Software Project Management
Prof. Rajib Mall
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

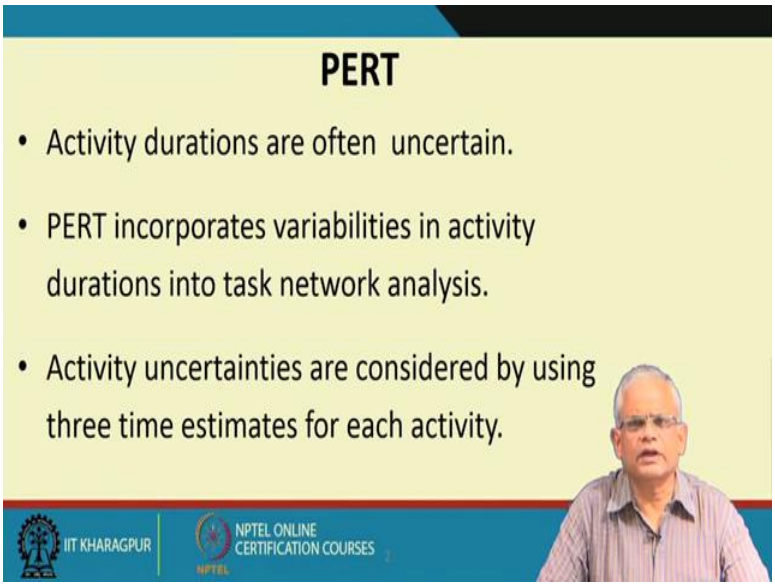
Lecture – 31
PERT, Project Crashing

Welcome to this lecture, in the last lecture we had looked at PERT CPM and how to identify determine the project characteristics, various characteristics of the activities critical path, critical tasks and so on.

(Refer Slide Time: 00:29)

In this lecture, we will look at PERT where statistical times for tasks are considered. We will just look at an overview idea of that, we will not go into details and then we will look at project crossing. That is if the customer wants the project duration to be reduced how does the project manager go about doing it and then we look at team management. Let us start with the today's topics. So, these are the PERT Project Crashing and then team management that is the plan for this lecture.

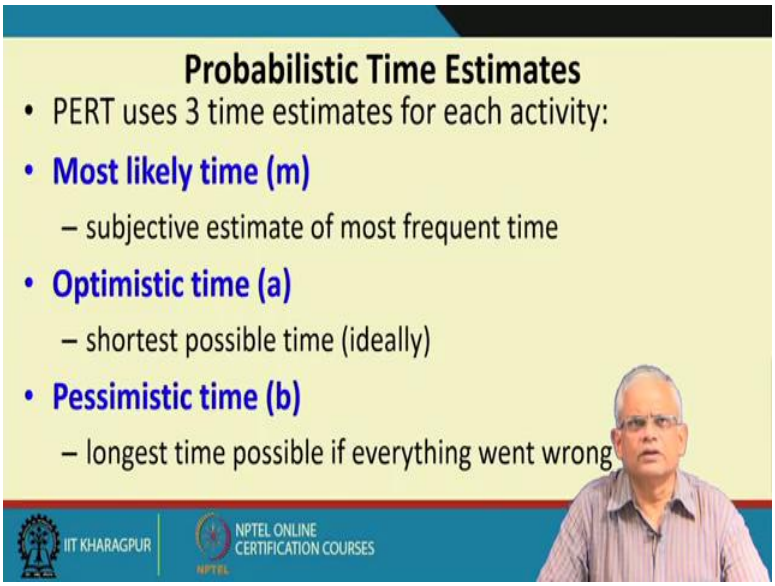
(Refer Slide Time: 01:17)



We had said earlier that in the PERT CPM, the task durations are deterministic; it is simple that all the tasks have estimated duration. But then often in development work things are uncertain; in a routine work like maintenance and so on; the durations are known quite accurately.

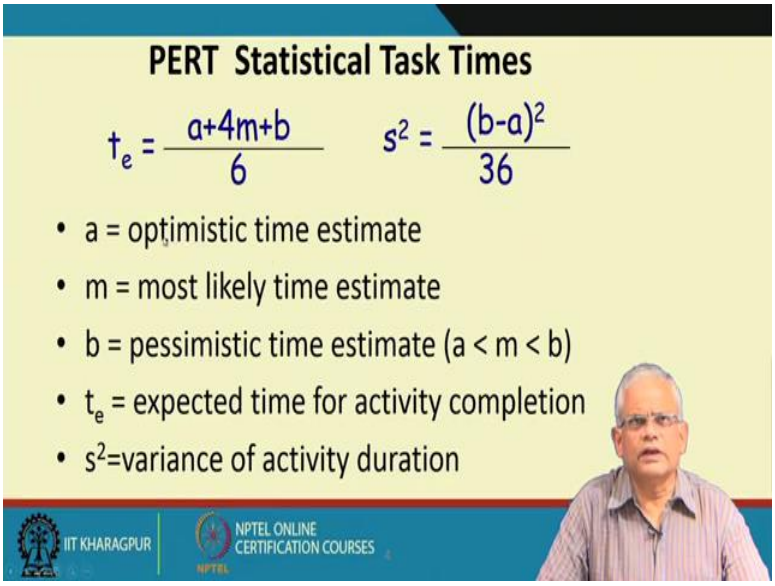
But in a development work often the project manager; it becomes hard to estimate the exact duration of activities. Only statistical values for activities can be given and the PERT can be used for that. Here the activity durations are uncertain and variability of the activity durations; that is probabilistic variations can be handled in PERT. Here there are three time estimates that are given to each activity; we will look at the three estimates and based on that various project parameters and the critical path and so on are determined.

(Refer Slide Time: 02:55)



The three probabilistic time estimates for the activities are the most likely time which we indicate by m. The optimistic time that is given that everything goes alright; the task will get completed in time a, this is the shortest possible time. And then we have b which is the pessimistic time that is if things do not work out obstacles appear and so on. This is the worst case time for the activity b; b is the worst case, a is the optimistic and m is the most frequent time.

(Refer Slide Time: 03:51)



And based on that; the task estimated time is given as a plus 4 median plus b by 6

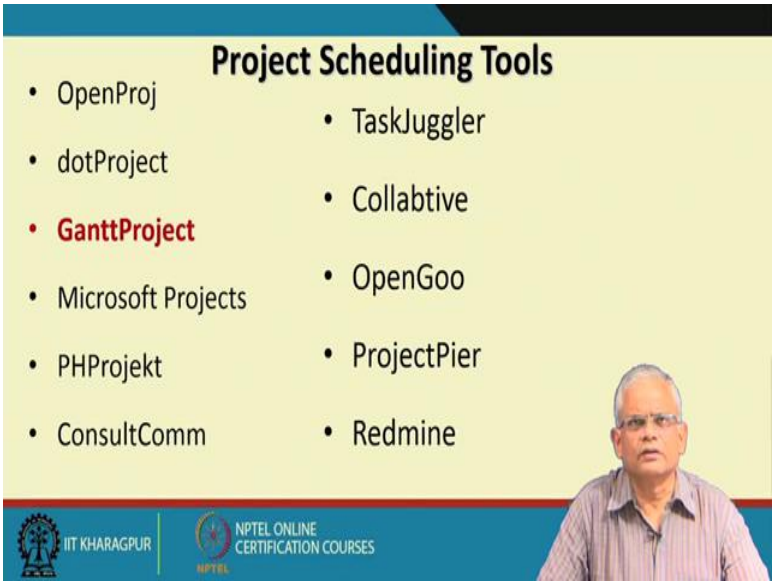
$$t_e = \frac{a + 4m + b}{6}$$

and the variance is given by

$$s^2 = \frac{(b-a)^2}{36}$$

b minus a whole square by 36. And based on these statistical parameters inferencing is done just like the way we were using in the PERT CPM.

(Refer Slide Time: 04:23)



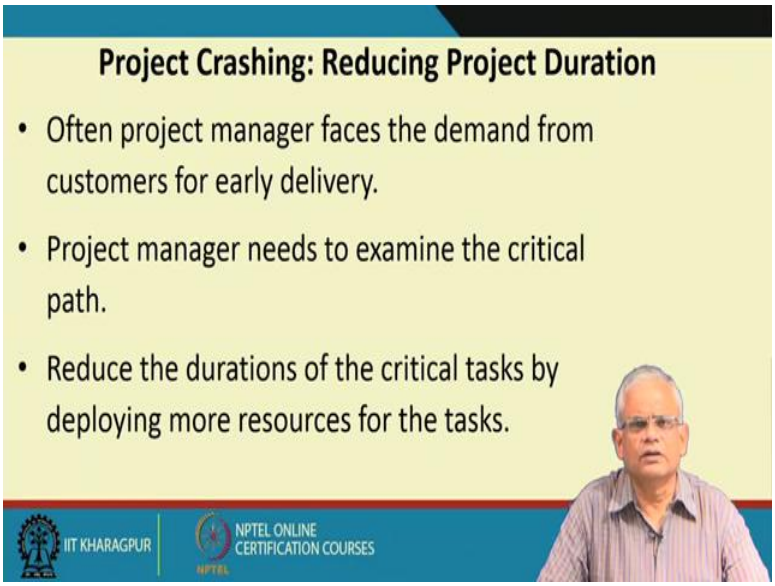
As I said that we will not go details into the statistical calculation of project times, but there are several tools available; where these can be done easily the PERT CPM and PERT can; there are large number of tools are available many are open source and some are price tools. But then some tools which we have used it is easily downloaded, takes very less space, does meaningfully well is the Gantt project. And if it is a task which is; Gantt project is typically the project manager uses alone on a desktop standalone machine.

But if there are multiple people who would like to use like the team members would like to examine their task characteristics, the tasks assigned to them, give inputs and so on then Redmine is a tool which is again open source; Gantt project in redmine. Redmine

runs on a server and there can be users and different computers whereas, Gantt project is a very simple tool, do not even need a user manual you can start right away using it.

The project characteristics are computed automatically the critical path is shown as you input. And of course, there are many other tools that are available; it will be nice as part of this course you can download some of these tools and get used to it. If you get used to one tool you will see that it becomes very easy to use the other tools.

(Refer Slide Time: 06:33)



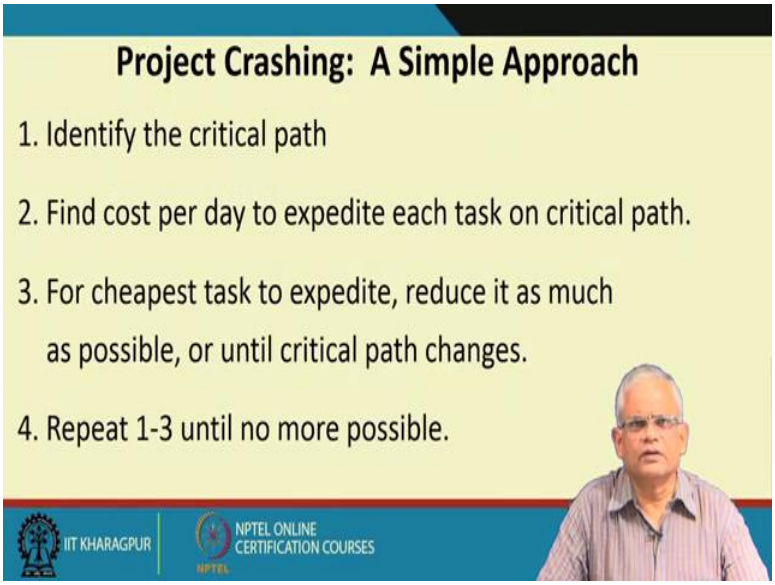
Now, let us look at project crashing that is common problem that the manager estimate certain time; let us say 6 months. And then the customer says that 6 month is too long can it be done in 5 months? Then what does the project manager do?

The project manager tries to reduce the project duration that is called as project crashing. As we have already discussed that the longest path in the task network is the critical path. To reduce the project duration, we need to reduce the time for the critical path, but then as we reduce the total duration for a critical path; we will see that there are other critical paths that appear that has to be considered. But how does the project manager reduce the duration of the critical tasks that appear on the critical path?

Of course, by deploying more resources if there was one tester testing was the task; then might deploy more testers, coding might deploy a additional coders. By assigning more resources to a task the duration of the critical task can be reduced, but then which critical

task to take up. If the requirement is let say 15 days reduction in a 6 month project which critical task to take up?

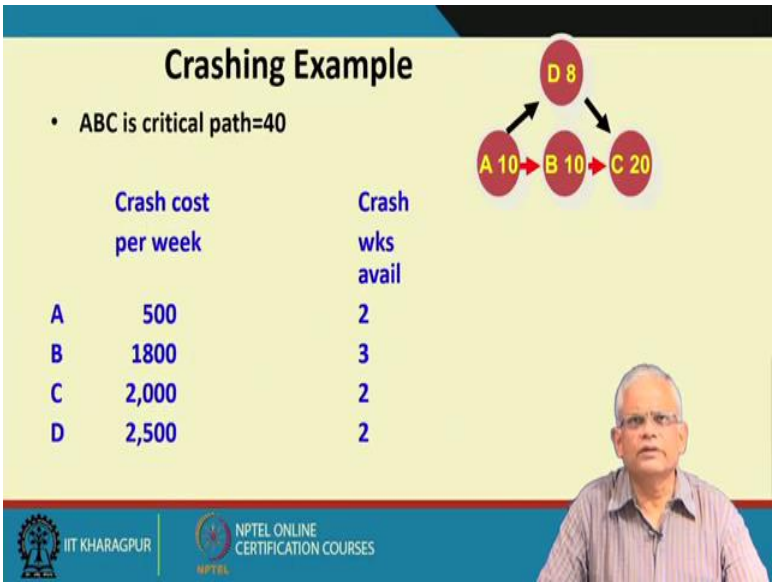
(Refer Slide Time: 08:39)



We can do that by a simple approach for project crashing. First is we must identify the critical path and then look at all the critical tasks on the critical path and then find the cost per day to expedite each task on the critical path. And then identify the cheapest task to expedite and then gradually reduce it, but then it may so happen that as we reduce it by a day or 2 day, 3 day etcetera we will find that the critical path changes.

The critical path which was let say 6 months and as it becomes 5 months 25 days; it may not remain the critical path, there maybe another path which becomes critical. And then we need to reduce that critical path the current critical path after reduction. And we keep on doing this steps until no more reduction is possible or the target is met.

(Refer Slide Time: 10:03)



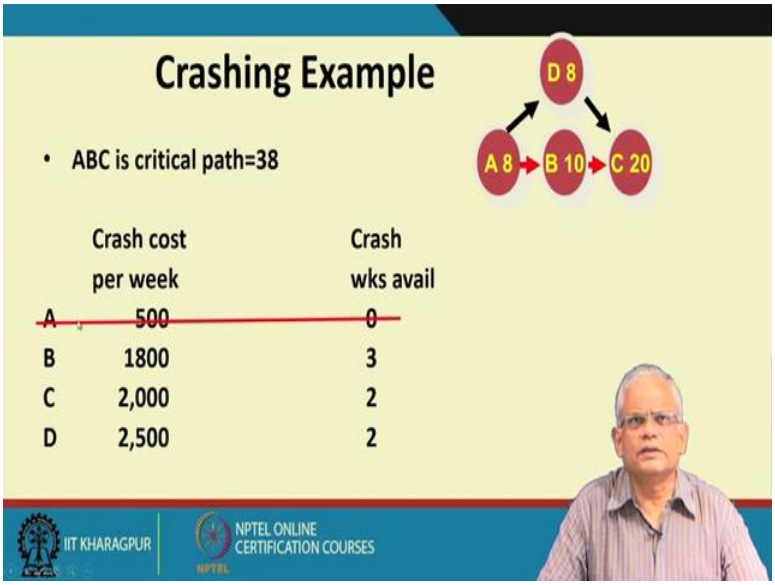
Just to give a small example of the crashing; let say a very simple projects with a four tasks A, B, C, D. The task durations are 10, 10, 20 and 8; we can easily see the critical path shown here in red the duration is 40 that is A, B, C is the critical path with 40 weeks as the duration. And then the project manager needs additional information that deploying additional resources here costs how much? The crash cost for a is 500 rupees, but it can at most be reduced by 2 weeks; it will not be possible to do it in less than 8 weeks. Task B, 1800 is the cost per week, but then it can at most be reduced by 3 weeks.

Tasks C, 2000 is the cost per week and only it can be reduced by 2 weeks and task D is 2500 per week and can at most be reduced by 2 weeks. Obviously, the project manager will choose A to reduce at first reduce it by 1 day; A becomes 9, B 10 and C 20. The critical path becomes 39, the overall project duration reduces by 1 day with an expense of 500 and the critical path does not change because the other critical path is 38.

Now, reduce it by 1 more day the critical path becomes 38, but the other path becomes 36; so still the critical path does not change. The next task to reduce possibly is 18; sorry B and the cost is 1800; now let us reduce it by 1; the critical path becomes 8 plus 9 plus 20; 37 and still remains the critical path the project duration reduces to 37. And we reduce it by one more it becomes 36 and there are two critical paths now; we have to reduce both just reducing this; does not reduce the project duration you need to reduce D as well.

So, this is basically the approach the project manager takes to reduce the project duration or the project crashing.

(Refer Slide Time: 13:35)

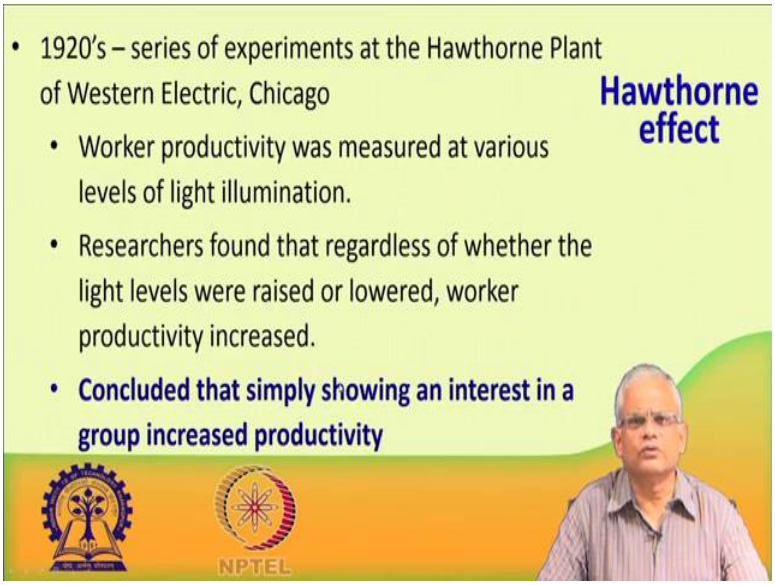


A is reduced as much as possible because that is the cheapest task to reduce and the duration for a becomes 8, the critical path becomes 38; the project duration is 38 now and the sub critical path is 36. Now, let us look at the topic of team management; how does the project manager go about doing the team management.

(Refer Slide Time: 14:13)

To manage the project team; the project manager has to put in daily effort, has to track the team member performance on a day to basis day to day basis, motivate the team members to improve their performance. Provide feedback which can help them and if there are issues and conflicts among the team members these need to resolved; so that the overall project performance improves.

(Refer Slide Time: 14:55)

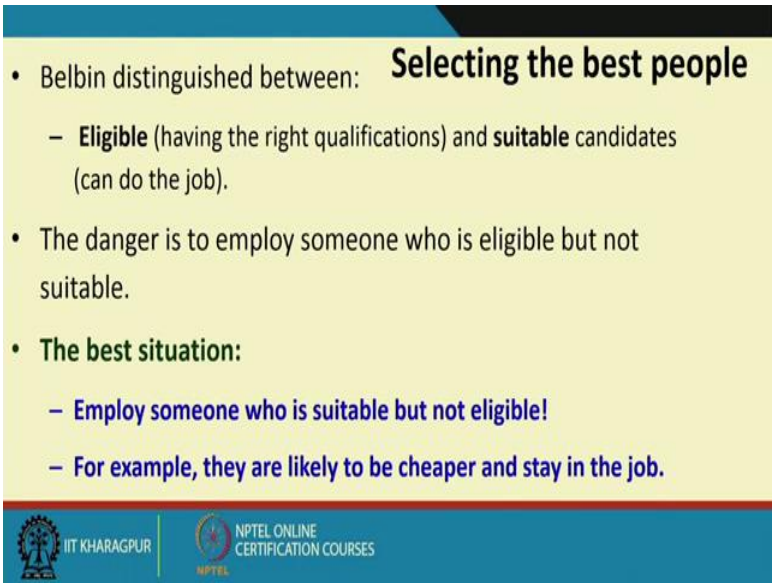


The project manager needs to be aware of some theories and observations which have been made over the last century or so; these are very fundamental results applicable even now and the project manager needs to be aware of this. One is the Hawthorne effect; way back in 1920; a series of experiments were conducted at the Hawthorne plant in western electric, Chicago. Here the idea was that how does the team performance change as the lighting conditions improve? More lights were added it was made very bright room, lights were reduced made relativity darker, but then the surprising observation the thing was actually repeated across many workers.

But something that was clearly identified was that regardless whether the light levels were raised or lowred; the productivity always increased, surprising that not only that the light level did not matter so much, but also for both cases the whether the light levels were raised or lowered; the productivity increased. What can be the cause for this? It was concluded that since the workers were under observation, their productivity was being measured; they were paid attention that is what increase the productivity.

The conclusion from here the Hawthorne effect is that; if the workers are given attention, the manager pays close attention that what they are doing their productivity increases. Simply showing an interest in a group increased it productivity.

(Refer Slide Time: 17:35)

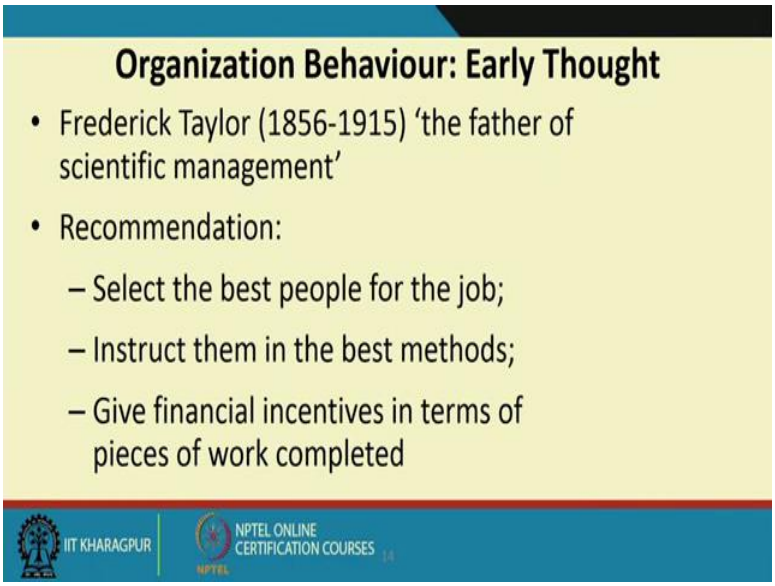


The manager has to select the best people for the project, here we must note that Belbin; he distinguished between eligible and suitable candidates. Eligible candidates are the ones who have the right qualification and suitable candidates are the one who can do the job.

May be the eligible candidate may not be suitable and a candidate who is not eligible; does not have the right qualification may be actually the suitable candidate. But the danger is to employ somebody who is not eligible sorry who is eligible, but not suitable. As per educational qualification other requirements the candidate is eligible, but then for the project is not suitable; the project suffers and also the candidate suffers, it does not help anyone.

But the best cases that employ someone who is suitable, but not eligible; they would be cheaper because they do not have the right educational qualification and they will stay on the job they will be motivated. So, this is the best case to employ somebody who is not eligible, but suitable, but then how to identify that is the; that is a problem.

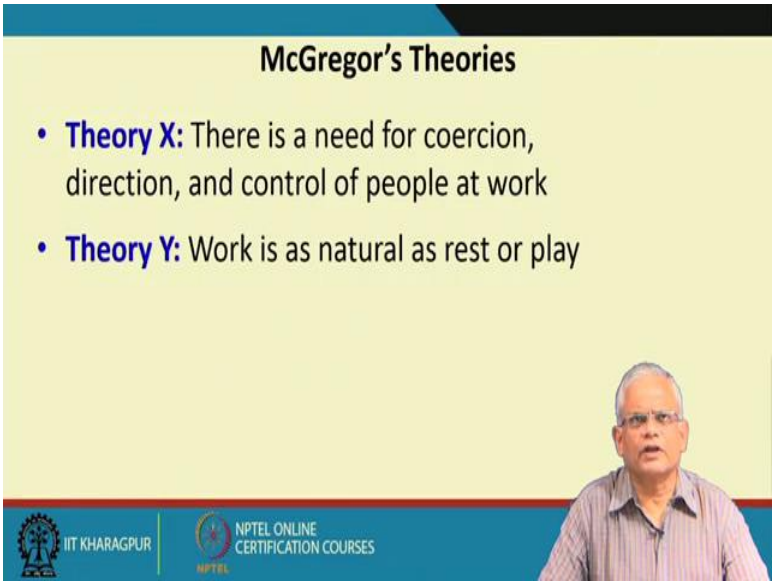
(Refer Slide Time: 19:13)



One of the early thoughts in organization behaviour was by Frederick Taylor; long back he recommended that select the best people for the job. And then instruct them in the best methods and then give financial incentives in terms of pieces of work completed.

Of course, this was for a manufacturing job for a software job it is not so well defined that what do you mean by pieces of work completed. But still let us see what he really meant that select the best people for job instruct them in the best methods and give financial incentive quite intuitive actually.

(Refer Slide Time: 20:17)

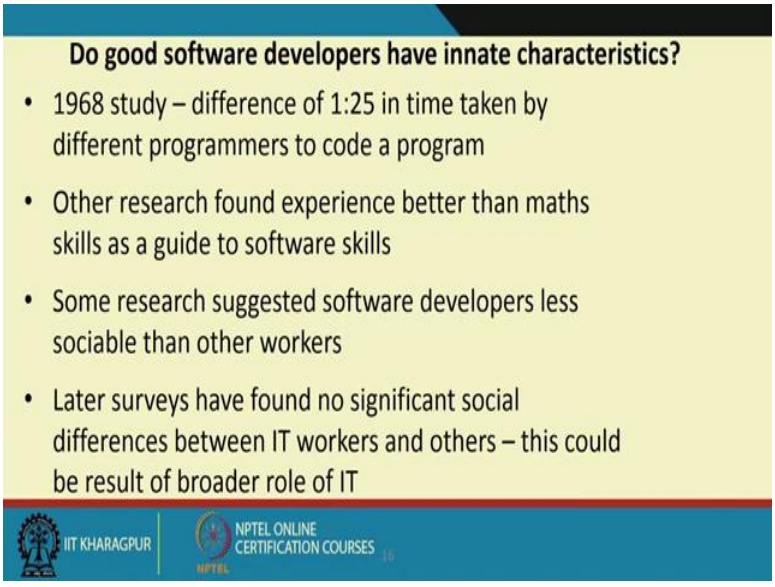


But then McGregor, he propounded two theories; theory X, there is no need for coercion, direction and control there is a need for coercion direction and control of people at work. He assumed that whenever the manager finds the worker; he needs to coerce monitor closely ask them to do the work because the workers by themselves do not work. They have a tendency to laze around not work and therefore, the manager needs to have coercive techniques, control the people.

But then the theory Y is a different type of manager who assumes that workers they love to work. Work is as natural as rest or play just needs to encourage them; these are two different ways in which managers work. One is the theory X manager; who assumes that people always cheat, they are not motivated, they will not work unless they are scolded they are forced and so on.

And theory Y is the manager assumes that the workers are efficient; they love to work. It will be very easy to spot what is the type of the manager if you visit a team and find that the day the manager is absent; if you find that everybody is relaxing feeling happy that the manager did not come and then that is a theory X manager; who is a dominating things that the people do not work, he needs to tell them to work and so on. And theory Y manager if you visit and see that even when the manager is not there; the workers are working as usual then the manager; you can think of him as a theory Y manager.

(Refer Slide Time: 22:49)



But do the software developers have some characteristics which make them do a good job; because that will give us a hint how to choose a good developer. In a very old study, 1968; the difference in time taken by different programmers to code the same program is 1 is to 25. Somebody who takes 1 hour to code a program, the other programmer takes 25 hours and these are employed by the same organization.

So, the proficiency level of the programmers vary widely that was the conclusion the study in 1968. Obviously, this can be interpreted to mean that during the selection; we need to find out who is a good coder, codes very fast writes good code. But then we also need to remember that nowadays software development is not just coding; coding is a small part of the activity may be in 1968; coding was on one of the major activities in program writing, but now coding is a small part of the activity of a software engineer.

For example, finding out which reusable libraries to use, which tools to use etcetera. So and of course, testing, design, requirements analysis and so on. So, this study is an indicator, but then we must remember that the times have changed; now we have not only coding, but other activities, but then still we can interpret this, to say that there is a wide variation in the competency of different engineers in a project. And we need to identify who is a good developer who can help the project and select the right people.

One of the study early study found that those who were good in maths; they had good software development skills, but then later it was found that this not may not be so. Another study found that the software developers are less sociable than other workers. But then later surveys found no social difference between IT workers and other; maybe when the study was done, it was the case where the programmer has to just write the code that was the major activity.

But now as you are mentioning software development has many other activities and possibly that may be the result that the developers have a broader role now and they have to be social. We are almost at the end of the time at this lecture, we will stop here and we will continue in the next lecture.

Thank you.