Empirical estimation technique:

Empirical estimation technique are based on the data taken from the previous project and some based on guesses and assumptions.

There are many empirical estimation technique but most popular are

Expert Judgement Technique

Delphi Cost Technique

Expert judgement technique:

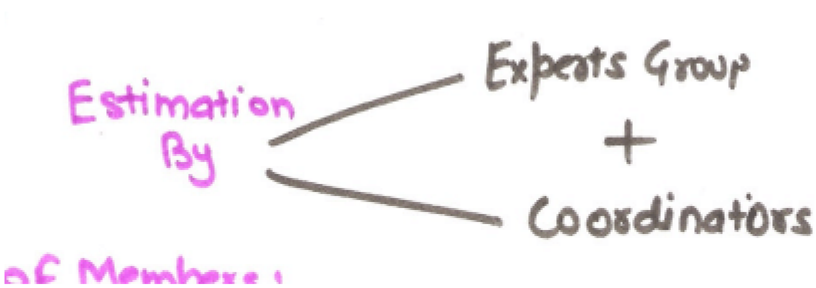
An expert makes an educated guess of the problem size after analyzing the problem thoroughly. Expert estimate the cost of different components that is modules and sub modules of the system.

Disadvantages:

Human error, considering not all factors and aspects of the project, individual bias, more chances of failure.

Estimation by group of experts minimises factors such as individual oversight, lack of familiarity with a particular aspect of a project, personal bias and desired to win a contract through overly optimistic estimates.

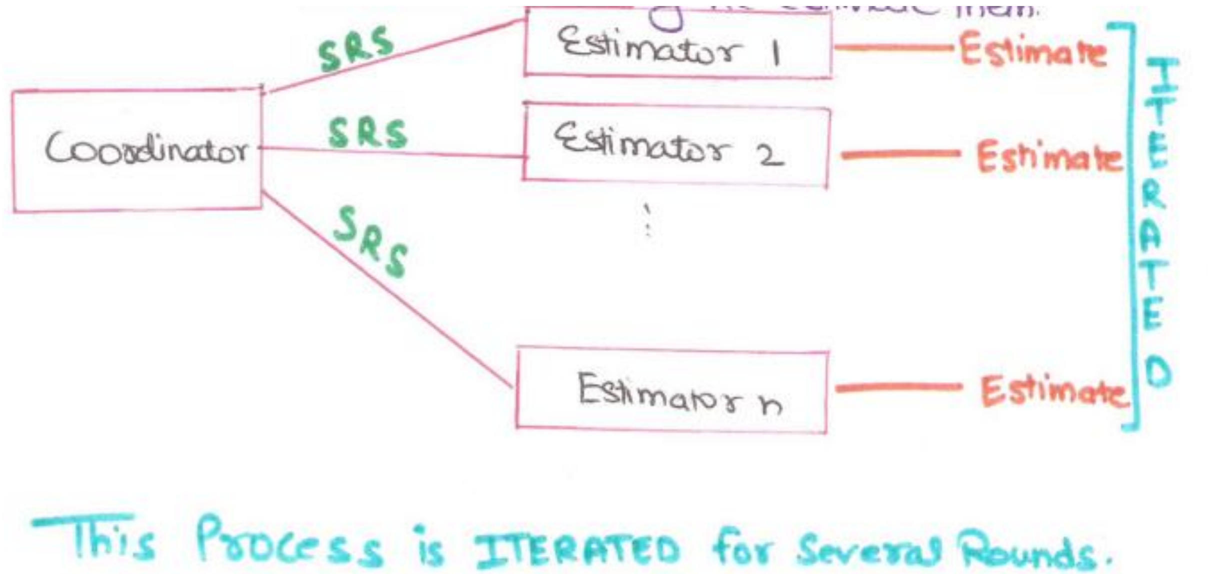
Delphi cost estimation:



Role of Members: Coordinator provide a copy of Software Requirement Specification(SRS) document and a form of recording it cost estimate to each estimator.

Estimator: It complete their individual estimate anomalously and submit to the coordinator with mentioning, if any, unusual characteristics of product which has influenced his estimation.

The coordinator and distribute the summary of the response to all estimator and they re-estimate them.



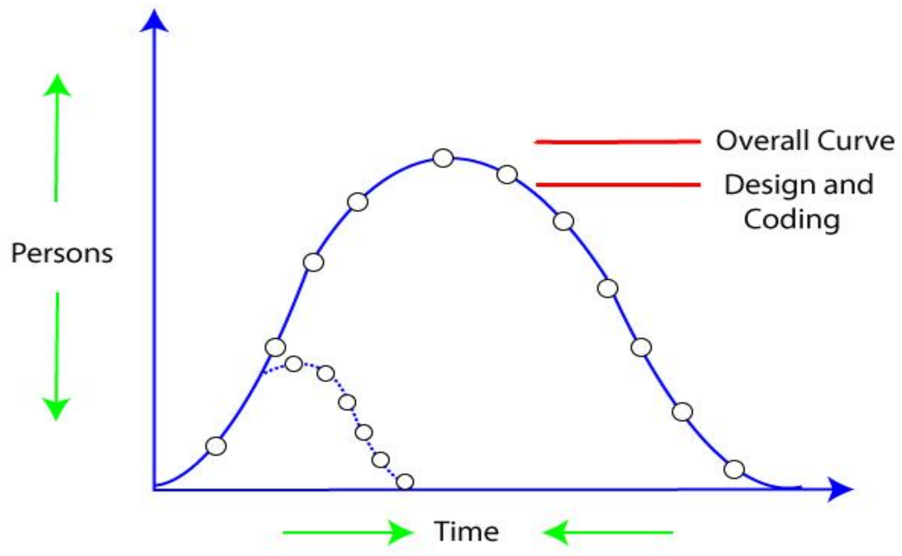
This process is Iterated for several rounds.

No discussion is allowed among the estimator during the entire estimation process because there may be many estimators get easily influenced by rationale of an estimator who may be more experienced or senior.

After the completion of several iterations of estimation, the coordinator takes the responsibility of compiling the result and preparing the final estimates.

Putnam Resource Allocation Model

The Lawrence Putnam model describes the time and effort requires finishing a software project of a specified size. Putnam makes a use of a so-called The Norden/Rayleigh Curve to estimate project effort, schedule & defect rate as shown in fig:



The Rayleigh manpower loading Curve

Putnam noticed that software staffing profiles followed the well known Rayleigh distribution. Putnam used his observation about productivity levels to derive the software equation:

$$L = C_k K^{1/3} t_d^{4/3}$$

The various terms of this expression are as follows:

K is the total effort expended (in PM) in product development, and L is the product estimate in **KLOC** .

$\mathbf{t_d}$ correlate to the time of system and integration testing. Therefore, $\mathbf{t_d}$ can be relatively considered as the time required for developing the product.

$\mathbf{C_k}$ Is the state of technology constant and reflects requirements that impede the development of the program.

Typical values of $\mathbf{C_k}$ = 2 for poor development environment

$\mathbf{C_k}$= 8 for good software development environment

$\mathbf{C_k}$ = 11 for an excellent environment (in addition to following software engineering principles, automated tools and techniques are used).

The exact value of $\mathbf{C_k}$ for a specific task can be computed from the historical data of the organization developing it.

Putnam proposed that optimal staff develop on a project should follow the Rayleigh curve. Only a small number of engineers are required at the beginning of a plan to carry out planning and specification tasks. As the project progresses and more detailed work are necessary, the number of engineers reaches a peak. After implementation and unit testing, the number of project staff falls.

Effect of a Schedule change on Cost

Putnam derived the following expression:

$$L = C_k K^{1/3} t_d^{4/3}$$

Where, **K** is the total effort expended (in PM) in the product development

L is the product size in KLOC

$\mathbf{t_d}$ corresponds to the time of system and integration testing

$\mathbf{C_k}$ Is the state of technology constant and reflects constraints that impede the progress of the program

Now by using the above expression, it is obtained that,

$$K = L^3 / C_k^3 t_d^4$$

Or $$K = C / t_d^4$$

For the same product size, $C = L^3 / C_k^3$ is a constant.

Or $$\frac{K_1}{K_2} = t_{d2}^4 / t_{d1}^4$$

Or $$K \propto 1/t_d^4$$

Or, $$\text{cost} \propto 1/t_d$$

(As project development effort is equally proportional to project development cost)

From the above expression, it can be easily observed that when the schedule of a project is compressed, the required development effort as well as project development cost increases in proportion to the fourth power of the degree of compression. It means that a relatively small compression in delivery schedule can result in a substantial penalty of human effort as well as development cost.

For example, if the estimated development time is 1 year, then to develop the product in 6 months, the total effort required to develop the product (and hence the project cost) increases 16 times.