Software Engineering

CA210

Unit 3:
Project Estimation
&
Scheduling

➢ Project management

- Software project management includes the tools, techniques, and knowledge essential to deal with the growth of software products.
- In Software Project Management, the end users and developers require to know the cost of the project, duration and length.
- It is a process of managing, allocating and timing resources to develop computer software that meets necessities.

It consists of eight tasks:

1. **Problem Identification**
2. **Problem Definition**
3. **Project Planning**
4. **Project Organization**
5. **Resource Allocation**
6. **Project Scheduling**
7. **Tracking, Reporting and Controlling**
8. **Project Termination**

What is Project?

A project is a sequence of unique, complex, and connected activities having one goal or purpose and that must be completed by a specific time, within budget, and according to specification. This definition tells us quite a bit about a project

Project Management:

- The methods and regulation used to define goals, plan and monitor tasks and resources, identify and resolve issues, and control costs and budgets for a specific project is known as project management.
- Project management is "the application of knowledge, skills, tools and techniques to project activities to meet the project requirements."
- A project should be initiated with a **feasibility study**, where a clear definition of the goals and ultimate benefits need to be determined.

➢ The Management Spectrum:

- The management spectrum describes the management of a software project or how to make a project successful.
- The Management Spectrum provides a holistic approach to managing software projects by addressing key areas: **People, Product, Process, and Project.**
- Effective management across these dimensions helps in reducing risks, improving productivity, and ensuring that the software meets its intended goals.
- Effective software project management focuses on the four P’s:
- **people, product, process, and project**. The order is not arbitrary.

The People :

- This involves defining the roles and responsibilities of every team member.
- Key roles might include **project managers, software engineers, testers, and stakeholders.**
- Ensuring that team members have the necessary skills and knowledge to perform their tasks effectively.
- The people capability maturity model (CMM) defines the following key practice areas for software people: staffing, communication and coordination, work environment, performance management, training, compensation, competency analysis and development, career development, workgroup development, team/culture development, and others.

❑**Product**

- It defines Project Scope: Clearly defining the scope of the software product, including functional and non-functional requirements.
- Establishing the goals and objectives that the product must meet, often tied to customer needs or business goals.
- Before a project can be planned, product objectives and scope should be established, alternative solutions should be considered and technical and management constraints should be identified
- Without this information, it is impossible to define reasonable(and accurate) estimates of the cost, an effective assessment of risk, a realistic breakdown of project tasks, or a manageable project schedule that provides a meaningful indication of progress.

❑The process

- Software process provides the framework from which a comprehensive plan for software development can be established.
- A small number of framework activities are applicable to all software projects, regardless of their size or complexity.
- A number of different tasks set—tasks, milestones, work products and quality assurance points—enable the framework activities to be adapted to the characteristics of the software project and the requirements of the project team.
- Finally, umbrella activities—such as software quality assurance, software configuration management, and measurement—overlay the process model .
- Umbrella activities are independent of any one framework activity and occur throughout the process
- Once the process model is chosen, populate it with the minimum set of work tasks and work products that will result in a high-quality product—avoid process overkill!

❑The Project

- Developing a detailed project plan that outlines tasks, timelines, resources, and budgets.
- The Project We conduct planned and controlled software projects for one primary reason—it is the only known way to manage complexity.
- To avoid project failure, a software project manager and the software engineers who build the product must avoid a set of common warning signs, understand the critical success factors that lead to good project management, and develop a commonsense approach for planning, monitoring, and controlling the project.
- Identifying potential risks that could impact the project and developing strategies to mitigate these risks.

➢ The WWWWWHH (W5HH) Principle

- **What questions need to be answered in order to develop a project plan? For this Bohem suggests W5HH Principle**
- In an excellent paper on software process and projects, Barry Boehm states: “you need an organizing principle that scales down to provide simple [project] plans for simple projects.”
- Boehm suggests an approach that addresses project objectives, milestones and schedules, responsibilities, management and technical approaches, and required resources.
- **He calls it the WWWWWHH principle, after a series of questions that lead to a definition of key project characteristics and the resultant project plan:**

Why is the system being developed?

The answer to this question enables all parties to assess the validity of business reasons for the software work. Stated in another way, does the business purpose justify the expenditure of people, time, and money?

What will be done, by when? The answers to these questions help the team to establish a project schedule by identifying key project tasks and the milestones that are required by the customer.

Who is responsible for a function? Earlier in this chapter, we noted that the role and responsibility of each member of the software team must be defined. The answer to this question helps accomplish this.

Where are they organizationally located? Not all roles and responsibilities reside within the software team itself. The customer, users, and other stakeholders also have responsibilities.

How will the job be done technically and managerially? Once product scope is established, a management and technical strategy for the project must be defined.

How much of each resource is needed? The answer to this question is derived by developing based on estimates. This question involves estimating the costs associated with the project, including development, resources, tools, and other expenses.

Boehm’s W5HH principle is applicable regardless of the size or complexity of a software project. The questions noted provide an excellent planning outline for the project manager and the software team.

➢Introduction of Estimation

- Estimation is the 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.
- Software Project Estimation is the process of estimating various resources required for the completion of a project.
- Estimation determines how much money, effort, resources, and time it will take to build a specific system or product.
- Estimation is based on −
    - Past Data/Past Experience
    - Available Documents/Knowledge
    - Assumptions
    - Identified Risk

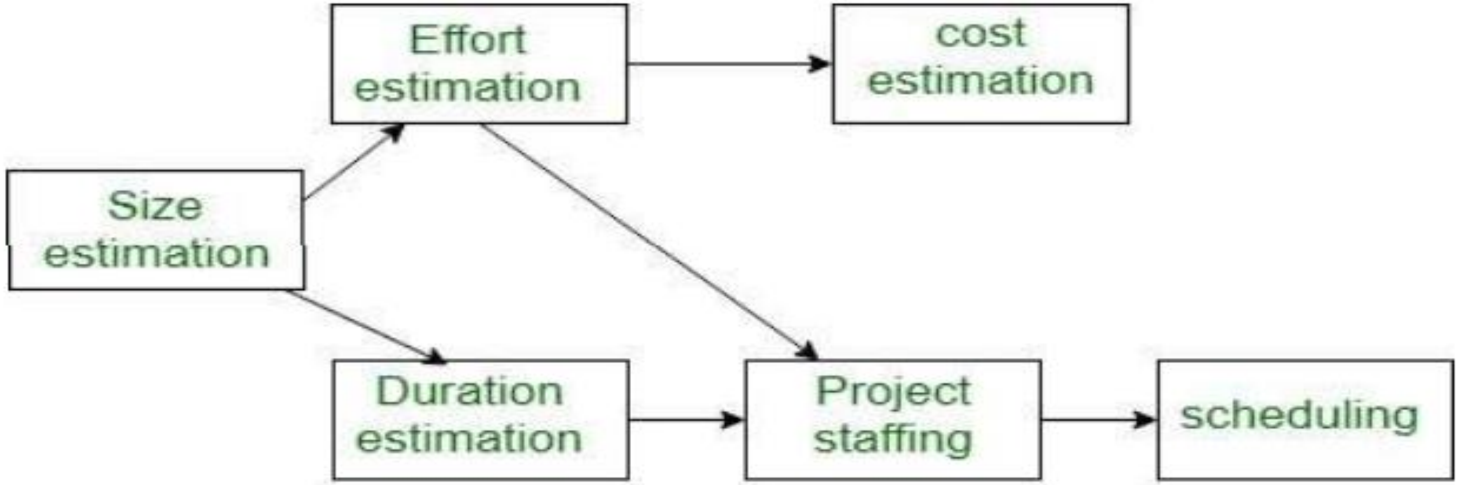
The four basic steps in software project estimation are:

1) **Estimate the size of the development product**. The size can be estimated by using either Lines of Code (LOC) or Function Points (FP).

2) **Estimate the effort** in person-months or person-hours (man-month or man-hour). Man-month is an estimate of personal resources required for the project.

3) **Estimate the schedule** in calendar days /months/ years based on total man-month required and manpower allocated to the project.

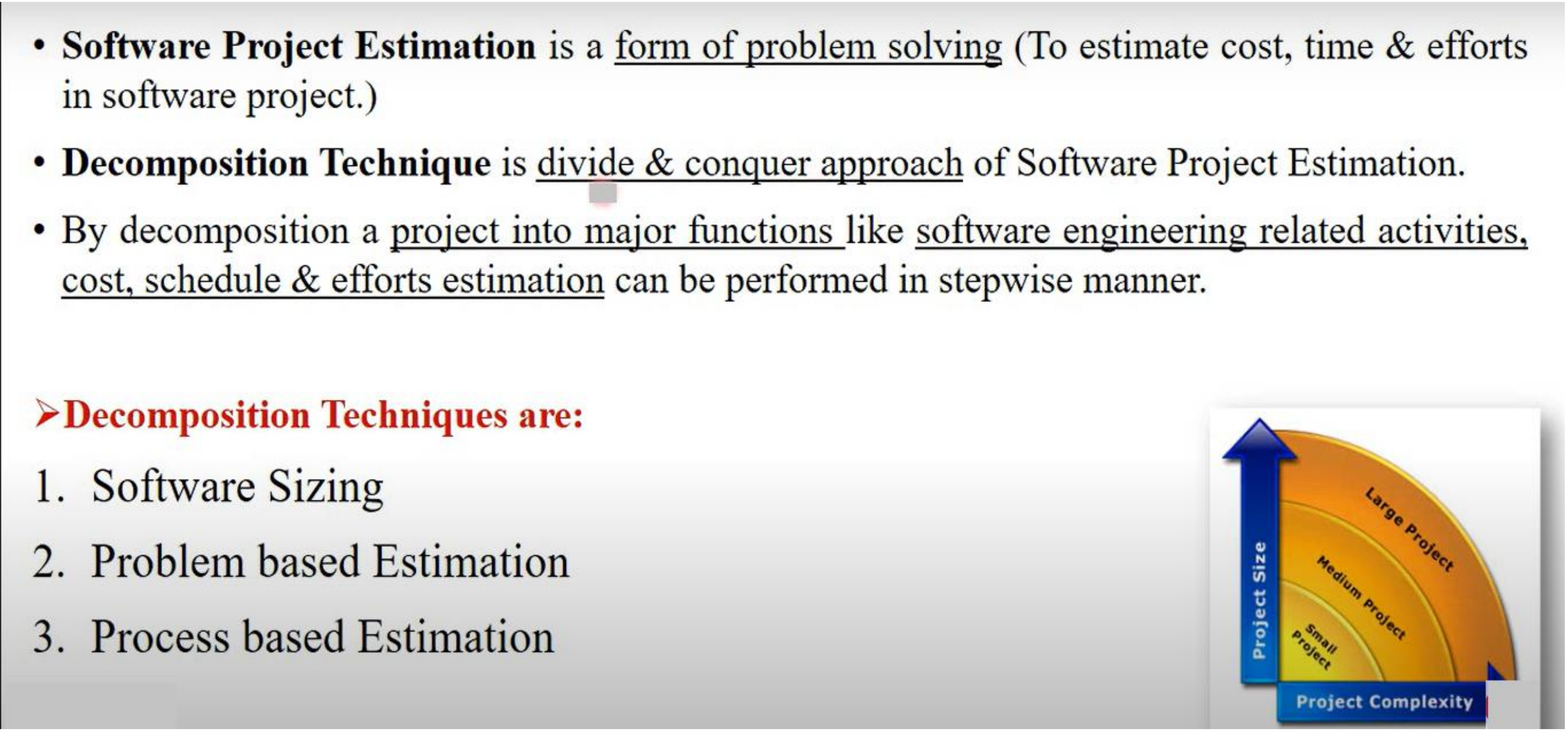
4) **Estimate the project cost** in local currency.



Precedence ordering among planning activities

Project Estimation Process

DECOMPOSITION TECHNIQUES

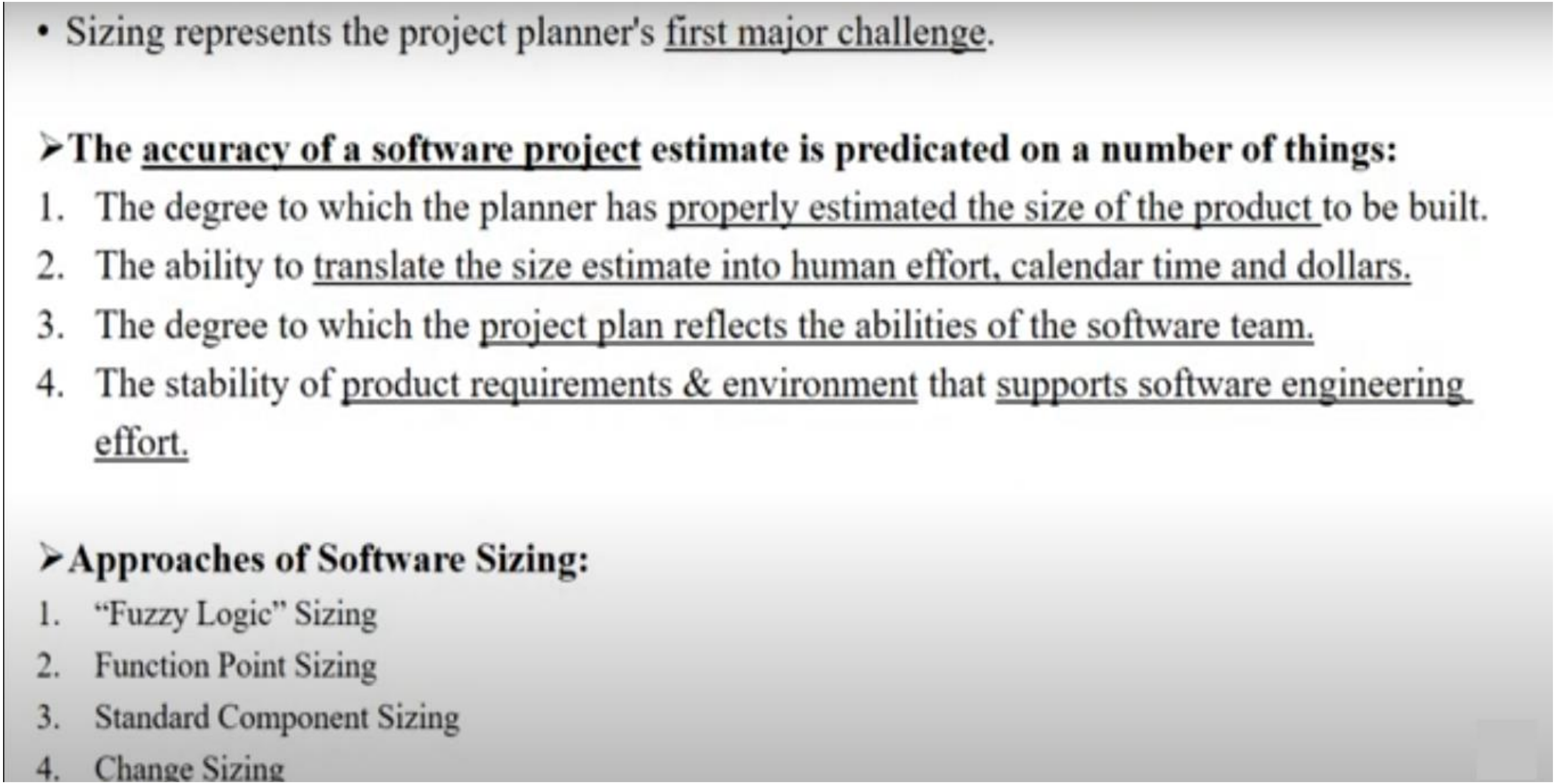


- **Software Project Estimation** is a form of problem solving (To estimate cost, time & efforts in software project.)
- **Decomposition Technique** is divide & conquer approach of Software Project Estimation.
- By decomposition a project into major functions like software engineering related activities, cost, schedule & efforts estimation can be performed in stepwise manner.

➢**Decomposition Techniques are:**

1. Software Sizing
2. Problem based Estimation
3. Process based Estimation

Software Sizing



- Sizing represents the project planner's first major challenge.

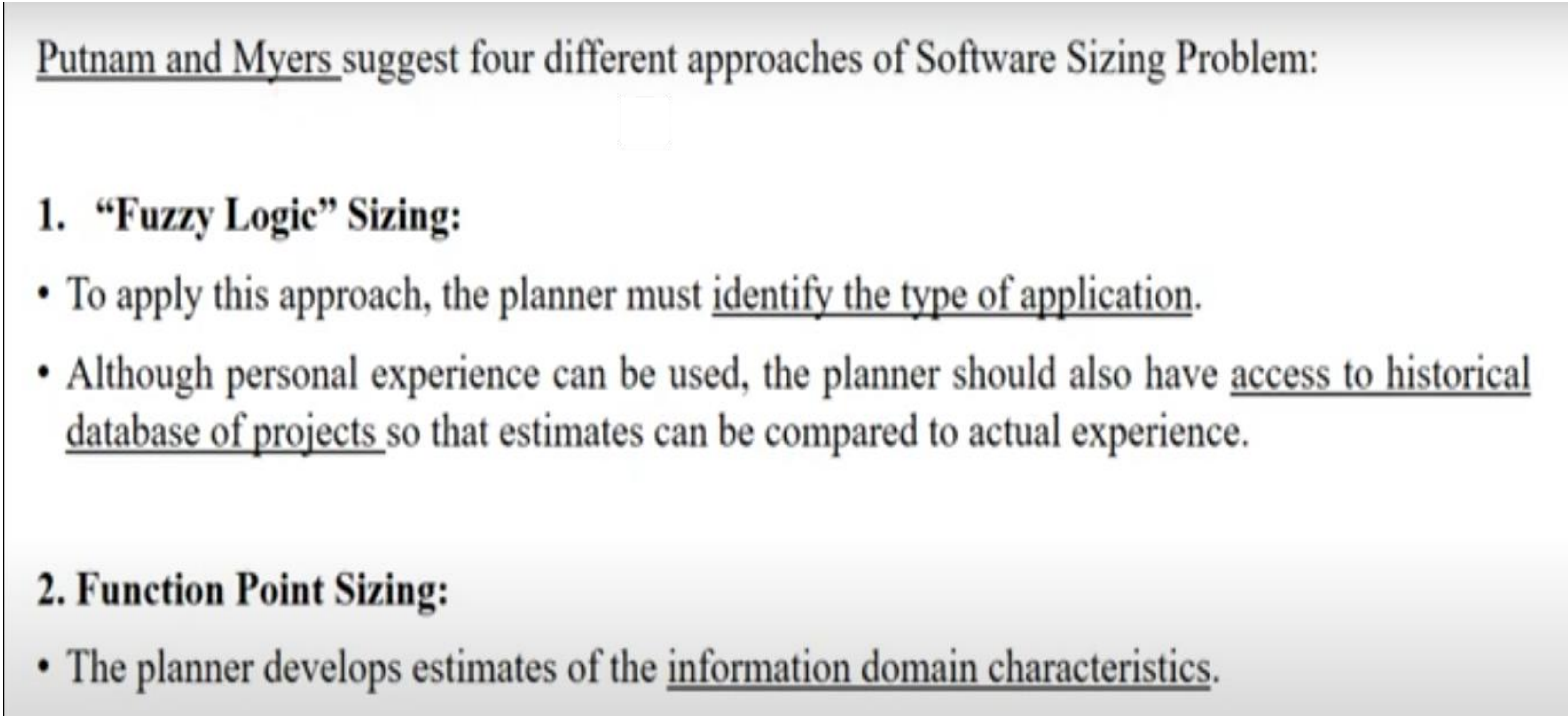
➢The accuracy of a software project estimate is predicated on a number of things:

1. The degree to which the planner has properly estimated the size of the product to be built.
2. The ability to translate the size estimate into human effort, calendar time and dollars.
3. The degree to which the project plan reflects the abilities of the software team.
4. The stability of product requirements & environment that supports software engineering effort.

➢Approaches of Software Sizing:

1. "Fuzzy Logic" Sizing
2. Function Point Sizing
3. Standard Component Sizing
4. Change Sizing

Approaches of Software Sizing



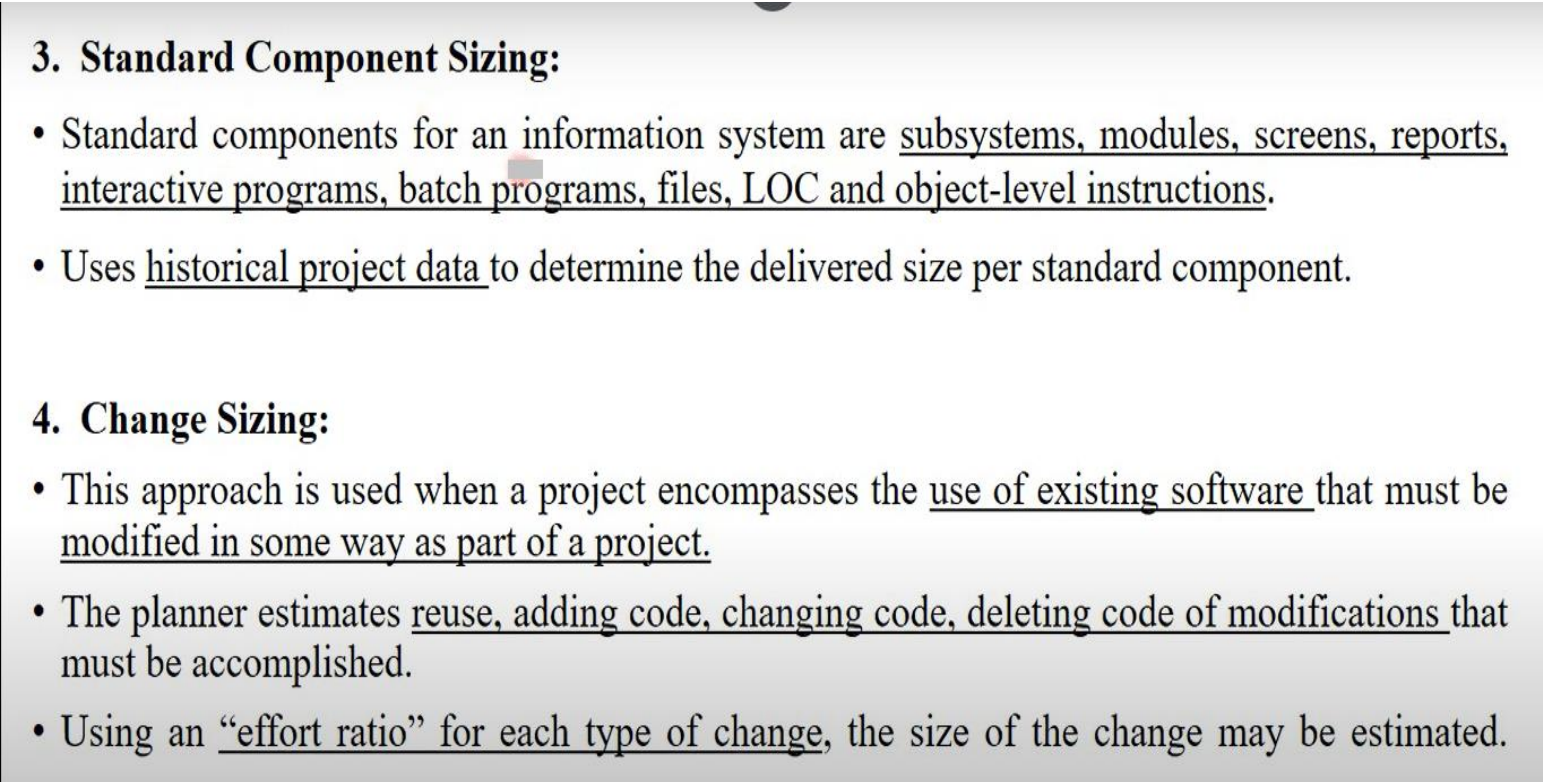
Putnam and Myers suggest four different approaches of Software Sizing Problem:

1. **"Fuzzy Logic" Sizing:**

- To apply this approach, the planner must identify the type of application.
- Although personal experience can be used, the planner should also have access to historical database of projects so that estimates can be compared to actual experience.

2. Function Point Sizing:

- The planner develops estimates of the information domain characteristics.



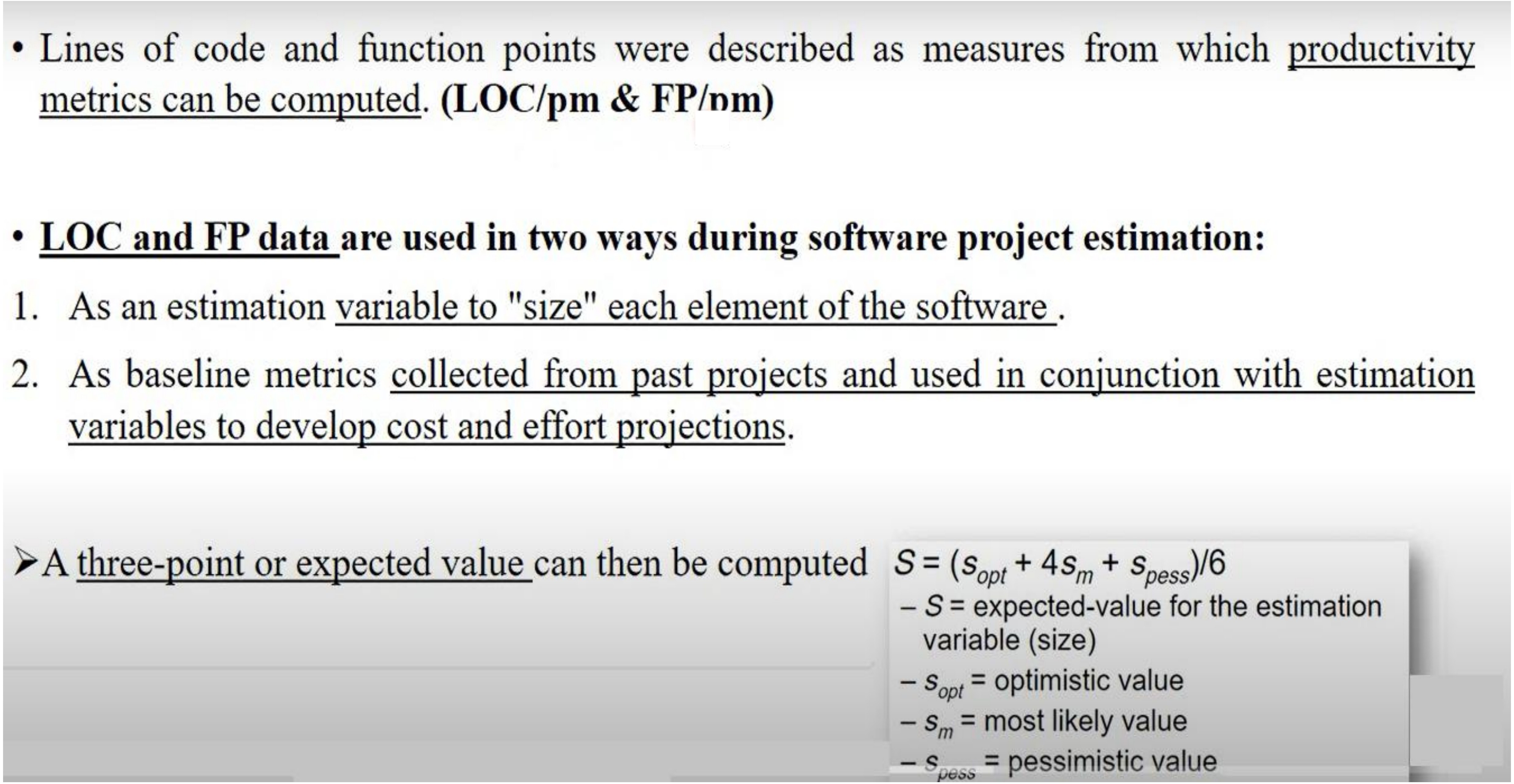
3. Standard Component Sizing:

- Standard components for an information system are subsystems, modules, screens, reports, interactive programs, batch programs, files, LOC and object-level instructions.
- Uses historical project data to determine the delivered size per standard component.

4. Change Sizing:

- This approach is used when a project encompasses the use of existing software that must be modified in some way as part of a project.
- The planner estimates reuse, adding code, changing code, deleting code of modifications that must be accomplished.
- Using an "effort ratio" for each type of change, the size of the change may be estimated.

Problem Based Estimation



- Lines of code and function points were described as measures from which productivity metrics can be computed. **(LOC/pm & FP/pm)**

- **LOC and FP data are used in two ways during software project estimation:**

1. As an estimation variable to "size" each element of the software .
2. As baseline metrics collected from past projects and used in conjunction with estimation variables to develop cost and effort projections.

➢A three-point or expected value can then be computed $S = (s_{opt} + 4s_m + s_{pess})/6$

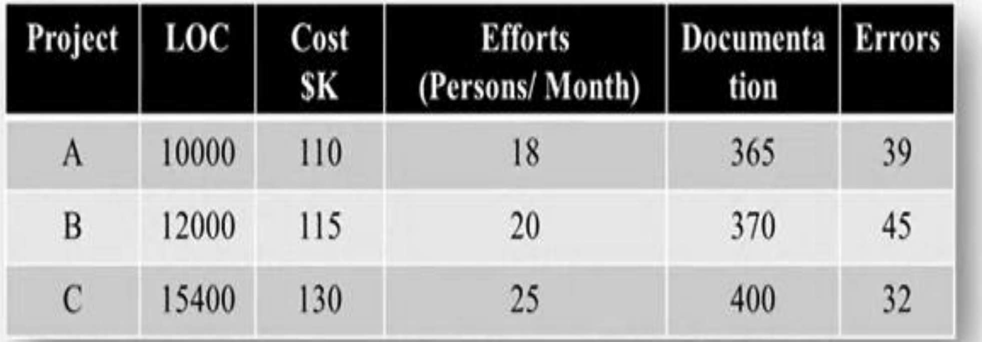
– S = expected-value for the estimation variable (size)

– s_{opt} = optimistic value

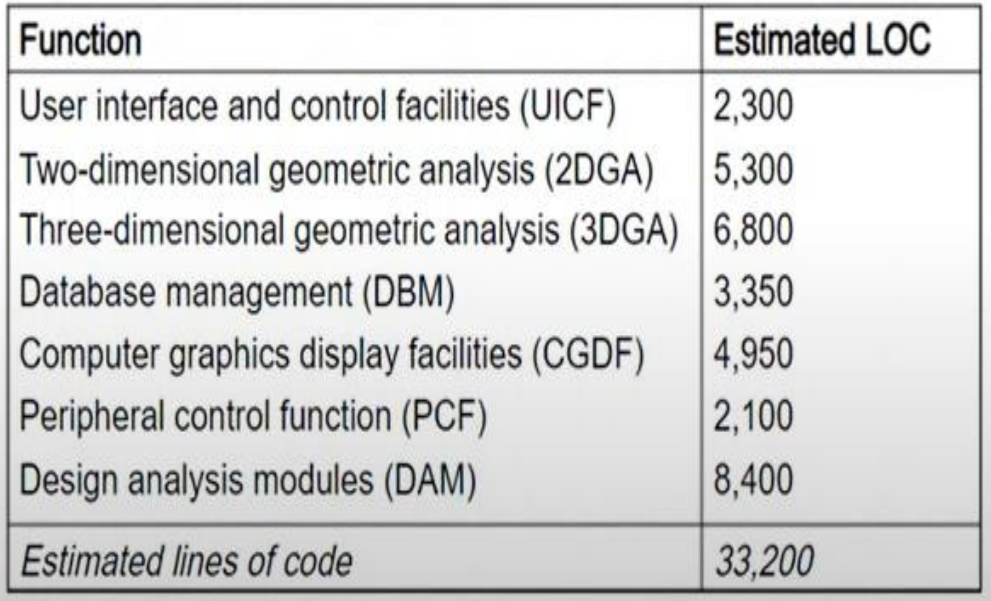
– s_m = most likely value

– s_{pess} = pessimistic value

1. LOC (Lines of Code):

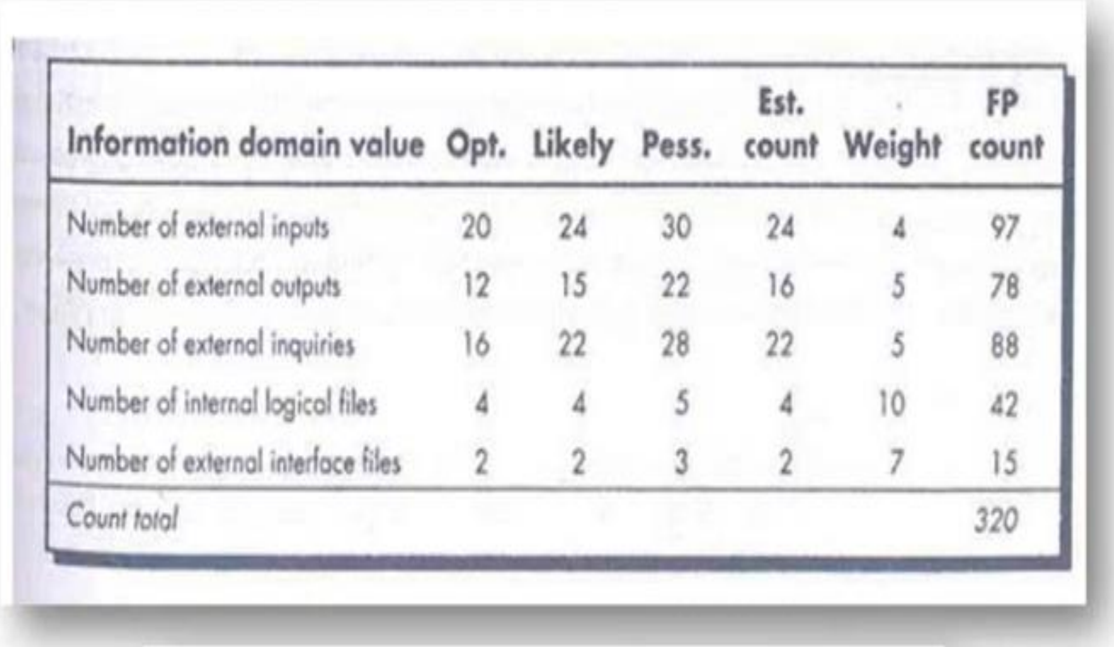


Project	LOC	Cost $K	Efforts (Persons/ Month)	Documentation	Errors
A	10000	110	18	365	39
B	12000	115	20	370	45
C	15400	130	25	400	32

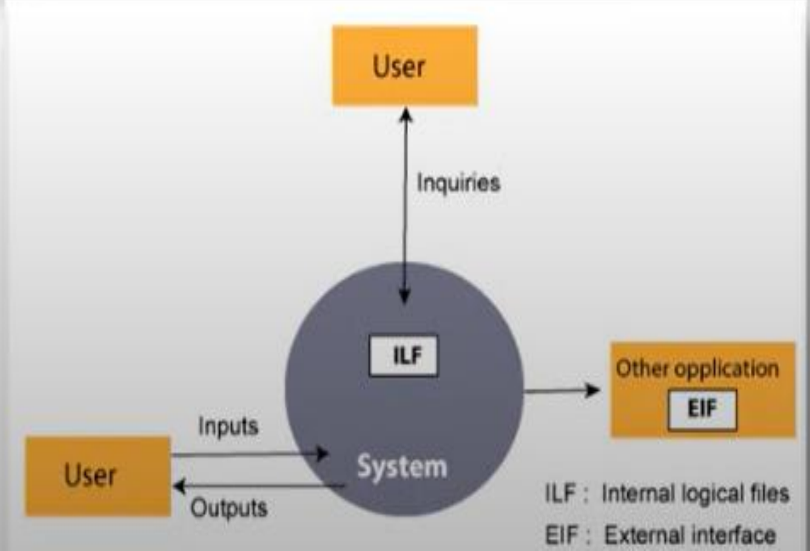


Function	Estimated LOC
User interface and control facilities (UICF)	2,300
Two-dimensional geometric analysis (2DGA)	5,300
Three-dimensional geometric analysis (3DGA)	6,800
Database management (DBM)	3,350
Computer graphics display facilities (CGDF)	4,950
Peripheral control function (PCF)	2,100
Design analysis modules (DAM)	8,400
Estimated lines of code	*33,200*

2. FP(Functional Points)

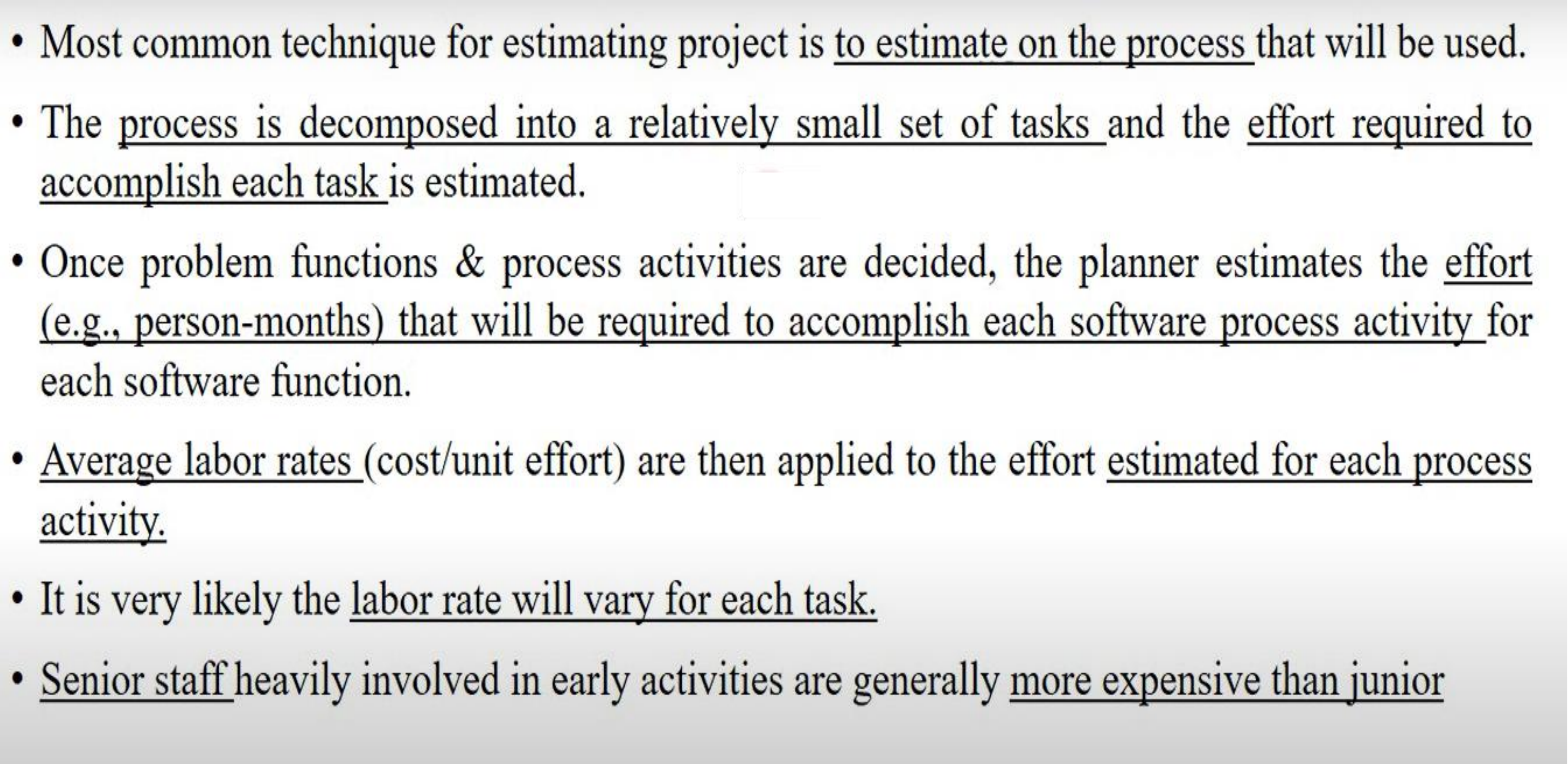


Information domain value	Opt.	Likely	Pess.	Est. count	Weight	FP count
Number of external inputs	20	24	30	24	4	97
Number of external outputs	12	15	22	16	5	78
Number of external inquiries	16	22	28	22	5	88
Number of internal logical files	4	4	5	4	10	42
Number of external interface files	2	2	3	2	7	15
Count total						*320*



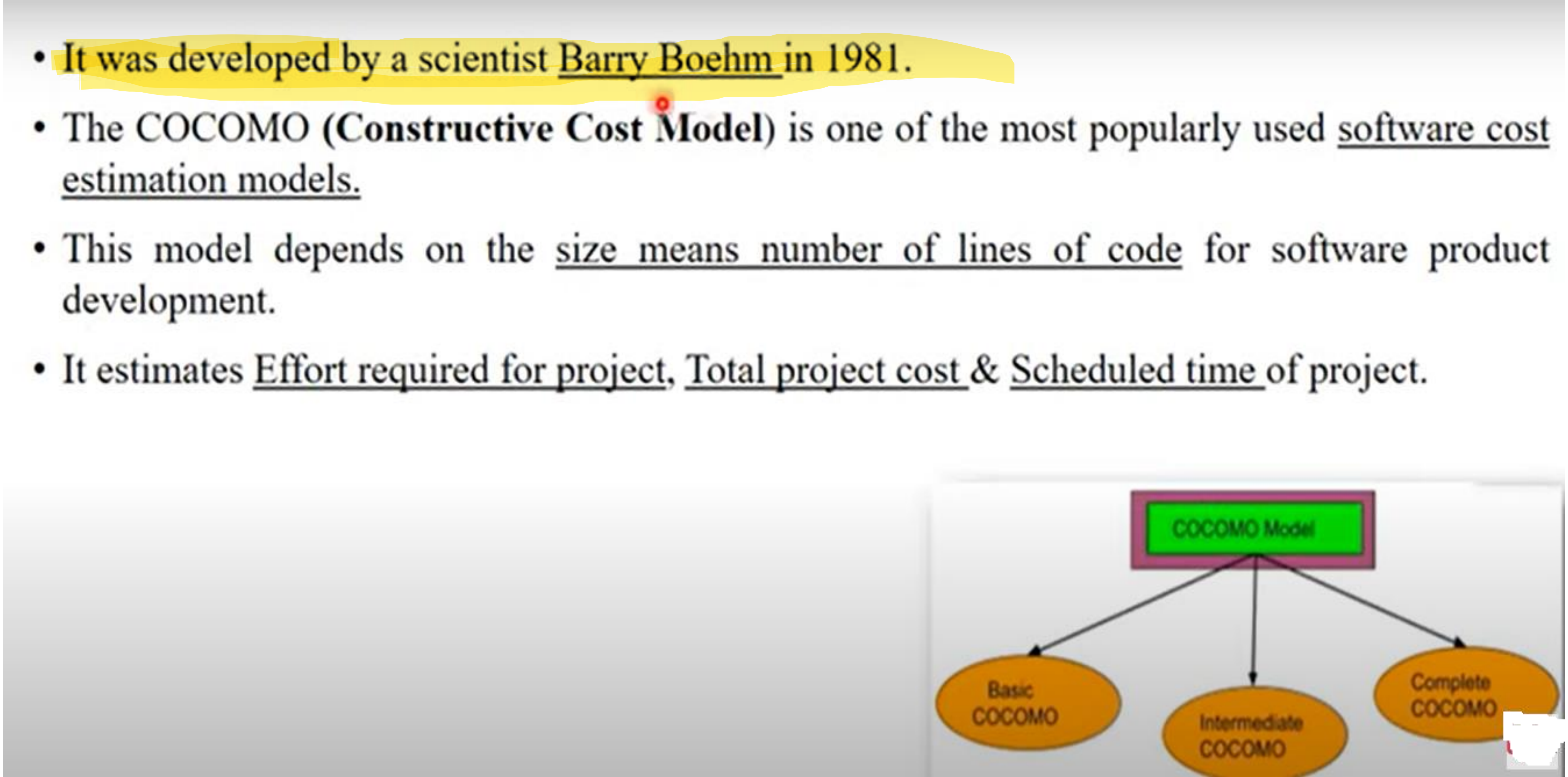
Cost is in from of $

Process Based Estimation

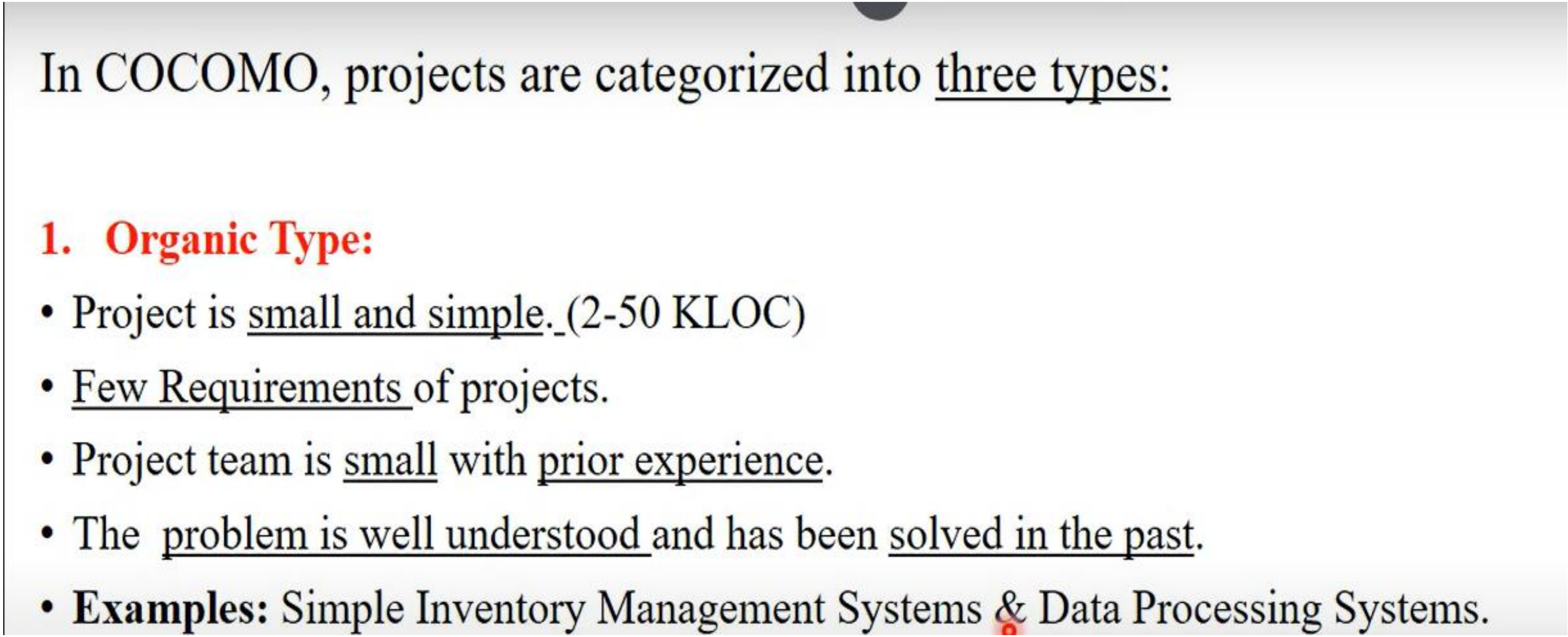


- Most common technique for estimating project is to estimate on the process that will be used.
- The process is decomposed into a relatively small set of tasks and the effort required to accomplish each task is estimated.
- Once problem functions & process activities are decided, the planner estimates the effort (e.g., person-months) that will be required to accomplish each software process activity for each software function.
- Average labor rates (cost/unit effort) are then applied to the effort estimated for each process activity.
- It is very likely the labor rate will vary for each task.
- Senior staff heavily involved in early activities are generally more expensive than junior

COCOMO Model



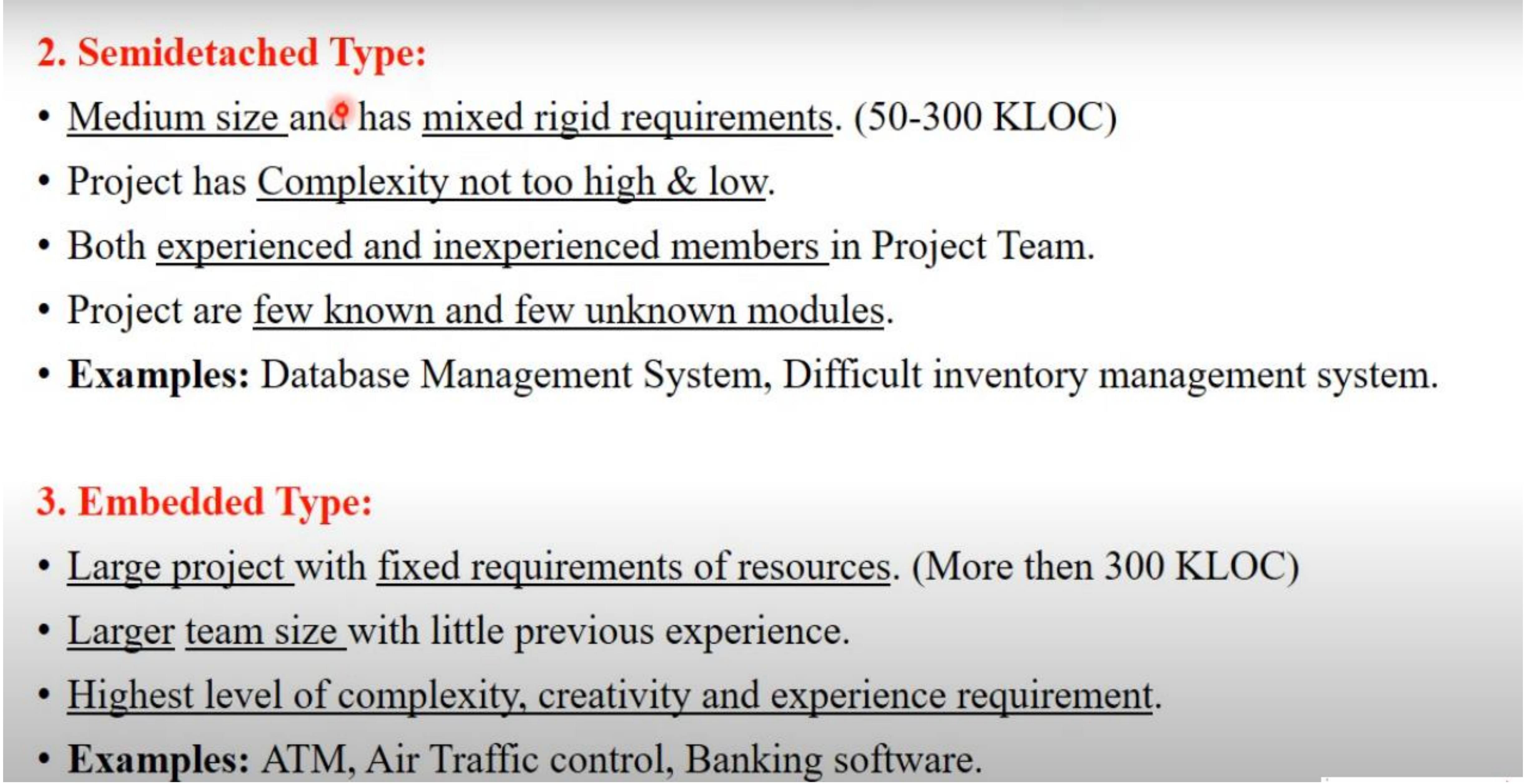
- It was developed by a scientist Barry Boehm in 1981.
- The COCOMO (**Constructive Cost Model**) is one of the most popularly used software cost estimation models.
- This model depends on the size means number of lines of code for software product development.
- It estimates Effort required for project, Total project cost & Scheduled time of project.



In COCOMO, projects are categorized into three types:

1. **Organic Type:**

- Project is small and simple. (2-50 KLOC)
- Few Requirements of projects.
- Project team is small with prior experience.
- The problem is well understood and has been solved in the past.
- **Examples:** Simple Inventory Management Systems & Data Processing Systems.

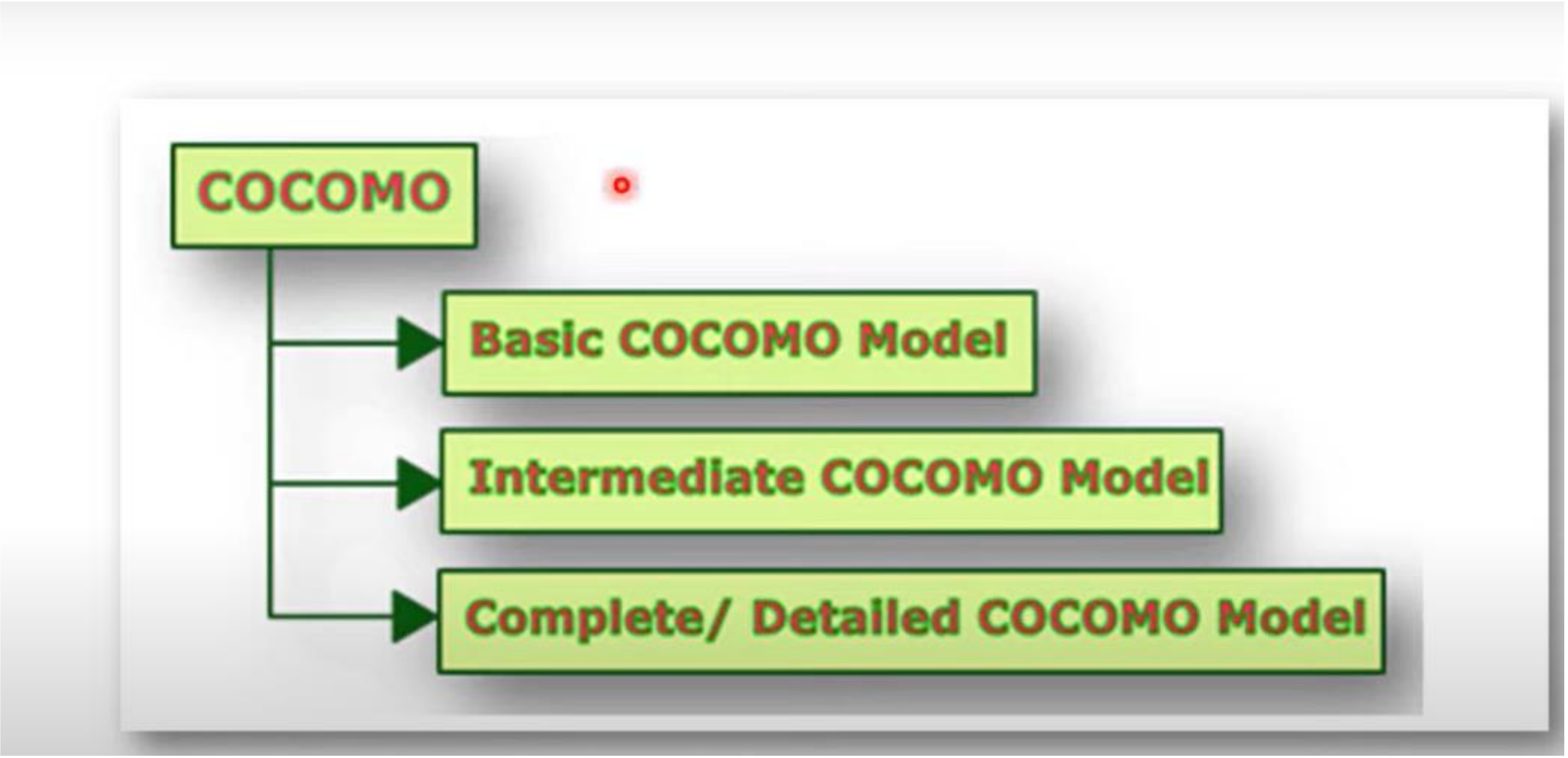


2. Semidetached Type:

- Medium size and has mixed rigid requirements. (50-300 KLOC)
- Project has Complexity not too high & low.
- Both experienced and inexperienced members in Project Team.
- Project are few known and few unknown modules.
- **Examples:** Database Management System, Difficult inventory management system.

3. Embedded Type:

- Large project with fixed requirements of resources. (More then 300 KLOC)
- Larger team size with little previous experience.
- Highest level of complexity, creativity and experience requirement.
- **Examples:** ATM, Air Traffic control, Banking software.



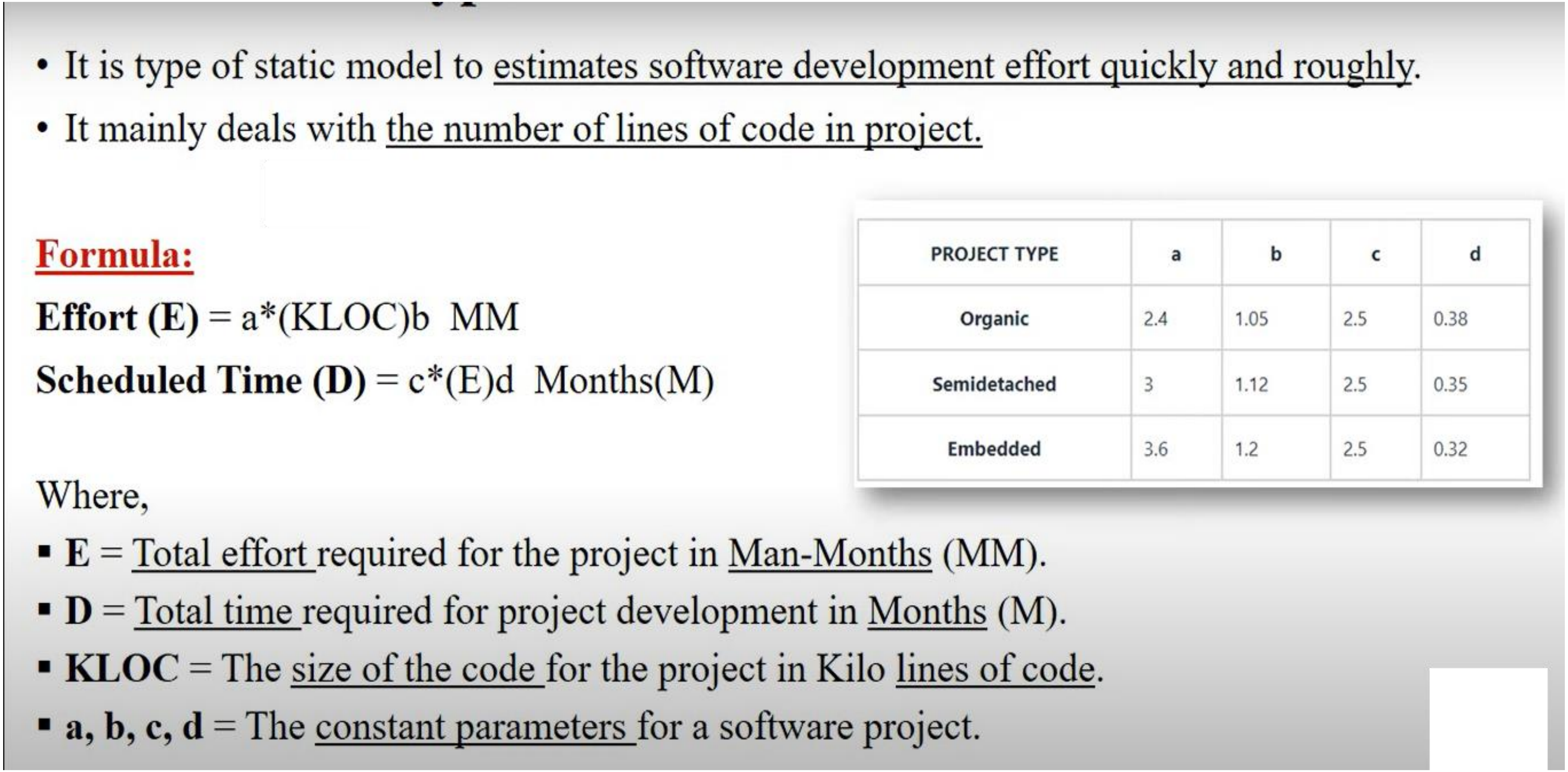
COCOMO

Basic COCOMO Model

Intermediate COCOMO Model

Complete/ Detailed COCOMO Model

Basic COCOMO



- It is type of static model to estimates software development effort quickly and roughly.
- It mainly deals with the number of lines of code in project.

Formula:

Effort (E) = a*(KLOC)b MM

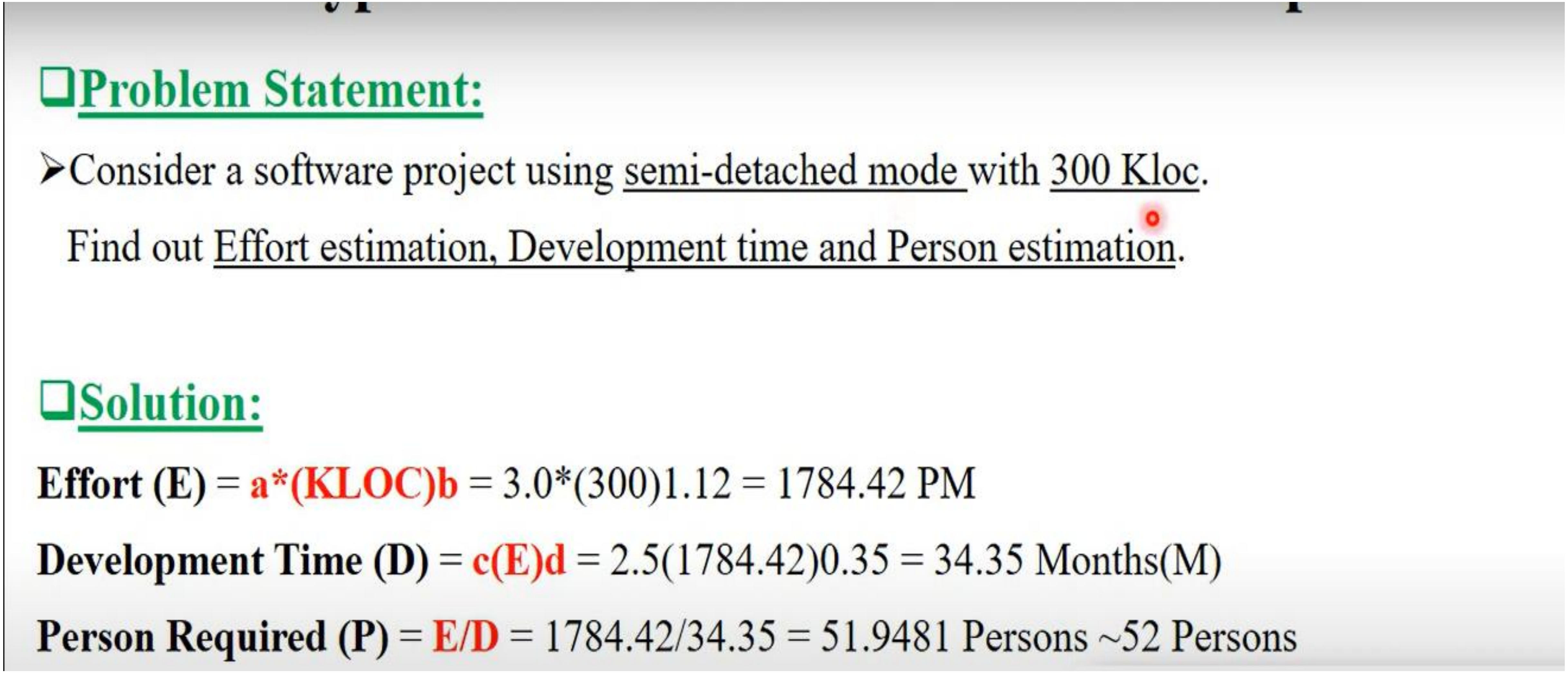
Scheduled Time (D) = c*(E)d Months(M)

PROJECT TYPE	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semidetached	3	1.12	2.5	0.35
Embedded	3.6	1.2	2.5	0.32

Where,

- **E** = Total effort required for the project in Man-Months (MM).
- **D** = Total time required for project development in Months (M).
- **KLOC** = The size of the code for the project in Kilo lines of code.
- **a, b, c, d** = The constant parameters for a software project.

Example of Basic COCOMO



❑Problem Statement:

➢Consider a software project using semi-detached mode with 300 Kloc.

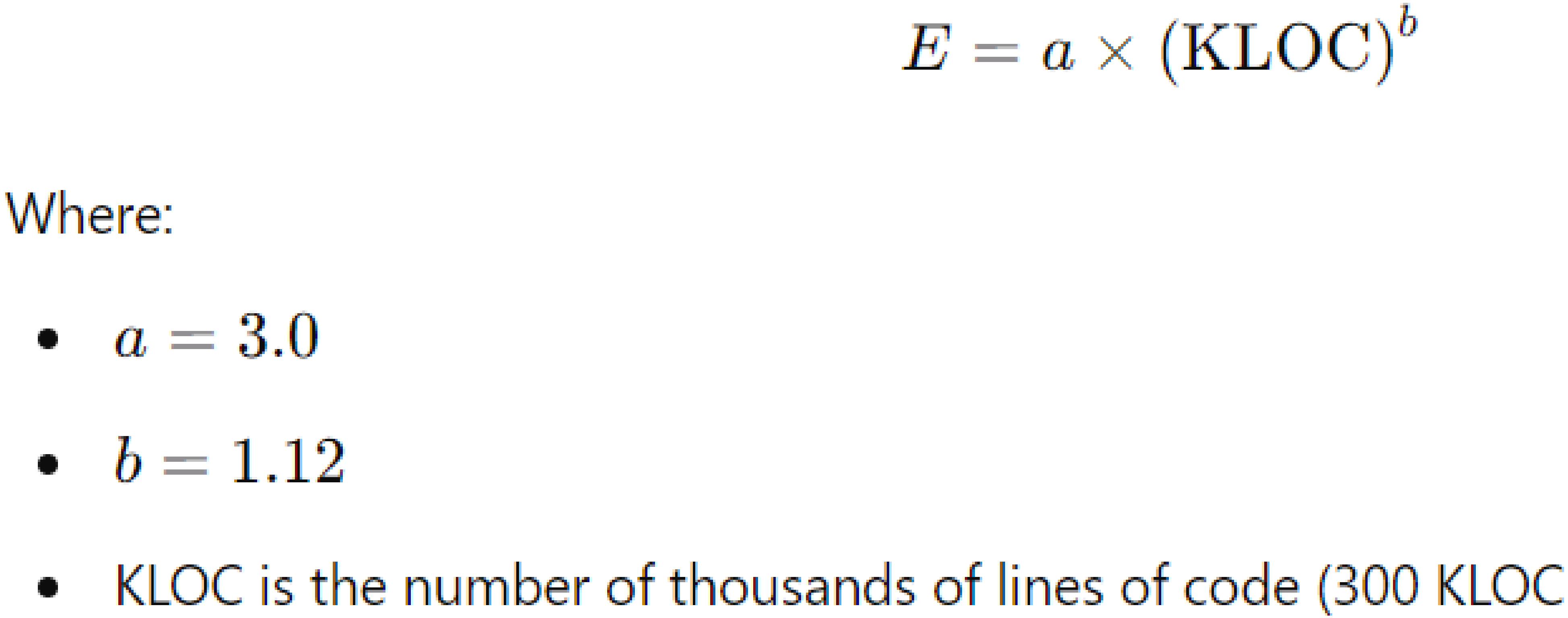
Find out Effort estimation, Development time and Person estimation.

❑Solution:

Effort (E) = **a*(KLOC)b** = 3.0*(300)1.12 = 1784.42 PM

Development Time (D) = **c(E)d** = 2.5(1784.42)0.35 = 34.35 Months(M)

Person Required (P) = **E/D** = 1784.42/34.35 = 51.9481 Persons ~52 Persons



$$E = a \times (\mathrm{KLOC})^b$$

Where:

- $a = 3.0$
- $b = 1.12$
- KLOC is the number of thousands of lines of code (300 KLOC

2. **Development Time (Tdev):**

$$Tdev = c \times (E)^d$$

Where:

- $c = 2.5$
- $d = 0.35$
- E is the effort calculated above.

3. **Number of Persons (P):**

$$P = \frac{E}{Tdev}$$

Where:

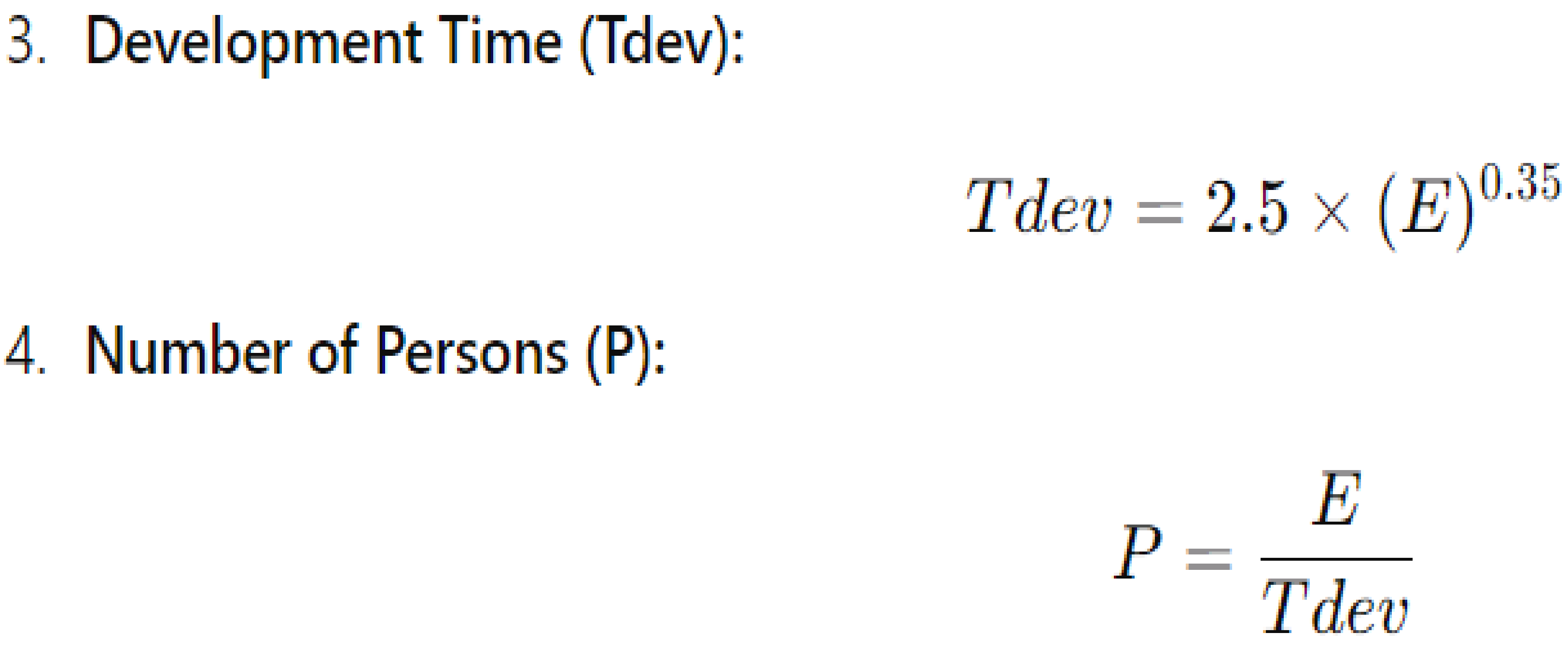
- E is the effort in person-months.
- $Tdev$ is the development time in months.

1. **KLOC (thousands of lines of code):**

$$KLOC = 300$$

2. **Effort (E):**

$$E = 3.0 \times (300)^{1.12}$$



3. Development Time (Tdev):

$$Tdev = 2.5 \times (E)^{0.35}$$

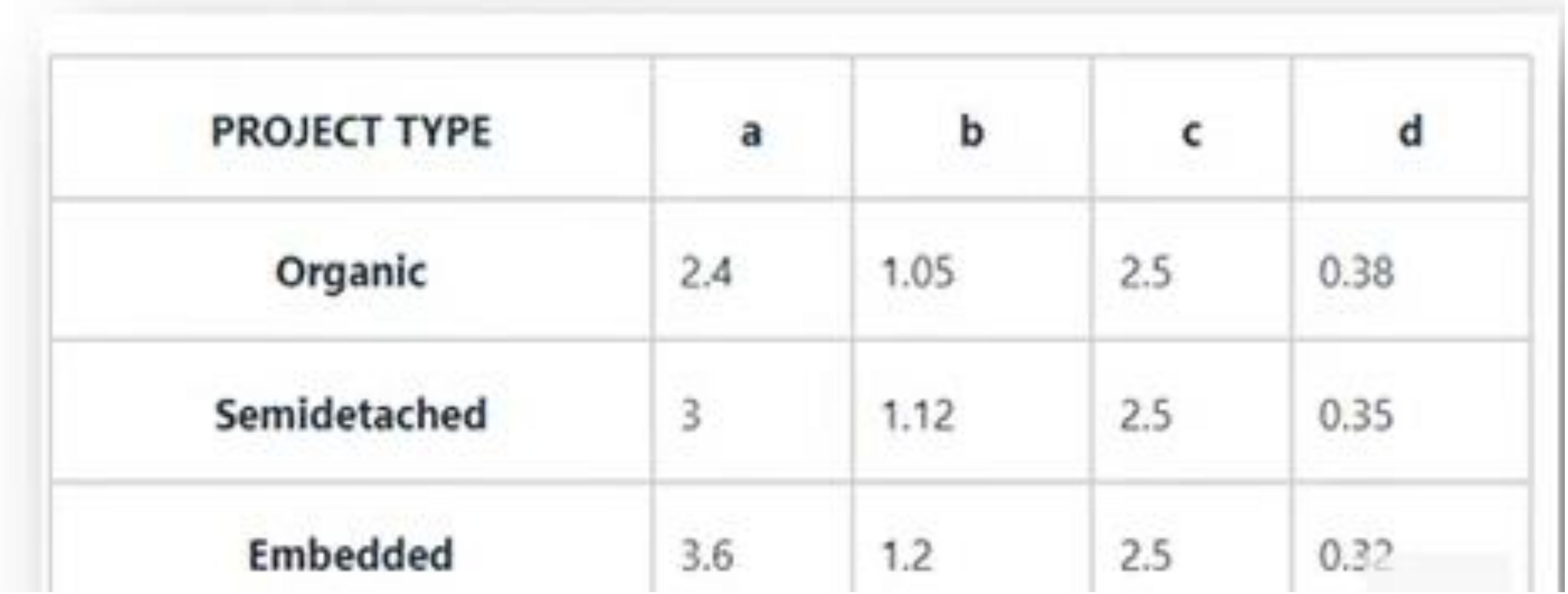
4. Number of Persons (P):

$$P = \frac{E}{Tdev}$$

For a software project with 300 KLOC in the semi-detached mode,

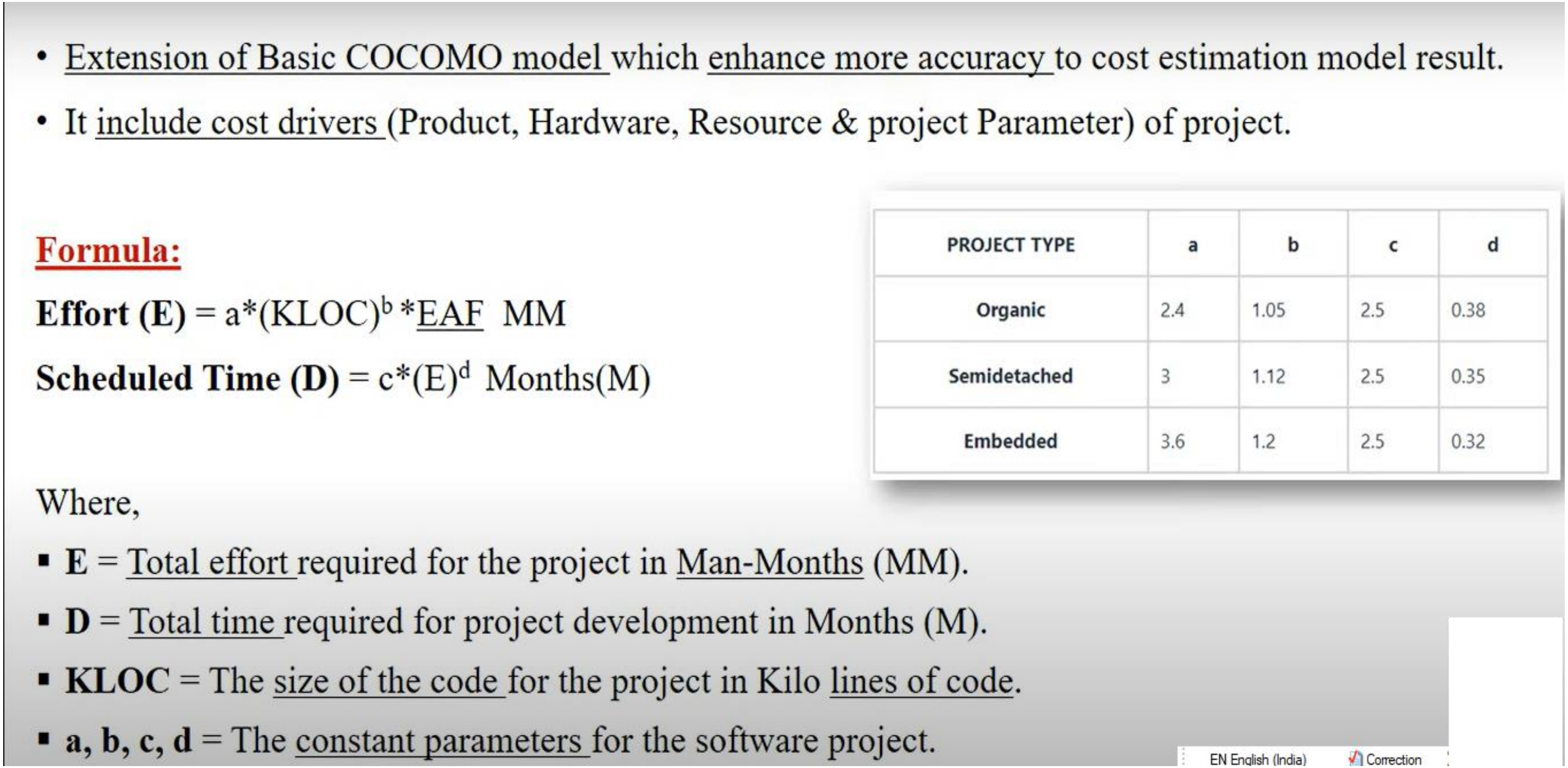
- **Effort (E)**: 1784.42 person-months
- **Development Time** : 34.35 months
- **Number of Persons (P)**: 51.94 persons = 52

Parameters for Basic COCOMO



PROJECT TYPE	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semidetached	3	1.12	2.5	0.35
Embedded	3.6	1.2	2.5	0.32

Intermediate COCOMO



- Extension of Basic COCOMO model which enhance more accuracy to cost estimation model result.
- It include cost drivers (Product, Hardware, Resource & project Parameter) of project.

Formula:

Effort (E) = $a*(KLOC)^{b}*EAF$ MM

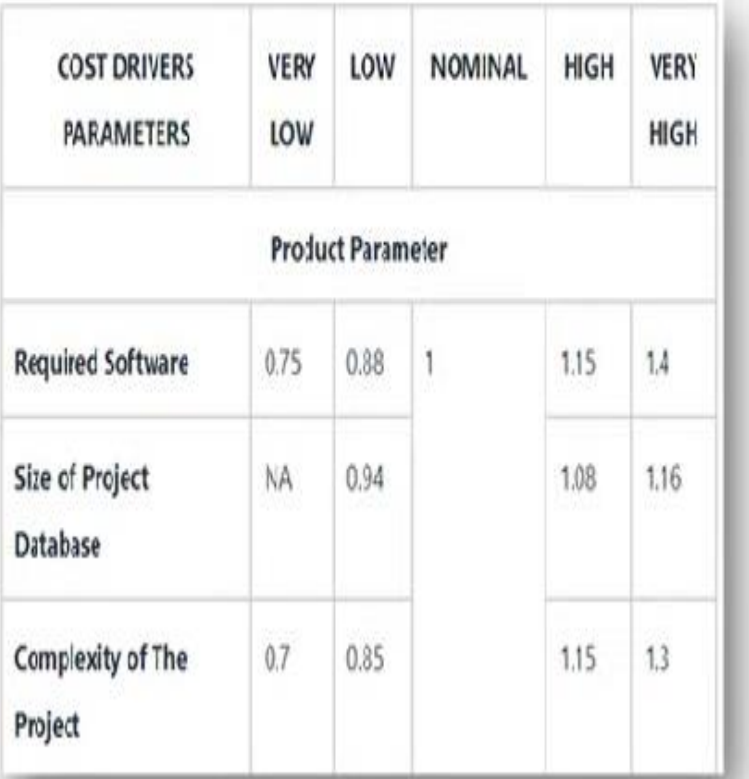
Scheduled Time (D) = $c*(E)^{d}$ Months(M)

PROJECT TYPE	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semidetached	3	1.12	2.5	0.35
Embedded	3.6	1.2	2.5	0.32

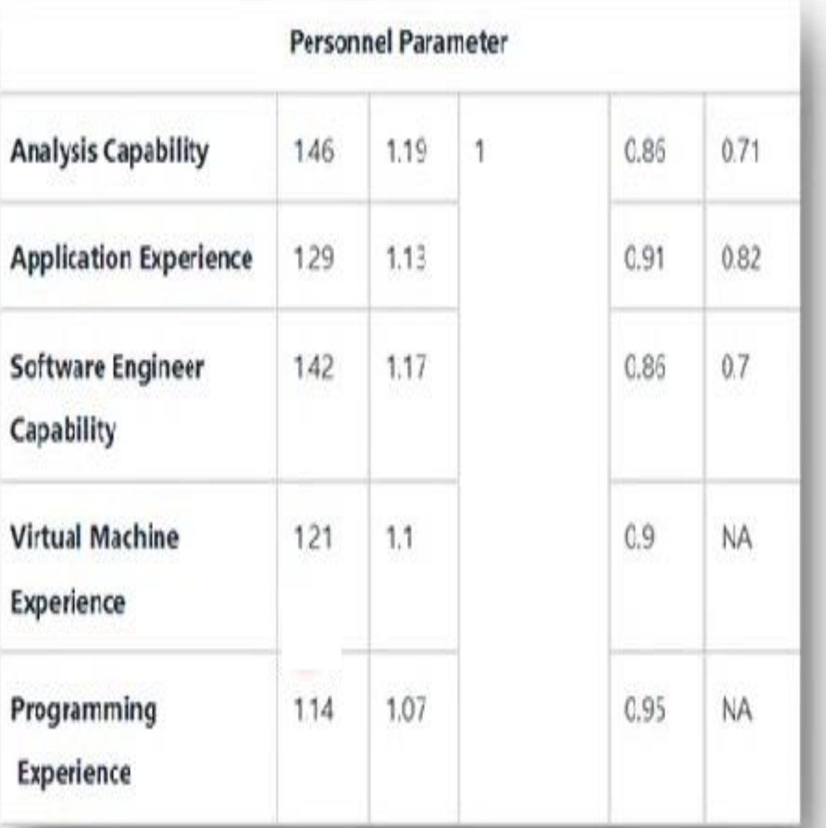
Where,

- **E** = Total effort required for the project in Man-Months (MM).
- **D** = Total time required for project development in Months (M).
- **KLOC** = The size of the code for the project in Kilo lines of code.
- **a, b, c, d** = The constant parameters for the software project.

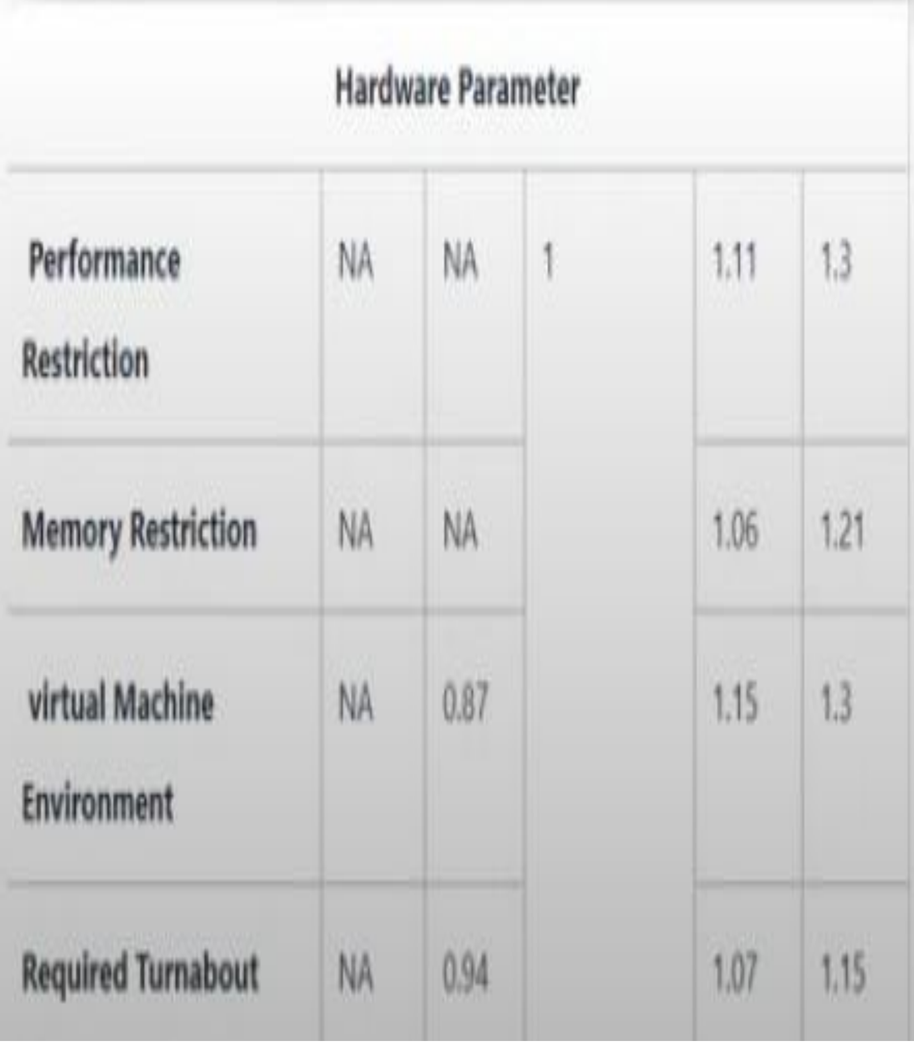
- **EAF:** Effort Adjustment Factor, which is calculated by multiplying the parameter values of different cost driver parameters. For ideal, the value is 1.



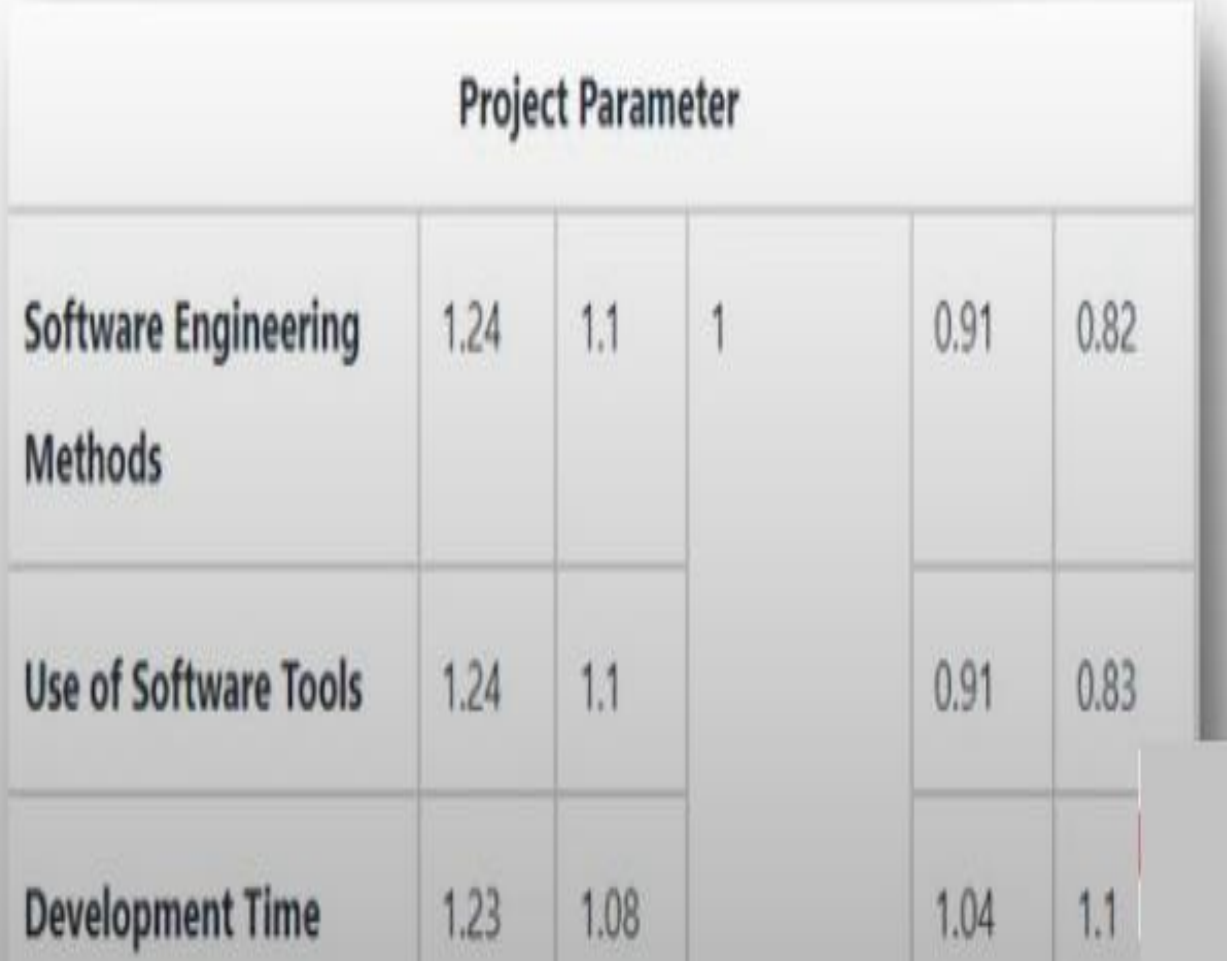
COST DRIVERS PARAMETERS	VERY LOW	LOW	NOMINAL	HIGH	VERY HIGH
Product Parameter					
Required Software	0.75	0.88	1	1.15	1.4
Size of Project Database	NA	0.94		1.08	1.16
Complexity of The Project	0.7	0.85		1.15	1.3



Personnel Parameter					
Analysis Capability	1.46	1.19	1	0.86	0.71
Application Experience	1.29	1.13		0.91	0.82
Software Engineer Capability	1.42	1.17		0.86	0.7
Virtual Machine Experience	1.21	1.1		0.9	NA
Programming Experience	1.14	1.07		0.95	NA



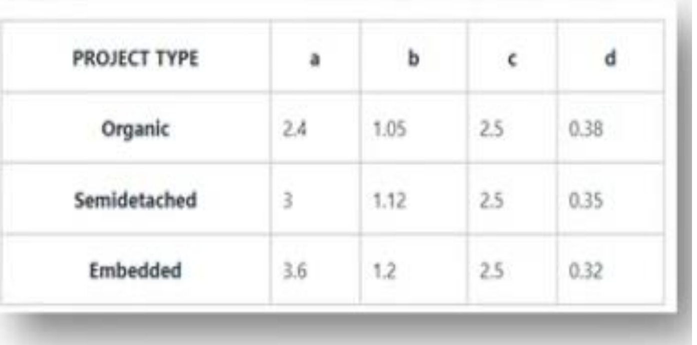
Hardware Parameter					
Performance Restriction	NA	NA	1	1.11	1.3
Memory Restriction	NA	NA		1.06	1.21
virtual Machine Environment	NA	0.87		1.15	1.3
Required Turnabout	NA	0.94		1.07	1.15



Project Parameter					
Software Engineering Methods	1.24	1.1	1	0.91	0.82
Use of Software Tools	1.24	1.1		0.91	0.83
Development Time	1.23	1.08		1.04	1.1

❑Problem Statement:

- For a given Semidetached project was estimated with a size of 300 KLOC. Calculate the Effort, Scheduled time for development by considering developer having high application experience and very low experience in programming.



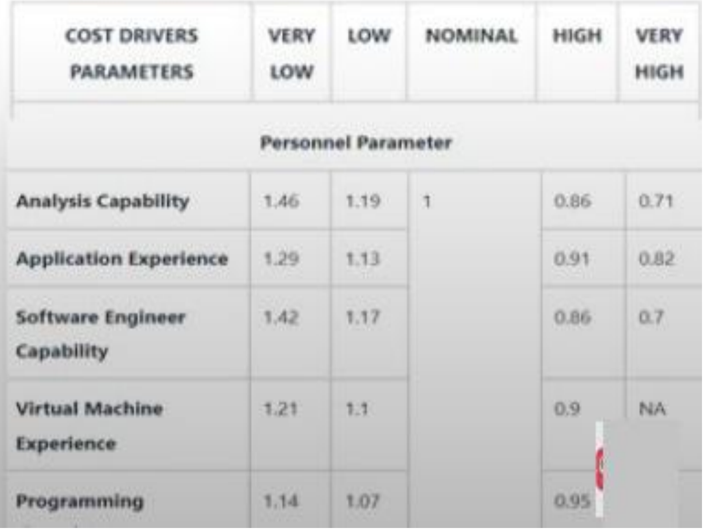
PROJECT TYPE	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semidetached	3	1.12	2.5	0.35
Embedded	3.6	1.2	2.5	0.32

❑Solution:

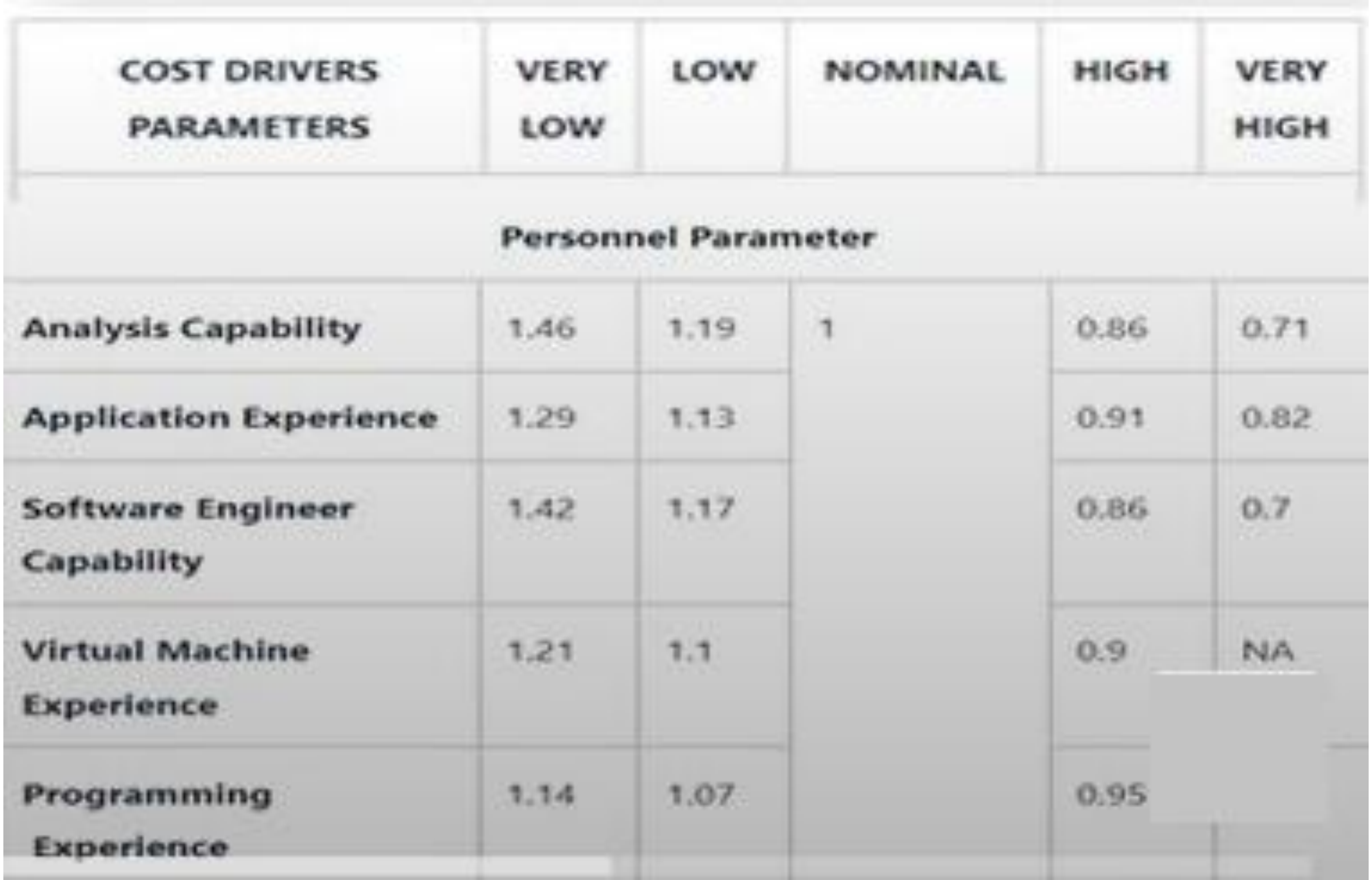
EAF = 0.82 *1.14 = 0.9348

Effort (E) = **a*(KLOC)^{b} *EAF** = 3.0*(300)$^{1.12}$ *0.9348 = 1668.07 MM

Scheduled Time (D) = **c*(E)^{d}** = 2.5*(1668.07)$^{0.35}$ = 33.55 Months(M)

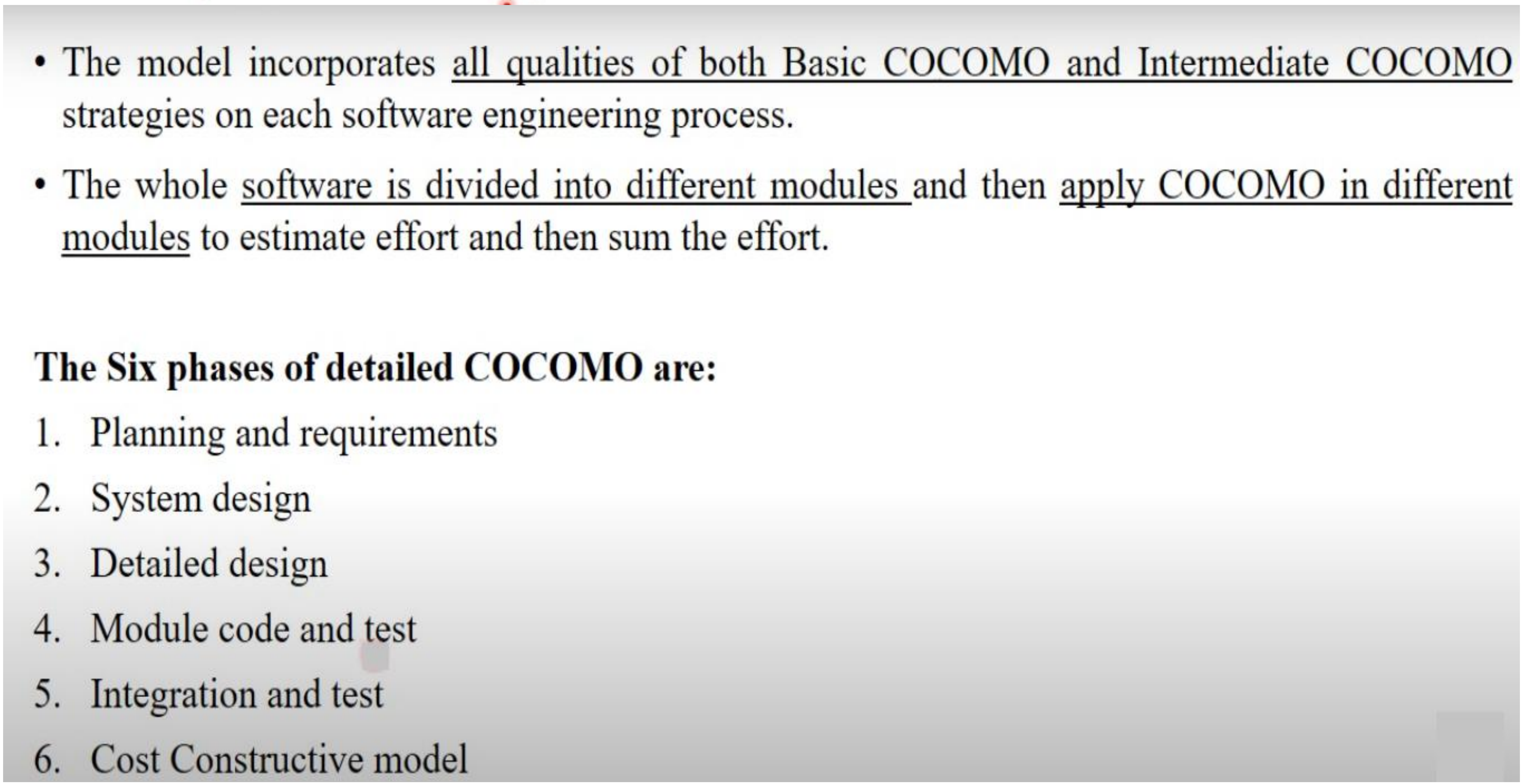


COST DRIVERS PARAMETERS	VERY LOW	LOW	NOMINAL	HIGH	VERY HIGH
Personnel Parameter					
Analysis Capability	1.46	1.19	1	0.86	0.71
Application Experience	1.29	1.13		0.91	0.82
Software Engineer Capability	1.42	1.17		0.86	0.7
Virtual Machine Experience	1.21	1.1		0.9	NA
Programming	1.14	1.07		0.95	



COST DRIVERS PARAMETERS	VERY LOW	LOW	NOMINAL	HIGH	VERY HIGH
Personnel Parameter					
Analysis Capability	1.46	1.19	1	0.86	0.71
Application Experience	1.29	1.13		0.91	0.82
Software Engineer Capability	1.42	1.17		0.86	0.7
Virtual Machine Experience	1.21	1.1		0.9	NA
Programming Experience	1.14	1.07		0.95	

Type 3: Detailed / Complete COCOMO Model



- The model incorporates all qualities of both Basic COCOMO and Intermediate COCOMO strategies on each software engineering process.
- The whole software is divided into different modules and then apply COCOMO in different modules to estimate effort and then sum the effort.

The Six phases of detailed COCOMO are:

1. Planning and requirements
2. System design
3. Detailed design
4. Module code and test
5. Integration and test
6. Cost Constructive model

❑Problem Statement:

A distributed Management Information System (MIS) product for an organization having offices at several places across the country can have the following sub-components:

- Database part
- Graphical User Interface (GUI) part
- Communication part

❑Solution:

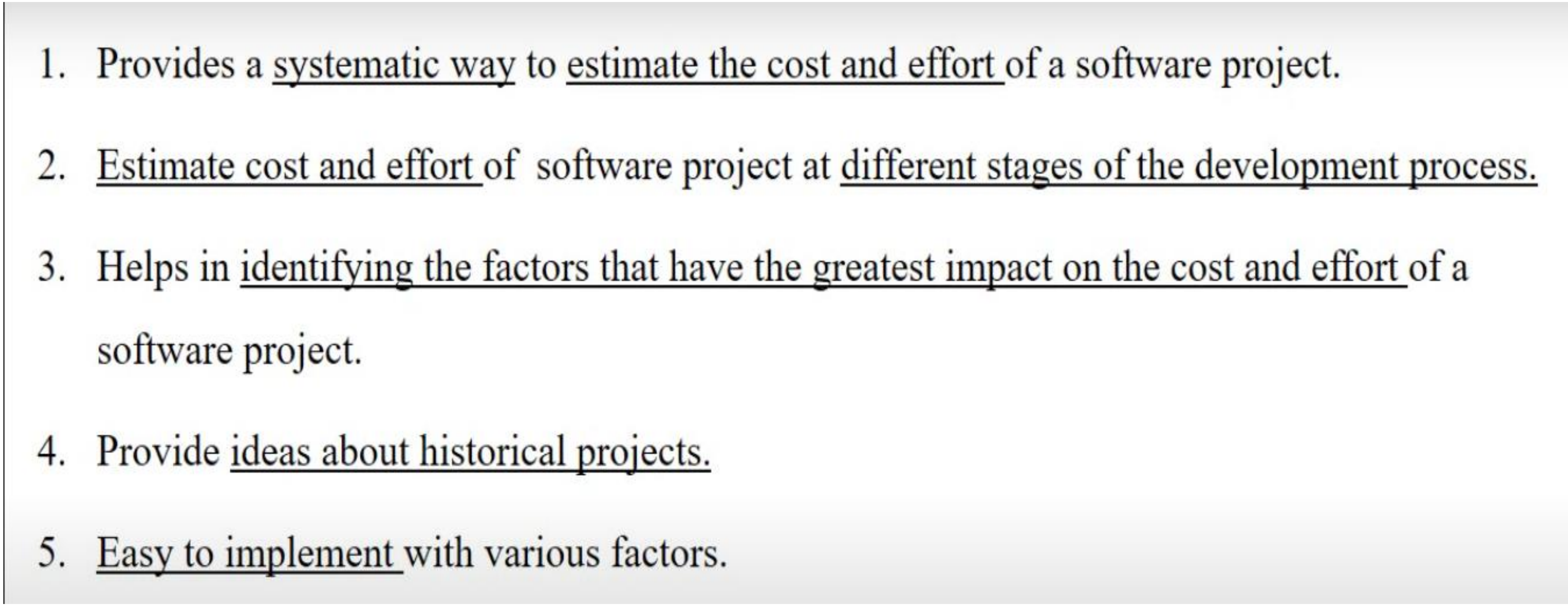
✓The communication part can be considered as Embedded software.

✓The database part could be Semi-detached software,

✓The GUI part Organic software.

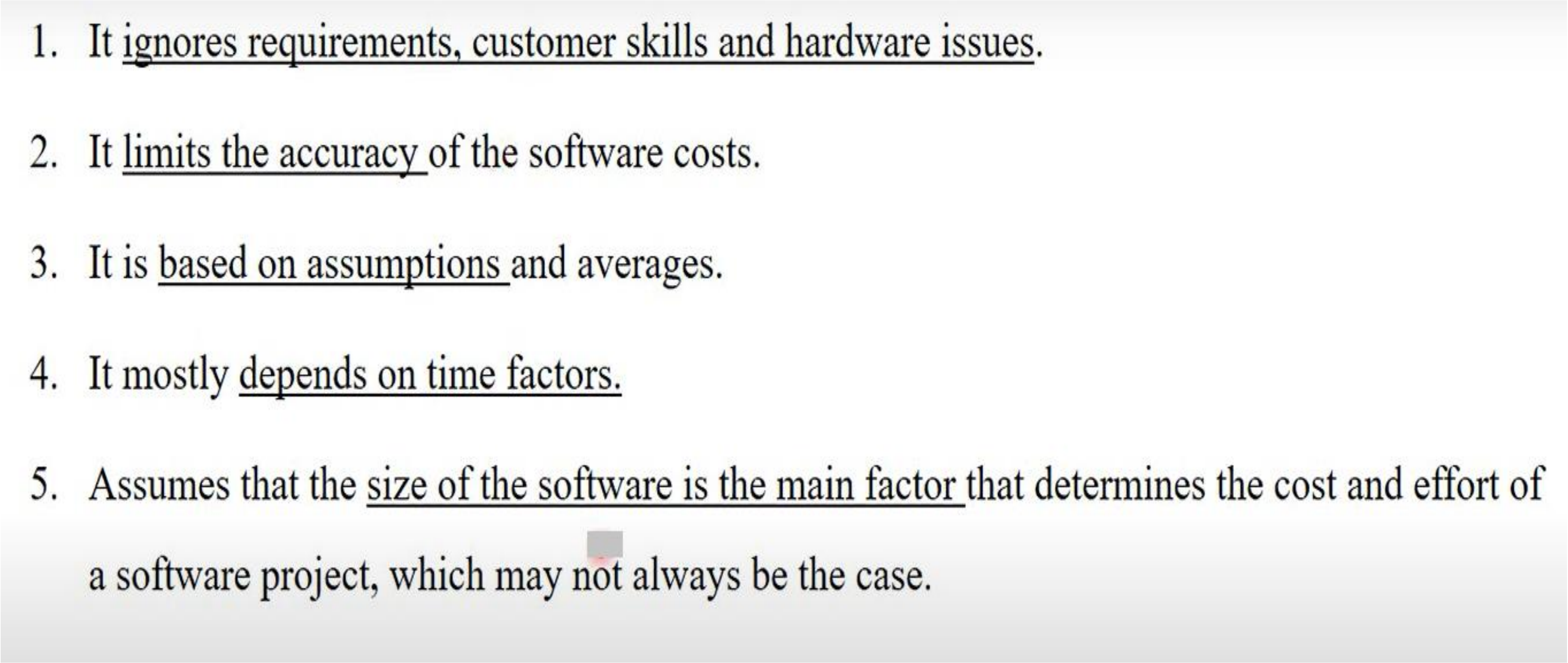
The costs for these three components can be estimated separately and summed up to give the overall cost of the system.

Advantages



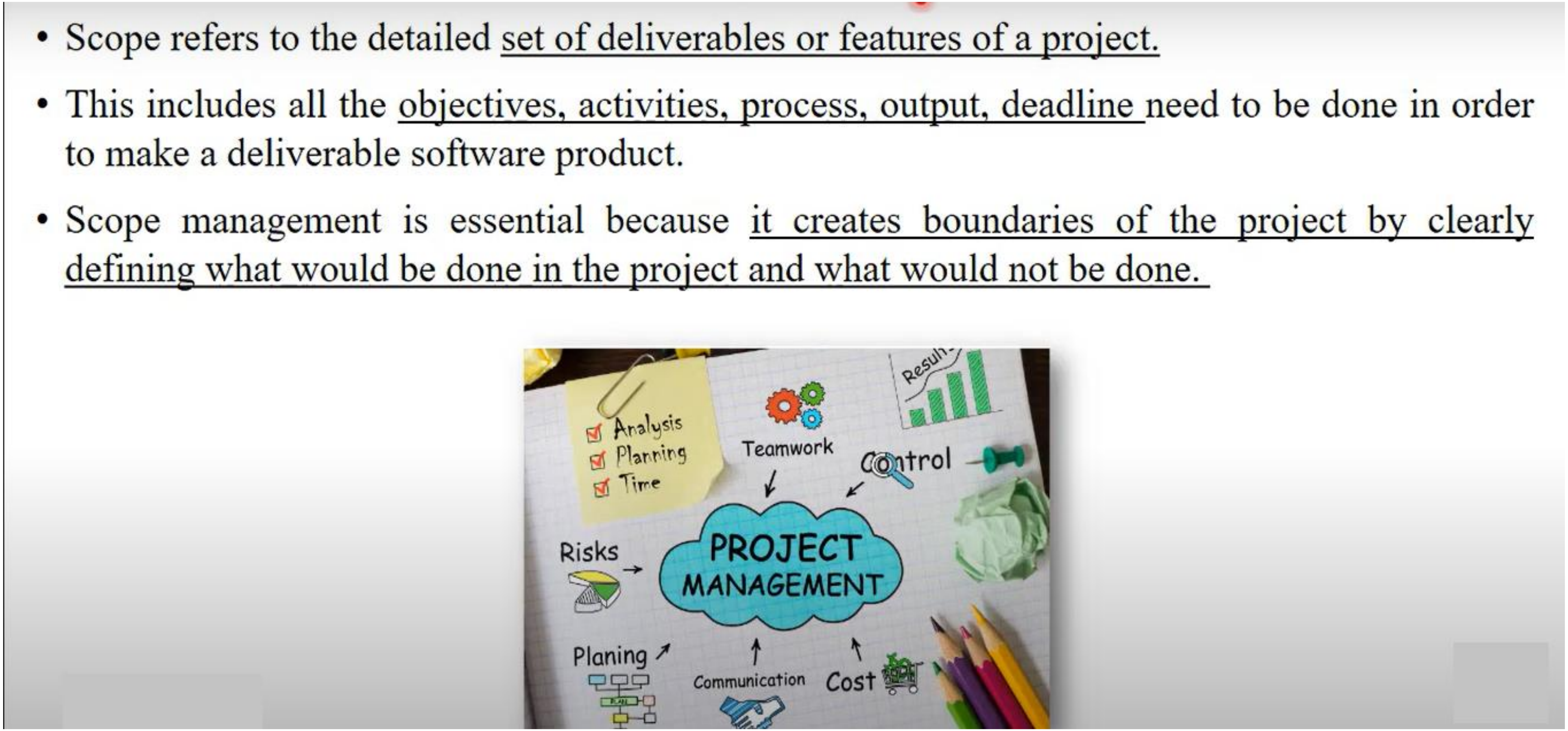
1. Provides a systematic way to estimate the cost and effort of a software project.
2. Estimate cost and effort of software project at different stages of the development process.
3. Helps in identifying the factors that have the greatest impact on the cost and effort of a software project.
4. Provide ideas about historical projects.
5. Easy to implement with various factors.

Disadvantages



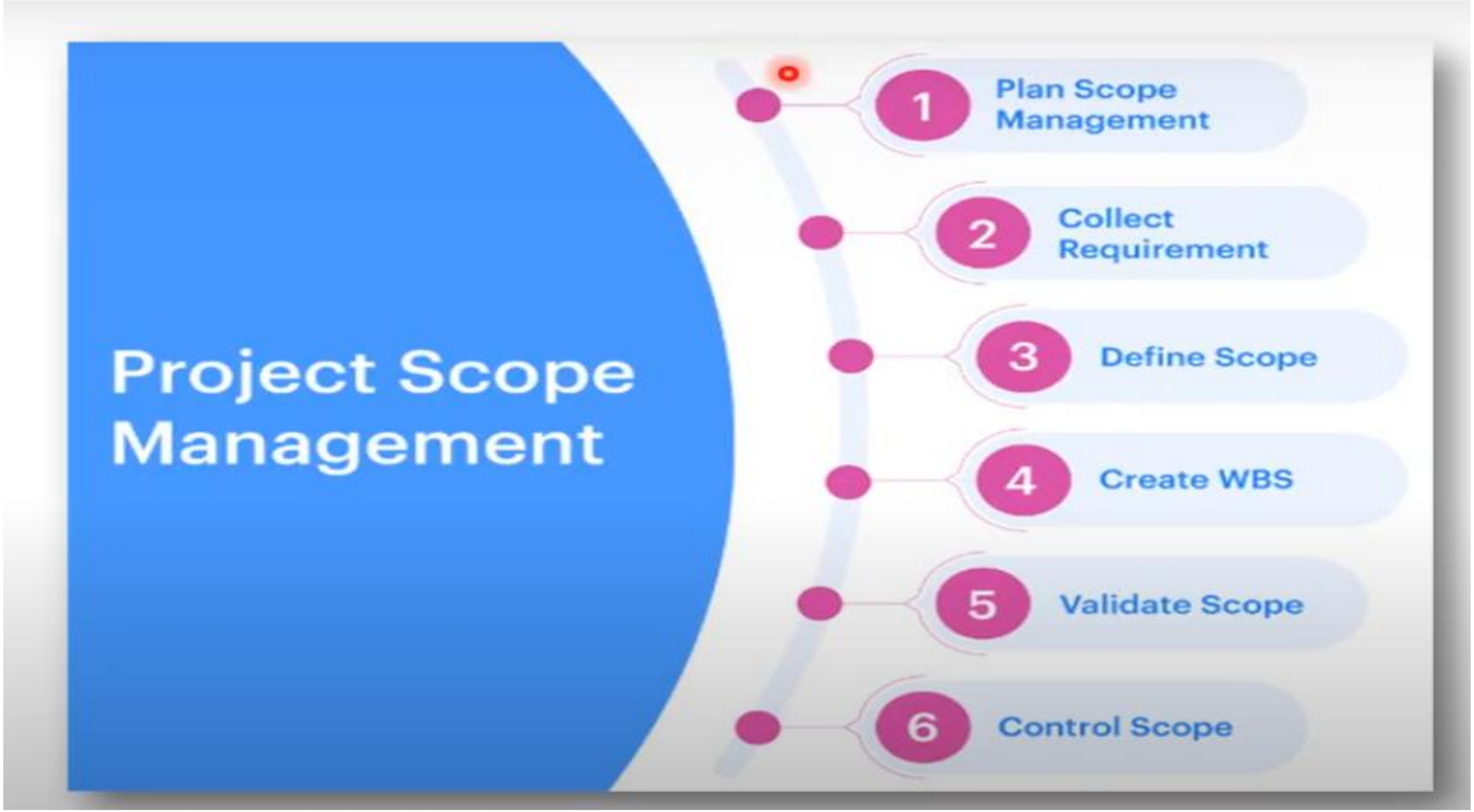
1. It ignores requirements, customer skills and hardware issues.
2. It limits the accuracy of the software costs.
3. It is based on assumptions and averages.
4. It mostly depends on time factors.
5. Assumes that the size of the software is the main factor that determines the cost and effort of a software project, which may not always be the case.

Project Scope / Project Scope Management



- Scope refers to the detailed set of deliverables or features of a project.
- This includes all the objectives, activities, process, output, deadline need to be done in order to make a deliverable software product.
- Scope management is essential because it creates boundaries of the project by clearly defining what would be done in the project and what would not be done.

Steps for Project Scope Management



Project Scope Management

1 Plan Scope Management

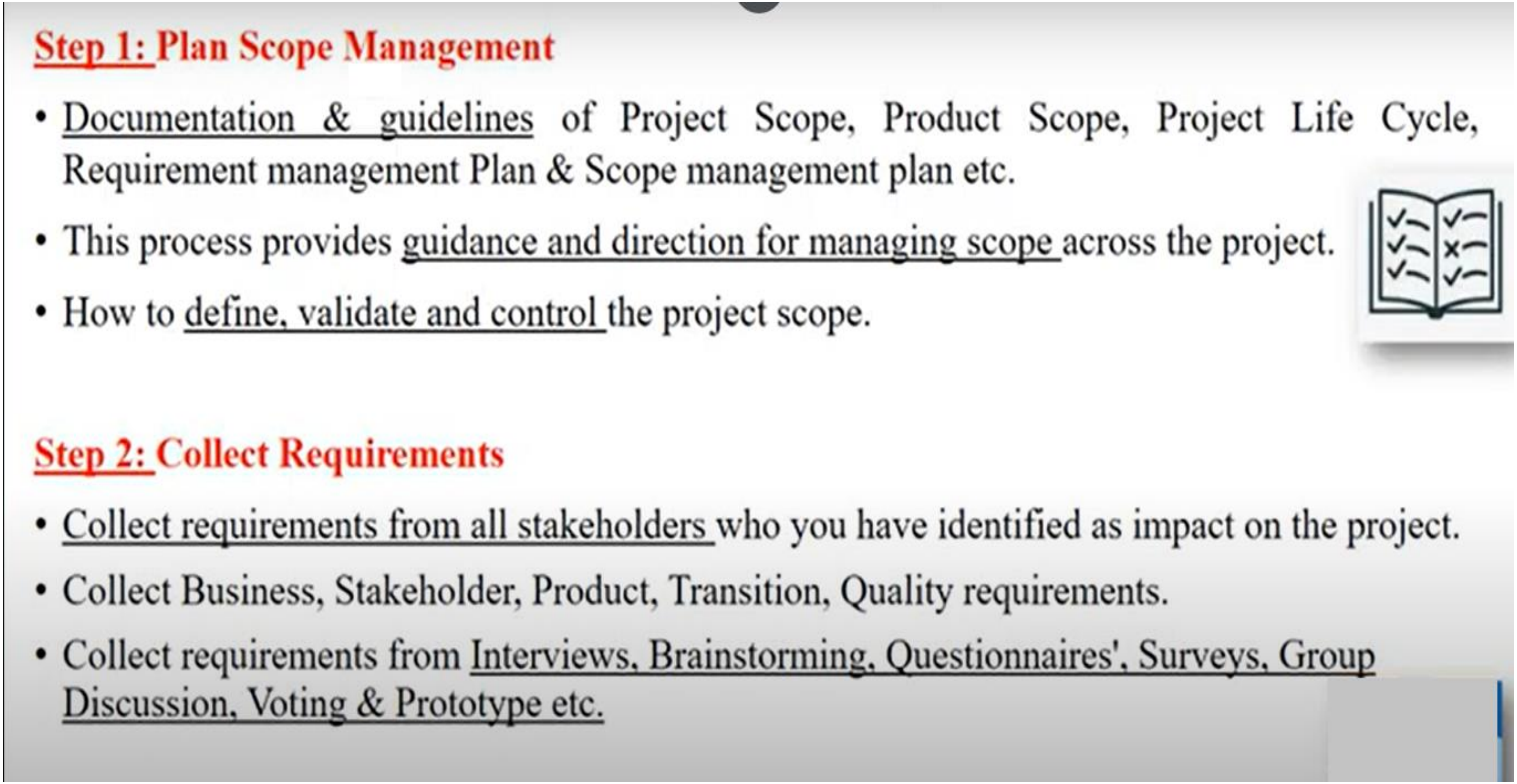
2 Collect Requirement

3 Define Scope

4 Create WBS

5 Validate Scope

6 Control Scope



Step 1: Plan Scope Management

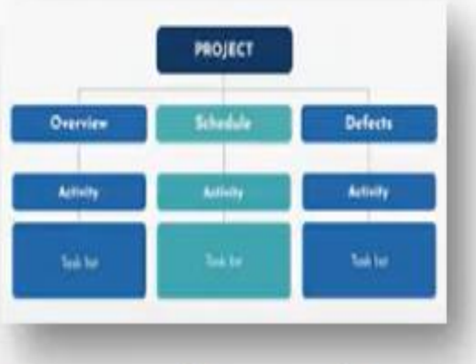
- Documentation & guidelines of Project Scope, Product Scope, Project Life Cycle, Requirement management Plan & Scope management plan etc.
- This process provides guidance and direction for managing scope across the project.
- How to define, validate and control the project scope.

Step 2: Collect Requirements

- Collect requirements from all stakeholders who you have identified as impact on the project.
- Collect Business, Stakeholder, Product, Transition, Quality requirements.
- Collect requirements from Interviews, Brainstorming, Questionnaires', Surveys, Group Discussion, Voting & Prototype etc.

Step 3: Defining Scope

- Define the project scope by identifying Project objectives, Goals, Sub-phases, Tasks, Budget Resources & Schedule.
- Identify Human Recourses like Inter-personnel and team skills.
- This ensures that the client, stakeholders, senior management, project manager & team members are all aware of what is expected.



Step 4: Create WBS

- The Work Breakdown Structure involves subdividing project deliverables into smaller units.
- Break down project into phases, including the priority tasks required in order to complete each phase.

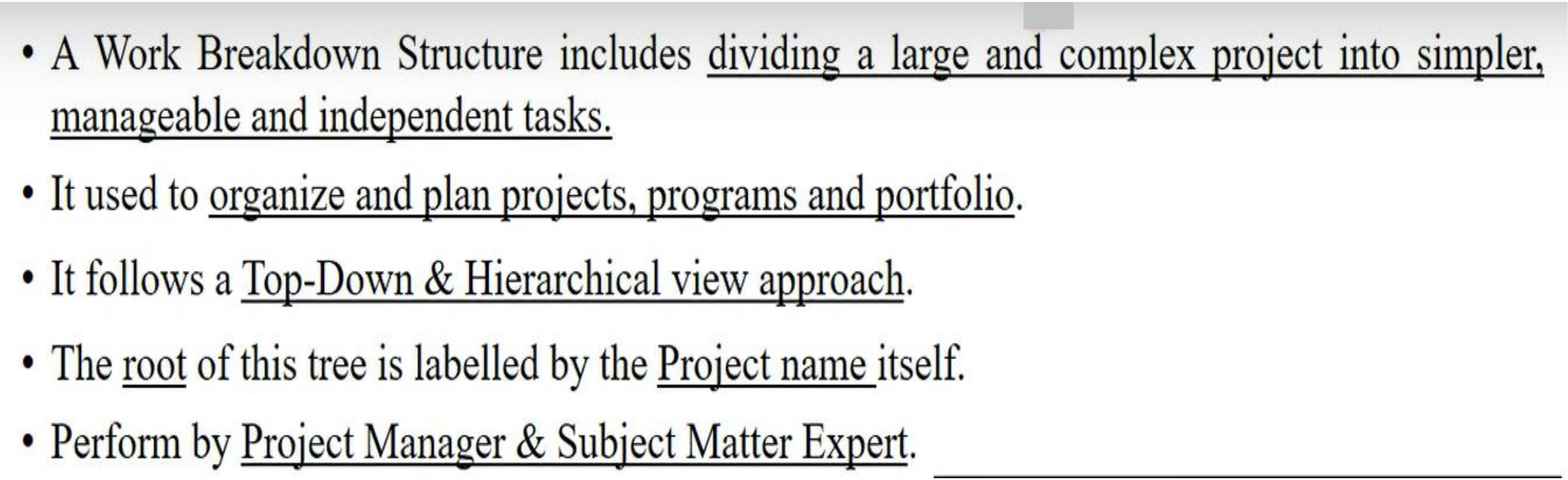
Step 5: Verify Scope

- The Validate Scope process focuses mainly on customer acceptance.
- It is when the project customer formally accepts all project deliverables & end of each phase.
- During the process, the customer gives feedback on the work that was performed.

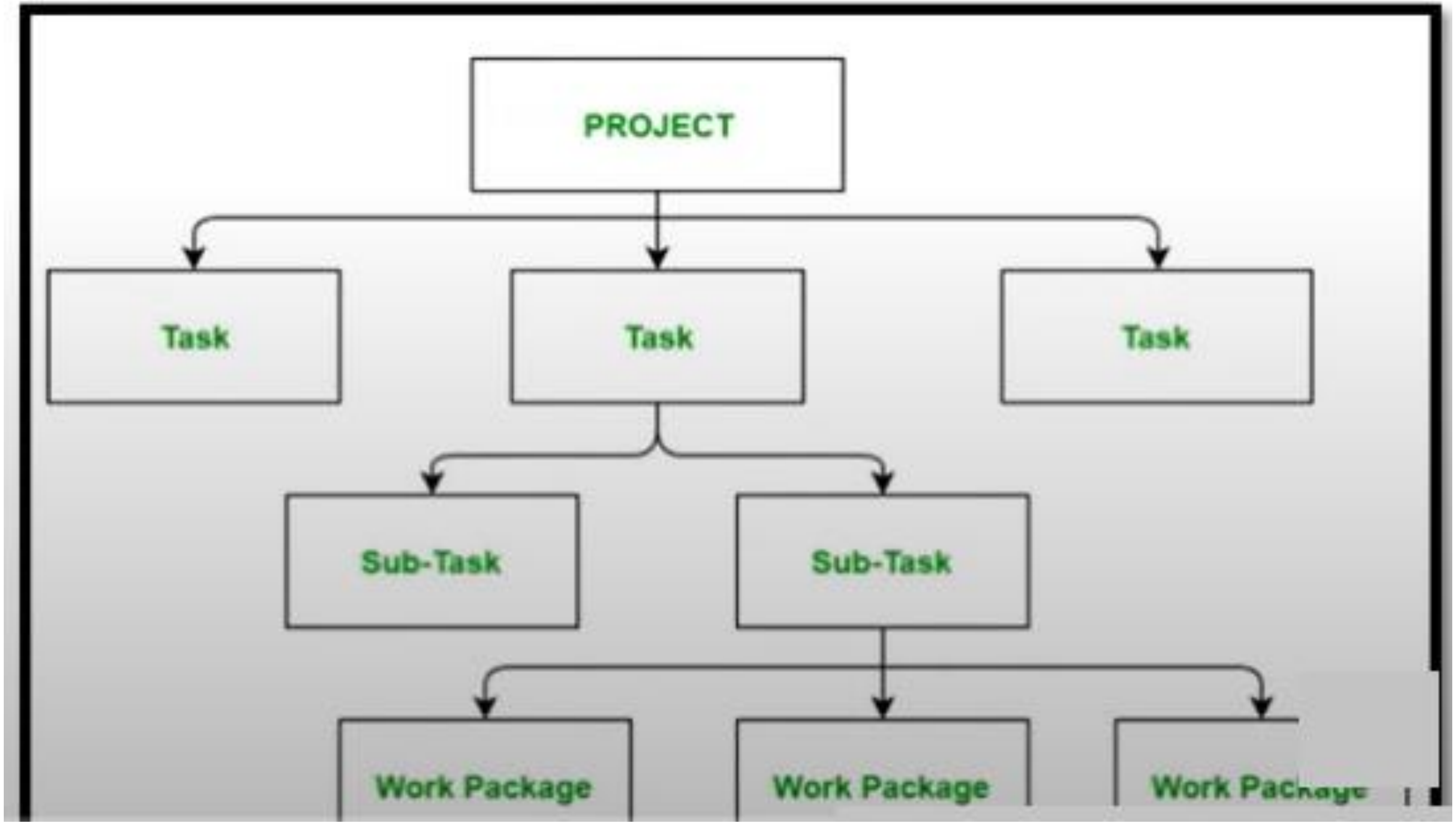
Step 6: Control Scope

- It involves monitoring the status of the project and managing changes in scope.
- Efficient in dealing with Time & Cost management.
- Calculate how does it impact the project and its process.

WBS



- A Work Breakdown Structure includes dividing a large and complex project into simpler, manageable and independent tasks.
- It used to organize and plan projects, programs and portfolio.
- It follows a Top-Down & Hierarchical view approach.
- The root of this tree is labelled by the Project name itself.
- Perform by Project Manager & Subject Matter Expert.


PROJECT
Task
Task
Task
Sub-Task
Sub-Task
Work Package
Work Package
Work Package

Step 1: Project managers decide project name at top / root of WBS.

Step 2: Project managers identifies the main deliverables of the project.

Step 3: These main deliverables are broke down into smaller higher-level tasks.

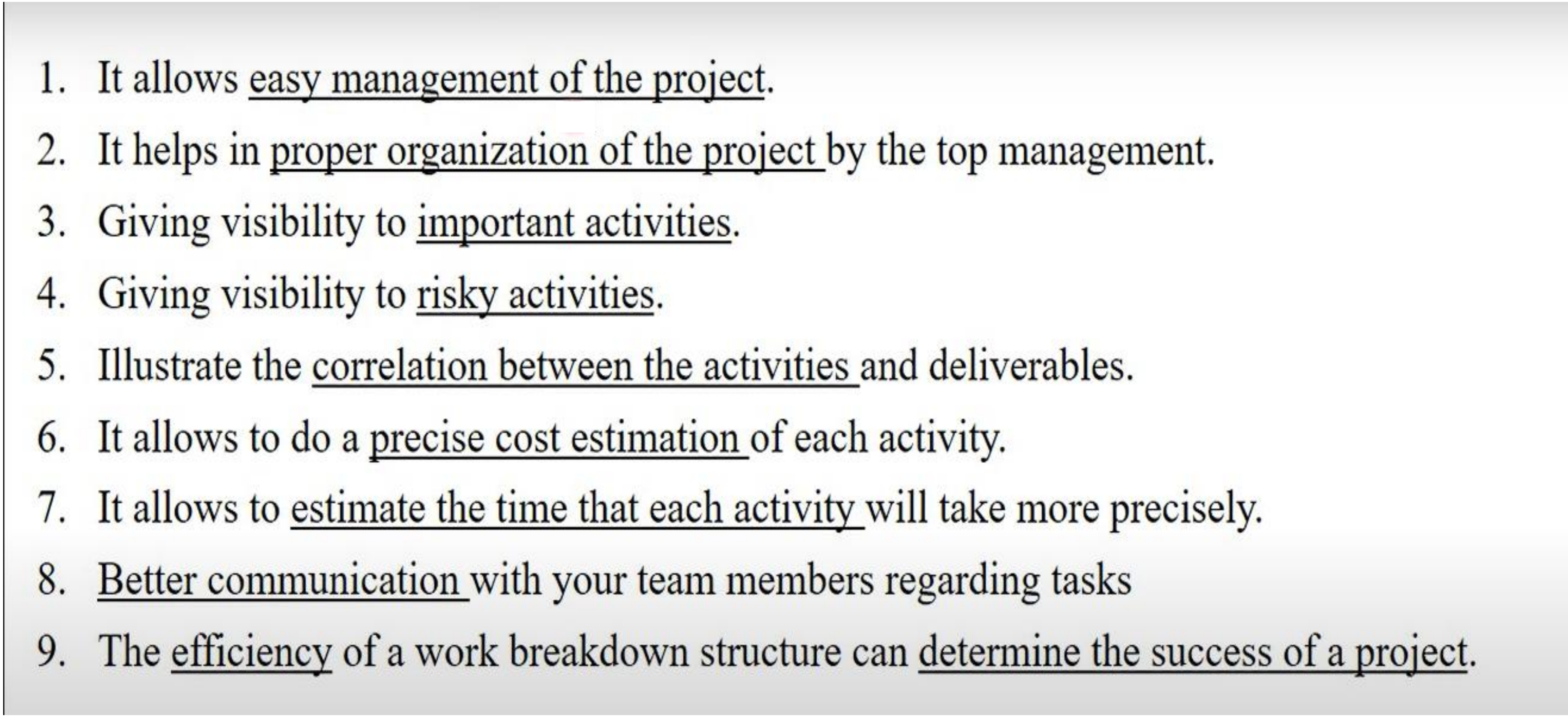
Step 4: This complete process is done recursively to produce much smaller independent tasks.

Step 5: Choose Task Owner. They need to get the job done.

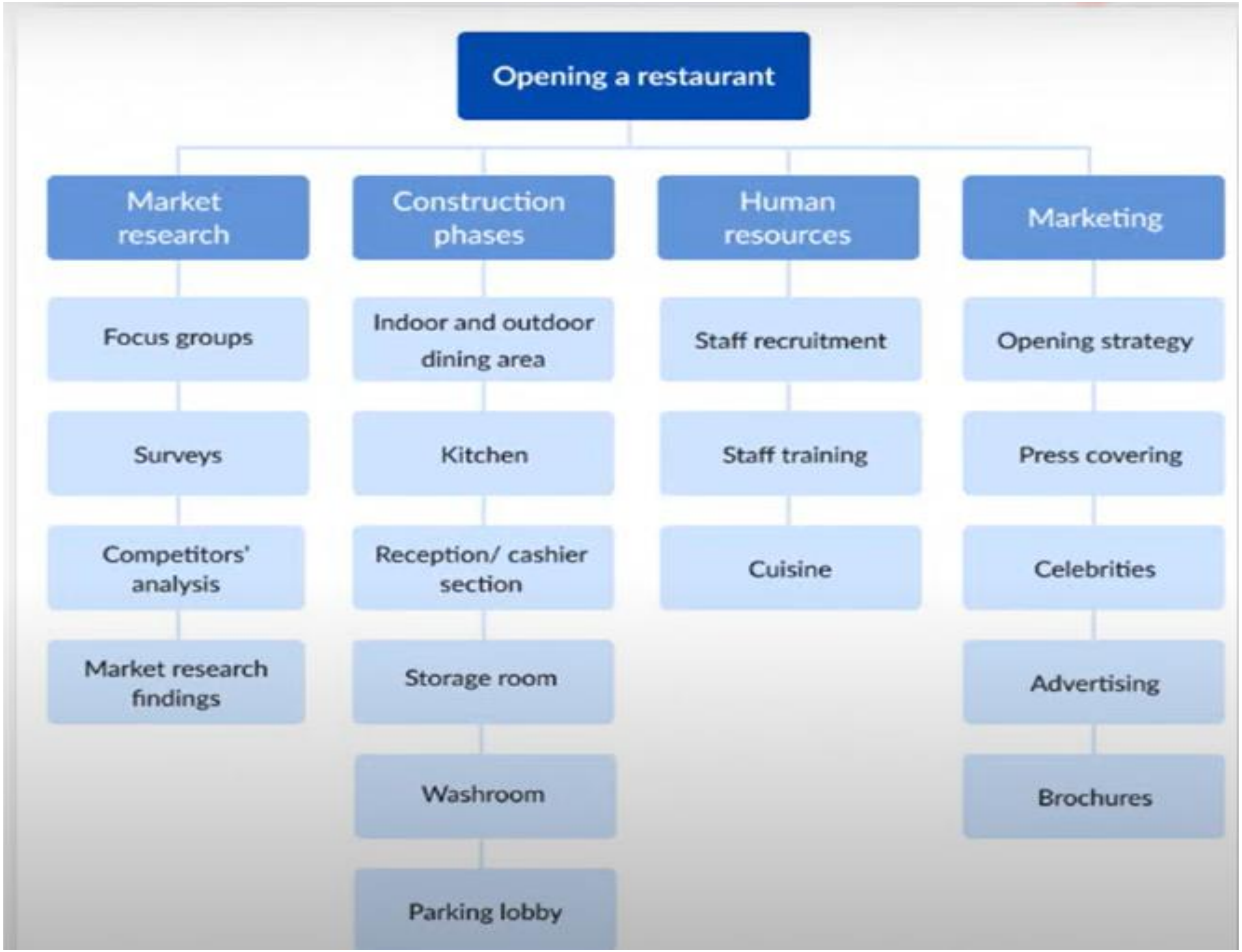
➢Depends on project manager that up to which level of detail they want to break down project.

➢Lowest level tasks are most simplest & independent tasks it takes less than weeks of work.

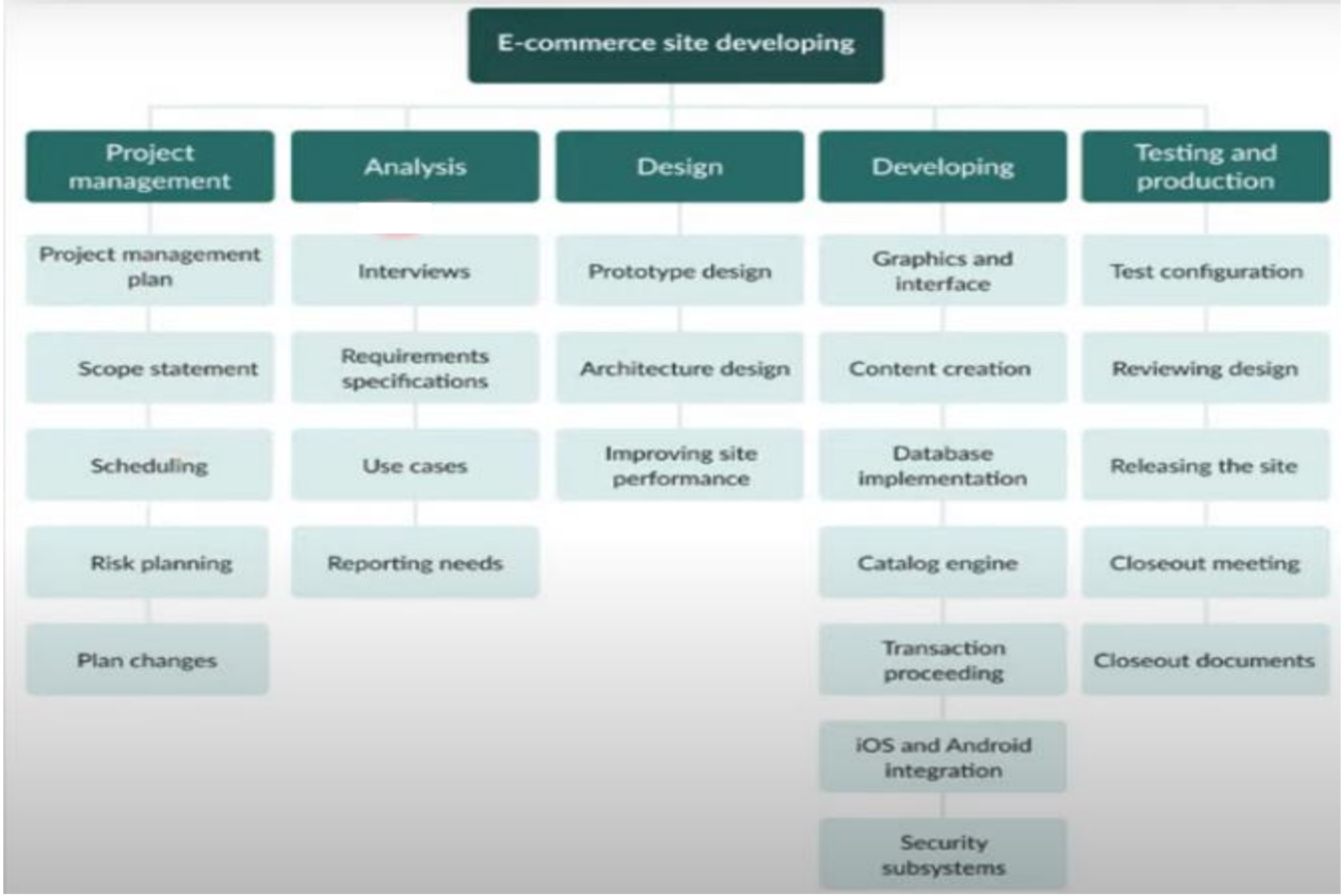
Why it is Required



1. It allows easy management of the project.
2. It helps in proper organization of the project by the top management.
3. Giving visibility to important activities.
4. Giving visibility to risky activities.
5. Illustrate the correlation between the activities and deliverables.
6. It allows to do a precise cost estimation of each activity.
7. It allows to estimate the time that each activity will take more precisely.
8. Better communication with your team members regarding tasks
9. The efficiency of a work breakdown structure can determine the success of a project.



- Opening a restaurant
  - Market research
    - Focus groups
    - Surveys
    - Competitors' analysis
    - Market research findings
  - Construction phases
    - Indoor and outdoor dining area
    - Kitchen
    - Reception/ cashier section
    - Storage room
    - Washroom
    - Parking lobby
  - Human resources
    - Staff recruitment
    - Staff training
    - Cuisine
  - Marketing
    - Opening strategy
    - Press covering
    - Celebrities
    - Advertising
    - Brochures

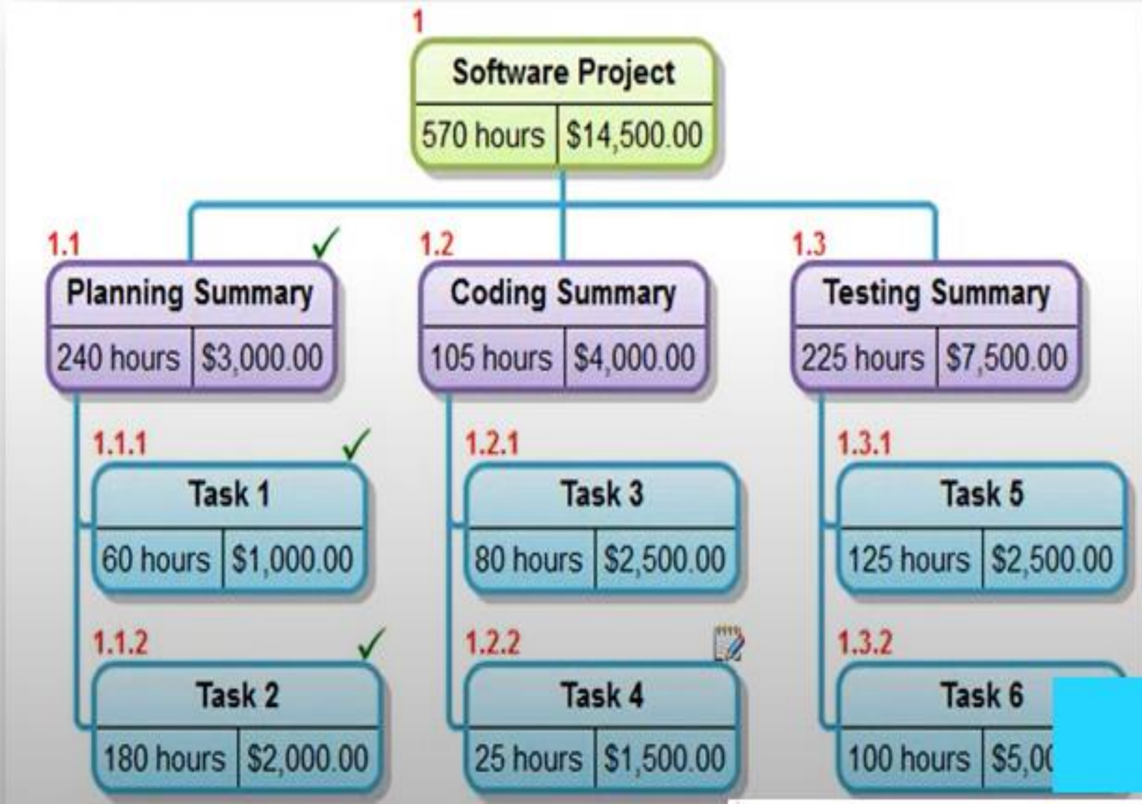


- E-commerce site developing
  - Project management
    - Project management plan
    - Scope statement
    - Scheduling
    - Risk planning
    - Plan changes
  - Analysis
    - Interviews
    - Requirements specifications
    - Use cases
    - Reporting needs
  - Design
    - Prototype design
    - Architecture design
    - Improving site performance
  - Developing
    - Graphics and interface
    - Content creation
    - Database implementation
    - Catalog engine
    - Transaction proceeding
    - iOS and Android integration
    - Security subsystems
  - Testing and production
    - Test configuration
    - Reviewing design
    - Releasing the site
    - Closeout meeting
    - Closeout documents

- Many free and paid software tools can help you create WBS for your projects

➢Common WBS Software tools are:

1. EdrawMax
2. Lucidchart
3. SmartDraw
4. Visual Paradigm
5. MindView
6. Creately



1 Software Project
570 hours | $14,500.00

1.1 Planning Summary ✓
240 hours | $3,000.00

1.1.1 Task 1 ✓
60 hours | $1,000.00

1.1.2 Task 2 ✓
180 hours | $2,000.00

1.2 Coding Summary
105 hours | $4,000.00

1.2.1 Task 3
80 hours | $2,500.00

1.2.2 Task 4
25 hours | $1,500.00

1.3 Testing Summary
225 hours | $7,500.00

1.3.1 Task 5
125 hours | $2,500.00

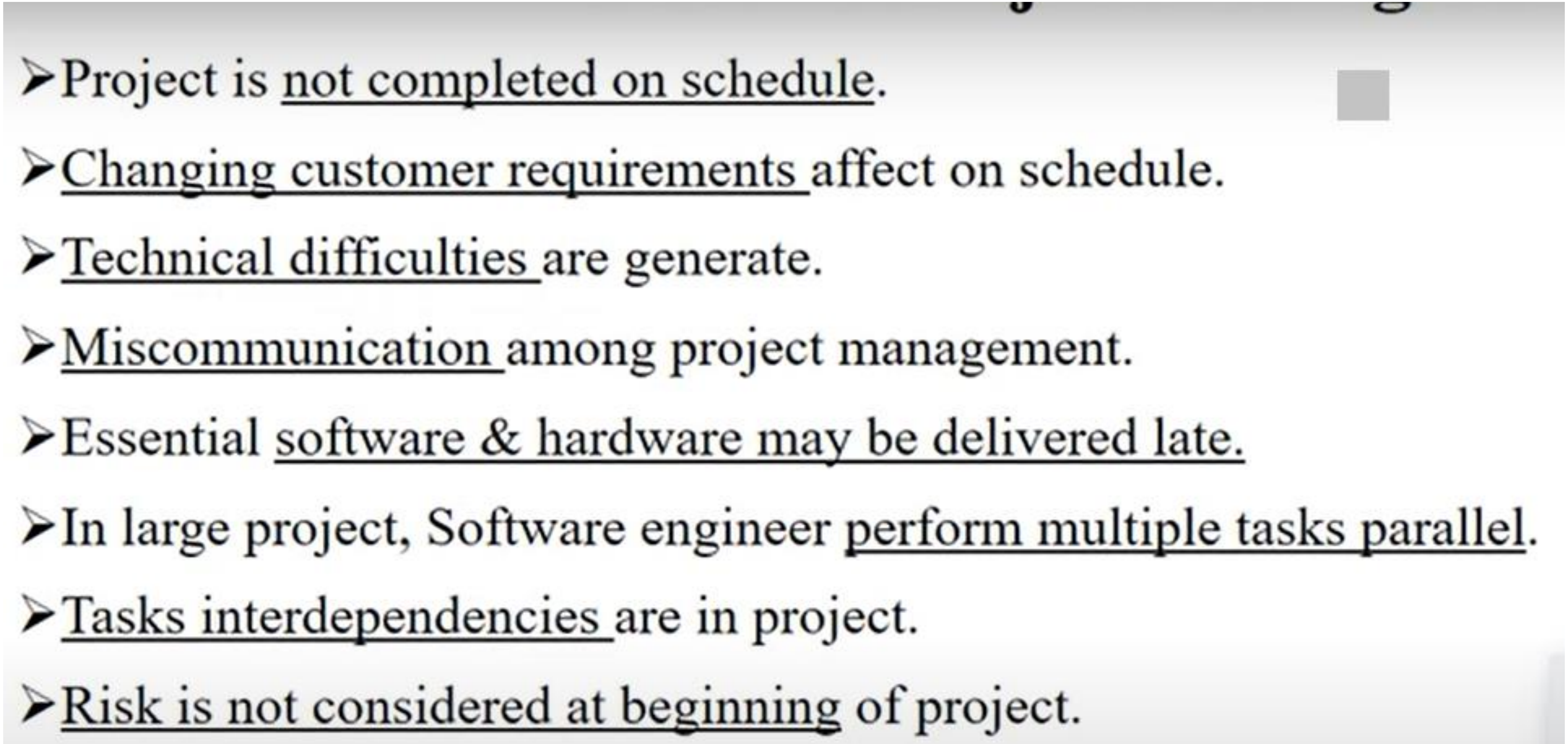
1.3.2 Task 6
100 hours | $5,0

Project Scheduling

- Software project scheduling is an activity that distributes estimated effort across the planned project duration by allocating the effort to specific software engineering tasks.

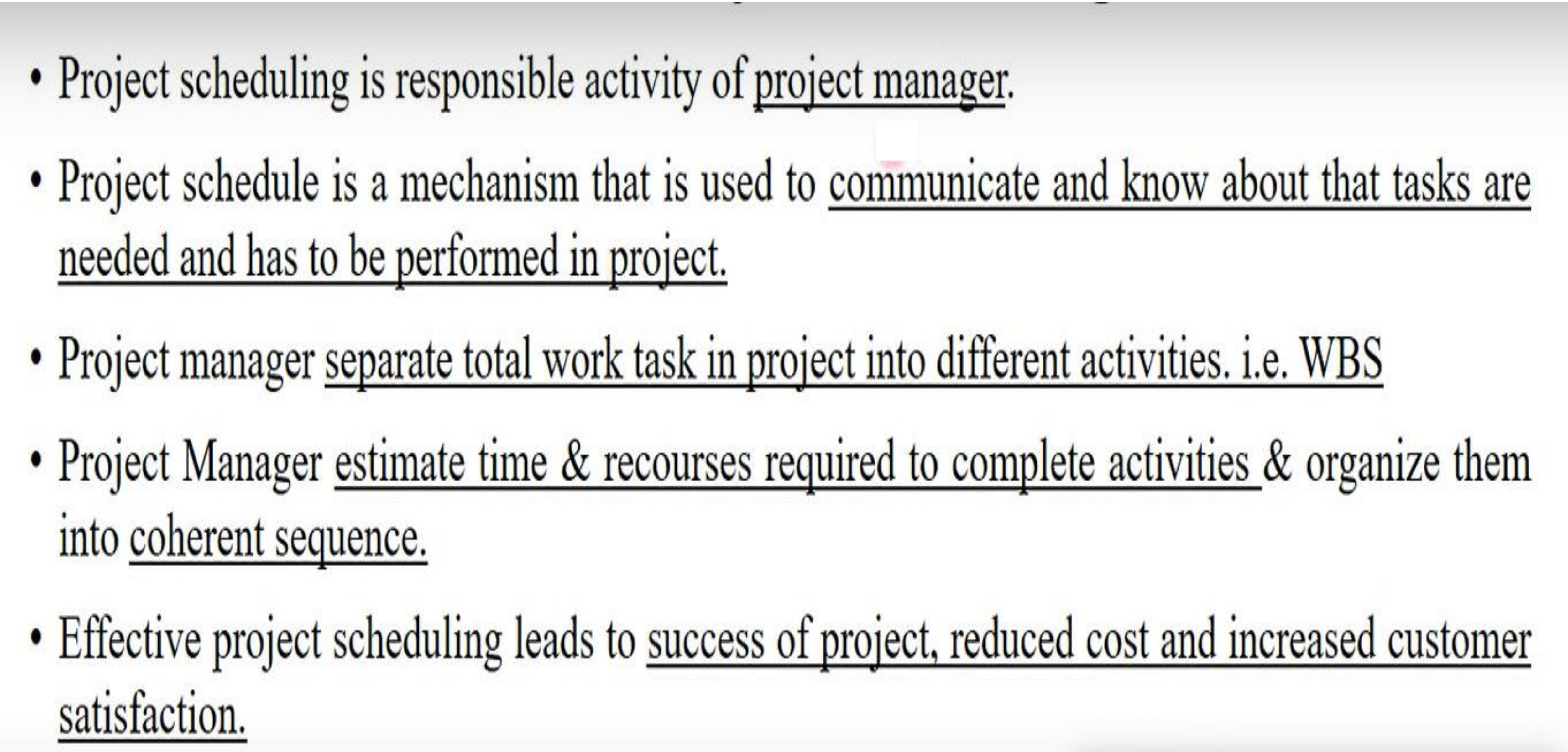
- Scheduling is the culmination of a planning activity that is a primary component of software project management. When combined with estimation methods and risk analysis, scheduling establishes a road map for the project manager

Issues in project management



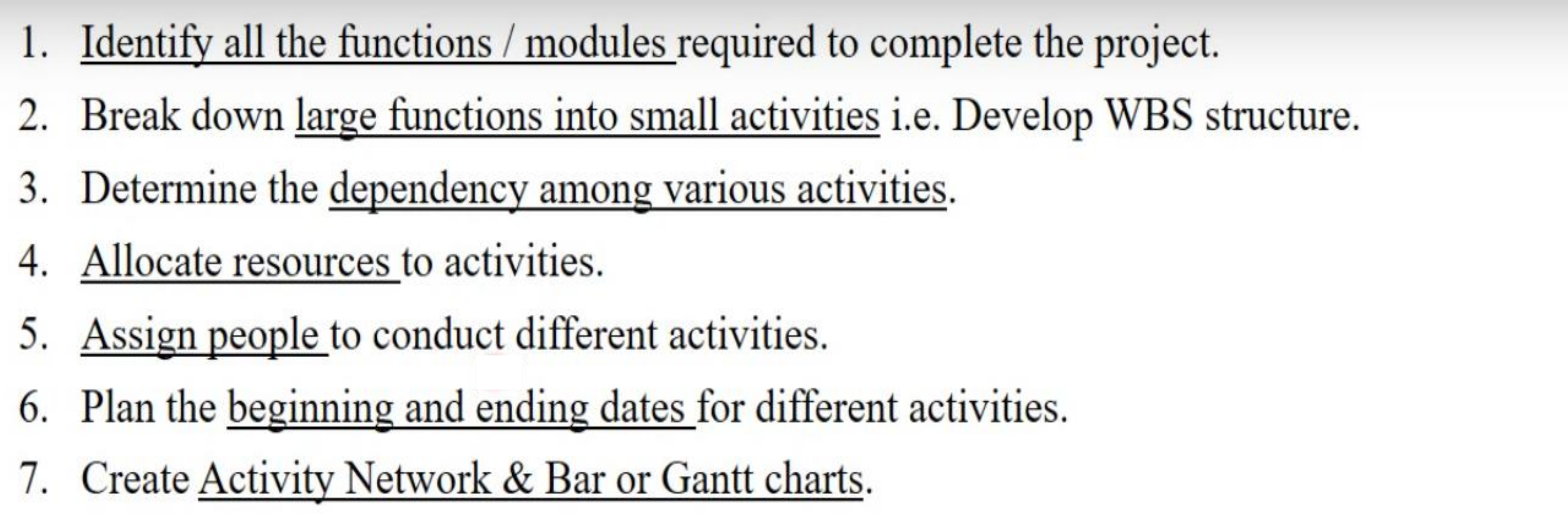
- Project is not completed on schedule.
- Changing customer requirements affect on schedule.
- Technical difficulties are generate.
- Miscommunication among project management.
- Essential software & hardware may be delivered late.
- In large project, Software engineer perform multiple tasks parallel.
- Tasks interdependencies are in project.
- Risk is not considered at beginning of project.

Project Scheduling

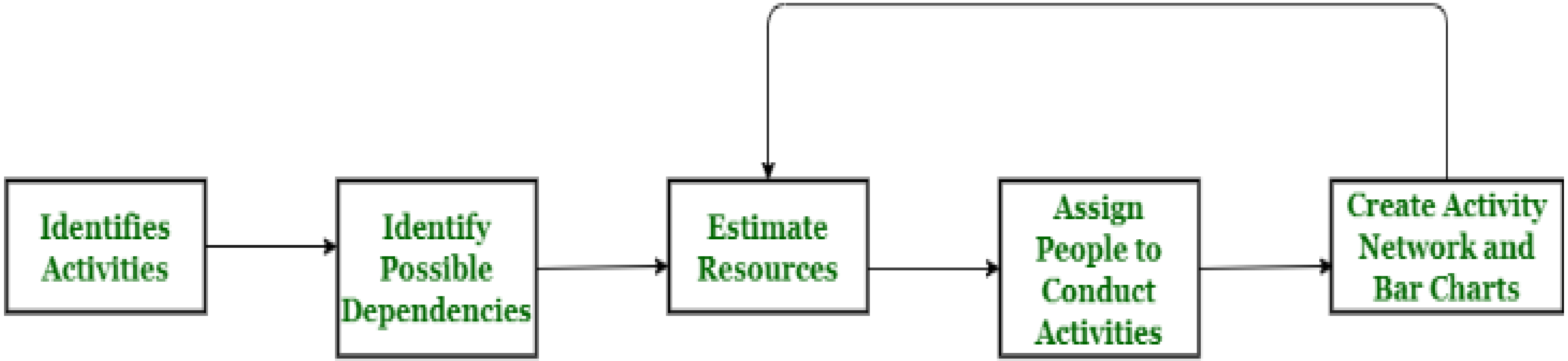


- Project scheduling is responsible activity of project manager.
- Project schedule is a mechanism that is used to communicate and know about that tasks are needed and has to be performed in project.
- Project manager separate total work task in project into different activities. i.e. WBS
- Project Manager estimate time & recourses required to complete activities & organize them into coherent sequence.
- Effective project scheduling leads to success of project, reduced cost and increased customer satisfaction.

Project Scheduling Process



1. Identify all the functions / modules required to complete the project.
2. Break down large functions into small activities i.e. Develop WBS structure.
3. Determine the dependency among various activities.
4. Allocate resources to activities.
5. Assign people to conduct different activities.
6. Plan the beginning and ending dates for different activities.
7. Create Activity Network & Bar or Gantt charts.



Identifies Activities

Identify Possible Dependencies

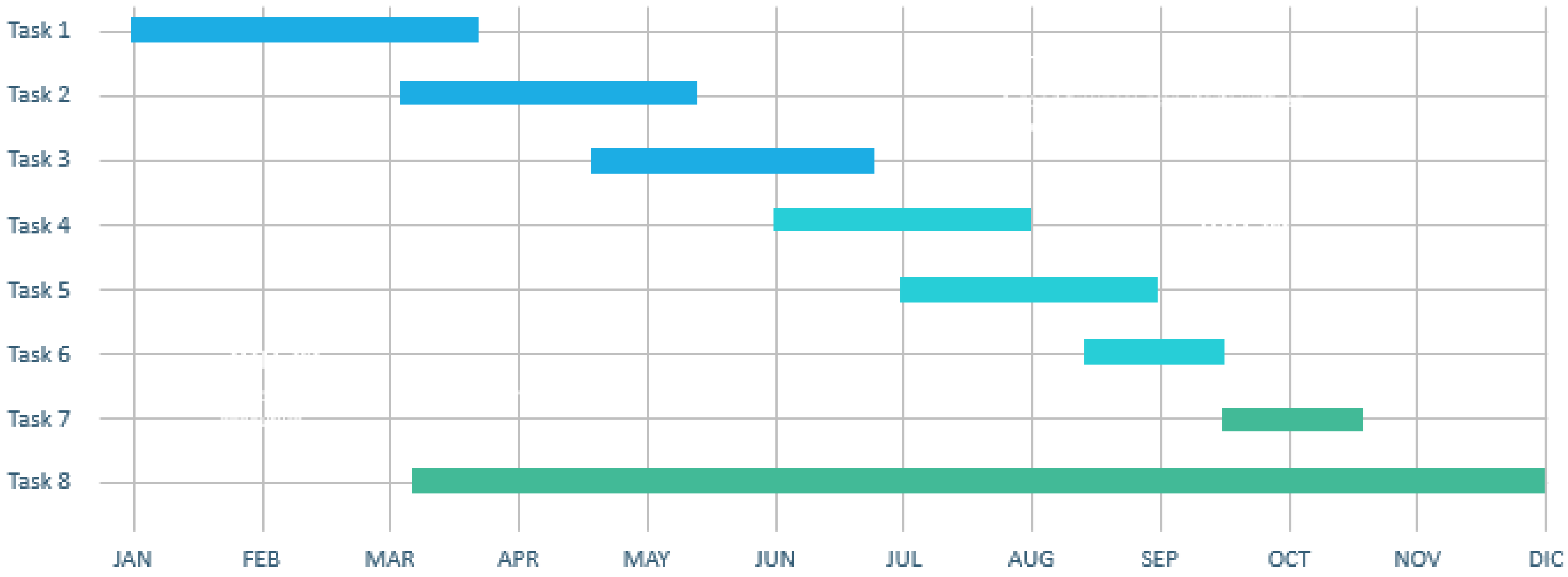
Estimate Resources

Assign People to Conduct Activities

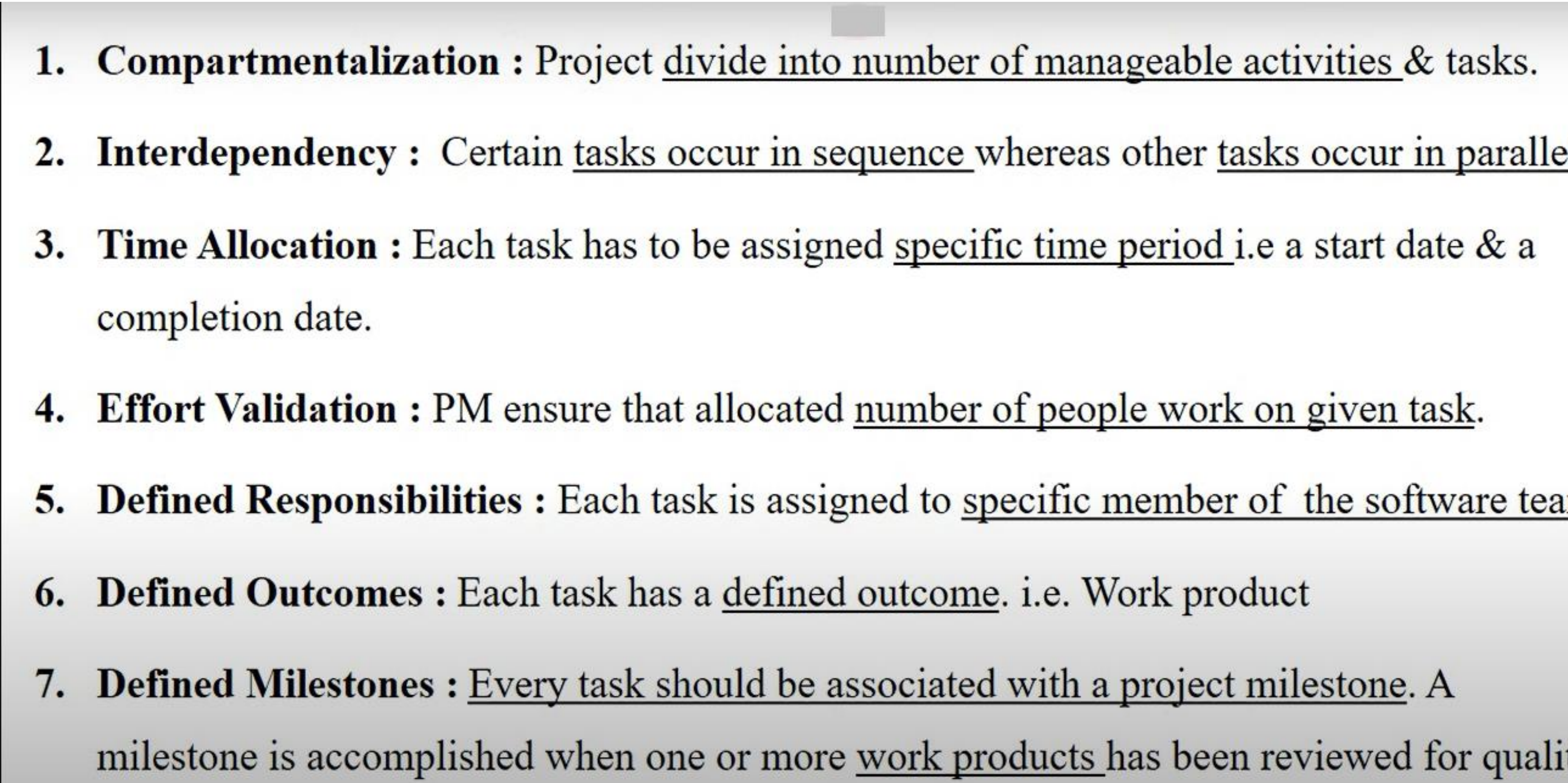
Create Activity Network and Bar Charts

Project Scheduling Process

Project Scheduling Management



Principles of project Scheduling



1. **Compartmentalization :** Project divide into number of manageable activities & tasks.
2. **Interdependency :** Certain tasks occur in sequence whereas other tasks occur in paralle
3. **Time Allocation :** Each task has to be assigned specific time period i.e a start date & a completion date.
4. **Effort Validation :** PM ensure that allocated number of people work on given task.
5. **Defined Responsibilities :** Each task is assigned to specific member of the software tea
6. **Defined Outcomes :** Each task has a defined outcome. i.e. Work product
7. **Defined Milestones :** Every task should be associated with a project milestone. A milestone is accomplished when one or more work products has been reviewed for quali

What is project Management? List task of project management
List and explain management Spectrum in details. Or Explain 4 P’s of Management
Explain w5HH Principle
Explain COCOMO Model for project estimation with suitable example
Explain COCOMO Model with its types and its levels.
What is project decomposition? Explain different techniques decomposition.
Explain project Scope management with its principles