Unit IV : Software Project Estimation
(weightage - 16 marks)

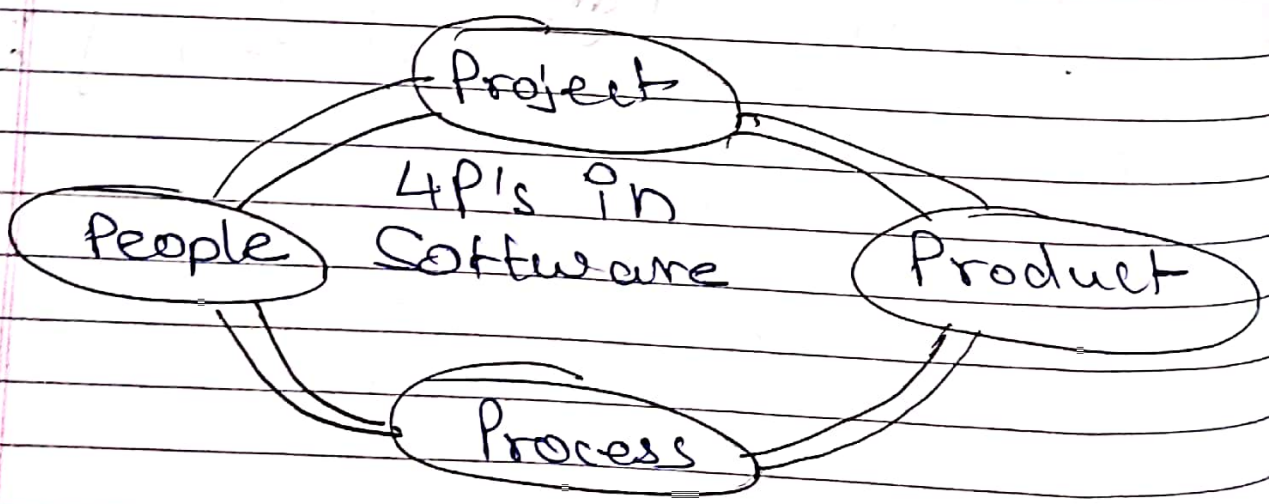
① Explaing following elements of management spectrum :-
i) People ii) Process iii) Product iv) Project

or

Explain the following 4P's management Spectrum.
.... 4 marks.

⇒ For properly building a product, there's a very important concept that we should know in software project planning while developing a product.
There are 4 critical components in software project planning which are known as 4'P namely.

- Product
- People
- Process
- Project



These common 4Ps are dependent on each other like an layer technology.

People

- The backbone of any project is team. A well managed team with clear roles ensures sucess, saving time, cost and effort.
- Roles include project manager, team leader, stakeholders, analyst and IT professions.
- Managing people effectively is crucial for sucess.

Product

- This is result of Project. ie This is what projects aims to deliever.
- The project manager ~~decides~~ defines the product scope, manages team, and addresses technical challenges.
- Products can be tangible or intangible.

Process

A clearly defined process is essential for sucess.
It outlines how the team will proceed through development, including phases like documentation, implementation, deployment and interaction.

Page No.:
Date: / /

Project

- This is overall plan.
- It is Blueprint of Product.
- The project manager leads the team, ensures goal are met.
- He keeps things on track.

(2) State and describe two metrics of project size estimation. ..... (4m)

⇒ A software metric is a measurement of quantifiable or countable software characteristics.
Software metrics are essential for various purpose, including measuring software performance, planning work items and measuring productivity.

i) LOC (Line of Code)

- LOC is simplest among all metrics available to estimate project size.
- This metric is very popular, being simple to use.
- A line of code (LOC) is any line of text in code that is not a comment or blank line, and also header lines.
- LOC clearly consist of all lines containing the declaration of any variable, and executable and non-executable statements.

- As Lines of code only counts the code, you can only use it to compare or estimate project that use the same language and are coded using same standard.
- In simpler terms, LOC counts the total number of lines of source code in project.

The units of LOC are :-
KLOC - Thousand line of code
NLOC - Non-comment line of code
KDSI :- Thousand of delivered source instruction.

Example :
```
// Hellow
for(i=0; i<100; i++)
  {
printf("Hello");
  }
```
We have,
5 physical LOC
2 logical LOC
1 comment LOC.

Page No.:
Date: / /

~~Unit 4~~ :

ii) Function point (FP)

Function point is the method that measure the software size as magnitude of functionality delievered by system to its users.

It is a standard unit of measuring and representing functional size of software application.

It used most widely metrics for measurement of Software Application is based on End users view.

It is unlike LOC, where LOC either measures complexity and lines of code it measures both.

It was developed by Allen J. Albrecht in 1979 at IBM.

It counts inputs, outputs, queries, internal files and external files, and adjusts the complexity of functions.

Point
Functional Type

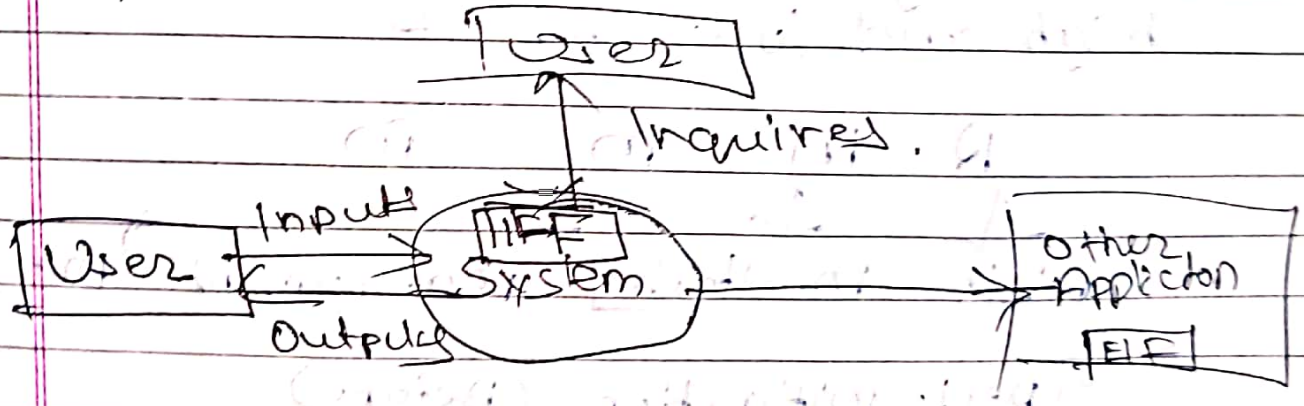
- Transactional Functional Type
- Data Functional Type

Transactional Functional Type

* External Input (EI): These process handle data or control information comming from outside application.
* External output: Generate data or control information and send it ouside application.

Data Functional Type

* Internal logic File: refers of data or control information within boundary of application identifiable by users.
* External Internal File: maintain within another software system but referenced by application.



③ Describe RMMM ~~stragety~~ Strategy. (4m)

⇒ The Risk Mitigation, Management and Monitoring (RMMM) is fundamental component of software project planning.

As overall project plan project managers generally uses this RMMM plan.

A risk management plan is a document that a prepares to foresee risks, estimate impacts and defines responses to risk. It also contains a risk matrix'.

A risk is "an uncertain event or condition, if it occurs, has positive or negative effect on project's objectives".

The risk management plan contains an analysis of likely risks with both high and low impact.

R M M M

Risk monitoring Management mitigation

i) Risk mitigation (Before)

- This is used to avoid Risk in advance
- To avoid risk first we must estimate the risk forehand.
- Removing causes that can cause a risk.

- Controlling / checking the documentation time to time.

ii) Risk Monitoring : (during)

An activity used for project tracking of risk.
It has few objectives like to check wheather predicated risk has occurs or not.
To collect data for future risk analysis to allocate the problem could cause by which risk from project.

iii) Risk Management

It assumes that the mitigation process failed and risks becomes reality. Now project manager are given the task of resolving the risk.
Project Managers used risk mitigation to remove risk. It will be easier to remove risk.

For example
Risk : Computer crash.

Mitigation
Backup software and documentation in multiple places to prevent data loss.

Monitoring
Regularly check the stability of the computing environment while working.

Management
Stop work if the computing environment is unstable until it's fixed or switch to a stable system.

④ Define risk. Enlist types of risk. ... (2m)
⇒ A possibility of suffering from loss in software development process is called as Risk.
Any action or activity that leads to loss of any type can be termed as risk.

Types of risk
Generic risk, Product specific risk, Budget/ Financial risk, Technical Risk, Operational Risk. Delivery Related Planning Risk.

(5) List the project cost Estimation Approaches. (2m)

⇒ Heuristic Estimation Approach
Analytical Estimation Approach
Empirical Estimation Approach.

(6) Define project cost estimation. (2m)

⇒ Project cost estimation is the process of predicting the quantity, cost, and price of the resources required by scope of project.

(7) Define Empirical estimation approach (2m)

⇒ Empirical estimation technique are based on making an educated guess of project parameters.
It is based on common sense.
There are many activites involves Empirical estimation technique is formalized.
Delphi technique is used as Emphirical estimate approach.
Method involves more interaction and communication between who are participation.
Sucessful for technical forecasting.

(8) Define Software cost estimation approaches.

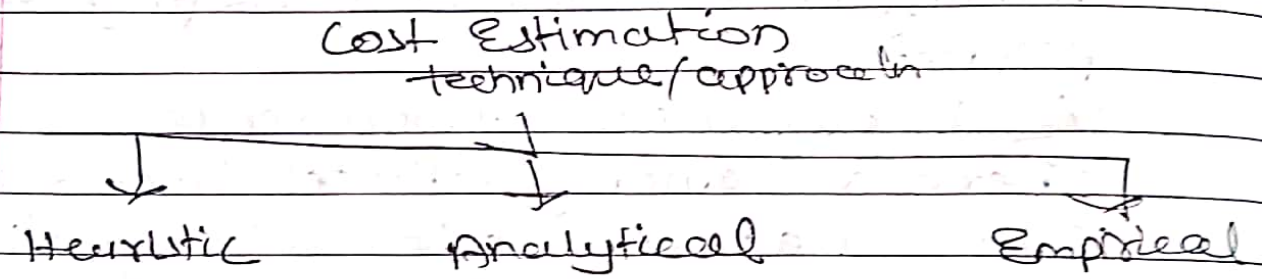
following :-

1) ~~Heuristic~~ Heuristic Estimation approach
2) Analytical Estimation approach
3) Empirical Estimation approach

-[4m/6m]

⟹ Cost estimation approach models are some algorithms or parametric equations that are used to estimate the cost of a product or a project.

Various techniques are available for cost estimation, also known as Cost Estimation Models.



Empirical

It uses formulae based on past data to project parameters in project planning.

It relies on past project data, guesses, prior experience and assumptions.

Estimates ~~effort~~ based on size of software.

Despite being formalized, it's grounded in common sense and educated guesses.

Despite it reliance common sense, the technique is structured and formalized.

Examples: Delphi & Expert Judgment

Page No.:
Date: / /

Heuristic technique

Heuristic technique aid problem-solving, learning, or discovery by offering practical methods to achieve immediate goals.

Heuristic word comes from Greek word meaning "to discover".

These technique are flexible and simple, allow for quick decesions through shortcuts and basic calculations, especially complex data.

Decesion made using it need to be satisfactory, if not optimal.

Example, Cocomo Model.

Relationship among project parameters are represented using mathematical equations.

Analytical

Analytical model/Technique is type of technique used for measuring work.

In this technique, firstly the task is divided or broken down into its basic component or operation or elements for analysing.

Second, if Standard time is available from some other source, then these sources are applied to each element or component of work.

In this technique, results are derived by making certain basic assumptions about the project.

Hence, the analytical estimation technique has some scientific basis.

Halstead's software science is based on an analytical estimation model.

(a) Describe Cocomo I & Cocomo II model. (6M)

⇒ Co Co mo
constructive cost model.

Cocomo

Cocomo is widely used software estimation model developed by Barry Boehm in 1981.

It is regression model that estimates effort and schedule for software projects based on size of software, typically measured in Lines of Code (LOC).

It provides a structured approach to predict various parameters essential for project management such as size, effort, cost, time and quality.

Cocomo is renowned for its reliability, stemming from its foundation on the analysis of 63 projects, making it most well-documented model in the field of software engineering.

Page No.:
Date: / /

Necessary step in model :

Estimate size : Start by figuring out how big your software project, by LOC.

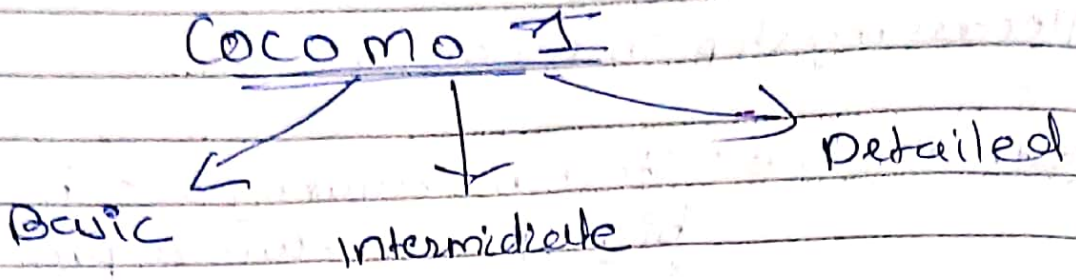
Consider Factors : Think about different aspect of your project. The one represented by 15 factors, Some are how complex it is, exprience of team etc.

Calculate Effort : Multiply your initial size estimate by all those factors are you considered.

Project are catogerized into three types

* Organic : Think of this like making simple program with a small team of exprienced developers.
* Semidetached : Imagine a project where some team members are experinced, but some are freshers.
* Embedded : This is when the software you're working on is connected to complex hardware or strict rules for how it works. Like ATM.

Cocomo I



Basic Cocomo model

- This model estimates effort and development time based on size of software (KLOC)
- It is divided into three classes: Semidetached, embedded and development Organic, each with its own formulas to estimate and development time.
- For example, for an Organic project, effort is estimated as 2.4 times the KLOC raised to the power 1.05, and development time is estimated as 2.5 times the effort raised to power of 0.38.

Intermidiate model

- Recognizes that effort and schedule can't be solely calculated based on lines of code.
- It considers 15th Cost Drivers that affect estimation, including factors like required reliability, complexity and exprience.
- These drivers help adjust the estimate based on various aspects of project, like database

Detailed model

It goes even deeper than the Intermidiate Model.

It looks how each factor affects different parts of development process.

It breaks the software into smaller parts and estimates efforts for each part seperately.

This model has 6 phases.

Pat Planning, System Design, Detailed Design etc.

Cocomo II

* Cocomo-II is an improved version of orginal cocomo model for estimating software development cost, effort and schedule.
* It offers three models tailored for different stages.

End User programming	Application generators and composite aid	Infrastructure
	Application Composition	
	System Integration	

Cocomo II

Application Composition Model

It is used for prototyping efforts to resolve high-risk issues such as user interface and technology maturity.
Estimates costs based on object Points
Suitable for projects built with modern GUI-builder tools.

Early Design Model

- Used for exploring alternative software/system architecture and concepts of operations.
- Utilizes a set of new Cost Drivers and Estimating equations, based on Unadjusted Function Points or KSLOC.

Post-Architecture Model

- Used for actual development and maintainence of software products.
- Estimate Effort in person months (PM) based on project size, a constant (A), and a scale factor (B).

Page No.
Date: / /

Parameters in COCOMO II

1) Person Month (PM)

This measures how much time one person spends working on project for a month.
It's estimated using a formula that considers the project size and other factors like complexity & team experience

PM nominal = A * (Size)^B

2) Maintenance Size

It is size of project code that needs to be changed.
Calculated using base code size, Maintened Change Factor (MCF) and Maltenance Effort (MAF)

Size = [(Basic Code Size) * MCF] * MAF

3) maintenance change Factor

The percentage of change to base code

MCF = (Size Added + Size Modified) / Base Code Size

4) Maitenance Effort

Helps account the impact of software structure and understandibility on maintenance work.

MAF = 1 + (SU * 0.01 * UNFM)

Cocomo Numerical

Q// Use cocomo model to calculate
i) Effort
ii) & Development Time
iii) Average Staff size.
iv) Productivity.

if estimated size of project 400 KLOC using embedded mode.

⇒ Effort = $a(\text{KLOC})^b$
$= 3.6(400)^{1.20}$
$= 4772.8137$ per month.

Development: $c(\text{Effort})^{0.32}$
$= 2.5(4772.8137)^{0.32}$
$= 37.5965$ months

Average Staff size

Effort / Development
$= 4772.8137 / 37.5965$
$= 126.94835$ person

Productivity

KLOC / Effort
$400 / 4772.8137$
$= 0.08380$

Page No.:
Date: / /

Q// Cocomo model

i) Effort
ii) Development time

KLOC: 500

Semidetached & Embedded.

	a	b	c	d
O	2.4	1.05	2.5	0.38
S	3.0	1.12	2.5	0.35
E	3.6	1.20	2.5	0.32

Semidetached

Effort : $a(KLOC)^b$

$= 3.0(500)^{1.12}$

$= 3162.0421$ per month

Embedded

Development $= c(Effort)^{0.35}$

$= 2.5(3162.0421)^{0.35}$

$= 41.9690$ months.

$\simeq 42$ round up

Embedded

Effort : $a(KLOC)^b$

$= 3.6(500)^{1.20}$

$= 6238.3035$ person-months.

Development time

$= c(KLOC)^d$

$= 2.5(6238.3035)^{0.32}$

$= 40.9601$

c) 200KLOC : Organic

Effort = $a(KLOC)^b$

$= 2.4(200)^{1.05}$

$= 625.5942$ per month

626

Project duration :

$: C\left(\frac{KLOC}{Effort}\right)^d$

$= 2.5(625.59)^{0.38}$

$= 28.8755$ month

Average staff size

: Effort / duration

$= 623.5942/28.8755$

$= 21.66522$ persons.