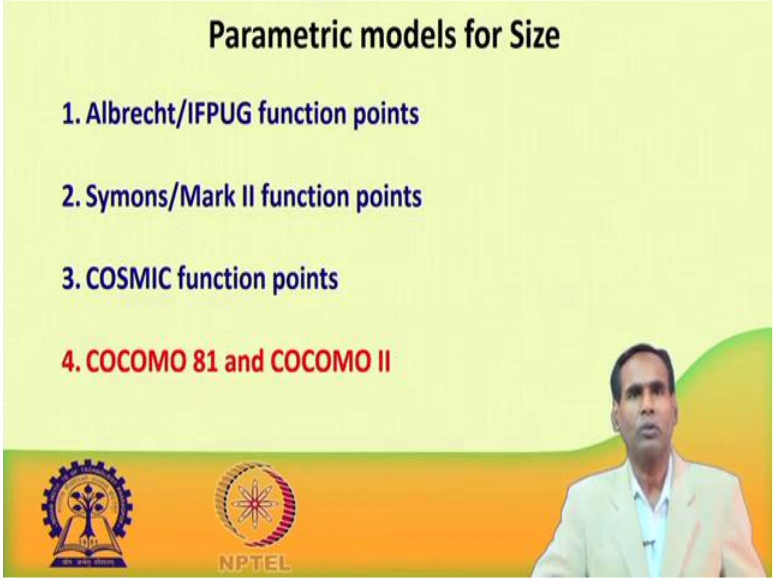
Software Project Management
Prof. Durga Prasad Mohapatra
Department of Computer Science and Engineering
National Institute of Technology, Rourkela

Lecture – 21
Project Estimation Techniques (Contd.)

Yes, good afternoon. So, now we will take up one more parametric model that is the COCOMO 1 and COCOMO 81 and COCOMO II.

(Refer Slide Time: 00:29)

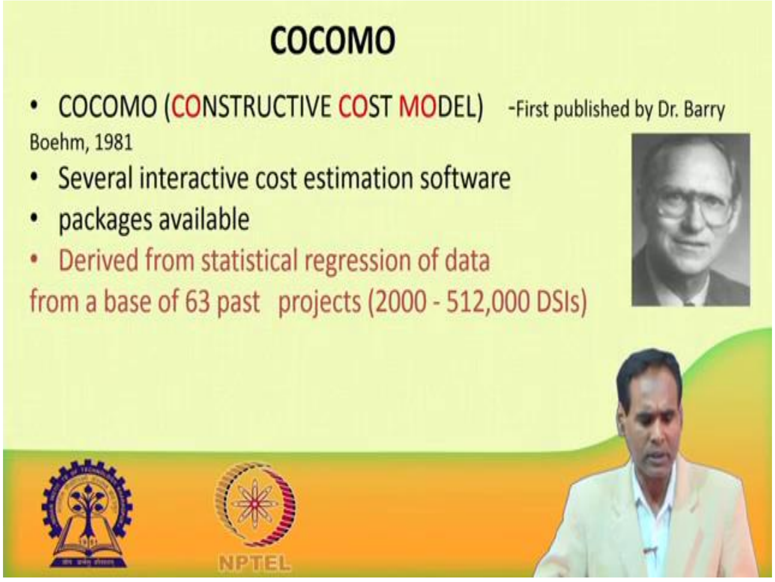


Parametric models for Size

1. Albrecht/IFPUG function points
2. Symons/Mark II function points
3. COSMIC function points
4. COCOMO 81 and COCOMO II

See last class we have seen Albrecht LFPUG function points, Symons Mark II function points, COSMIC function points. So, one was remaining that is COCOMO. So, in this class we will discuss the parametric models for size which is COCOMO 81 and COCOMO II.

(Refer Slide Time: 00:51)

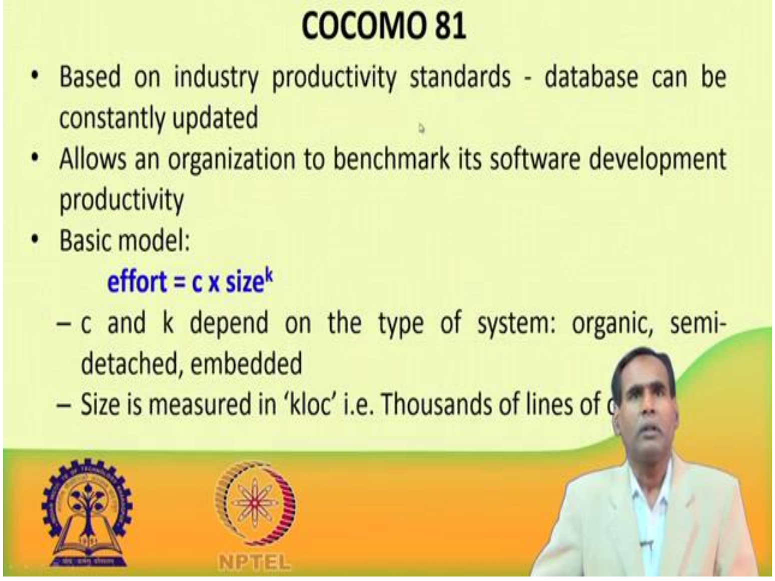


COCOMO

- COCOMO (CONSTRUCTIVE COST MODEL) -First published by Dr. Barry Boehm, 1981
- Several interactive cost estimation software
- packages available
- Derived from statistical regression of data from a base of 63 past projects (2000 - 512,000 DSIs)

So, let us first see what does COCOMO stands for what does COCOMO stand for. COCOMO stands for Constructive Cost Model, it was first published by Doctor Barry Boehm in 1981, several interactive cost estimation softwares are there the packages available. This is derived from the statistical regression of data, this COCOMO model or coco yes this COCOMO model was derived from statistical regression of data from a base of 63 past projects they have and they there are almost 2000 to 5 lakh 12000 delivered source you know instructions there from this data they have derived this COCOMO model from the statistical regression of this data and there was almost he has conducted studies from 63 past projects.

(Refer Slide Time: 01:50)



COCOMO 81

- Based on industry productivity standards - database can be constantly updated
- Allows an organization to benchmark its software development productivity
- Basic model:

  $\text{effort} = c \times \text{size}^{k}$

  - c and k depend on the type of system: organic, semi-detached, embedded
  - Size is measured in 'kloc' i.e. Thousands of lines of c

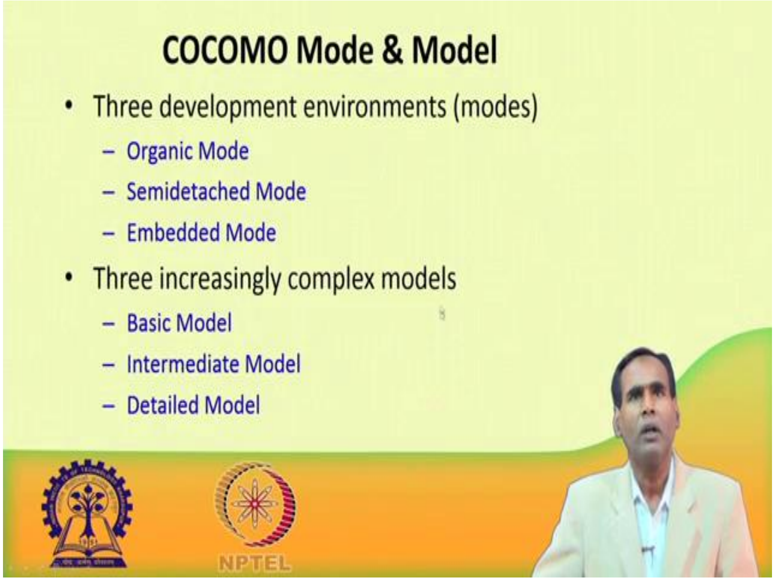
We will first see about COCOMO 81 then, we will see COCOMO II, COCOMO 81 based on industry productive standards.

The data base can be a constantly updated. It allows an organisation to bench mark, it is software development productivity. The basic model is like this. We want to calculate the effort, the basic model says that effort can be calculated as the product of 2 things; c and size, where effort is represented as

$$\text{effort} = c \times \text{size}^{k}$$

where c and k they depend on the type of the system the type of the product at the type of the project to going to develop, the product type could be organic could be semi-detached or could be embedded and the size is measured in KLOC; that means, in terms of thousands of lines of code.

(Refer Slide Time: 02:53)

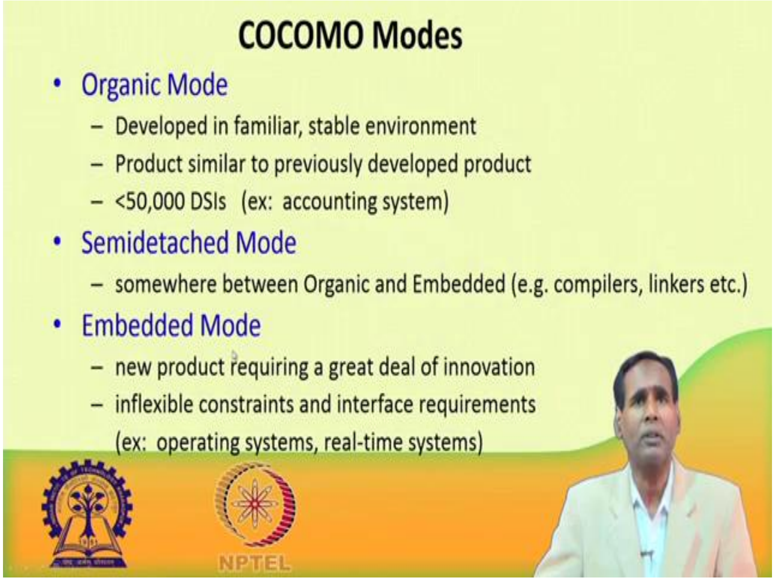


COCOMO Mode & Model

- Three development environments (modes)
  - Organic Mode
  - Semidetached Mode
  - Embedded Mode
- Three increasingly complex models
  - Basic Model
  - Intermediate Model
  - Detailed Model

Now, let us see what are the different COCOMO modes and the models, 3 different environments or modes are there for COCOMO. I have already told you they are organic mode, semidetached mode and embedded mode, 3 increasingly complex COCOMO models are there. First one is the basic model then intermediate model then, detailed model and today we will discuss only the basic model and other two intermediate model and detailed model we will discuss in the next class.

(Refer Slide Time: 03:25)



COCOMO Modes

- Organic Mode
  - Developed in familiar, stable environment
  - Product similar to previously developed product
  - <50,000 DSIs (ex: accounting system)
- Semidetached Mode
  - somewhere between Organic and Embedded (e.g. compilers, linkers etc.)
- Embedded Mode
  - new product requiring a great deal of innovation
  - inflexible constraints and interface requirements (ex: operating systems, real-time systems)

See let us first see the COCOMO modes. As I have already told you that are 3 modes or 3 environments for COCOMO, one is organic mode, then semidetached mode, embedded mode. In organic mode what do we when you will call a product as organic mode? A product which can be developed in familiar and stable environment we call it is an organic mode.

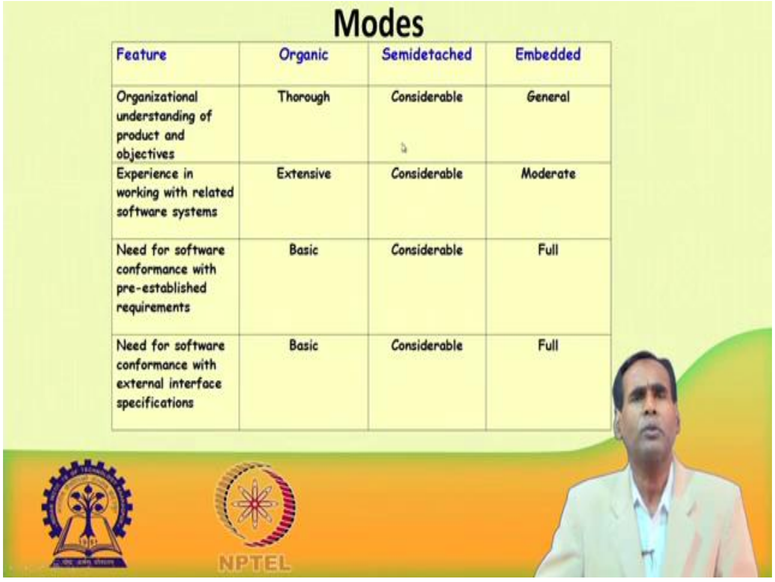
So, the products are similar or the product that you are going to develop is similar to the previously developed products, normally it is size is less than 50000 delivered source of instructions. So, normally it is size is less than or should be less than 50000 delivered source instructions. For example, people see your accounting system, all the information systems normally coming under organic mode.

So, for example, accounting system, student information system these things pay roll information system, these all are coming under organic modes. And what is about semidetached mode? Before going to semidetached mode let us see about the embedded mode. Embedded mode actually, a product which is completely new, a new product. So, embedded mode means, a product which is completely new, a new product requiring a great deal of innovation, a great deal of effort inflexible constrains and interface requirements are there.

So, normally all the what we can say your systems kind of things like operating systems, real time systems so, they are coming under a these system software kind of things like operating systems and real time systems, embedded systems etcetera they are coming under the embedded mode. Now, we will see this semidetached mode. Semi the products which are coming in between organic mode and embedded mode they normally we call them as a semidetached mode. So, semidetached product semidetached mode products they lie in between what somewhere these organic and embedded modes semidetached mode.

So, these products somewhere lie in between the organic and embedded and examples are the utility applications such as, compilers, linkers etcetera.

(Refer Slide Time: 05:45)



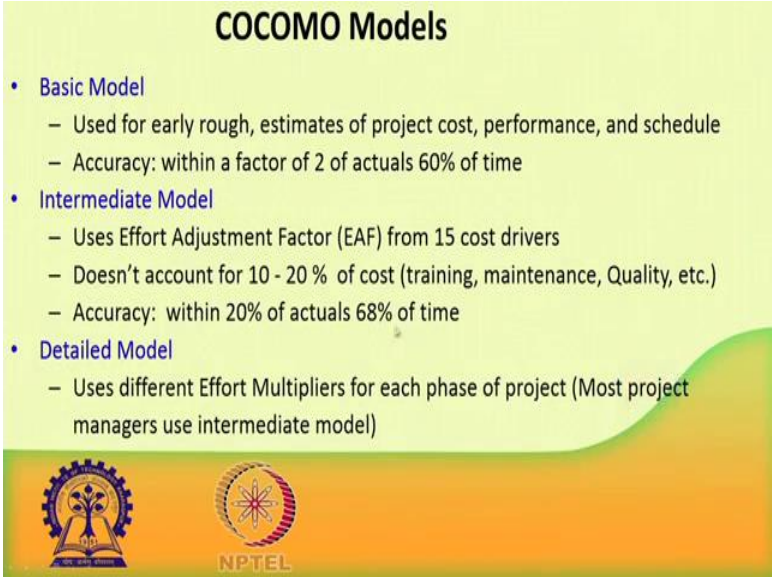
Modes

Feature	Organic	Semidetached	Embedded
Organizational understanding of product and objectives	Thorough	Considerable	General
Experience in working with related software systems	Extensive	Considerable	Moderate
Need for software conformance with pre-established requirements	Basic	Considerable	Full
Need for software conformance with external interface specifications	Basic	Considerable	Full

So, here basically in this table we have shown the different features of the products under different modes for example, the organisational understanding of product and objectives in case of organic it is a thorough in semidetached it is considerable and it is embedded it is general and experience in working with related software systems, you can see organic it is extensive, semidetached it is considerable and embedded it is moderate.

So, need for software conformance with pre established requirements here organic yes it is a basic, where semidetached it is considerable and embedded it is full Similarly need for software conformance with external interface specifications in organics is just it is basic, but semidetached it is moderate or considerable and in embedded applications it is full.

(Refer Slide Time: 06:37)



COCOMO Models

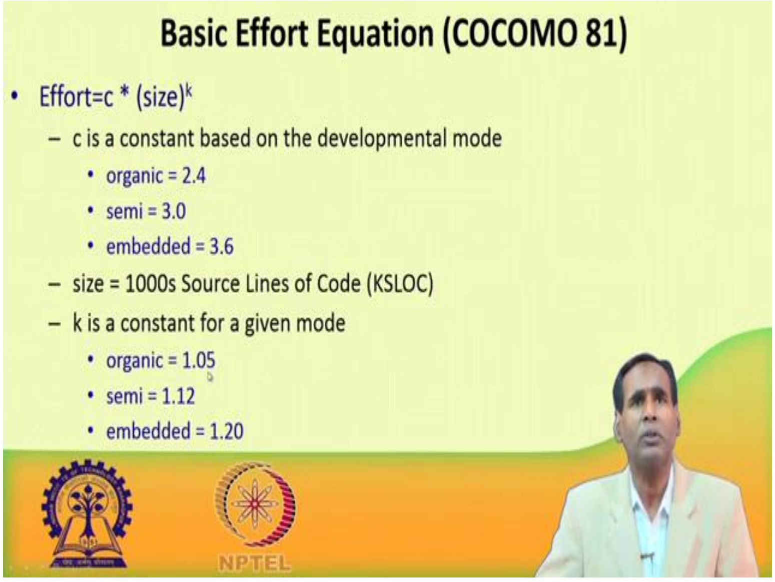
- Basic Model
  - Used for early rough, estimates of project cost, performance, and schedule
  - Accuracy: within a factor of 2 of actuals 60% of time
- Intermediate Model
  - Uses Effort Adjustment Factor (EAF) from 15 cost drivers
  - Doesn't account for 10 - 20 % of cost (training, maintenance, Quality, etc.)
  - Accuracy: within 20% of actuals 68% of time
- Detailed Model
  - Uses different Effort Multipliers for each phase of project (Most project managers use intermediate model)

As I have already told you there are different models of COCOMO, one, the fundamental one the basic model, then intermediate model and detailed model. Let us see about first basic model. Normally, this basic model is used for early rough estimates of a project cost, performance and schedule.

So, using this model or this model can be used for estimation of early and rough estimates ok, this can be used for the early rough estimates of project cost, the performance and the schedule. It is accuracy is within a factor of 2 of actuals 60 percent of time. Intermediate model it uses effort adjustment factor which is known as EAF from 15 cost drivers, it does not account for 10 to 20 percent of cost; that means, training. It does not account for 10 to 20 percent of the cost for example, training maintenance and quality etcetera.

Here the accuracy is within 20 percent of the actuals 68 percent of time. The another model called detailed model or the COCOMO complete model or complete COCOMO, we say, it uses different effort multipliers for each phase of project, most project managers they use intermediate model.

(Refer Slide Time: 08:07)



Basic Effort Equation (COCOMO 81)

- Effort=c * (size)^k
  - c is a constant based on the developmental mode
    - organic = 2.4
    - semi = 3.0
    - embedded = 3.6
  - size = 1000s Source Lines of Code (KSLOC)
  - k is a constant for a given mode
    - organic = 1.05
    - semi = 1.12
    - embedded = 1.20

Let us first see the basic effort equation or the basic COCOMO model, what will be the equation for estimating effort using basic COCOMO, it is known as COCOMO 81.

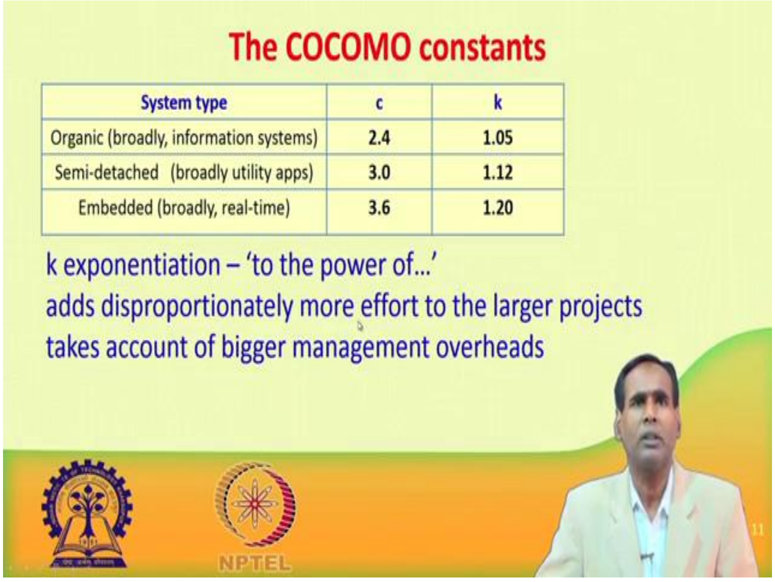
So, here the according to COCOMO 81 the basic the fundamental equation for estimating effort is given a like this:

$$\text{effort} = c \text{ x } \text{size}^k$$

where c is a constant based on the developmental mode, for organic mode the value of c is 2.4, for semi organic semidetached it is 3 3.0 and for embedded mode it is 3.6.

Similarly, the size it is normally considered as is normally considered in KSLOC, that means, 1000 source lines of code, k is constant that also varies for the different modes, for organic mode the value of k is 1.05, for semidetached it is 1.12 and for embedded it is 1.20.

(Refer Slide Time: 09:14)



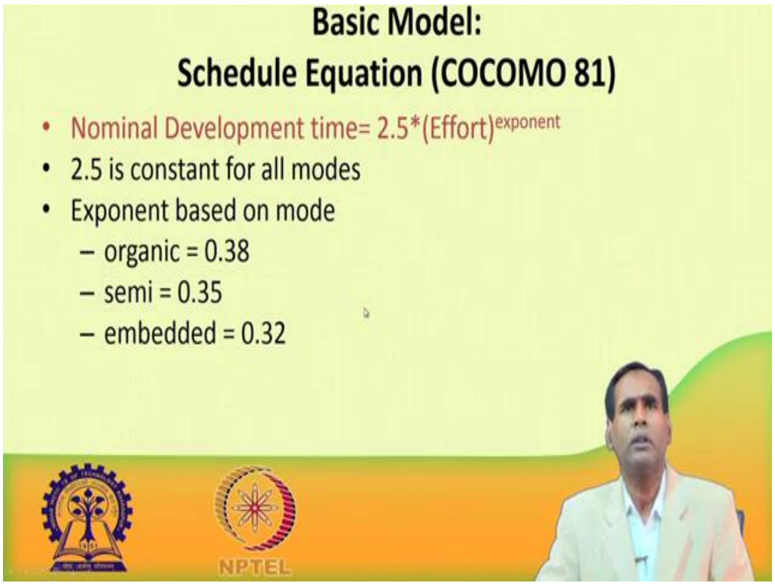
The COCOMO constants

System type	c	k
Organic (broadly, information systems)	2.4	1.05
Semi-detached (broadly utility apps)	3.0	1.12
Embedded (broadly, real-time)	3.6	1.20

k exponentiation – 'to the power of...'
adds disproportionately more effort to the larger projects
takes account of bigger management overheads

So, all those values that I have shown here different values of c and k, they are presented in a table for a better understanding, here k is the exponentiation to the power of something it adds disproportionately more effort to the larger projects, it also takes account for bigger management overheads.

(Refer Slide Time: 09:36)



Basic Model: Schedule Equation (COCOMO 81)

- Nominal Development time= $2.5*(\text{Effort})^{\text{exponent}}$
- 2.5 is constant for all modes
- Exponent based on mode
  - organic = 0.38
  - semi = 0.35
  - embedded = 0.32

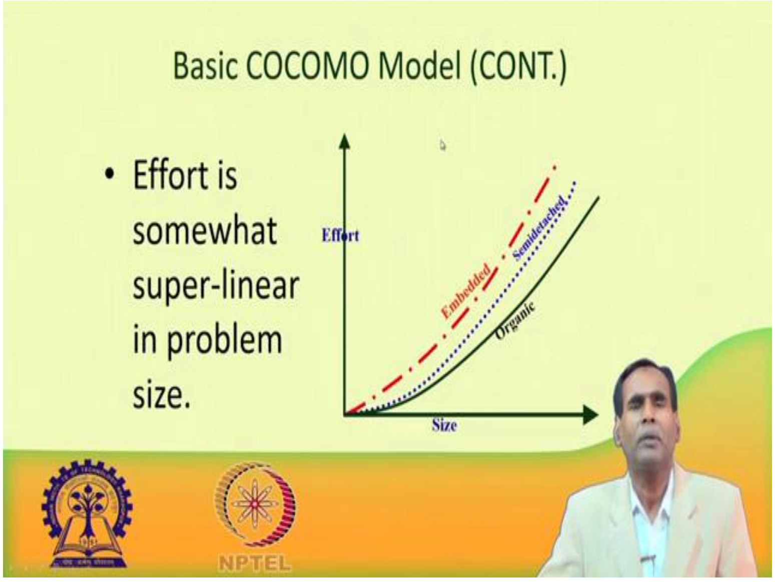
Now, let us find out the equation for the schedule using the basic COCOMO model that is COCOMO 81. The nominal development time can be represented as 2.5 into effort to the power exponent, where effort we have already calculated using this fundamental formula that effort is equal to c into size to the power k.

So, this value of effort will be put in this equation so that we can get in the nominal development time. So, the nominal development time will be equal to

$$\text{Nominal Development time} = 2.5*(\text{Effort})^{\text{exponent}}$$

where 2.5 is constant for all the modes that is organic, semidetached and embedded. The exponent, this factor it again depends upon the mode that you are using for organic mode, the value of the exponent is 0.38 for semidetached mode it is 0.35 and for embedded it is 0.32.

(Refer Slide Time: 10:34)

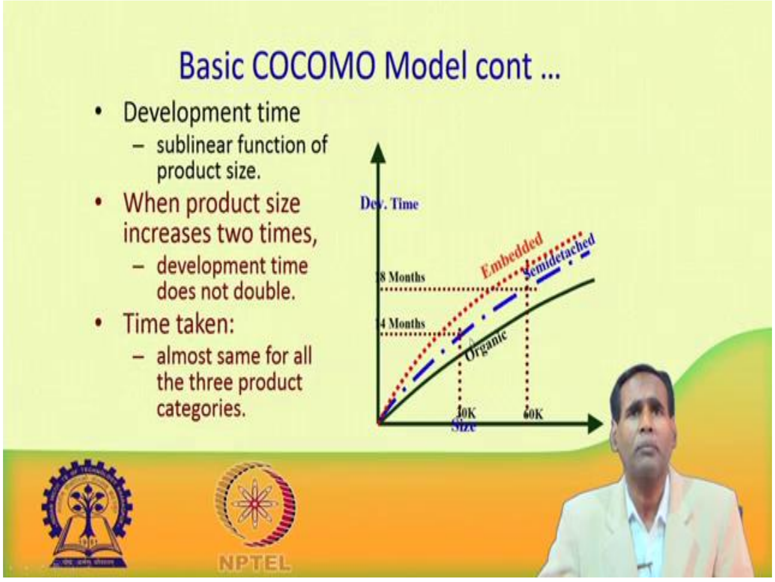


Basic COCOMO Model (CONT.)

- Effort is somewhat super-linear in problem size.

So, now let us see we have already seen this what these 2 equations for estimating effort and duration, now let us see how the graph will look like for the different products. You can see that in x axis in this graph, we have taken x axis along x axis we have taken size and along y axis we have taken effort. You can see that effort is somewhat superliner in problem size, just the problem size is increasing it is effort in is just increasing, but not in that proportionate. So, effort is somewhat superliner in problem size.

(Refer Slide Time: 11:16)

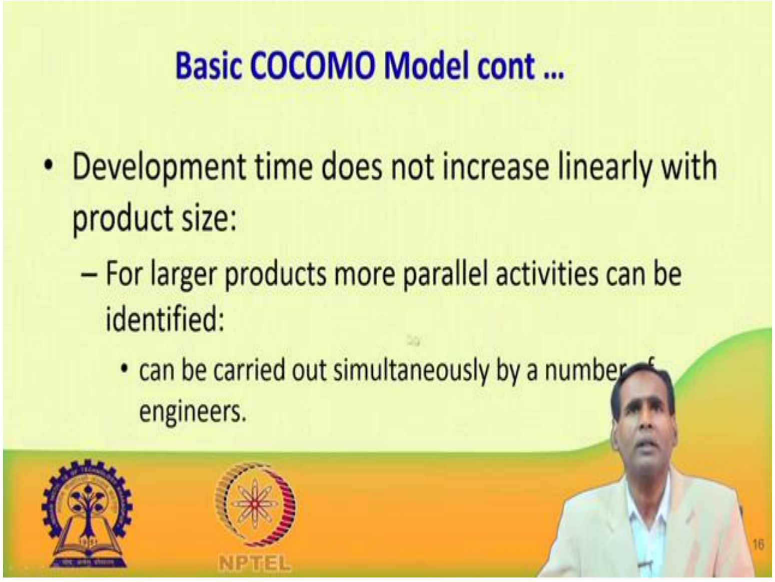


Basic COCOMO Model cont ...

- Development time
  - sublinear function of product size.
- When product size increases two times,
  - development time does not double.
- Time taken:
  - almost same for all the three product categories.

Similarly, you can draw a graph between the what size and the development time, you can see that the development is a sub linear function of the product size. What do you mean by what is this? That when the product size increases in what in which proportion the development time does not increase in that way. When the product size increases 2 times please mark here the development time does not double it. The time taken is almost the same you another observation you can make from this graph it is like that, the time taken is almost same for all the 3 product categories whatever may the product may organic or semidetached or embedded the time taken to develop this different kinds of a products it almost same.

(Refer Slide Time: 12:12)



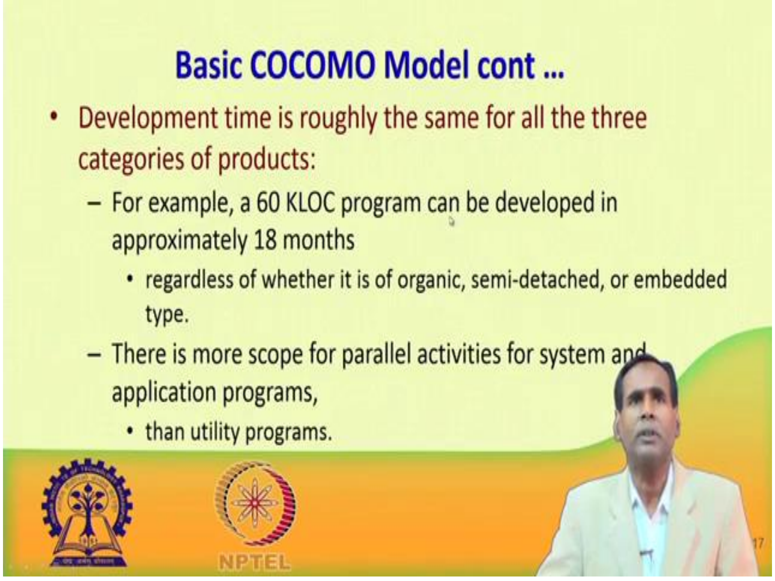
Basic COCOMO Model cont ...

- Development time does not increase linearly with product size:
  - For larger products more parallel activities can be identified:
    - can be carried out simultaneously by a number of engineers.

And you can say that let us say, that development time it does not increase linearly with the product size I have already told you, because for larger products more parallel activities can be identified and they can be carried out simultaneously by a number of engineers.

So, hence the development time it will not increase linearly with product size that is why the graph is what is coming like this.

(Refer Slide Time: 12:39)



Basic COCOMO Model cont ...

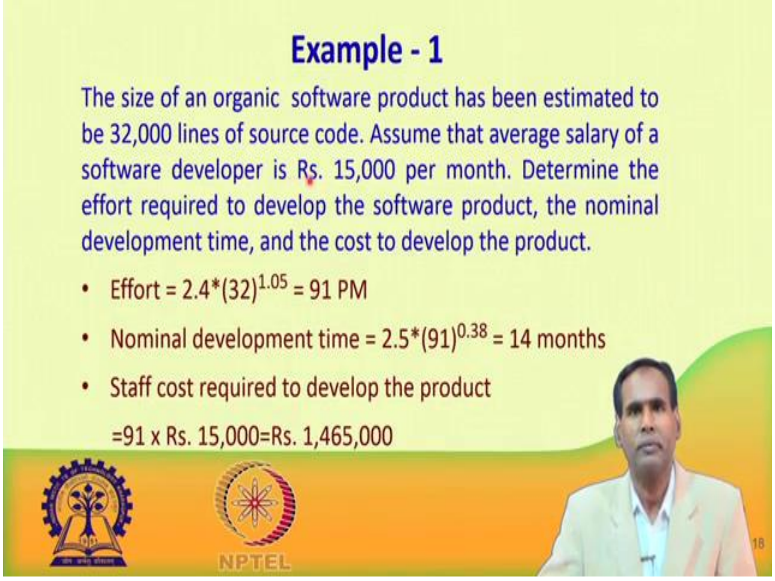
- Development time is roughly the same for all the three categories of products:
  - For example, a 60 KLOC program can be developed in approximately 18 months
    - regardless of whether it is of organic, semi-detached, or embedded type.
  - There is more scope for parallel activities for system and application programs,
    - than utility programs.

For example, you see the developmental time is roughly the same for all the 3 category of the products, I have already told you. Let us take an example for example, a 60 KLOC program, it can be developed in approximately 18 months irrespective of whether the product is organic or semidetached or embedded type. So, there is more scope, why? Because I have already told you that for larger programs there is more scope for parallel activities for larger programs such as, systems and application programs than the utility programs.

So, another thing you can mark. So, this is a graph which shows that if the exponent is greater than 1, so, how this what graph looks likes for the effort and the if the exponent is less than 1 how the graph is look like for the duration. See duration for the increasing effort the and graph, for the duration for the increasing effort when these power the exponent is less than 1 0.38, see how this what look like because you are the what exponential is less than 1 and in case of a effort here the this exponent is greater than 1.

So, for effort you see the graph it looks like this. So, this is how the graph can look like for the effort and duration depending upon on the value of the exponent, whether it is greater than 1 or less than 1 the graph size also changes.

(Refer Slide Time: 14:18)



Example - 1

The size of an organic software product has been estimated to be 32,000 lines of source code. Assume that average salary of a software developer is Rs. 15,000 per month. Determine the effort required to develop the software product, the nominal development time, and the cost to develop the product.

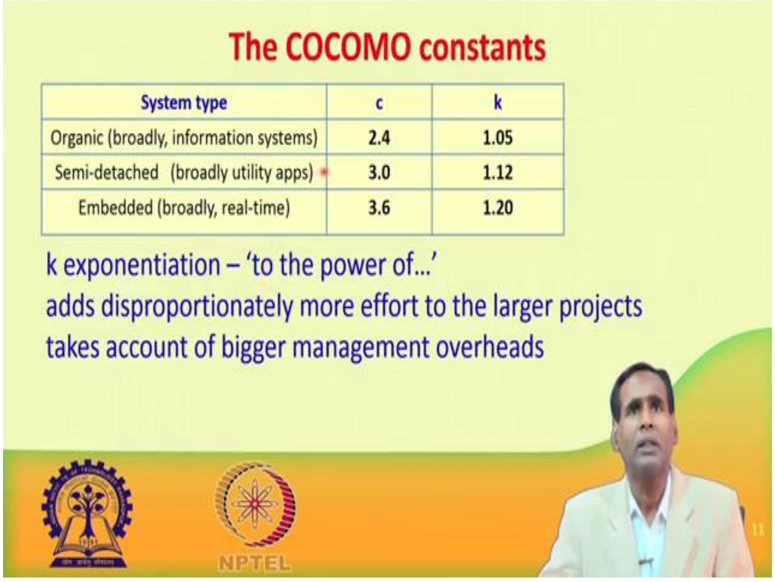
- Effort = $2.4*(32)^{1.05}$ = 91 PM
- Nominal development time = $2.5*(91)^{0.38}$ = 14 months
- Staff cost required to develop the product =91 x Rs. 15,000=Rs. 1,465,000

Now, let us take some of the examples and try to elaborate this. We will use some the equations of the basic COCOMO and we will try to find out, we will try to estimate the effort and the nominal development type.

So, the first example says that the size of an organic software product has been estimated to be 32 lines of source code. Assume, that the average salary of a software developer is 15000 per month, what you have to do? Determine the effort required to develop the software product, the nominal development time and the cost to develop the product. So, as you we can see from the problem it is given as organic software product.

So, now for effort the formula is you know: c into size to the power k, now we have to find out what is the value of c and k for the organic software. You can see the value of c and k I have already given you a table for the value of c and k are given like this.

(Refer Slide Time: 15:22)



The COCOMO constants

System type	c	k
Organic (broadly, information systems)	2.4	1.05
Semi-detached (broadly utility apps)	3.0	1.12
Embedded (broadly, real-time)	3.6	1.20

k exponentiation – ‘to the power of...’
adds disproportionately more effort to the larger projects
takes account of bigger management overheads

For organic product, the value of c is 2 to the power 2.4 and the value of k is how much? 1.05. So, we can see that here I will just put the value of c and k in the equation.

So, for organic product value of c is 2.2, for organic product the value of k is 1.05, the size is already given 32000 lines of code; that means, 32 KLOC. So, effort can be estimated as 2.4 into 32 to the power 1.05 and I have already told you the unit of effort in the last class, that effort is on normally what expressed as in terms of person months or man months.

So, effort is equal to after solving this equation it is you coming to be 91 approximately, 91 person month. The nominal development time again we have seen the equation for nominal development time earlier this is coming to be how much? The nominal development time can be expressed as 2.5 into effort to the power exponent and the value of exponent varies as the mode changes, it is the organic type of product, So, the value of exponent or the constantly 0.38.

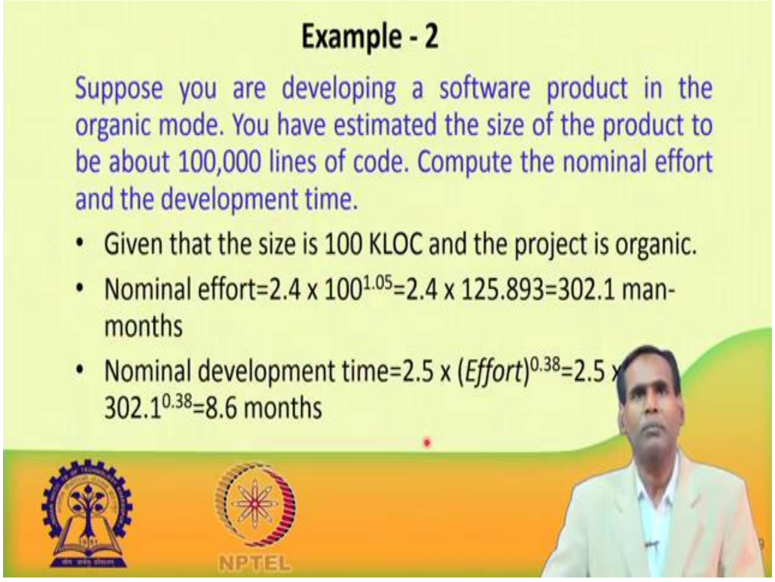
So, now I have to use the value of effort and the value of this exponent is 0.38 in the equation. So, I will get how much? So, see we will get the value of nominal development time is equal to 2.5 which is a constant and what is this? Effort is 91 to the power the exponent and for the this product which organic software product the value of exponent is 0.38 and I am solving what this equation you will get the nominal development time is equal to 14 months, approximately 14 months. So, now, what the question is find out also how much cost will be spent to develop the product. So, we have to see that what will be the staff cost. I have only in the problem says that one software developer will take the average salary of a software developer is 15000 will require, how much? What is the person month? We require 91 person months to develop the software. So, the cost that will be required the staff cost that will be required to develop these organic product will be 91 into 15000, which is coming to be how much? 14 lakh 65000.

So, in this way given a product how you can estimate the effort and nominal development time and the cost? First step is to identify what kind of product is, because in the examination it may not be given clearly organic software it might be an information system.

So, we have to apply your common sense that information system means this an organic software. So, if it is a real time system means, this is an embedded software. Accordingly, these value of the constants we have to remember and put in the equation.

So, then after finding out the effort by putting the value of the c and k in the equation, find out the effort then nominal time use this value of the effort obtained in the previous step put that value and then value of the constant again it will vary from the for the different modes such as, organic, semidetached and what your embedded put the appropriate value of the constant here. So, that you will get the nominal development time and then in order to find out the staff cost what you have to do? In the question, it may be given what is the average what salary of the developer multiply it by the effort, you will get the total cost that will require to develop the project.

(Refer Slide Time: 18:57)



Example - 2

Suppose you are developing a software product in the organic mode. You have estimated the size of the product to be about 100,000 lines of code. Compute the nominal effort and the development time.

- Given that the size is 100 KLOC and the project is organic.
- Nominal effort=2.4 x $100^{1.05}$=2.4 x 125.893=302.1 man-months
- Nominal development time=2.5 x $(Effort)^{0.38}$=2.5 x $302.1^{0.38}$=8.6 months

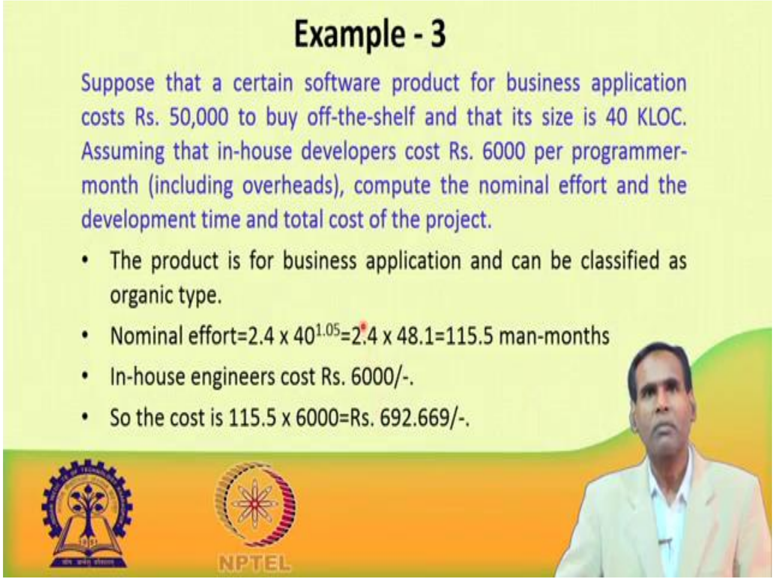
Now let us quickly take another example. Suppose you are developing a software product in the organic mode you have estimated the size of the product to be about how much 1 lakh lines of code we have to compute the nominal effort and the development time.

So, we can see that here it is given that it is a organic product. So, we can easily choose the value of the c and k, the size is also given it is 1 lakh lines of code; that means, 100 KLOC.

So, nominal effort could be we know already for organic product value of c is 2.4 and for organic product, the value of k is equal to 1.05, putting the values in the equation we will get and the size already given 100 KLOC. So, nominal effort will be called 2.4 into 100 to the power 1.05; on solving, you will get 302.1 man months or no you can round off it to 302 what man months.

So, this is the effort then nominal development time will be equal to how much? 2.5 into effort to the power some constant and for organic mode the constant is 0.38; so, which will give you 8.6 months. So, in this case what for this project the nominal effort is found to be 302.1 man months or person months and the nominal development time is coming to be 8.6 months.

(Refer Slide Time: 20:25)



Example - 3

Suppose that a certain software product for business application costs Rs. 50,000 to buy off-the-shelf and that its size is 40 KLOC. Assuming that in-house developers cost Rs. 6000 per programmer-month (including overheads), compute the nominal effort and the development time and total cost of the project.

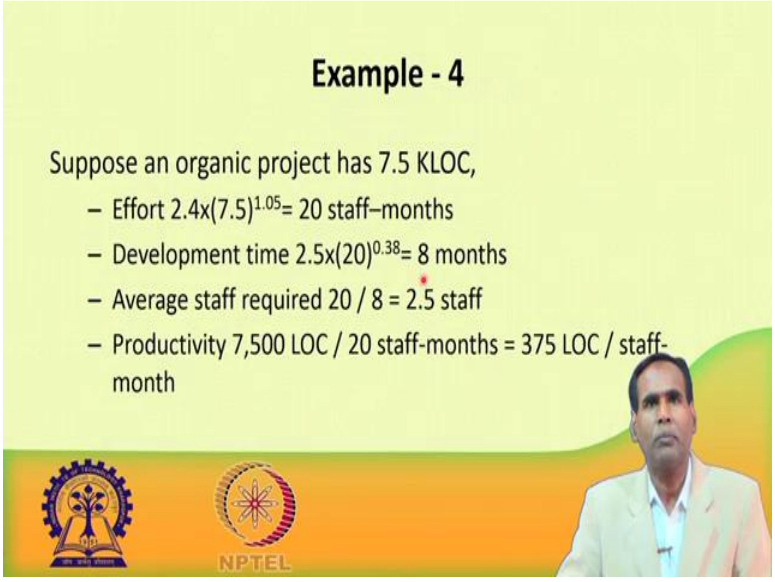
- The product is for business application and can be classified as organic type.
- Nominal effort=2.4 x $40^{1.05}$=2.4 x 48.1=115.5 man-months
- In-house engineers cost Rs. 6000/-.
- So the cost is 115.5 x 6000=Rs. 692.669/-.

You can take a another example here. Suppose that certain software product for business application cost rupees 15000 to buy off-th-shelf and that it is size is 40 KLOC, assuming that in house developers cost rupees 6000 for per program or month or for person month including overheads compute the nominal effort and the development time and the total cost of the project. See the product is see indirectly it is not told what whether it is organic or semidetached it is not directly given, but it is given is an business application.

So, business application means, it is an information system. So, indirectly you can say that this is organic type. So, indirectly all are given. So, you have to start from this point that the product is for business application and hence it can be classified as organic type, the nominal product the nominal effort we have to now find out and we know for organic products the value of constant c and k are as follows c will be 2.4 and k is equal to 1.05 and it is given that what is the size already given, please whatever the size is given it may be given in KLOC or simple LOC, if it is given LOC you convert into KLOC, here it is directly given 40 KLOC.

So, I will use in place of size 40. So, this is coming to be this many man months then in house engineer cost is how much? It is given that the in house developers, they take 6000 per programmer month and I have to find out then the total cost. So, the effort is coming to be 115.5 and per month cost is 6000. So, the program will take how much cost? Total cost would be required this much cost 692.669 rupees it will be taken for developing this product.

(Refer Slide Time: 22:11)



Example - 4

Suppose an organic project has 7.5 KLOC,

- Effort $2.4x(7.5)^{1.05}$= 20 staff–months
- Development time $2.5x(20)^{0.38}$= 8 months
- Average staff required 20 / 8 = 2.5 staff
- Productivity 7,500 LOC / 20 staff-months = 375 LOC / staff-month

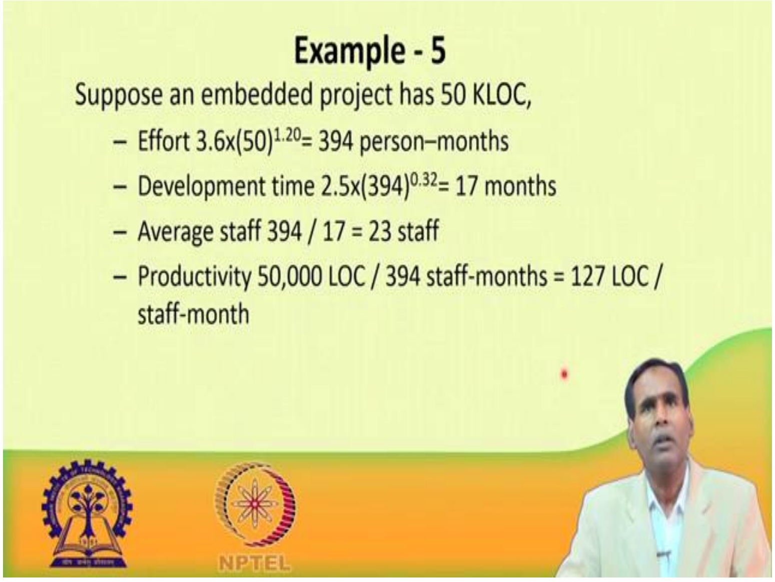
Still another example you can take. Again, suppose an organic project you want going to construct it is of 7.5, 7.5 KLOC it is size.

So, effort will be for organic product we know that these what the value of c is 2.4 and k is equal to 1.05. So, effort is equal to straight forward 2.4 into 7.5 to the power 1.05 that is nearly equal to 20 staff months; development, time is equal to 2.5 into 20 to the power constant for normal product, the value of constant is 0.38 which is coming to be on simplification it is coming to be approximately 8 months.

So, now the average staff required will be how you can calculate it? It will be calculated by dividing the effort by the development time. So, 20 by 8 so; that means, roughly or approximately 2.5 staff members will be required. So, the productivity if you will want to find out what will be the productivity then the productivity will be coming to as how much the total size is 7.5 or 7005 LOC, 7500 LOC divided by 20 staff months.

So, this is we got that 20 staff months will be required here, because effort is 20 staff months. So, that we get how much? So, for staff 375 LOC may be developed. So, the productivity is 375 LOC for staff month.

(Refer Slide Time: 23:39)



Example - 5

Suppose an embedded project has 50 KLOC,

- Effort $3.6x(50)^{1.20}$= 394 person–months
- Development time $2.5x(394)^{0.32}$= 17 months
- Average staff 394 / 17 = 23 staff
- Productivity 50,000 LOC / 394 staff-months = 127 LOC / staff-month

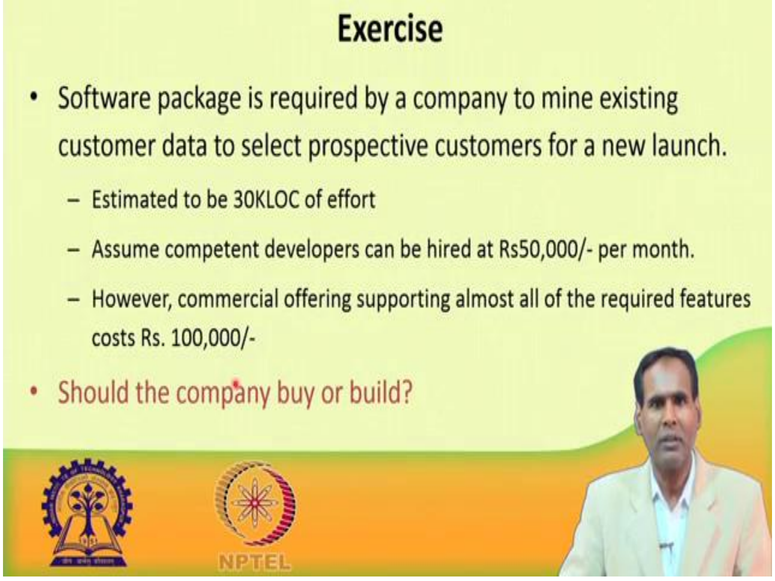
And the other example say it is like this. Suppose an embedded project has 50 KLOC and again we want to develop this value for the nominal effort to nominal development time and the average staff and the productivity here, this product size is already given 50 KLOC and type is embedded project, for embedded project, we know the value of c is 3.6 value of k is 1.20 by putting the values here we can get the value of effort.

So, let us put the value. So, 3 it will be now the equation will be simplified as 3.6 into 50 to the power 1.0, on solving you will get approximately 394 person months for this embedded project. The nominal development time will be 2.5 into effort to the power the exponent the constant. So, here effort is coming to be 394 and the value of constant or the value of the exponent for embedded type of project is 0.32. So, this require how much? On simplification you will get this is 17 months. Average staff how you will get?

So, what is the effort divided by the nominal development time is coming to be approximately 23 staff members, productivity can be obtained by dividing the total size of the project divided by what the productivity can be obtained by dividing what the total size by the effort which is in person months or staff months. So, this is equal to a 50000 LOC divided by 394 staff months, which is equal to how much? Approximately 127 LOC per staff months; that means, the productive is it is 127 LOC for I think it is yes the productivity is equal to how much for staff month this is or for person month this is 375 LOC and similarly, this here in this example the productivity is 121 approximately 127 lines of code for staff month.

So, in this way in the examination some questions may be asked simple questions, where you have to first identify what kind of product it is then what are the values of the constants for those products, then we can easily what then you have to remember the required equations for finding the value of effort and development time, then you can easily estimate the values for the effort and development time and then you can calculate the average staff, productivity and the total cost etcetera. So, please solve some of the examples and exercises from new book or from internet sources etcetera.

(Refer Slide Time: 26:30)



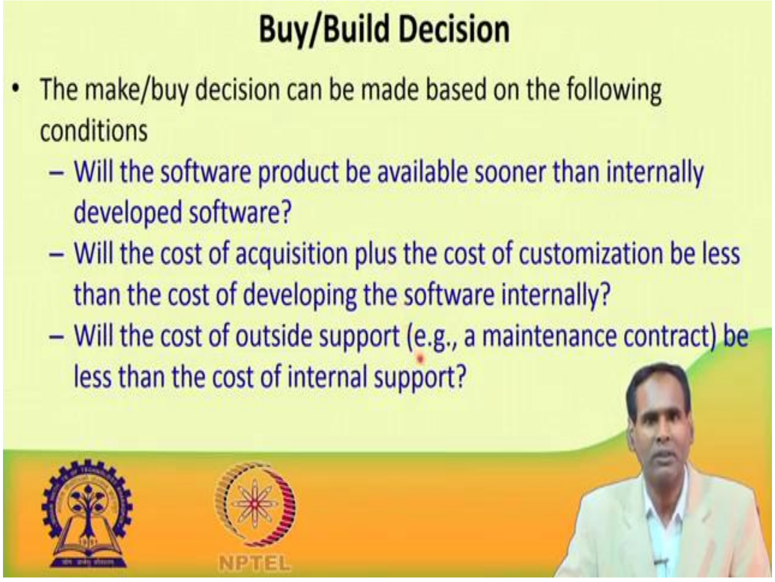
Exercise

- Software package is required by a company to mine existing customer data to select prospective customers for a new launch.
  - Estimated to be 30KLOC of effort
  - Assume competent developers can be hired at Rs50,000/- per month.
  - However, commercial offering supporting almost all of the required features costs Rs. 100,000/-
- Should the company buy or build?

So, a small exercise we have given to you like a software package is required by a company to mine the existing customer data to select prospective customers for a new launch. The estimated what it is estimated to be 30 KLOC of effort assume that the competent developers can be hired at 50000 per month. However, commercial offering support supporting almost all the required features cost, this how much 1 lakh. So, here you see that the developers can be hired at rupees 50000 per month, but here if you will purchase from outside; however, commercial offering if you market you will purchase it is total is coming 1 lakh.

So, the question say that should we or should the company develop it or build it this software or they will buy it. So, we have to think a decision. So, hence I am giving identify first what type of product if this is identify then calculate the effort, then what the what average, what salary or the developers is developer salary can be given per month, then find out what is the total cost if it will be developed in house then compare with this market value, if somebody will come what purchase from the market it is 1 lakh compare this values, then take a decision whether the company should go for buying it or building it.

(Refer Slide Time: 28:01)



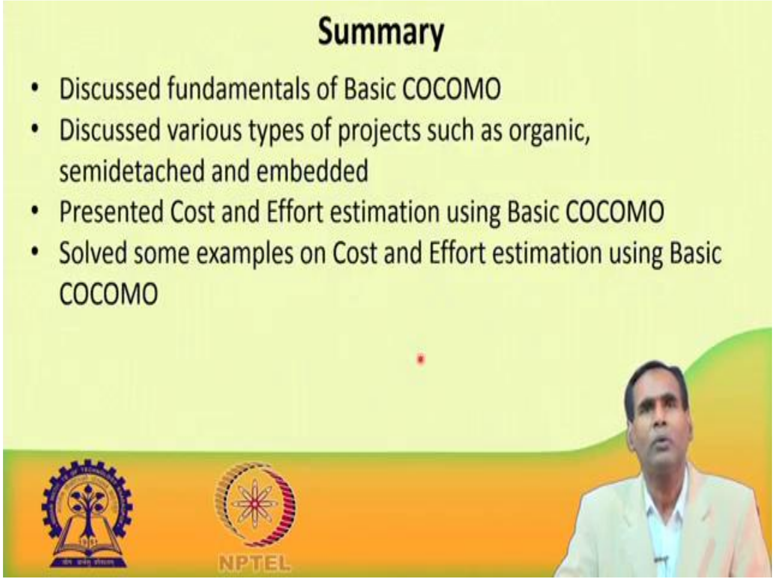
Buy/Build Decision

- The make/buy decision can be made based on the following conditions
  - Will the software product be available sooner than internally developed software?
  - Will the cost of acquisition plus the cost of customization be less than the cost of developing the software internally?
  - Will the cost of outside support (e.g., a maintenance contract) be less than the cost of internal support?

These guild lines will help you to take a viable decision to the make or buy decision can be made based on the following conditions, the will the software product available sooner than internal developed software, if you will go to market definitely you will you can easily very soon you can purchase it.

So, we have to consider will the software or whether the software product be available sooner than internal developed software similarly, we have to think of when the cost of acquisition plus the cost of the customisation both they will be less than the cost of developing the software internally will the cost of outside support that is the maintenance contract etcetera will less than the cost of internal support.

(Refer Slide Time: 28:44)



Summary

- Discussed fundamentals of Basic COCOMO
- Discussed various types of projects such as organic, semidetached and embedded
- Presented Cost and Effort estimation using Basic COCOMO
- Solved some examples on Cost and Effort estimation using Basic COCOMO

Say if we can what. So, before going for a buy build decision these issues we must have to consider, that this we must have to look what.

So, that then we can take a proper decision whether to buy or build a software. So, finally, we have come to the summary that we first discussed the fundamental concept of basic COCOMO, we have also discussed the various types of projects or products such as, organic semidetached and embedded we have also presented the cost and effort estimation that formula that formulae using basic COCOMO finally, we have solved some examples on estimation of cost and effort using basic COCOMO.

(Refer Slide Time: 29:30)

References :

1. B. Hughes, M. Cotterell, R. Mall, *Software Project Management*, Fifth Edition, McGraw Hill Education (India) Pvt. Ltd., 2018.
2. R. Mall, *Fundamentals of Software Engineering*, Fifth Edition, PHI Learning Pvt. Ltd., 2018.

We have taken the references from these books, the details you can find in these books.

Thank you very much.