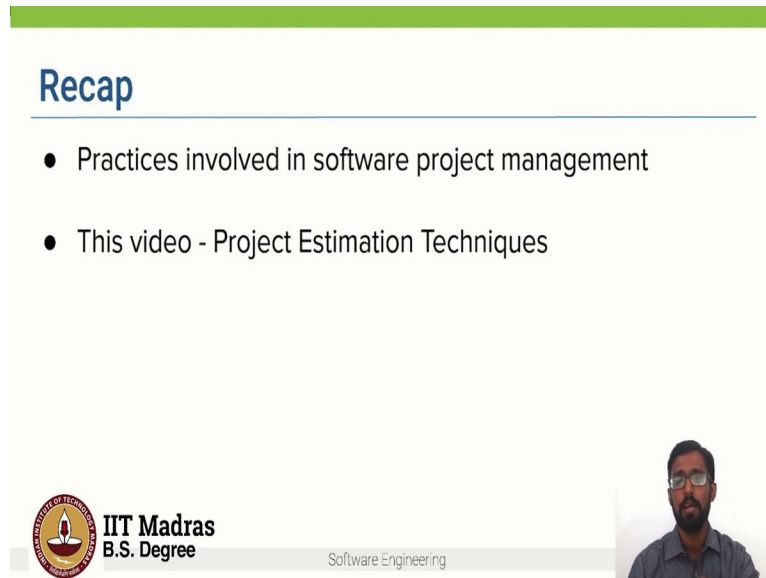



Software Engineering
Professor Sridhar Iyer
Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
Professor Prajish Prasad
Computer Science
FLAME University
Project Estimation Techniques

(Refer Slide Time: 00:24)




Recap

- Practices involved in software project management
- This video - Project Estimation Techniques

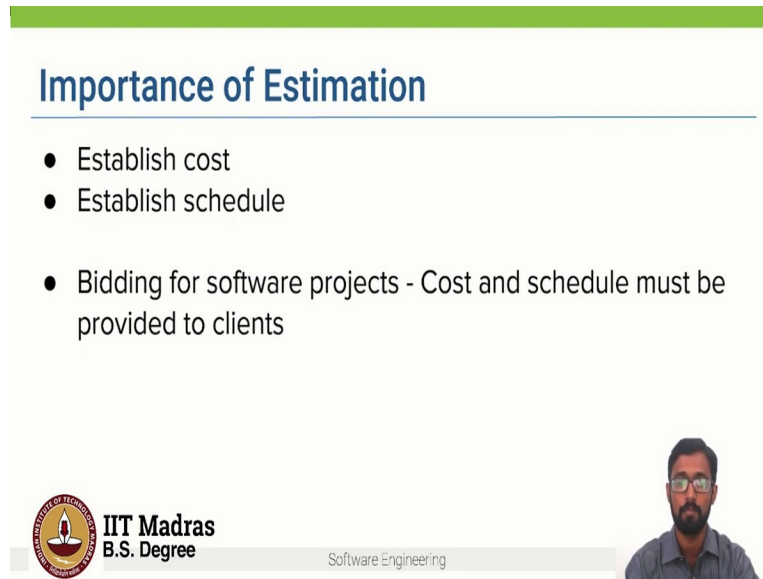
 **IIT Madras**
B.S. Degree

Software Engineering




In the previous video, we looked at the practices involved in software project management. So, what all are the important tasks that a software project manager does, such as forming and managing the development team, project scheduling, project estimation, risk management and so on. So, in this video, we will be talking about project estimation techniques. So, before we look at the actual techniques used for estimation, let us reflect on why it is important to do project estimation.

(Refer Slide Time: 01:00)




Importance of Estimation

- Establish cost
- Establish schedule
- Bidding for software projects - Cost and schedule must be provided to clients

 **IIT Madras**
B.S. Degree

Software Engineering




So, one is to establish the cost of the project. So, I need to know how much money is required to build the software and this information has to be communicated to the clients. Second, I also need to establish a schedule for the project, I need to know when the project is likely to get completed and as the project is moving along, is it on schedule, is it behind schedule, how many people are required, all these depends on the estimates of the project.

Also, from the client's perspective, when they bid for software projects, these cost and schedule estimates must be provided to the clients for the development contract. And of course, it is not possible to accurately establish or estimate the exact cost and the schedule and these estimates they depends on the level of information available about the project. So, the more detailed information we have about the project, the more accurate the estimation can be.


(Refer Slide Time: 02:24)

Reflection Spot

What can be estimated in a project? What are key parameter that you will provide an estimate for?




Please pause the video and written down your responses



IIT Madras
B.S. Degree

Software Engineering




So now, let us reflect on this question on what all can be estimated in a project. So, let us say you are running a company and you are bidding for software projects. So, what are the key parameters that you will provide an estimate for? So, what are the key parameters that are required to do project estimation? You can pause this video and think about some parameters before proceeding.

(Refer Slide Time: 03:08)


Key Estimation Parameters

- Size of code
 - Number of lines of code
 - KLOC - number of 1000 lines of code
- Effort
 - How many people are required in the team
 - Person-month - effort an individual can typically put in a month
 - E.g. -
 - A project requires 12 person-month of development time
 - 4 developers - 3 months
 - 2 developers - 6 months



IIT Madras
B.S. Degree

Software Engineering



So, let us look at some key estimation parameters. So, some of you might have thought about the size of the code that is the number of lines of code. Yes, this is important to estimate the

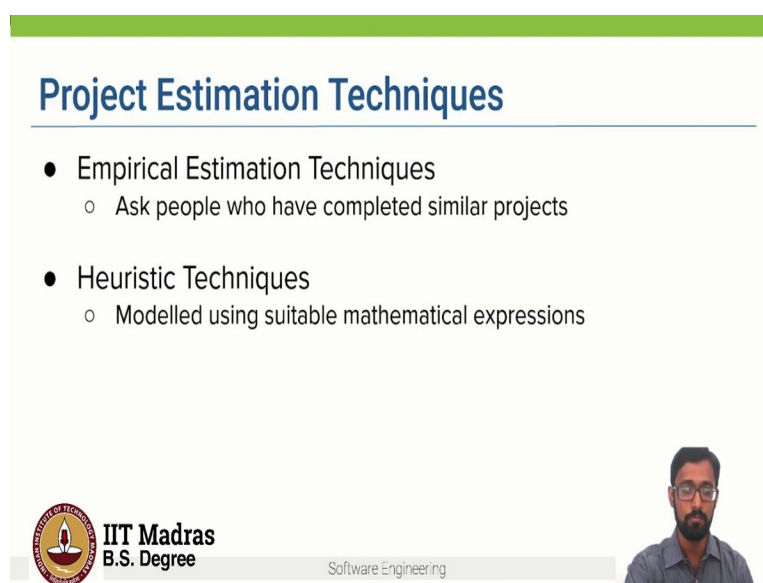
number of lines of code. And the number of lines of code is usually represented as KLOC, which is number of 1000 lines of code. So, 1 KLOC is 1000 lines of code.

So, this is one estimation parameter. But if you think about it, the size of the code alone cannot be provided to the clients to specially to justify the cost. So, another key estimation parameter is effort. And this is also an important parameter. And based on this we can determine how many people are required to form the team. And in the software industry effort is usually measured in terms of person-month.

So, one-person-month is the effort an individual can typically put in a month. So, let us take an example. Let us say project it requires 12-person-months of development time. So, let us assume that we have estimated this quantity, we will see how we can do this estimation but let us say we have got an estimate that this project requires 12 person-months of development.


So, what that means is if there are 12 people, then this project and if all these 12 people work on this project for one month, then this project can get completed. So, if we have only four developers that means this project will require 3 months. And if we have just two developers then it will require 6 months. So, you might think it is better to think in terms of total hours, how much as this entire project take, but work is often expressed in person-month units, because it is easier to get a feel for how many people might be needed to finish the development work in a fixed time.

(Refer Slide Time: 05:48)




Project Estimation Techniques

- Empirical Estimation Techniques
 - Ask people who have completed similar projects
- Heuristic Techniques
 - Modelled using suitable mathematical expressions

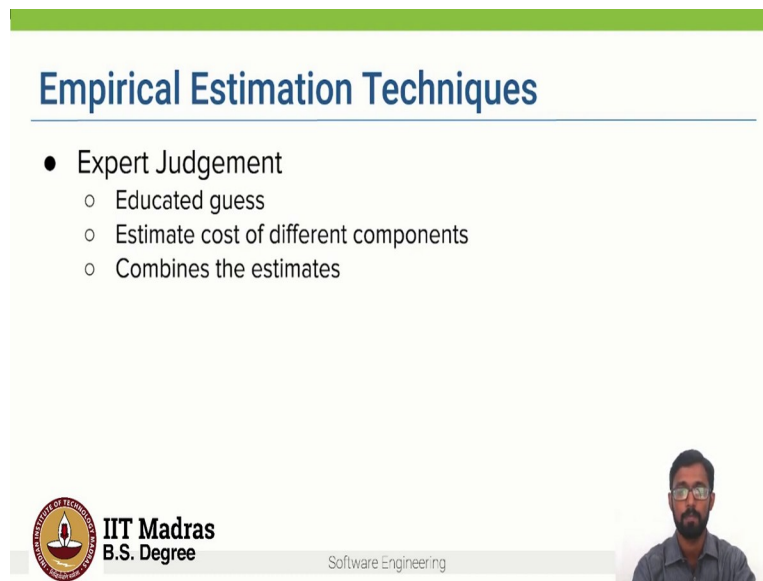
 **IIT Madras**
B.S. Degree

Software Engineering



So, now that we have looked at some key estimation parameters, let us look at project estimation techniques. So, one broad estimation technique is empirical estimation. And what this means is we ask experts, we ask people who have completed similar projects in the past and they provide estimates for us for this new project. Another type of estimation technique is using heuristics. And what this means is relationships that exist between different parameters like we saw effort, time, these can be modeled using suitable mathematical expressions. So, we will be looking at heuristic techniques also in this video.

(Refer Slide Time: 06:56)



The slide features a green header bar at the top. Below it, the title "Empirical Estimation Techniques" is displayed in a bold, blue font. A horizontal line separates the title from the content area. The content area contains a bulleted list with the following items:

- Expert Judgement
 - Educated guess
 - Estimate cost of different components
 - Combines the estimates


In the bottom left corner, there is a circular logo of IIT Madras, followed by the text "IIT Madras B.S. Degree". To the right of this, the text "Software Engineering" is visible. In the bottom right corner, there is a small video feed showing a man with a beard and glasses, wearing a blue shirt.

So now, let us look at some empirical estimation techniques, one popular empirical estimation technique is to use expert's judgment. So, an expert he or she makes an educated guess about the problem size after analyzing the problem thoroughly. They estimate the cost of different components in the system and then they combine all these estimates for the individual modules to arrive at the overall estimate.


(Refer Slide Time: 07:33)

Reflection Spot

What are the drawbacks of using expert judgement for estimation?




Please pause the video and written down your responses



IIT Madras
B.S. Degree

Software Engineering




So, now that we have seen what expert judgment is, what do you think are drawbacks of this method? What are the drawbacks of asking an expert to give their estimates for the project? You can pause this video and think about some drawbacks before proceeding.

(Refer Slide Time: 08:08)


Drawbacks of Expert Judgement

- Human errors, individual bias
- Optimistic estimates
- Overlook some factors, lack of adequate knowledge
- Solved by a group of experts?



IIT Madras
B.S. Degree

Software Engineering

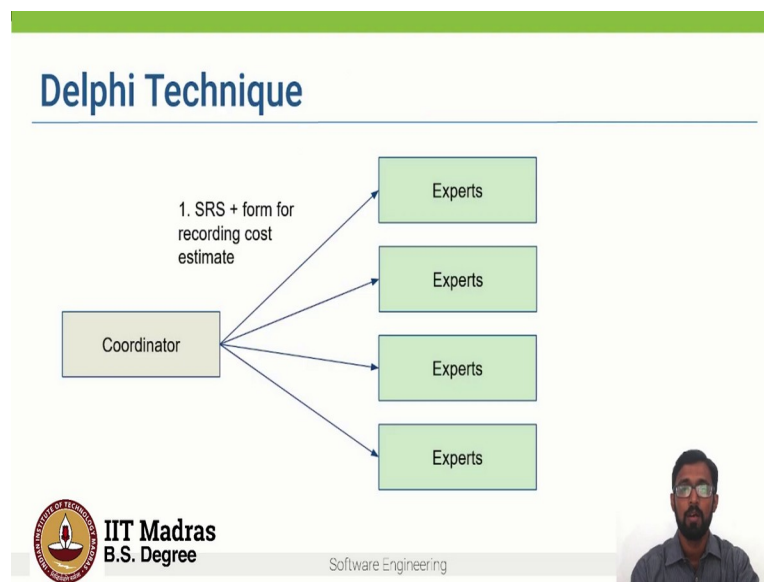


So now, let us look at some drawbacks of expert judgment. So, one is these experts are humans and hence they can make errors, they can have some individual biases towards a particular technology or a particular framework. And hence, because of this, they might give some optimistic estimates based on their expertise. And also, maybe if they want the contract to be won, then they might give even more optimistic estimates.

Another drawback is that they might overlook some factors, they might not have adequate knowledge in particular areas. So, for example, they might be very knowledgeable in the database and the user interface part, but they might lack knowledge about maybe additional modules or additional plugins, and because of this, they cannot accurately provide an estimation. So, but do you think this can be solved by a group of experts?

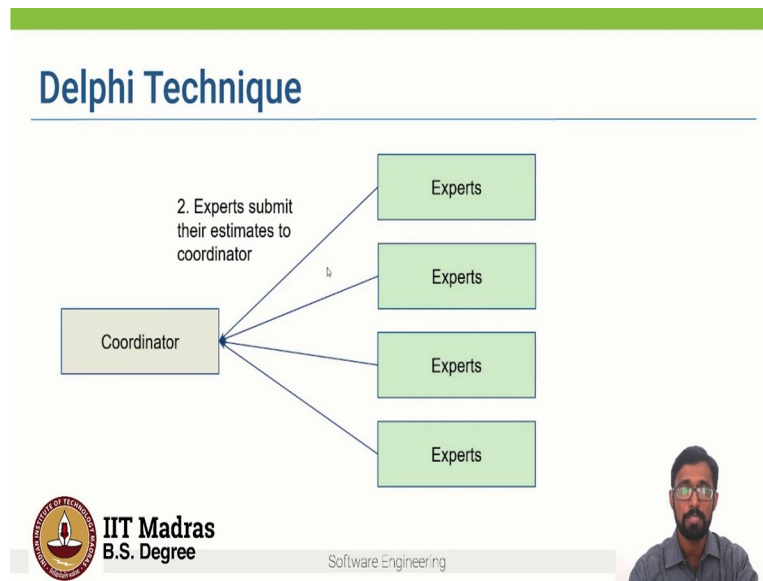
So, yes maybe to an extent, but still the entire group can also be biased. And the decision made by the group it may be dominated by some members who are overly assertive or maybe some senior members. So, they might influence the decision of the entire group.

(Refer Slide Time: 09:47)



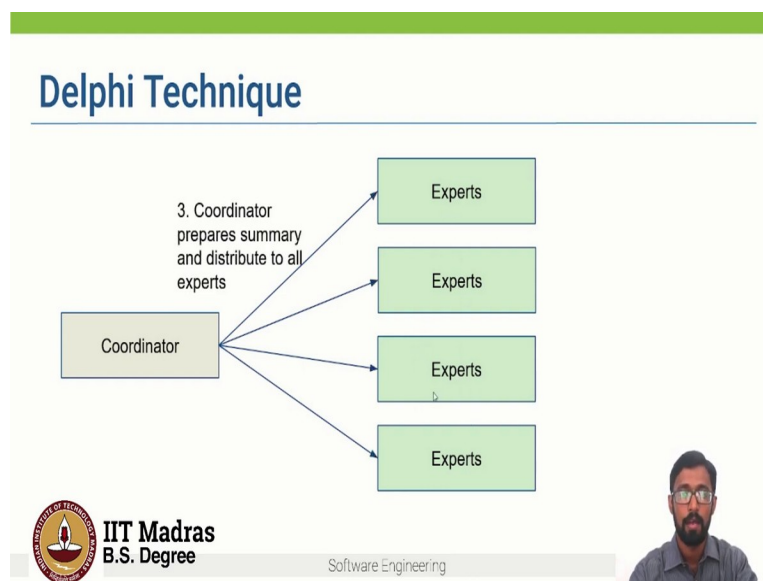
So, to overcome the drawbacks of expert judgment, the Delphi technique can be used. So, what is the Delphi technique? So, in this technique we have a group of experts and we have coordinator. So, the coordinator provides each expert with a copy of the SRS, the requirements and a form for recording the cost estimates. So, the experts go through the requirements, the SRS, the different modules and then they provide their estimates and they submit their estimates to the coordinator.

(Refer Slide Time: 10:18)



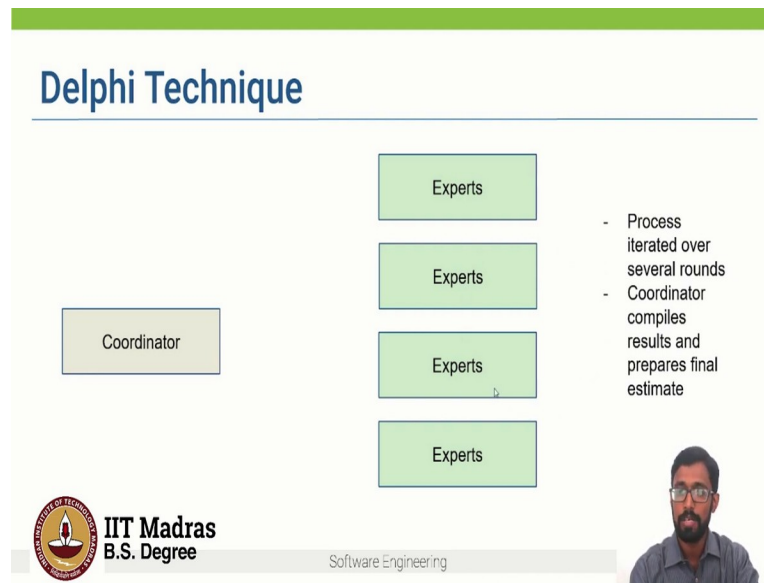
Note that each of these experts are submitting it individually they are not discussing with each other.

(Refer Slide Time: 10:32)



And what the coordinator does is the coordinator prepares a summary and distributes it to all the estimators to the experts. Now, the experts look at the summary and based on this they re-estimate based on all the comments from all the experts, and note that these comments are still anonymous experts do not know who has provided the estimation.

(Refer Slide Time: 11:01)




And this process is iterated over several rounds. And the coordinator compiles the results and prepares the final estimate. So, as we saw, there is no or no discussion between the experts is allowed during the entire estimation process. Because some experts or some people can get influenced by an experienced or a senior estimator. And after several rounds of iteration, the coordinator takes responsibility of compiling all the results and prepares the final list.

(Refer Slide Time: 09:41)


Project Estimation Techniques

- Empirical Estimation Techniques
 - Ask people who have completed similar projects
- Heuristic Techniques
 - Modelled using suitable mathematical expressions



IIT Madras
B.S. Degree

Software Engineering




Now, the next important type of project estimation techniques is known as heuristic techniques. So, in heuristic techniques we provide estimations by doing some modelling using some suitable mathematical expressions.

(Refer Slide Time: 10:02)


COCOMO Estimation Model

- Constructive Cost Estimation Model (COCOMO)
- Proposed by Boehm in 1981
- $\text{Effort} = a \times \text{SIZE}_k^b$



IIT Madras
B.S. Degree

Software Engineering



So, popular model which has been used to estimate effort is known as the constructive cost estimation model or the COCOMO model, this model was proposed by Boehm in 1981. And the formula which is used to estimate effort is effort is nothing but a multiplied by size raised to b where size is the size of the project in kilo lines of code.

(Refer Slide Time: 10:33)

COCOMO Estimation Model

- $\text{Effort} = a \times \text{SIZE}^b$
- a and b depend on the type of project
 - Organic
 - Well-understood application program
 - Team size - small and experienced
 - Semi-detached
 - Mix of experienced + inexperienced people
 - Embedded
 - Strongly coupled with hardware



IIT Madras
B.S. Degree

Software Engineering



And what is this a and b. So, the values of a and b it depends on the type of project or the type of software you are building. So, Boehm, he classified projects as one organic. So, an organic project involves building a very well understood application program and the size of the team is also small and all most of them are experienced. Another type of project can be a semi-detached project, which has a mix of both experienced and inexperienced people and they might have some limited experience building related systems.

And the third type of project is an embedded project where the project is strongly coupled with hardware and the team size also might be large and there might be many inexperienced people. So, based on the type of project there we assign different values for a and b, and what we have seen is that the effort it not only depends on the size of the project, but also the type of project, the composition of the team is also an important factor and which the estimate depends on.

(Refer Slide Time: 12:08)

COCOMO Estimation Model

- Organic:
 $\text{Effort} = 2.4 \times (\text{SIZE in KLOC})^{1.05} \text{ Person Month}$
- Semi-detached:
 $\text{Effort} = 3.0 \times (\text{SIZE in KLOC})^{1.12} \text{ Person Month}$
- Embedded:
 $\text{Effort} = 3.6 \times (\text{SIZE in KLOC})^{1.20} \text{ Person Month}$



IIT Madras
B.S. Degree

Software Engineering

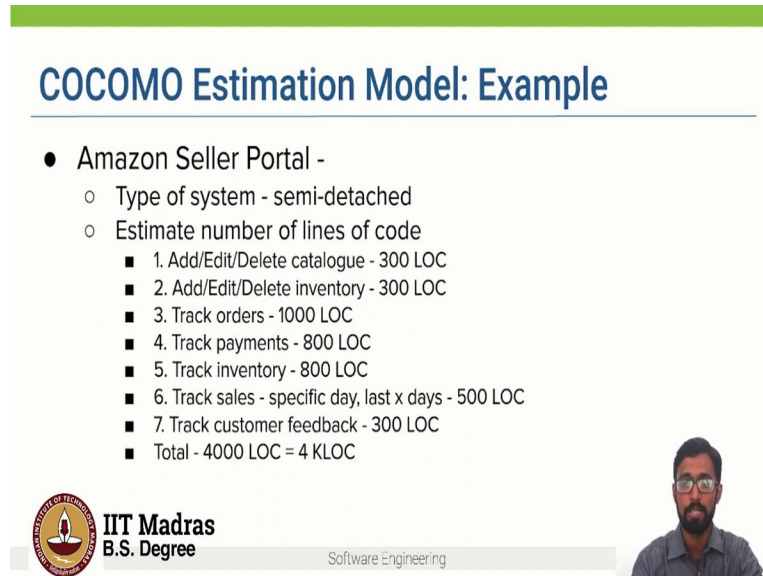


So, based on an empirical analysis of several projects Boehm came up with these values for a and b for each of the types of projects. So, you can see that the values of the constants of a

and b, they are based on the type of project. And for organic projects, the values are the least, it is 2.4 and 1.05. And it is highest for embedded projects.

And this makes sense because embedded projects as we saw it involves coupling with hardware, the team is large, there might be many inexperienced people and because of this effort increases exponentially as the complexity of the project and the team increases. So, based on these constants, the initial estimate or the nominal estimate is calculated in terms of the effort.


(Refer Slide Time: 13:18)




The slide is titled "COCOMO Estimation Model: Example" in blue text. It lists the details for an "Amazon Seller Portal" project. The project is semi-detached and has an estimated 4000 LOC. The modules and their LOC are: 1. Add/Edit/Delete catalogue (300 LOC), 2. Add/Edit/Delete inventory (300 LOC), 3. Track orders (1000 LOC), 4. Track payments (800 LOC), 5. Track inventory (800 LOC), 6. Track sales - specific day, last x days (500 LOC), 7. Track customer feedback (300 LOC). The total is 4000 LOC = 4 KLOC. The slide also features the IIT Madras B.S. Degree logo and a small photo of a man.

COCOMO Estimation Model: Example

- Amazon Seller Portal -
 - Type of system - semi-detached
 - Estimate number of lines of code
 - 1. Add/Edit/Delete catalogue - 300 LOC
 - 2. Add/Edit/Delete inventory - 300 LOC
 - 3. Track orders - 1000 LOC
 - 4. Track payments - 800 LOC
 - 5. Track inventory - 800 LOC
 - 6. Track sales - specific day, last x days - 500 LOC
 - 7. Track customer feedback - 300 LOC
 - Total - 4000 LOC = 4 KLOC

 **IIT Madras**
B.S. Degree

Software Engineering



Now, let us take the example of the Amazon Seller Portal and try to come up with the initial estimated effort to build such a system. The first thing which we need to identify is what type of system is this Amazon Seller Portal. So, according to me, this is not a straightforward software system, there are several modules which have to interface with the main Amazon system with the banks for payment, etcetera.

So, I would maybe classify it as a semi-detached system. In some cases, it can also be an embedded system. So, now that we have identified what type of system it is, now, let us try to second thing to do is to estimate the number of lines of code. So, recall that we had identified some requirements in the previous weeks. So, we have adding, editing, deleting catalog, inventory, tracking orders, payments, sales, customer feedback.

So, all of these things are there. And for each of these modules, I estimate the number of lines of code required. So, how does one do it. So, this itself is a very difficult task. And usually the project manager looks at data from previous projects to come up with these estimates. So, for example, in a previous project, there would have been a module which captured customer feedback and based on that the software manager makes estimates for that module. So, I add up all the lines of codes for all the module and I get the total of 4000 lines of code which is nothing but 4 kilo lines of code.

(Refer Slide Time: 15:22)

COCOMO Estimation Model: Example

- Amazon Seller Portal -
 - Type of system - semi-detached
 $\text{Effort} = 3.0 \times (\text{SIZE in KLOC})^{1.12} \text{ Person Month}$
 - Estimate number of lines of code - 4 KLOC
 - $\text{Effort} = 3.0 \times (4)^{1.12} = 15.83 \text{ Person Months}$ - Initial Estimate
 - $\text{Cost} = 16 \times 75,000 = \text{Rs. } 12 \text{ lakhs}$



IIT Madras
B.S. Degree

Software Engineering



So, now, let us look at the estimations which we have till now. We know that the system is semi-detached, we have the size in kilo lines of code, we plug these values in and we get the effort is roughly around 16 person-months or 15.83 person-months. So, what does this mean? This is our initial estimate. So, we require around 15.83 person-months that is around 16 people working full time can finish this project in one month. And based on this we can estimate the cost. So, let us say that the average salary of the 16 people is 75,000 per month. So, then the total cost comes up to around 12 lakhs.

(Refer Slide Time: 16:35)

Reflection Spot

Apart from the size of the code, what other factors have to be considered to estimate **effort required for a project**?



Please pause the video and written down your responses



IIT Madras
B.S. Degree

Software Engineering



So, now, apart from the size of the code, we need to consider other factors also to estimate the effort required for a project. So, let us reflect on this. So, what are the other factors which have to be considered to estimate the effort required for a project? You can pause this video and think about some other factors before proceeding.

(Refer Slide Time: 17:15)

Effort Estimation Parameters

- People working in the project
- Technical attributes of the project
- Tools and practices used by the team



Software Engineering



So, apart from the size of the project, so, what other factors have to be considered? Let us look at that. So, some of you might have said about the people working in the project. Some team members may be experience, but not in the programming framework required for this project. Others may not have domain experience. So, in this case, in the case of the Amazon Seller Portal, you know how an E-commerce application works, and hence some time we will go in learning about the domain.

So, another factor can be the technical attributes, the finer details about the product. So, is the database large, is it complex? How complicated is the entire product? So, is the reliability an important part of the system? So, these factors also have to be considered when we estimate the effort. And estimation also depends on the tools and practices which are used by the team. If the practices and tools are not very effective, then maybe more effort is required for the project.

(Refer Slide Time: 18:39)

Cost driver attributes

Cost Drivers	Rating				
	Very Low	Low	Nominal	High	Very High
Product Attributes					
RELY, required reliability	.75	.88	1.00	1.15	1.40
DATA, database size		.94	1.00	1.08	1.16
CPLX, product complexity	.70	.85	1.00	1.15	1.30
Computer Attributes					
TIME, execution time constraint			1.00	1.11	1.30
STOR, main storage constraint			1.00	1.06	1.21
VTTR, virtual machine volatility		.87	1.00	1.15	1.30
TURN, computer turnaround time		.87	1.00	1.07	1.15
Personnel Attributes					
ACAP, analyst capability	1.46	1.19	1.00	.86	.71
AEXP, application exp.	1.29	1.13	1.00	.91	.82
PCAP, programmer capability	1.42	1.17	1.00	.86	.70
VEXP, virtual machine exp.	1.21	1.10	1.00	.90	
LEXP, prog. language exp.	1.14	1.07	1.00	.95	
Project Attributes					
MODP, modern prog. practices	1.24	1.10	1.00	.91	.82
TOOL, use of SW tools	1.24	1.10	1.00	.91	.83
SCHED, development schedule	1.23	1.08	1.00	1.04	1.10

- Initial estimate = 15.83 PM
- Effort adjustment factor = $1.40 \times 1.08 \times 1.13 \times 0.95 = 1.62$
- Final estimate = $15.83 \times 1.62 = 25.65 \text{ PM}$



Taken from - A Concise Introduction to Software Engineering - Pankaj Jalote - Table 4.1 pg 72

Software Engineering



So, in the COCOMO model after determining this initial estimate, some other factors are added in order to obtain the final estimate and these factors are known as cost drivers. So, this

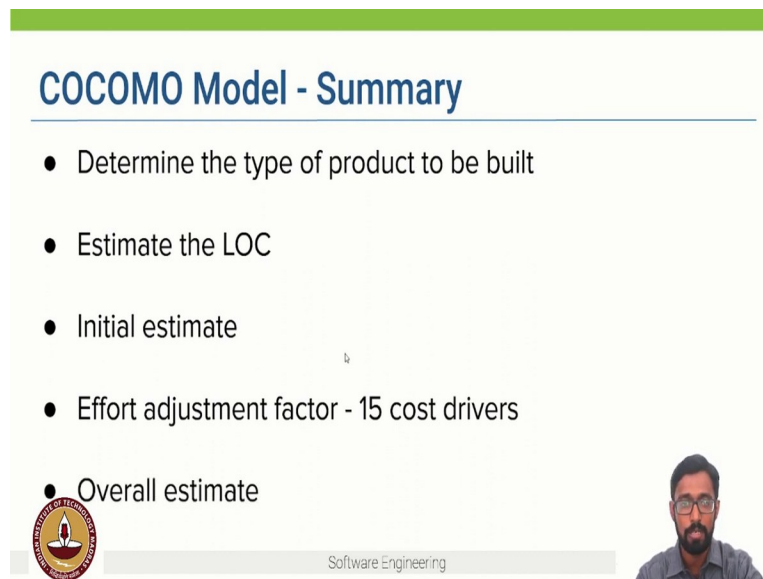
table shows 15 such cost drivers. And these attributes like the product, computer attributes, personal attributes, these are similar to what we discussed earlier. Like software reliability, so if the reliability required is very high, then we have a higher cost driver compared to when it is very low. So, let us look at our example.

So, let us say we want to estimate the effort for the Amazon Seller portal. Let us look at some of these attributes. So, for example, we want the system to be highly reliable because orders and money are involved. So, we will choose a cost driver which is very high for reliability. And maybe my database size it is also quite large. So, I will choose a cost driver which is high.

Now, if we look at some personal attributes, let us say my application experience of my team is low. So, I will choose this as 1.13, which is higher than when if it is very high. And let us say my team has good programming language experience, so I rate it as high. So, multiplying, so the multiplying factor for all 15 cost drivers, they are multiplied to get the effort adjustment factor.

So, in this case, I am just taking four drivers for simplicity. So, let us recall that our initial estimate was 15.83 person-month. Now, I take these cost drivers multiply all of them together. And this gives me the effort adjustment factor, which is 1.62. And now, my final estimate is calculated by multiplying the initial estimate with this effort adjustment factor. And so, I get final estimate of 25.65 person-months.

(Refer Slide Time: 21:23)



The slide is titled "COCOMO Model - Summary" in blue text. It contains a bulleted list of five steps: "Determine the type of product to be built", "Estimate the LOC", "Initial estimate", "Effort adjustment factor - 15 cost drivers", and "Overall estimate". At the bottom left is a circular logo with a lamp and the text "UNIVERSITY OF TECHNOLOGY". At the bottom center is a grey bar with the text "Software Engineering". At the bottom right is a small video feed of a man with a beard and glasses.

- Determine the type of product to be built
- Estimate the LOC
- Initial estimate
- Effort adjustment factor - 15 cost drivers
- Overall estimate

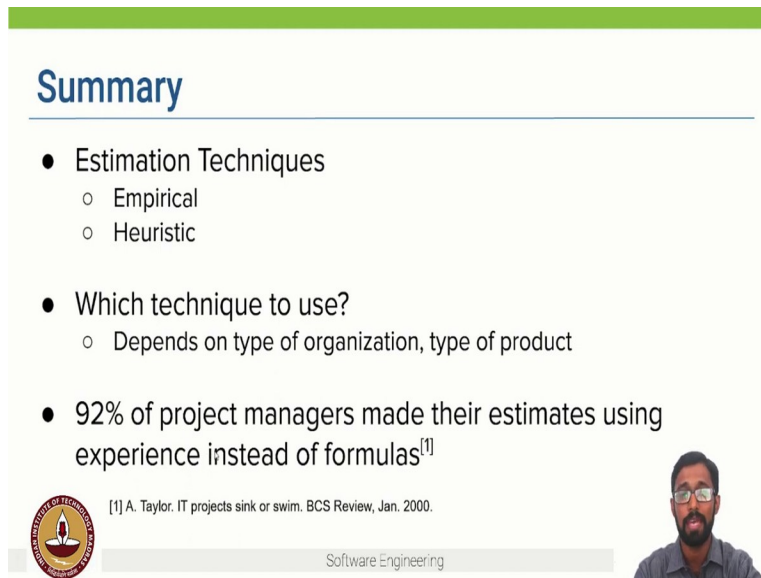
UNIVERSITY OF TECHNOLOGY

Software Engineering

So, to summarize the COCOMO model, first we determine the type of product which has to be built. Then we estimate the number of lines of code for each module. And based on the formula, we get the initial estimate. And from this initial estimate, we calculate the effort adjustment factor by considering the 15 cost drivers and then we multiply the initial estimate with these adjustment factors to get the overall estimate.


So, if you have noticed one important thing to note is that the overall estimate it depends on the estimation of the lines of code for each module. And so, it is important to get as much detail about the software and we should also be aware of our biases when estimating the size of various components.

(Refer Slide Time: 22:33)




Summary

- Estimation Techniques
 - Empirical
 - Heuristic
- Which technique to use?
 - Depends on type of organization, type of product
- 92% of project managers made their estimates using experience instead of formulas^[1]

 [1] A. Taylor: IT projects sink or swim. BCS Review, Jan. 2000.

Software Engineering



So, to summarize, in this video, we looked at several estimation techniques both empirical and heuristic. And another question now to ask is which technique to use for a particular project. So, what estimation techniques project managers use it depends on various factors like the type of organization, the type of product which is being built. And it depends on the practices in your organization as well.

So, for example, the British Computer Society Survey of more than 1000 projects, found that 92 percent of the project managers they made their estimates using experience instead of formulas. So, now that we have learned these estimation techniques, you will encounter these or maybe some variations of these as you start working in the software industry.