

2) **Schedule :** It is the time to complete the project the most frequent project errors are reflected due to missed deadlines in complete activities & late supporter, reports. So, proper scheduling is req. as per duration & req. processes of the project.

3) **Budget :** It is the cost approve for all the necessary expenses to deliver the project. Poorly executed budget can cause last min. rush to many req.

4) **Quality :** It is delivering the project outcome according to the stated or implied needs. In order to meet stake holders satisfaction.

(iv) Majority budget is divided into 3 parts :

Quality includes professional standards, law regulation & service level agreements.

OBJECTIVE OF PROJECT MANAGEMENT

① Scope Management : (i) Control of what is & what not is included in the project.

(ii) Defining clear boundary of the project.

(iii) Goals & objectives of the project.

(iv) Managing the changes in the objectives of the project.

(v) The activities inside scope management include assessment, impact analysis, authorisation, inclusion & exclusion of objects.

(vi) We create a work breakdown structure in scope management.

② Schedule Management : (i) Action required to ~~Enable~~ ensure the timely completion of the project.

(ii) Estimated time for each activity.

(iii) Relationship b/w each activity.

(iv) Procedure to control schedule changes if any.

(v) Monitoring schedule to meet the deadline.

(vi) It includes project baseline with actual dates & variance.

③ Budget Management : (i) It is req. to ensure that the project is completed within the approved budget.

(ii) It manages financial resources with proper assignment of expenses.

(iii) It has activities like proposal for req. monitoring for accounts & proj. failure estimation.

(iv) Majority budget is divided into 3 parts :
- Budget Monitoring
- Budget update
- Budget Control Plan

(4) Quality Management : (i) It is the process that ensure that proj. will satisfy the needs of supporters or users.

(ii) For quality check cond^n in which the proj. will be used are created so analysis its behaviour.

(iii) Quality assurance execution plan is executed by quality audits.

(iv) A quality control plan is developed when quality is judged on the output required as per the standard.

OTHER MANAGEMENT FACILITIES

(1) Team Management : (i) Identifying roles & responsibilities of an individual.

(ii) Training req. or resource allocation as per team.

(iii) Combination of technical management skilled people to balance the team.

(2) Information Management : (i) The flow of info. as per responsibilities & authorisation of the person.

(ii) Effective distribution of the information.

(iii) Analysis of the given info. to produce effective plans.

(3) Rise Management : (i) It defines the of the system.

It works in 3 phase:

(i) RISK IDENTIFICATION : Finding all the possible risk & there impact on the system.

(ii) RISK ANALYSIS : The imp. of risk identified & priorities of risk to be metigated (eliminated) out of the system.

(iii) RISK RESPONSE : Stratergy developed for the actions that is to be taken for risk elimination.

RESOURCES

22/9/18

The development of the software depends on resources, infrastructures & development effort as per the use characteristics availability & duration of use we divide the resources in three major parts.

People
Reuseable Software Resources
Hardware / Software Tools

→ PEOPLE

They are also termed as human resources. There planning began by evaluation of scope, requirement of skills, duration of the project & speciality required. The number of people req. for a software project can be determined only after an estimate of development efforts (person per month).

→ REUSABLE SOFTWARE RESOURCES

These are the components which can serve as the building block for any software & termed as the component.

Categories of Reusable Resources :

→ 1) Off the shelf Components : existing software that can be acquired from a third party or that has been developed internally for a past project & not in current use.

→ 2) Full experience Components : existing specification, design, code or test data for the past project that are similar to the software to be built for the current project.

→ 3) Partial experience Components : Members of the current software team have only limited experience in the appⁿ area of the components. These type of components have a fair degree of risk.

→ 4) New Components : Software Components that has to be built specially for the current need of the software project contains very high risk.

→ HARDWARE AND SOFTWARE TOOLS

Software tools consists of development environment, req. softwares, design tools & support tools for the project. Hardware requirements consist of control equipments, development machines, computer based system & special required hardware.

ESTIMATION

An effective software estimation provides the info. needed to design a workable software development plan. Its primary goal is success of the project. It provides imp. info. for project decisions, project performance, defining objects & future plans. The estimate should realistic & workable when problem appears.

Project Estimation Process

1. Establish estimate scope
2. Establish technical baseline, ground Rules, Assumptions
3. Collect Data
4. Size Software
5. Prepare base line estimates
6. Quantify Risk & Risk Analysis
7. Review, Verify, Validate
8. Generate a project plan
9. Document estimate
10. Track project through Development

1. EES : Participants understands the escpections of the project.
2. tOgRA : Identify functionality, assumptions, cost & reusability.
3. CD : Collecting info. from different sources.
4. SS : Effective estimation of size over functionality, reusability & expectation.
5. PBFE : Baselines are models for guessing the estimate.
6. QRRA : To understand events that can impact quality, money, & control.
7. RVV : Validation by someone who is not involved in estimation process & changing values as per ~~vic~~ review
8. GPP : It includes allocation, costing, scheduling as per task oriented structure.
9. DE : Everytime a task is completed, document of estimate should be generated.
10. TPD : Information of project completion compare to the original plan.

25/9/18

1) Size Estimation

Customer requirements & specification forms a base line for the size of the software. Project size can be determined through the past data from the earlier developed projects. This is called estimation by size analogy. The other approach is through product features & functionalities. The system is divided into several subsystems & depending on the size of each sub-system size of whole system is estimated. Software size

depends on 2 major factors :

① Line of Code ② Function Points

2) Effort Estimation

The estimation of the effort can be done in number of person per month. The best way to estimate it is based on historical data & development life cycle of similar applications. Different strategy should be implied, adopt estimation model for different algorithmic systems

3) Schedule Estimation

This will generally depend on human resources involved in the process. Schedule can be calculated from the following model.

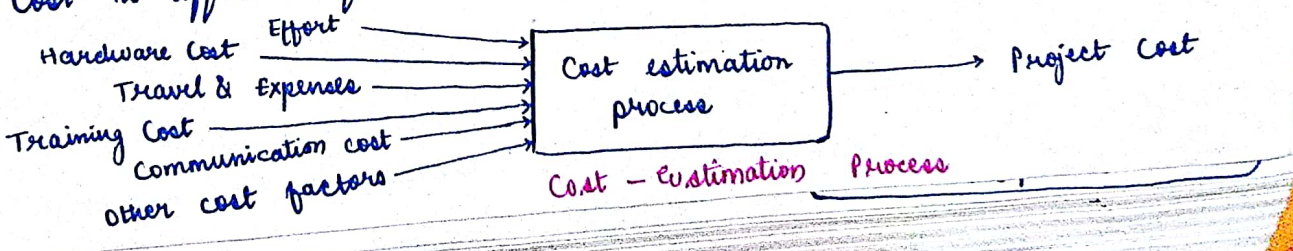
$$\text{Schedule in Calendar Month} = 3.0 * (\text{person - Month})^{1/3}$$

3.0 depends on security, increases with more security & vica versa as well as standard is 3.0.

The parameter 3.0 is a variable depends upon the situation & criticality of the system for the organisation.

4) Cost Estimation

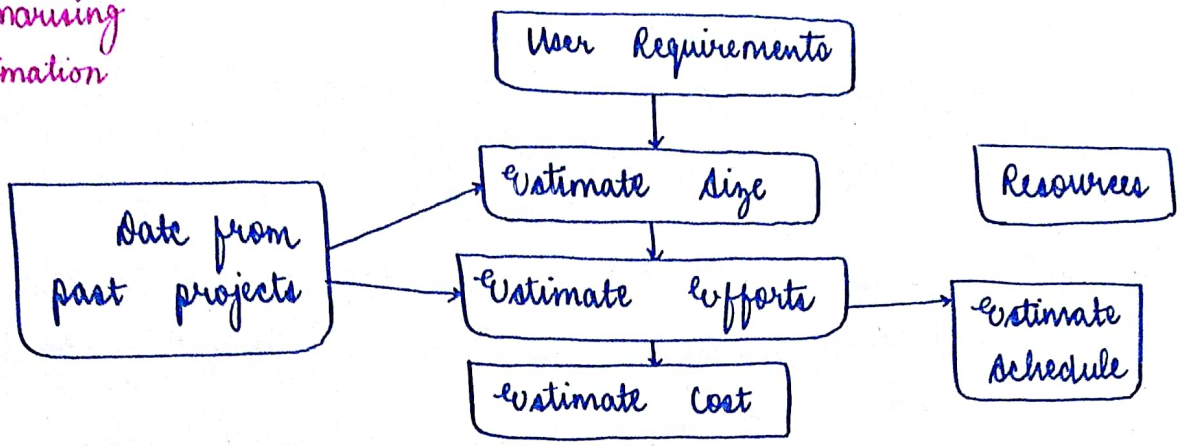
Cost is affected from multiple factors in the system.



Cost – Estimation Process

Factors using estimation are used in synchronisation to find the estimation of the process.

Summarising Estimation



The accuracy of estimation depends on :

1. Historical data used for the project estimation.
2. Accuracy of input data to various estimates.
3. Maturity of organisation's software development process.
4. Understanding of real time issues that can appear during project implimentation.
5. Linear Association between multiple factors of project development.

TECHNIQUES FOR ESTIMATION

1) Decomposition Technique :

This uses two methods of software sizing directly by estimating lines of code & indirectly by using function points. Meaningful line of code & function point estimation attempts to decompose the project along functional lines then estimate the size of

each sub-function individually. This approach is also termed as Problem-Based Estimation. Number of estimations are used to get a fair degree of accuracy. The decomposition technique uses following estimations for the project.

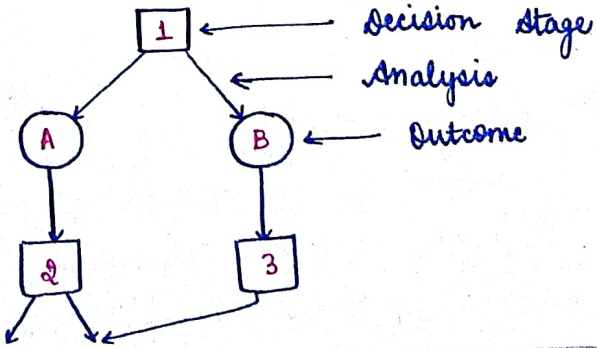
→ **Software Sizing** : This includes prediction of ability of the software, human effort, time, money & environment required to engineer the software.

→ **Problem Based Estimation** : Line of code & function point variables are used to estimate the size of the software.

→ **Software Equation** : Based on the historical data of other software efforts required by a person in number of months is calculated.

$$\underset{\text{Efforts person month}}{EPM} = \left(\underset{\text{Numbers of line in code}}{L} \times \underset{\text{Factor of special skills}}{Sk^{(1/3)}}\right) / \underset{\text{parameter of productivity}}{(PP)^3} \times \left(\underset{\text{duration}}{1/d^4}\right)$$

→ **Decision Tree** : It is helpful in analysing a software problem which have multiple decision stages.



→ Automated Estimation Tool : In this we use What-if analysis for different set of variables in a project to estimate cost & efforts.
eg: staffing & delivery rate.

SOFTWARE SIZE ESTIMATION

27/9/18

1) LINE OF CODE :

It is declaration actual code including logics & computation.

- What is not a line of code :

1) Blank Lines 2) Comments

→ Blank lines are only used for increasing the readability of the code.

→ Comments are used for better understanding & maintainance of the code.

- Why these are not included in line of code : Blank line & Comments does not contribute anything to the functionality of the code & they can be misused by the developers to give a false notion about productivity.

eg 1:

```
sum = 0
// this func" is to call sum
for (i=0 ; i <10 ; i++)
{
    sum = sum +i
}
printf (sum)
```

LOC = 6

: ADVANTAGES :

① It is very easy to calculate & count from the developed code.
② It is simple & does-not require much of an effort.

DISADVANTAGES OF LOC :

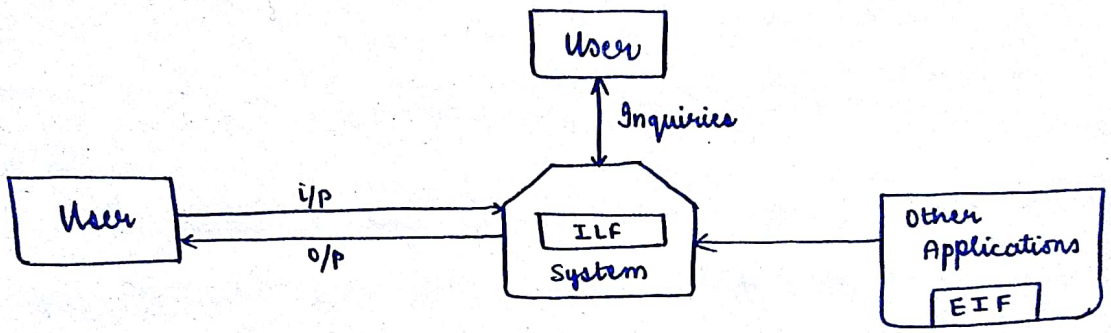
① LOC is language & technology dependent.

② It is very difficult to find out what makes an LOC when a single expression can be written into multiple ways.

2) FUNCTION POINTS :

→ It measures the functionality from users point of view. i.e. what user recieves from the software & what user requests from the software.

→ It focuses on what functionality is being delivered.



• A system has 5 types of functional units :

1) Internal logical ~~Logical~~ Files [ILF] :

→ It includes the control information or logically related data that is present within the system.

2) External Interface files [EIF] :

→ (Similar as above) The control data or other logical data that is referenced by the system but present in some other system.

3) External input :

→ The data or control input that comes from outside the system

4) External output :

→ The data that goes out of the system after generation of output.

5) External Inquiries :

→ It is the combination of input & output.

(1) & (2) are data function types

(3) & (4) & (5) are transaction function types : defines transaction of data from one to other.

Counting the points

① Each function point is ranked acc. to the complexity.

② There exists pre-defined weights for each function points.

* Low & Avg & High defines the need of a particular factor

Function Units	Low	Avg	High
EI	3	4	6
EO	4	5	7
EQ	3	4	6
ILF	7	10	15
EIF	5	7	10

- Weights are assigned as per the organisation & on the past project experience

Step ② Calculate unadjusted function points by multiplying each factor point by its corresponsing weights.

$$UFP = \sum_{i=1}^{5} \sum_{j=1}^{3} (3ij * wij)$$

Step ③ Calculate final function points

Final FP = UFP x CAF

CAF (Complexity Adjustment factor): To calculate this we use 14 accepts of complexity. Questions are answered on the scale of 0 - 5.

0 → No influence (not at all)
1 → Incidential (depends)
2 → Moderate (neither less nor more)
3 → Average (OK)
4 → Significant (visible)
5 → Essential (compulsory)

Eg 1: Is the project high performance critical

Eg 2: Is code reusable

Eg 3: Is internal processing complex

$$CAF = [0.65 + 0.01 * \Sigma fi]$$

ADVANTAGES OF FP:

① Size of the software calculated is independent of language & technology.

② Factor points are directly estimated from requirements before design & code so we get an estimate of software size even before major design or coding takes place. Any change in the requirements can be easily reflected in function points count.

③ It is useful for the users without technical expertise as it is based on external structure of the software to be developed.

Q1. Given are the factors to compute function points when all the complexity factors & weighing factors are average.

→ ① Calculating avg :

EI = 50
EO = 40
EQ = 35
ILF = 6
EIF = 4

$UFP = 50 \times 4 + 40 \times 5 + 35 \times 4 + 6 \times 10 + 4 \times 7 = 628$

② $CAF = [0.65 + 0.01 \times (14 \times \underset{\text{avg}}{3})] = 1.07$

③ Final P = $628 \times 1.07 = 671.96 \approx 672$

4/10/18

COCOMO Model

It was developed by Barry Bohem. It contains the heirarchy of software cost estimation model. It means it consists of three models.

① Basic Model ② Intermediate Model ③ Detailed Model

1) Basic Model : ① It estimates the cost of the software in a rough & quick manner.

② It is mostly used in cost estimation for small to medium sized softwares. It has three modes of development.

KLOC : Kilo lines of Code

Factors	Organic	Semi- Detached	Embeded
1) Size of project	2-50 KLOC	50 - 300 KLOC	300 and above KLOC
2) Team Size	Small	medium	large team
3) Developer Experience	Experienced	Avg experienced	Very little experienced
4) Environment	Familiar	Less similar	Significant envi. changes
5) Innovation Requirement	Little	Medium	Major
6) Deadline	Not tight	Not less nor high	Liniant

Examples :

Organic : Pay Roll Management Services (Providing salary to employe)

Semi-Detached : Travelling / Online Booking System

Embeded : Official Aircraft Management

Basic Equations is COCOMO Model :

1) Effort = $a\,(KLOC)^b$ person month

2) Development Time = $c\,(effort)^d$ months

3) Average size of staff = $\frac{\text{Effort}}{\text{development time}}$ person

4) Productivity = $\frac{KLOC}{effort}$ KLOC / pm

Modes	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi - Detached	3.0	1.12	2.5	0.35
Embeded	3.6	1.20	2.5	0.32

Learn these basic values

Q1.) Suppose a project was estimated to be 400 KLOC. Calculate effort & development time for all three modes of development.

→ Effort :

① For organic = $2.4\,(400)^{1.05}$ = 1295.311

② For Semi - De. = $3.0\,(400)^{1.12}$ = 2462.796

③ For embeded = $3.6\,(400)^{1.20}$ = 4772.813.

→ Development time : ① For organic = $2.5(1295.311)^{0.38}$ = 38 months

② For semi-detached : $2.5(2462.796)^{0.35} = 38$ months

③ For embedded : $2.5(4772.813)^{0.32} = 37$ months

Difference b/w basic & intermediate cocomo Model

1) Basic Model is very quick but inaccurate & phase insensitive
2) Intermediate Model includes a set of 15 additional predicate also known as cost drivers.
3) Basic Model lacks development environment into account during cost estimation.

Use of Cost Drivers : These are the facts that adjust nominial cost of project to actual product environment & hence increase the accuracy of estimate. 15 cost drivers are :

I.) ~~Personal~~ Product Attributes
a.) Required Software reliability (RELY)
b.) Database Size (DATA)
c.) Product Complexity (CPLX)

II.) Computer Attributes
a.) Execution time constraints (TIME)
b.) Main Storage Constraints (STOR)
c.) Virtual Machine Volatility (VIRT)
d.) Computer turn around time (TURN)

III.) Personal Attributes
a.) Analyst Capability (ACAP)
b.) Application Experience (AEXP)
c.) Programmer Capability (PCAP)
d.) Virtual Machine Experience (VEXP)
e.) Programming language Experience (LEXP)

IV.) Project Attributes
a.) Modern Prog. Practices (MODP)
b.) Use of Software Tools (TOOL)
c.) Required development schedule (SCED)

→ Each cost driver is rated for given project environment acc. to the impact which is divided in the categories : Very low, low, ~~Nominal~~ Normal, High, Very high, Extra high.

Cost Drivers	Very Low	Low	Nominal	High	Very High	Extra High
Product Attributes						
RELY	0.75	0.88	1.00	1.15	1.40	—
DATA	—	0.94	1.00	1.08	1.16	—
COMPLX	0.70	0.85	1.00	1.15	1.30	1.65
Computer Attributes						
TIME	—	—	1.00	1.11	1.30	1.66
STOR	—	—	1.00	1.06	1.21	1.65
VIR	—	0.87	1.00	1.15	1.30	—
TURN	—	0.87	1.00	1.07	1.15	—
Personal Attributes						
ACAP	1.46	1.19	1.00	0.86	0.71	—
AEXP	1.29	1.13	1.00	0.91	0.82	—
PCAP	1.42	1.17	1.00	0.86	0.70	—
VEXP	1.21	1.10	1.00	0.90	—	—
LEXP	1.14	1.07	1.00	0.95	—	—
Project Attributes						
MODP	1.24	1.10	1.00	0.91	0.82	—
TOOL	1.24	1.10	1.00	0.91	0.83	—
SCED	1.23	1.18	1.00	1.04	1.10	—

8/10/18

EAF : Effort Adjustment Factor

It is calculated by multiplying all the values that have been obtained after categorising each cost driver.

◇ Equations for 1st COCOMO Model :

Effort $a_i (KLoc)^{b_i} * EAF$

Development Time $c_i (effort)^{d_i}$

Mode	a_i	b_i	c_i	d_i
Organic	3.2	1.05	2.5	0.38
Semi Detached	3.0	1.12	2.5	0.35
Embeded	2.8	1.20	2.5	0.32

Q. A new project with estimated 400 KLOC has to be developed. A project manager has choice from 2 pools of developers.

(1) With very high capability with application & with little exp. is maintaining.

(2) Developers of low quality but with a lot of programming exp.

Find out which option will be a better choice in b/w these 2 pools:

→ AExp. → 0.82 LExp → 1.14 EAF = 0.82 X 1.14 = 0.9348

Organic : effort = $3.2 (400)^{1.05} * 0.9348 = 1614$

DT = $2.5 (1614)^{0.38} = 41.89$

Semi Detached : effort = $3.0 (400)^{1.12} * 0.9348 = 2302$

DT = $2.5 (2302)^{0.35} = 37.55$

Embeded : effort = $2.8 (400)^{1.20} * 0.9348 = 3470$

DT = $2.5 (3470)^{0.32} * = 33.95$

EAF = 1.13 x 0.95 = 1.0735

Organic : effort = $3.2 (400)^{1.05} * 1.0735 = 1854.022$

DT = $2.5 (1854.02)^{0.38} = 43.63$

Semi Detached : effort = $3.0 (400)^{1.12} * 1.0735 = 2643.811$

DT = $2.5 (2643.811)^{0.35} = 39.42$

Embedded : effort = $2.8 (400)^{1.08} * 1.0735 = 1941.71$

DT = $2.5 (1941.71)^{0.32} = 28.194$

DETAILED COCOMO MODEL

→ Detailed mode is face & cost drivers are calculated for each face.

→ It uses phase sensitive effort multiplier for each cost driver so detertime effort & time req. for each phase.

→ It has a module subsystem hierarchy to be followed.

→ The rating of cost drivers is done at that level only when cost driver is the ~~total~~ most sensitive variable.

⊙ When we need to find the reusability of the core we use adjustment factor:

Adjustment factor:

C = core I = Integration & testing

DD = Design Documentation

Adjustment Size = $6 \times \frac{A}{100}$ → actual size

$A = 0.4(DD) + 0.3(C) + 0.3(I)$

Table for different phases of a software in detailed cocomo Model:

Mode & Code Size	Plan & Requirement	System Design	Detailed Design	Code & text	Integration & text	~~Mode~~ Size
2	0.6	0.16	0.26	0.42	0.16	Org. Small
12	0.6	0.16	0.24	0.38	0.22	" Medium
22	0.7	0.17	0.25	0.33	0.25	Semi Detached Small
128	0.7	0.17	0.24	0.31	0.28	" Medium
126	0.8	0.18	0.25	0.26	0.31	Embeded Small
320	0.8	0.18	0.24	0.24	0.34	" Medium

1	0.10	0.19	0.24	0.39	0.18	
2	0.12	0.17	0.21	0.34	0.26	
3	0.20	0.26	0.21	0.27	0.26	I_p
4	0.22	0.27	0.19	0.25	0.29	
5	0.26	0.36	0.18	0.18	0.28	
6	0.40	0.38	0.16	0.16	0.30	

Effort = $\mu_p \times E$

D.T. = $I_e \times D$

Q1. Consider a project with following name compounds:

① Screen Edit ② CLI ③ File i/p & o/p ④ Cursor Movement

⑤ Screen Movement

The sizes for these are 4K, 2K, 1K, 3K & 1K LOC

Find the overall cost & schedule where the cost drivers are:

① High Software reliability ② Low Language Exp.

③ High Product Complexity ④ High analist capabilities.

Find the cost & scheduled estimate for difference.

→ Total = 4+2+1+3+1 = 11 K

1) 1.15 HSR 2) 1.07 LLE 3) 1.15 HPC 4) HAC 0.86

Effort = $\mu_p \times E$

EAF = $1.15 \times 1.07 \times 1.15 \times 0.86 = 1.21$

Effort = $3.2\,(11)^{1.05} \times 1.2169 = 48.29$

D.T. = $2.5\,(48.29)^{0.38} = 11$

P&R = $0.6 \times 48 = 2.6$ $0.10 \times 11 =$

SD = $0.16 \times 48 = 7.68$ $0.19 \times 11 =$

D.T. = $0.26 \times 48 =$ $0.24 \times 11 =$

C&T = $0.42 \times 48 =$ $0.39 \times 11 =$

I&T = $0.16 \times 48 =$ $0.18 \times 11 =$

RISK ANALYSIS

9/10/18

Risk are potential problems or uncertainity that might attack the successful complition of a software project.

* Risk Analysis is used to help software item & understand & manage the uncertainity during the development process work product is called risk mitigration, monitoring & management plan. There are 2 risk strategies:

① Reactive Strategy: In this the software team does nothing until the risk becomes real.

② Proactive Strategy: Risk management begins long before technical work starts. Risk are identified & managed as priority then a plan is created to avoid risk so that they can minimize the probability of occurance of risk.

Categories of Risk: It contains:

1) PROJECT RISK: ① Schedule & Cost increament in the project.
② Problems related to resources.

2) TECHNICAL RISK: ① Quality of the software.
② Problems related to design, implementation & maintainance of code.

3) BUSINESS RISK: ① Building a product that no one wants.
② Product no longer fits in the business strategy.
③ Product sales are not upto the mark.
④ Losing budget.

Steps for Risk Analysis

① RISK IDENTIFICATION: ① List the risk associated with specific project.

(2) Use risk item check list to identify the risk. eg of list: If we are staffing for a particular project the risk check list will contain: Q1. Are enough people available?
Q2. What is the working duration of the employee?
Q3. Is the staff properly trained?

Number of -ve response to the questions ∝ Degree to which the project staffing is at risk

(2) RISK PROJECTION: It is also termed as Risk Estimation. It rates risk in 2 ways:

- 1) Find probability of occurance of each risk.
- 2) Impact of the problems associated with the risk.

Process of Risk Estimation

(1) Prepare the risk table. Req. things here: [A] List all the risk identifier [B] Category of the risk identified in the second column.
Categories: 1) PS: Project Size 2) TE: Technical Risk
3) BU: Business Risk
[C] The third column will contain probability of occurance of risk.
[D] Impact of each risk. The impact depends on 4 components:
(1) Performance (2) Cost (3) Support (4) Schedule
After analysing the impact it can be divided into 4 categories as per the value calculated:
(1) Catastrophic: A very high risk (2) Critical
(3) Marginal (4) Negligible

Risk Analysis

Example : Risks | Category | Probability | Impact | RMMM

Example left to be done by own

Step 1: After drawing a table next the table is sorted by high probability and high impact. The project manager will define a cut of line impling only risk above this line will be managed.

RMMM contains M: Mitigation, M: ~~Migrating~~ Monitoring, M: Management Plan

Step 2: It includes assessment of risk impact. (1) estimate cost for each risk in the table. Example: In risk identification process we found out that only 70% of the software components could be used & rest has to be developed in a customized manner. Risk probability of this problem is 80% likely. Risk Impact is 60 reusable components were planned but only 70% of these could be used. The average component LOC is 100 & each LOC cost is 14. Find the overall cost?

→ 1) 60 × 30% to be developed = $60 \times \frac{30}{100} = 18$

Cost = $18 \times 100 \times 14 = 25200$ $

- Calculate the risk exposure?

→ Probability of occurance of risk × Cost of risk

= 0.80 × 25200 = 20160

- Calculate the total exposure of all the risk? The total risk exposure provides a mean for major adjusting in the final cost estimation of a project. If RE > 50% cost of product Then the project has to be reevaluated for development.

③ RISK REFINEMENT: During the early project stage the risk may be stated quite generally. As time passes the risk could go higher or lower as per the refined properties. The process of risk refinement includes: ① Risk in condition: It includes transition & consequences.

Example: If we are working with 40% reusable components & 30% are left. This can be refined as?

→ SUBCONDITION 1: Reusable components were designed by third party with no knowledge of internal design.

→ SUBCONDITION 2: Reusable components were developed in a language that is not supported by the environment.

10/10/18

RMMM

All the risk analysis is done to develop a strategy to deal with them. An effective strategy consist of:

① RISK MITIGATION: (avoidance)

Risk starts with assuming the risk that could appear in the system.

Eg: We assume that the probability of risk of having more no. of users,

then expected is likely to have a probability of 70%. Risk mitigation steps could be:
(i) Meeting with a staff.
(ii) Reorganization of the team.
(iii) Documentation of the standard of code will re-evaluated.

(2) RISK MONITORING:

These are the activities comments for risk appearing while project is progressing - manager needs to monitor whether the risk is becoming less or more likely.

Eg: The risk that needs to be monitored if we are need to deal with staffing problem.

→ Attitude of team members.
→ Degree of Compatibility of co-workers.
→ Monitoring has to be done regularly to reach the conclusion of risk category.

(3) RISK MANAGEMENT:

If migration is failed then management has to be taken place, backup resources should be analysed & knowledge of previous conditions must be taken into account.

⇒ RMMM prepares a document (Risk info. sheet) which consist of:

Risk	Date	Problem	Impact
Description:			
Refinement:			
Cond^n 1:			
Cond^n 2:			
Monitoring (monitoring)			Current Status
Management Plan			

Scheduling

It is required to manage no. of smaller task that occur to establish a larger goal.

Objective of project manager in scheduling

1. Define all task in project
2. Build activity network
3. Build a timeline to detect plan & progress of each task.
4. Track the progress to control & monitor the project.

Scheduling of the projects can be done in 2 ways :

1) End date of release is fixed & establish distributive is req. in persived time.
2) Rough bounds are decided for project delivering. Project will be completed in 7-8 months.

Basic Principles of project Scheduling

1) **Compartment** : No of management activities, actions & task should be divided into compartments.

2) **Interdependency** : Interdependency each compartment must be determined.

3) **Time Allocation** : Each task must be allocated some no. of work units & calculated completion date.

4) Effort Validation : It defines no. of people on the team & work required from that.

5) Define Responsibilities : Every task must be allocated to specific team members.

6) Define Outcomes : Every task must have defined & desired outcome as ~~this~~ task completion.

7) Define Milestone : Milestone defines the completion of phases of the project.

TASK NETWORK

(i) A task network is a work break down structure for a project.

(ii) It depends on project type & degree of influential factors from which team plan should be created.

(iii) Quality & completion should be taken as the highest priority.

TYPES OF SOFTWARE PROJECTS

1) Concept Development Project:

It consist of new business concept or application of new technology.

TASK NETWORK

11/10/18

It is also called as activity network. It is the graphical representation of activity flow of a project.

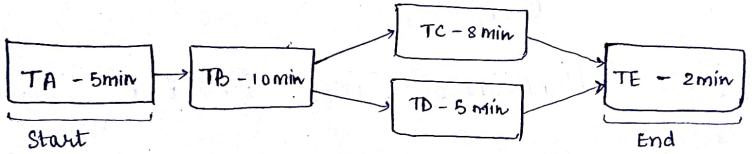
It depicts task length, sequence, concurrency & dependency.

It points out intertask dependency to help to monitor continuous process of project.

- For task scheduling we use two different methods:
  (i) Critical Path Method (CPM) (ii) Program Evaluation & Review Technique (PERT)

CPM ⟹ It is a single path leading from start to finish in a task network. It provides sequence of task in which they should be completed.

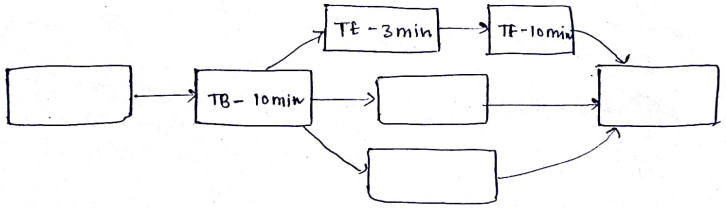
Eg 1 ⟹



TA(1) = 5
TB(2) = 10
TC(3) = 8
TE(4) = 2
25 min

:: Critical Path ::

Eg 2 ⟹



TE(3) = 13

PERT ⟹ It is a probabilistic method because it uses a weighted average of 3 estimates of a duration of an activity. It is prefered when there is high degree of uncertainity regarding the duration of task. Formula for finding the scheduling time:

(Effective Time) $T = (O + 4M + P) \div 6$

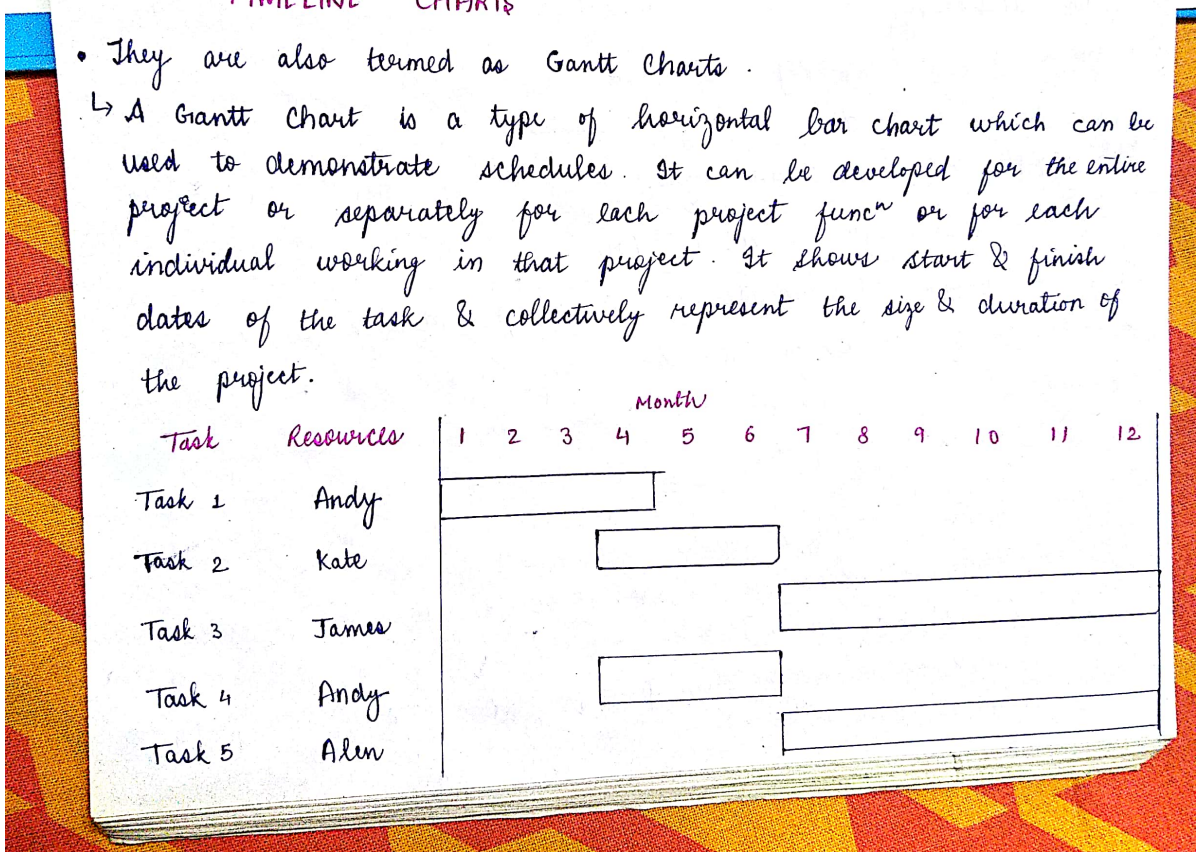
O: Most optimistic estimate
M: Most likely estimate
P: pessimist estimate

Difference B/w PERT & CPM:

1)	PERT is used for critical time dependent projects as it was originally designed for projects of US Navy.	It is mostly used for control & maintainance as it was developed for chemical plants.
2)	PERT is applied for research & development (R&T) projects.	CPM is used for construction projects.
3)	Majorly utilised when the projects are complex & result is uncertain.	Best utilized for repeatative & non-complex projects.

TIMELINE CHARTS

- They are also termed as Gantt Charts.
  - ↳ A Gantt Chart is a type of horizontal bar chart which can be used to demonstrate schedules. It can be developed for the entire project or separately for each project func^{n} or for each individual working in that project. It shows start & finish dates of the task & collectively represent the size & duration of the project.



i) Interdependency
ii) Concurrency
iii) Total duration

Task 1 = 4 days
Task 2 = 3
Task 3 = 6
Task 4 = 4
Task 5 = 6

23 days : Individually

But actually 12 days are req. to complete tasks.
∴ Gantt charts used.

ADVANTAGES:

1) Ease of Communication
2) Efficiency : Most optimal way to predict the work load.
3) Managibility : Entire timeline could be visualised for diff. components that enables for a better management & decision tradeoff

DISADVANTAGES:

1) Time Consuming
2) Using Ability / Usability : It is no longer useful for a task which is having multiple subtask.
3) Blindness towards occured fault.

EARNED VALUE ANALYSIS

It is a method for measuring project performance. It indicates how much of the budget should have been spent to finish the amount of work done so far. Factors used for measuring earned value :

① The budgeted cost of work schedule (BCWS): the amount that is planned to be spent on a task between start & end date.
② Actual cost of work performed (ACWP): the amount that is actually spent on the task from starting date to ending date.
③ Budgeted cost of work performed (BCWP): The % age of the budget that should have been spent for given % age of work performed on a task. Calculations done :

1) Cost Variance — CV = BCWP - ACWP

2) Schedule Variance — SV = BCWP - BCWS

3) Cost performance Index — CPI = BCWP / ACWP

4) Schedule performance Index — SPI = BCWP / BCWS