PROJECT MANAGEMENT (SPM)

Software Project Mngment aims at delivering the software successfully to the end users. The developer has not only to develop the softwares but the successful completion (within time constraints & within budget) is also very necessary. So a management team is also needed to keep a check on various conditions & issues.

Responsibility OF S/w PROJECT MANAGER

Various job responsibilities and key skills required for a Project Manager are listed below:-

(1) Software Planning: The project manager has to arrange the effective development plans to complete the project within cost and time constraints. He also has to estimate the no. of persons required for the s/w development. The effort estimate with respect to the time required is measured in the units - PERSON MONTH.

(2) Software Project Monitoring and controlling: This is needed to ensure the timely & successful development of software. It helps in checking whether all the activities are proceeding according to the planned criteria or not? The project manager can make changes in the schedule whenever he feels a need to do so.

- The spiral model involves Risk manage. It detects all the risks involved at any point of time and resolve them.

Different types of Risks involved are :-

1) Change in technology.
2) Cost and time schedule are overrun.
3) Labour Turnover - the manpower can go on strike due to some reason or may move to some other company. Then project is disturbed & needs to be started again with new developer
4) Unexperienced staff. - They can make many mistakes leading cost increment and customer dissatisfaction.
5) Software must be backward compatible, mean it should support all old as well as new platforms.

SKILLS NEEDED: ⇒ A theoretical knowledge of diff proj mgnt techniques like cost estimation, risk mgnt etc.

- Decision making skills.
- Communication skills
- Knowledge of technology.
- Management of Risks involved: He must be able to handle all types of situations.
- He must be expert in this domain.
- Leadership Qualities.
- Cost and Time scheduling
- Tracking & controlling: He must be aware of the various methods of making observations, the approach followed for monitoring the software development activities.

Planning and Resourcing a Project

PROJECT PLANNING

Project planning is the initial step towards the development of a software. In this, all the factors & aspects involved in the development process are estimated & development approach is planned.

Steps involved in project planning are:-

1) Estimation: The pre-judgement of the following factors -

Estimating some attributes of the project

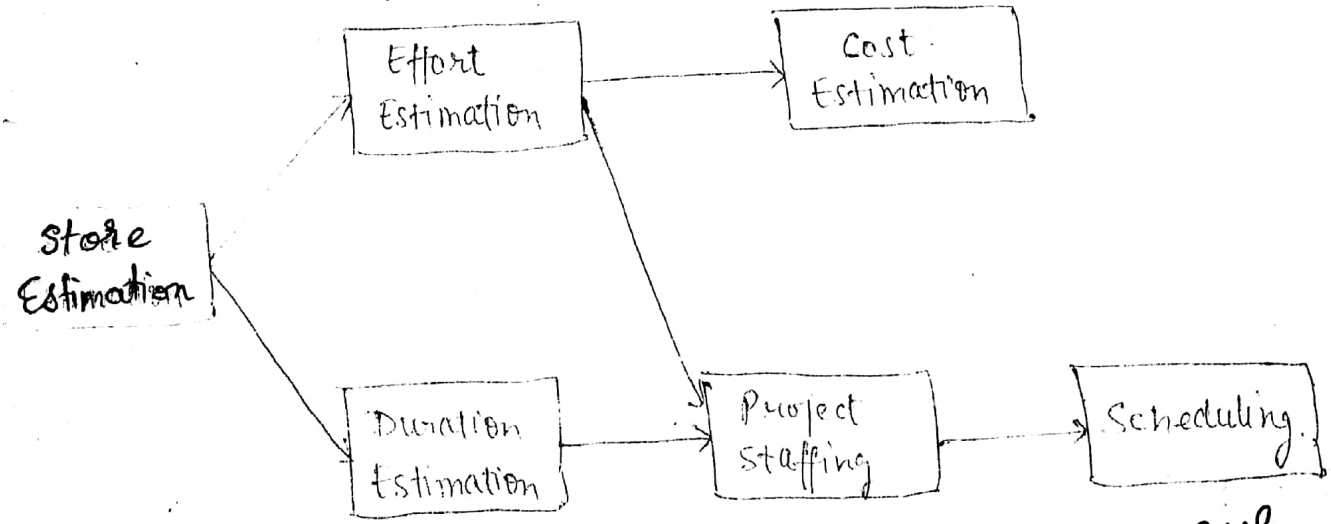
- Cost – How much financial assets are needed?
- Time – What will be the project development duration?
- Effort – How much manpower is required?

(2) Scheduling :- This is a pre-planned chart that divides various resources - manpower & time - proportionally, so as to carry out all the development activities in time.

(3) Staffing :- It involves the recruitment of the capable accurate developers needed for software coding and then organizing the hired staff. Organization of staff includes the assigning of task to each team member in the development team.

(4) Risk Management :- It involves the identification of the risks involved during project development & the removal of those risks. We will study the risk analysis in the next chapters.

(5) Miscellaneous Plans :- The other plans like Quality Assurance Plan, Testing plan etc. are to be developed under miscellaneous category to facilitate the manager.



Precedence ordering among planning activities.

The SPMP Document.

Once planning is complete, Project managers document their plans in a SPMP document.

The "Software Project Management Plan" document is an ~~legal~~ official document where the project managers list their planned strategy. This document is then reffered by project managers for controlling and tracking activities.

The SPMP document include following items :-

1) Introduction : Objective, functions, technique to be used.
2) Project estimation : Cost, Time, Effort
3) Schedule : WBS (work breakdown structure), Activity n/w, gantt chart, PERT chart.
4) Project Resources : manpower, Hardware, Software.
5) Staff Organization ⇒
6) Risk Management plans → Risk analysis, Risk identification.
7) Project tracking and monitoring activities.
8) Miscellaneous plans. (Cost+time), System Testing plan, Delivery, Installation and Maintenance plans.

(METRICS) FOR 'PROJECT SIZE' ESTIMATION.

10/1/13

The project size estimation is the measure of the complexity in terms of effort and time required to develop the software. These metrics are widely used to estimate size: LOC, FPM, Feature Point.

1) LOC - LINES OF CODE.

This is the simplest method to estimate project size. LOC count the number of lines / instructions in the source code. The comment lines are ignored while counting the source code lines.

Disadvantages of LOC

DIFFERENT CODING STYLE : The coding style of different

1. programmers vary from each other, so the same program can be written in multiple ways and therefore the number of lines vary from each other. So we can't define a specific complexity level for a software.

FOCUS ON CODING : This method only concentrate on the coding phase of the development process. But the overall complexity not only depends on the coding, but also on planning, design and management phases.

2. LOC do not consider the quality and efficiency of the code written by the programmers.

3. It is very difficult to estimate the exact number of instructions in the final program beforehand.

4. Two programs with the same LOC count may not have the same complexity. Larger code size does not mean that code will be of better quality i.e. a smaller program can be more efficient than the program with larger LOC count.

Thus LOC is not an efficient or standardized method.

FUNCTION POINT METRIC (Module wise Counting)

FPM overcome the drawbacks of the LOC method as it do not depend on the number of lines to calculate the size of progeam.

FPM estimate the size of a project by the total number of (functions or features) supported by it. The program that support large number of features will be of larger size than a s/w that include lesser no. of functions.

Function point is calculated in three steps:

(i) UFP (Unadjusted Function Point)

It is the weighted sum of 5 characteristics -

UFP = (number of inputs) × 4 + (number of outputs) × 5 + (number of inquiries) × 4 + (number of files) × 10 + (number of interfaces) × 10.

(ii) TCF (Technical complexity factor)

TCF = 0.65 + 0.01 * DI

where DI is Degree of Influence that varies from 0° to 84°.

(iii) FP (Function Point)

Function Point is yeilded by multiplying UFP to TCF.

FP = UFP * TCF

one of the imp advantages of using the function Point metric is that it can be easily estimate the size of a s/w product directly from the problem specification. This is in contrast to the LOC metric, where the size can be accurately determined only after the product has fully been developed.

Di

FPM incorporates an extra parameter into algorithm complexity

FEATURE POINT METRIC

Feature point metric assumes that all the functions do not have same complexity, therefore the overall complexity can not be calculated only by counting the number of functions.

Feature Point metric thus include the complexity of each function separately. It aims at calculating the algorithm complexity to determine the overall size and effort.

21/11/13

(FP)

PROJECT ESTIMATION TECHNIQUES.

Project estimation is necessary to have an estimate of the parameters like - cost, time, effort & project size. These estimates form the basis for resource planning and scheduling and also helps in informing the cost of the project to the customer.

We can broadly classify these methods as follows:-

Empirical Estimation Techniques.

↳ Expert Judgment Method

↳ Delphi Cost Estimation

Heuristic Estimation Technique:

↳ COCOMO model

1- Basic COCOMO Model

Analytical

Analytical Estimation Technique.
↳ Halstead software.

Now we will have a detailed view of the Project Estimation Techniques mentioned above.

EMPIRICAL ESTIMATION TECHNIQUES

Empirical estimation is based on making an educated guess of the parameters involved by using the experienced developers. In this technique we mostly take help of the developers who have an experience of many years and have developed similar projects beforehand.

This technique also use user manuals and guides of the similar type of projects. The judgment is made on the basis of common sense and prior experience of the developers.

This is further classified into two types —

1) Expert Judgment Method.

In this only a single expert developers is the incharge and take all the decisions after analyzing the problem. But this technique subjects to certain drawbacks :-

- It is subject to human errors as only a single person take all the decisions.
- Single expert may overlook some features i.e. may miss some functions while planning for s/w development.

- That judge may not have come across that particular type of project, so he may not be an expert in that field. In such circumstances he may take wrong decisions.

DELPHI COST ESTIMATION.

Though its name is cost estimation technique, but it has to measure the duration & efforts required for development for estimating the cost of the project.

This is different from expert judgment because it involves MULTIPLE EXPERTS that are authorized to make collective decision.

PROCESS:

- Panel consists of multiple experts that analyze the problem thoroughly.
- The SRS (software Requirement Specification) document is distributed among the panel and all the experts write their ideas ~~on the document.~~ and ~~estimates~~ estimates to meet the requirement written on SRS document.
- At last the coordinator recheck all the ideas and makes a summary of selected estimates to be offered throughout development process.

HEURISTIC TECHNIQUES

It is an estimation techniques based on the mathematical expressions and modelling of the relationships among different project parameters. A general mathematical expression is derived for the project parameters and is then replaced by the basic parameters of the project for which the estimation is to be made.

Heuristic models are of two types :-

- Single variable estimation model

This uses the previously estimated parameters of the similar projects to estimate the desired characteristic of new project. These previously estimated parameters are the independant characteristics and the new parameter is dependant on these previous characteristics.

Estimated Parameter = $c_1 * e^{d_1}$

where -

e is the independant, previously estimated parameter

Estimated Parameter is the dependant (new) parameter.

c_1 and d_1 are the constants whose values is determined from the data collected for previous projects.

c_1, d_1 etc are where collected data from

Multivariable cost estimation model

These models suffer more than one basic (independant) parameters for estimating the new parameters. These are found to be more accurate than the previous model.

Estimated parameter = $c_1 \times ep_1^{d_1}$ + $c_2 \times ep_2^{d_2}$ + ...

where -

ep_1, ep_2 are the already estimated parameters from past projects.

c_1, c_2, d_1, d_2 are the constants.

COCOMO model.

22/1/13

COnstructive COst Estimation MOdel was proposed by Boehm in 1981. COCOMO model mainly aims at classifying the software into 3 broad categories :

1) Organic - The softwares under this category are mainly application programs & are usually developed by a small team of developers. The team members are very experienced for developing organic softwares.

2) Semi-detached - They involves the development of utility softwares. The team members are a combination of experienced and inexperienced staff.

Utility s/w
also written software
is a type of system software
used to support the comp infrastructure,
eg Anti-virus
backup s/w, Debuggers

3) Embedded - It uses very complex hardware and new/modren techniques for developing system softwares. It involves the fresh and inexperience staff to work with the new technology.

The complexity level in these 3 types of project is given as follows :-

Organic : Semidetached : Embedded

1 : 3 : 9.

COCOMO model can also be further classified into 3 categories -

BASIC COCOMO Model.

This gives an approximate estimate of the project parameters by using following formulas :-

KLoC

$$\text{Effort} = a_1 \times (\text{KLOC})^{a_2} \text{ PM}$$

$$T_{dev} = b_1 \times (\text{Effort})^{b_2} \text{ months.}$$

where

- KLOC is estimated size of project expressed in Kilo lines of code.
- a_1, a_2, b_1, b_2 are constants whose values are derived from previous projects.
- T_{dev} is development time in months.
- Effort is the total manpower required with respect to time. It is measured in the units PERSON - MONTH (PM).

To estimate the effort and time for BASIC COCOMO in the above formulas are calculated as follows:-

EFFORT

Organic: Effort = $2.4 (KLOC)^{1.05}$ PM

Semidetached: Effort = $3.0 (KLOC)^{1.12}$ PM

Embedded: Effort = $3.6 (KLOC)^{1.20}$ PM

DEVELOPMENT TIME

Organic: Time = $2.5 (effort)^{0.38}$ months

Semidetached: Time = $2.5 (effort)^{0.35}$ months

Embedded: Time = $2.5 (effort)^{0.32}$ months.

The above values are research based values.

INTERMEDIATE COCOMO (Q&A) 23/01/13

The BASIC cocomo only consider two factors - time and effort for the software development but actually many more factors are involved for estimating the accurate size of the project.

The intermediate cocomo model recognizes this drawback & fix it by adding many more features/ parameters for estimating the project size. Some of the parameters are:-

Product: complexity of s/w & reliability needs are considered.

Computer: h/w requirements, memory space needed or processing speed are checked.

Personal: skills, experience or capability of developers.

Development Requirement: Use of ....

COCOMO

COMPLETE COCOMO (7b)

The complete cocomo states that not only the software are of different types, but also the different modules / parts / components of a software can be of the different types with different complexities.

The Basic and intermediate COCOMO consider the whole model as a single entity, but complete COCOMO divide the whole software system into subsystems & determine their individual estimate of effort & time. These individual estimates are then combined to calculate overall cost of the project.

The major components or subsystems into which the main system is divided are :-

1) Database part : where all the backups of system & user records is kept.
2) GUI (graphical user interface) part : used for making interactions with the end user. The main purpose is to accept the input & deliver the output to the user.
3) Communication part : It is the internal interface b/w the hardware and software parts of the system.

Therefore, complete cocomo subdivides system into -

DB part + GUI part + Interface
(semi-detached) (organic) (embedded).

HALSTEAD TECHNIQUE (2p) 24.1.13

It is an analytical technique to measure size, development effort & cost of s/w project. It used conventional expression & formulas to calculate certain parameters. Various terms & expressions used are listed below :-

Length and Vocabulary.

The length of a program is total no. of operators and operands used in the program.

$$\therefore \quad N = N_1 + N_2$$

where
- N – total length
- N_1 – no. of operators
- N_2 – no. of operands

The program vocabulary is the no. of unique operators and operands used in the program.

$$\text{Vocabulary} \quad \eta = \eta_1 + \eta_2$$

Program Volume

Volume is the minimum number of bits needed to encode the program.

$$V = N \log_2 \eta$$

where η (eta) is the program vocabulary.

Potential Minimum Volume.

It is defined as the volume of shortest program in which a problem can be loaded.

- Program Level

The program level is given by :

$$L = V^*/V.$$

- Effort & Time

Effort required to develop a software is calculated by dividing volume by level.

$$E = V/L$$

OR $$E = \frac{V^2}{V^*} \quad (\text{since } L = V^*/V)$$

- Estimated length.

$$\eta_1 * \log \eta_1 + \eta_2 * \log \eta_2$$

PROJECT SCHEDULING

This is the most important planning activity that perform the following tasks:-

1) Identify the activities needed for project completion.
2) Break down of large activities

ex:→ Design phase

↙ GUI design ↘ DB design.

3) To find dependancy of activities on each other.
4) Establishing most likely estimates.
5) Allocate the resources to activities.
6) Plan for starting & end date for activities.
7) Determine the critical path. A critical path is the chain of activities that determines the duration of the project.

PROJECT SCHEDULING METHODS

1.) Work Breakdown Structure (WBS)

In this method, we break the larger set of activities into a smaller set. The major problem is made the root of the tree and its subdivisions form the child / leaf nodes. (The root of the tree is labelled by the problem name).

Eg :→

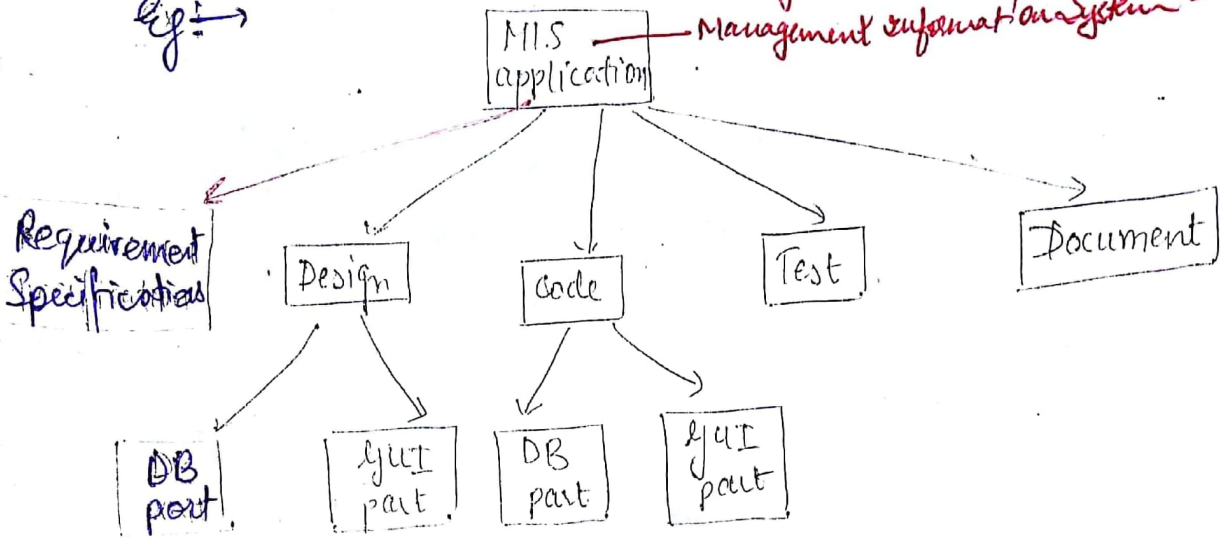
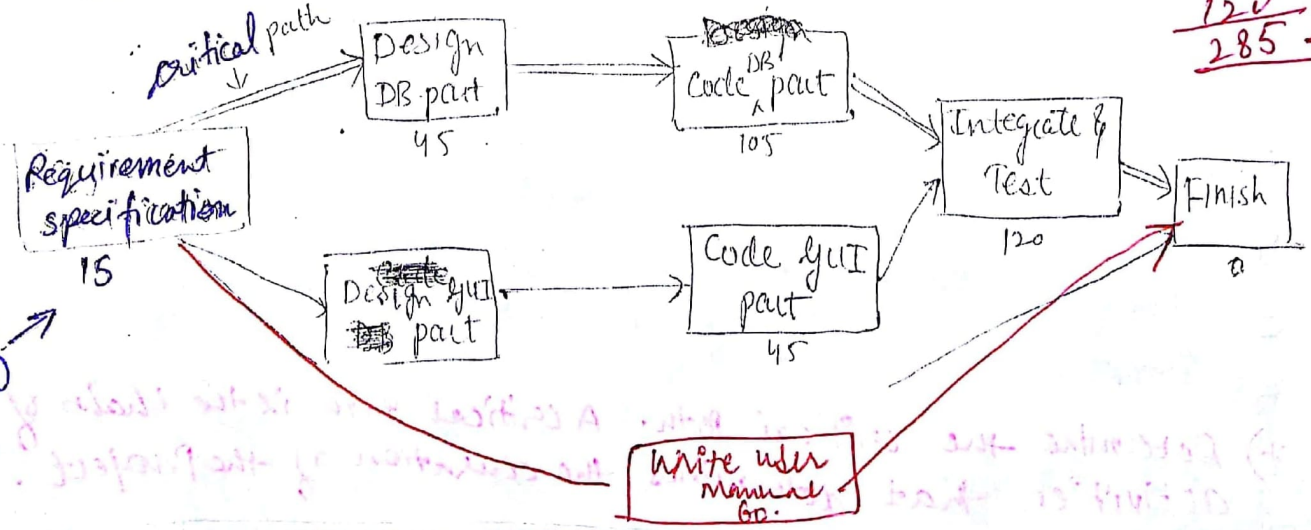


Fig: WBS for a MIS application.

(2) Activity Network / Critical path method

In this method, along with the activities performed the number of days for each activity is specified. Also the critical path is also defined but the start and end date for an activity is not specified.

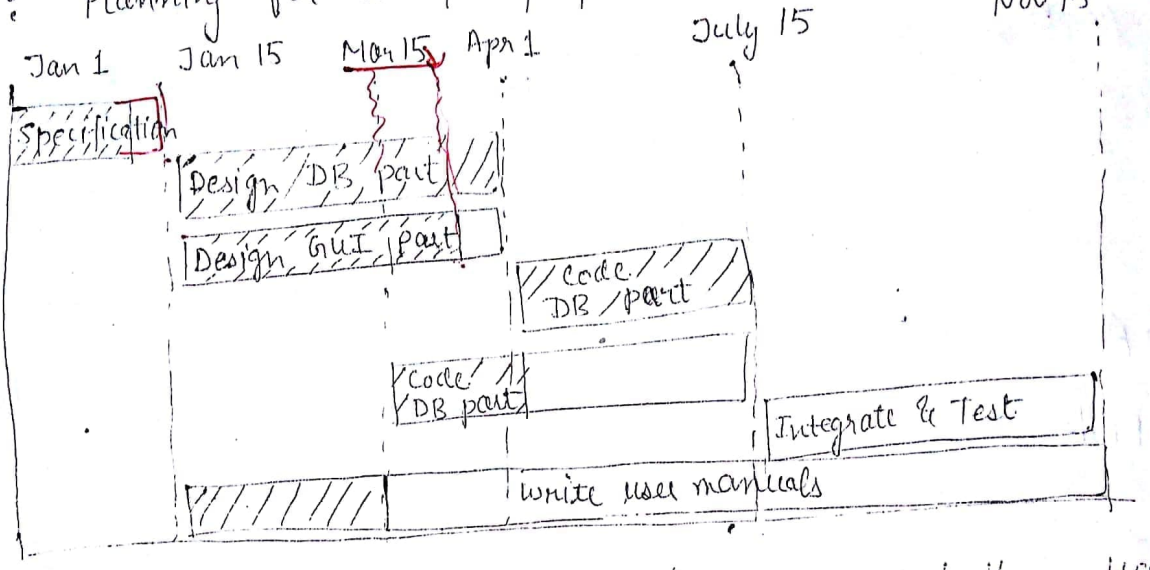


Monday - 1:10-2:50 — DBMS IT-2.
Wed -> 2:3:40 — OS
— DBMS Lab.

(3) GANTT CHARTS

Gantt chart represent the data / activities performed in the form of bars. They are used to allocate the resources to activities, i.e. they are useful for resource planning.

Ex: Planning for a s/w project:

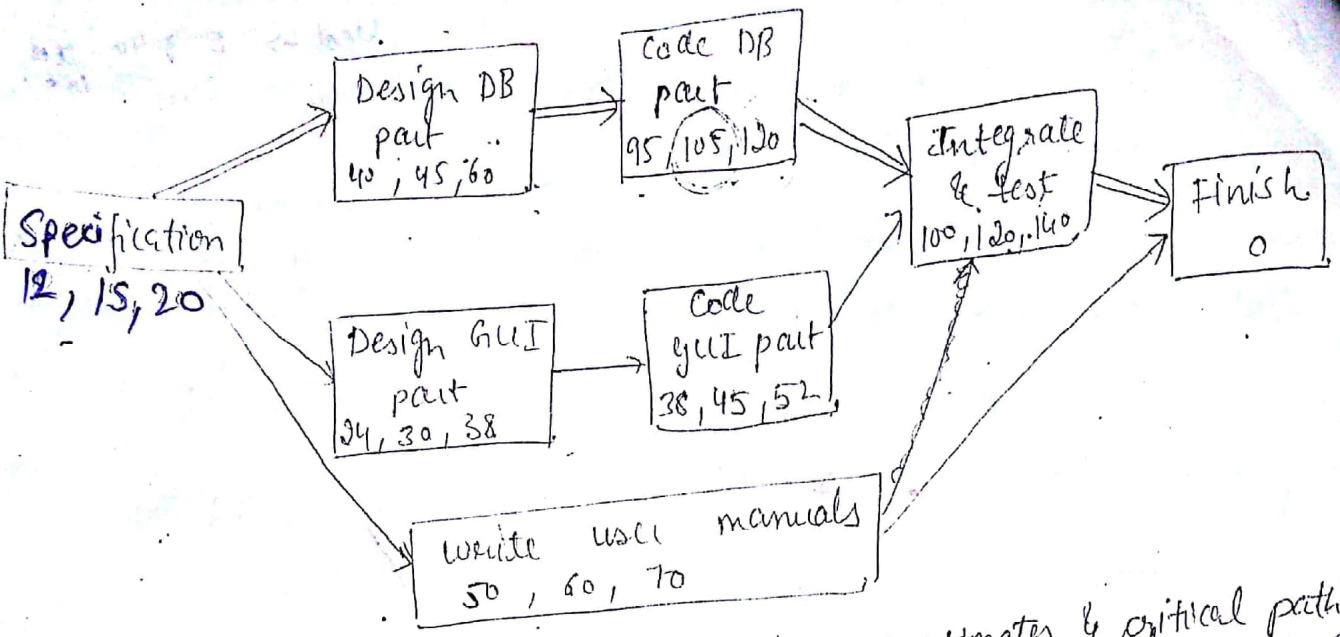


Each bar consist of shaded or white part or both. Here the shaded part represents the estimated time to be taken to perform each activity. The white portion is called the slack time, which means the maximum time allowed by which the activity must complete.

(4) PERT CHART

PERT chart is Project Evaluation and Review Technique similar to the Activity n/w., But the difference lies in the point that it (PERT chart) not only define a single estimate. but it gives the minimum, likely and maximum no. of days in which an activity can be completed. Critical path is also specified.

Pessimistic, likely optimistic.



1/10/12

Fig: PERT chart (with 3 estimates & critical path)

ORGANIZATION & TEAM STRUCTURE.

ORGANIZATION :- It refers to the arrangement and division of the staffs according to the approach followed. It can be done in two ways :-

a) Project organization.

In this the developers start from the very begining with the project & work in all the phases till the completion of project. The ~~division of staffs is project oriented~~ the development staffs are divided based on the project for which they work.

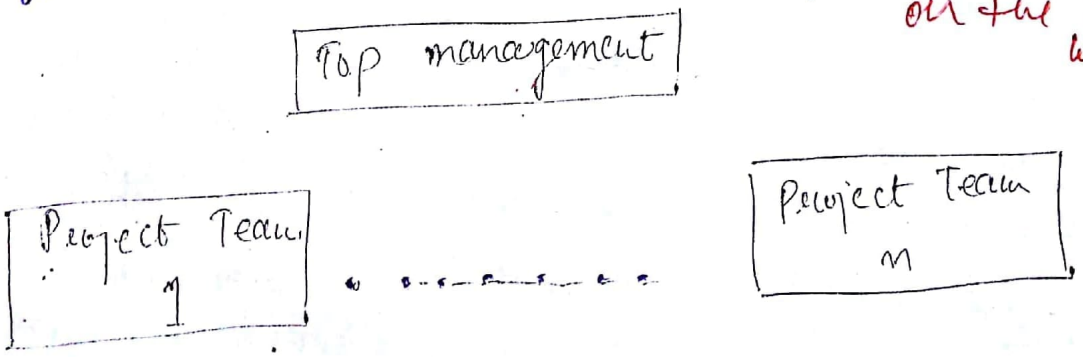
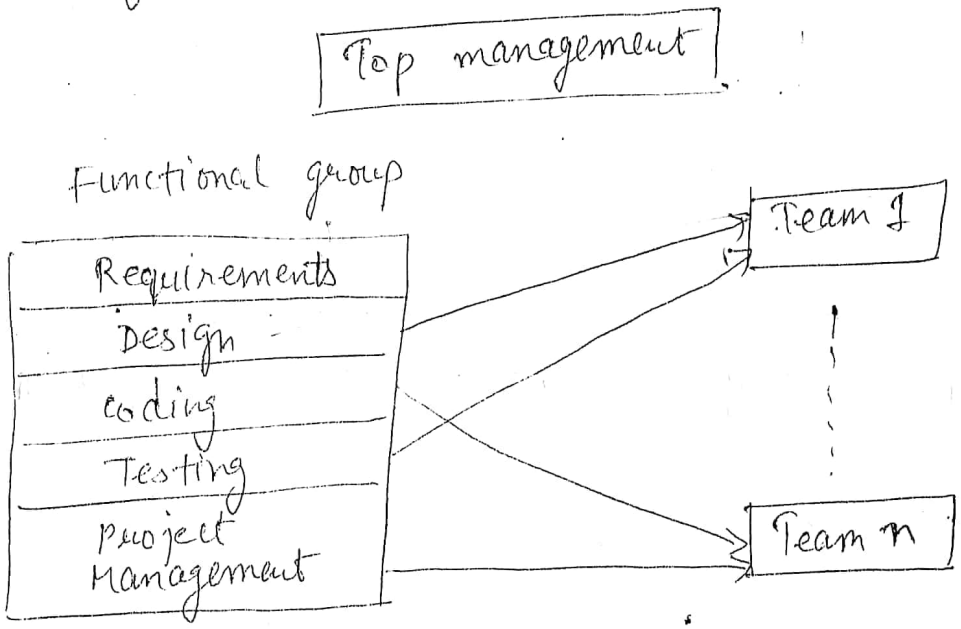


Fig: Project organization.

b) Functional organization.

In this, the skills and functional area of the team members is considered. The developers or team members are assigned the task according to their interested feilds. Some members may carry out specification phase, the other may go for designing and some other may be assigned the task of coding. The staff division is function-oriented.



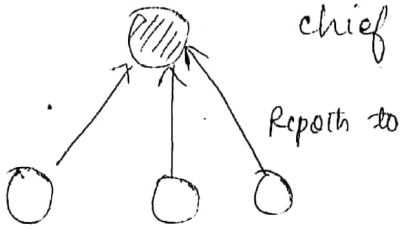
c) TEAM STRUCTURE :- Team structure refers to the internal interaction and organization of the development team. A good team structure helps in :-

- Better performance
- Generation of new ideas.
- Prosperous work culture.

The development teams can be structured in three ways :-

Chief Programmer Team Structure.

In this method there is only one team manager and software engg. directly works under him. The chief programmer have the authority to take all the decisions.

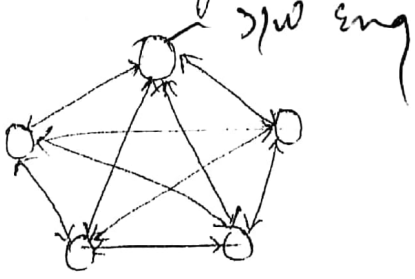


DRAWBACKS :

A single manager have a narrow view of the problem. The single person may have no knowledge or experience related to the kind of s/w to be developed. Morale of the s/w engg. may go down in such strict environment.

Democratic Team Structure

This method overcomes the orthodox nature of previous method. There is no chief manager & no supervision is there. S/w developers have a wider view and take their independant decisions through discussion. No stress or burden is there. It is a two-way process.



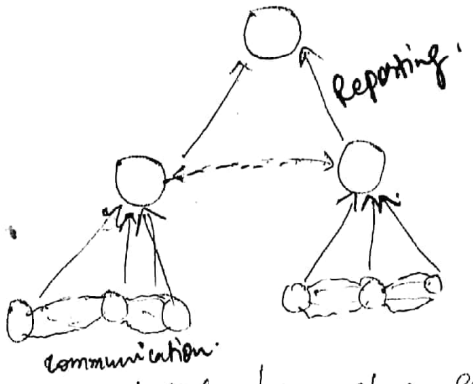
Democratic Team Structure.

DISADVANTAGES :-

1. More conflicts arise.

Mixed Team Structure:

It is very important to have proper management for maintaining rules, regulations & discipline in a team. The Mixed structure approach have heads at different levels but no supervision is done directly by the manager.



The work completed by s/w engg. is submitted to senior engg. & then to the project manger.

ADVANTAGES:

- It is a two way process.
- Discipline & rules are maintained properly.
- Developers at same level are in direct contact.
- There is not stress on developers.

RISK MANAGEMENT

20/01/13

A risk is an expected unfavourable condition that can occur during any development phase. A risk can hamper the successful completion of a project, so it is very necessary to identify the possible risks and avoid / contain those risks ... the... become a problem.

Risk Mgmt consists of 3 essential activities.

RISK IDENTIFICATION

This is a process of listing all the possible events that can harm the development process. We can classify the ~~we~~ risks into various categories —

- Budgetary Risk / Financial Risk → exceeding of cost than the defined limit.
- Project Risk → include the overrun of time, shortage of resources and labour turnover (strike of manpower/developers due to some reason).
- Business Risk → This means success or failure of the end product. It might happen that a developer spends a lot of cost to develop such a product for which there is no demand or we can say that it do not fullfill the needs of any user. Then there is a big loss in such condition.
- Technical Risk → These risks are incurred due to the change in technology, problem in implementation, interfacing or testing of the product. The developing team may also not have sufficient knowledge about the new technology.

2. RISK ASSESSMENT

This is the process of evaluating the risk according to the level of damage they cause. We find the priority of every risk based on two factors -

r → probability of a risk coming true.
s → sverity / damage caused by the risk.

priority (P) = r * s.

3. RISK CONTAINMENT

After the consequencies / results of a risk are assessed, we have to make plans to face and control that risk.
Main strategies to control / contain the risks are -

- Avoid the risk.

We can convince the customer to change their requirement in which there is a risk, we can suggest some other alternative or substitute for that activity.

- Transfer the risk.

we can transfer the risk to a third party by buying insurance for our project.

- Risk Reduction.

This involves the pre-planning for the risk. ... must make different

strategies to handle the risks.

For eg:— If there is a risk of labour turnover, we can recruit new people as a pool of extra developers in case if needed.

OR

If there is a technical risk in project, we must try the peices of new technology beforehand to make our developers efficient and competetive in the market.

Now after making certain strategies, we have to choose the best one from all suggestions. For this we have to (1) calculate the total cost of reducing the risk and (2) the cost of damage caused by that risk.

Now we can compute the RISK LEVERAGE

$$\text{Risk leverage} = \frac{\text{Risk exposure before reduction} - \text{Risk exp. after reduction}}{\text{Cost of Reduction.}}$$

17

Software configuration

SOFTWARE CONFIGURATION MANAGEMET.

Software configuration refers to the documents, manuals and SRS document needed in all the phases design coding, testing, maintenance) by the developers to cross check their product with the needs of the customer.