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

Lecture - 34
Team Structure (Contd.) and Risk Management

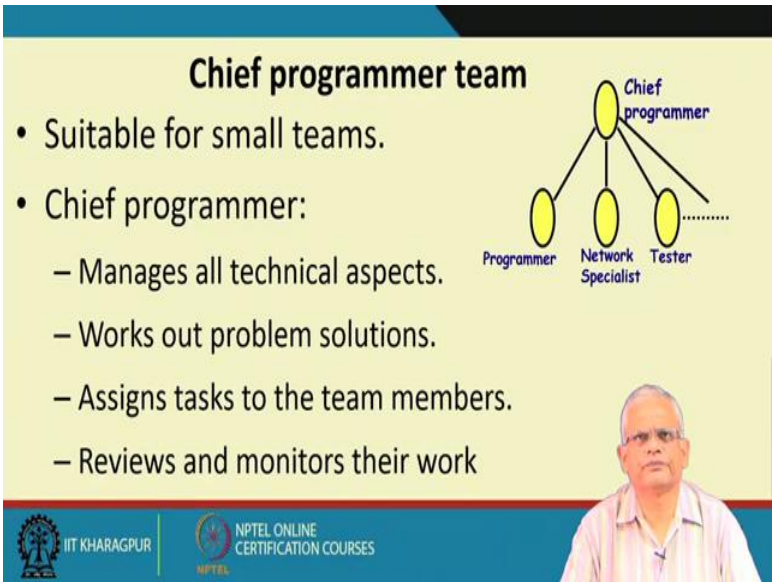
Welcome to this lecture. In the last lecture we were discussing about the organization structure and team Structure. In the team structure we were discussing that each team has some structure, the way the members do their job report to somebody there in the team and so on. We found that there are three major team organization; one is the democratic team suitable for small projects where there are about 7, 8 persons.

Rarely used for large projects, because the decisions there is collectively taken everybody consults everybody. It is suitable for research type of projects where there is lot of innovation, new ideas are required; but then it is inefficient in the sense that lot of time goes in discussions, meetings and so on.

We were looking at the chief programmer team. In the chief programmer team, these are also suitable for small projects where there is one person designated as the chief programmer. He is the expert for the project, he is in overall charge of the project, provides the high level solution, divides the project into small pieces of work, assigns to the team members, helps them the team members monitors and supervises them and then gets the work from them and integrates.

This is suitable for very simple programmers, it is a very efficient team structure for simple problems because hardly much time goes in discussions meetings and so on. Let us look at the chief programmer team first; we were already looked at the democratic team in the last lecture. We will start with the chief programmer team and then we will look at the hybrid organization.

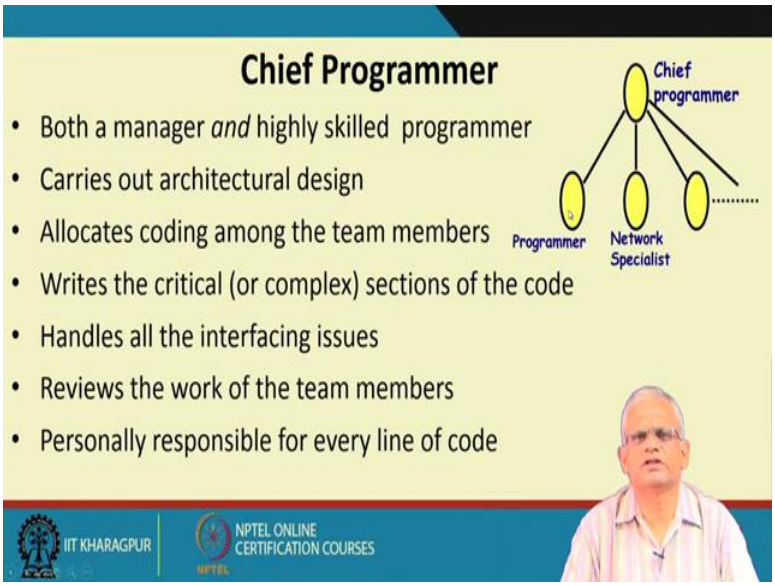
(Refer Slide Time: 02:44)



The chief programmer team as the name says we have one the programmers designated as the chief programmer. All other team members report to the chief programmer; the chief programmer is responsible for the project. He is the technical lead and also the managerial lead; all other team members report to him, he visualizes the solution, works out the design, explains the work to different team members. For example, the programmers he might explain them what coding exactly they would have to do, to the network specialist he may say what network solution is needed, to the testers he would say that how to test and so on, what aspects to test and so on.

So, here the entire project solution is with of chief programmer. The other team members have only partial views of the project, they do not fully understand what is going on; but whatever has explained the small tasks that has been explained to them they only know that and they do that. The chief programmer monitors the work, provides any help that may be needed further explanation and so on. And as the different team members complete their work, reviews the work and once these are done satisfactorily, integrates and slowly the full system develops.

(Refer Slide Time: 04:44)



Here the chief programmer is the main person responsible; he is both a manager and a technical expert, carries out the architectural design detailed design and then allocates the coding among the team members helps the team members and the critical parts of the code himself writes, handles all interfacing issues how to integrate the different parts, reviews the work of team members, anything deficient explains them and asks them to do those things and the chief programmer alone understands the complete project every line of code, every design everything is known to the chief programmer. All of the team members have only partial views, only small parts of the project they know.

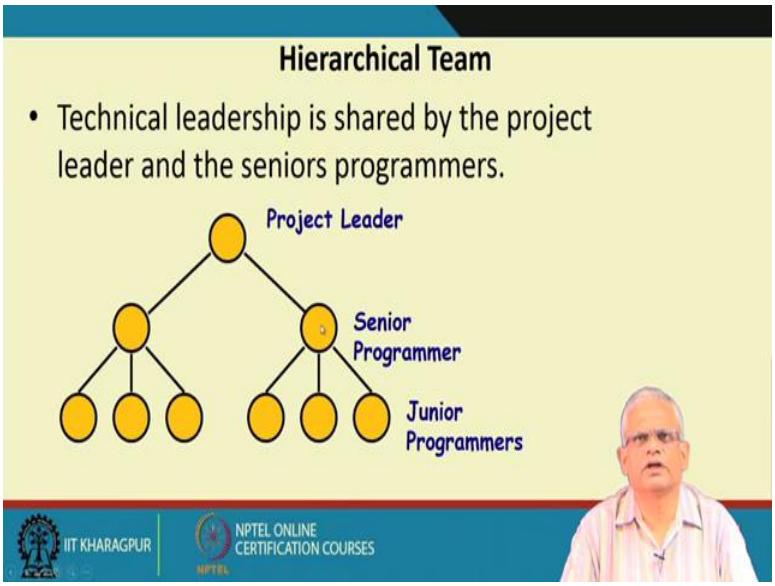
As you can see that one of the major problem here is that what if the chief programmer becomes indisposed. The project would stall, but then this is a very efficient way of doing simple projects, because hardly there is much time wasted in discussions and so on. The chief programmer does everything, effectively supervises gets all the work done on time, he knows how much time it takes because he is the expert there and therefore, this is the fastest way of doing the simple projects.

(Refer Slide Time: 06:31)

But what about large projects? Neither the chief programmer team nor the democratic team is suitable. We said that both of these work for small projects. The chief programmer cannot supervise too many other programmers; the democratic everybody discusses with everybody and therefore, as the size of the team increases, the number of communication path increases is N square.

And quickly it becomes very inefficient for even moderately large projects. And for this reason the hierarchical team structure is used for large projects. This is as the name says this is the hierarchical organization; there are junior programmers who report to senior programmer, the senior programmer report to a project leader, the project leader reports to a senior leader and so on.

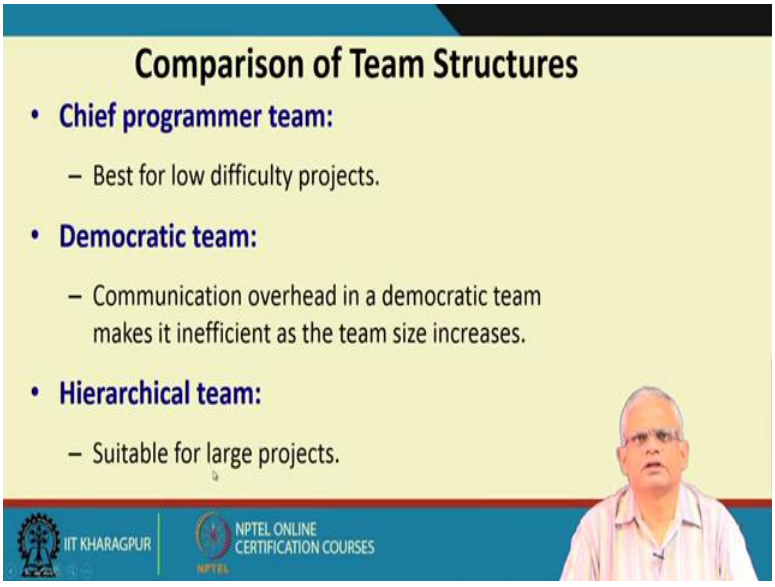
(Refer Slide Time: 07:44)



So, as you can see this is the tree structure where the junior programmers for a specific part; the project leader assigns work to the senior programmers. The different parts of the project are done by different senior project programmers, they have their own group of junior programmers and they talk to each other.

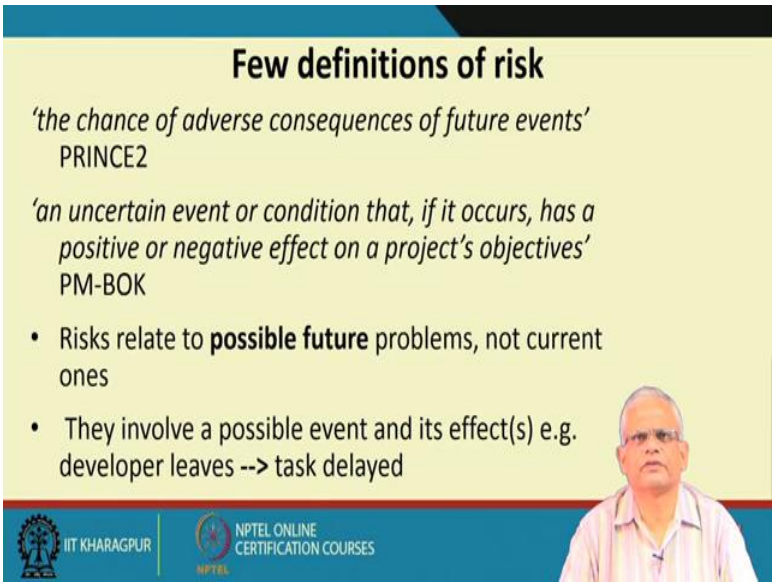
See here there is a democratic setup at the bottom level and a chief programmer set up at the top level. The technical leadership unlike the chief programmer here it is given by the project leader and the senior programmers.

(Refer Slide Time: 08:34)



The chief programmer is the best structure for simple low difficultly projects. The democratic team, its suitable for research oriented, innovative projects, but the communication overhead increases rapidly with team size; and therefore, democratic team is also used for small projects, the hierarchical team is used for large projects. We will now discuss about the Risk management issues; how does a project manager carries out risk management?

(Refer Slide Time: 09:19)



The first thing that to know here is, what is a risk? A project suffers from many types of risks; in the PRINCE 2 risk is defined as the chance of adverse consequences of future events as the project progresses many events occur and the risk is the adverse consequence of a future event.

For example, one event may be that a team member leaves that is a event and that has an adverse consequence; that who will do his work. Another future event maybe that the budget becomes very constraint, the customer faces financial difficulties and is not able to provide the funding and so on. Now, let us look at the PM-BOK definition Project Management Book of Knowledge; the definition there is an uncertain event or condition that, if it occurs, has a positive or negative effect on a project's objectives.

So, this is again similar definition about an event which may or may not occur, it is an uncertain event or condition; if it occurs has a positive or negative effect. So, there is a positive risk; the positive risk is that something good happens. For example, the

customer may say that you can take an additional 2 months because our actual product launch is delayed, for your part you may take another 2 months, so that is a positive risk.

The negative risk maybe that the customer says, that you need to complete it by one month early because our project launch has been prescheduled. But in all these definitions the risk relate to some events, that may occur in the future not the current one's. So, the risk is some event that occurs in the future and the problem that may arise because of that.

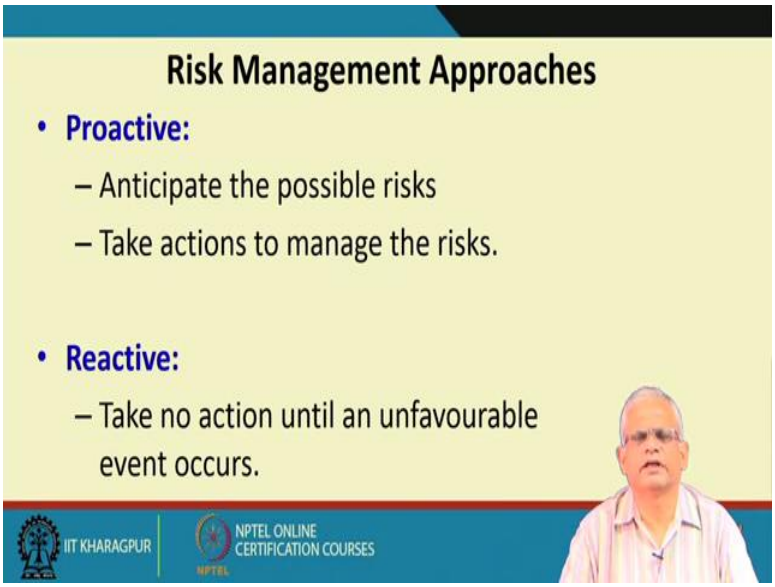
(Refer Slide Time: 12:32)

There are many many types of risks that a project may suffer; one possible categorization of the risks is the market risk. The market risk is the project complete successfully, but then the market conditions have changed and the product is not accepted in the market; may be the competitors have come up with the better product, maybe the technology has changed and it is not no more acceptable in the market, for the specific type of technology it was being developed it has further advancement and the product is not acceptable in the market. Those are all market risks examples of market risk.

Financial risk is risk of financial commitment by the customer not being honored; the customer gone bankrupt cannot pay for the project halfway through the project and so on. The technology risk is that the project assumes some technology and then the project proceeds based on that technology; but it may so happen that better technology gets developed and comes into use before the project completes. So, the technology

obsolescence is a example of a technology risk. And then there are project risks like project personal leave unable to find replacement for a specific role, the schedule gets delayed and so on.

(Refer Slide Time: 14:42)

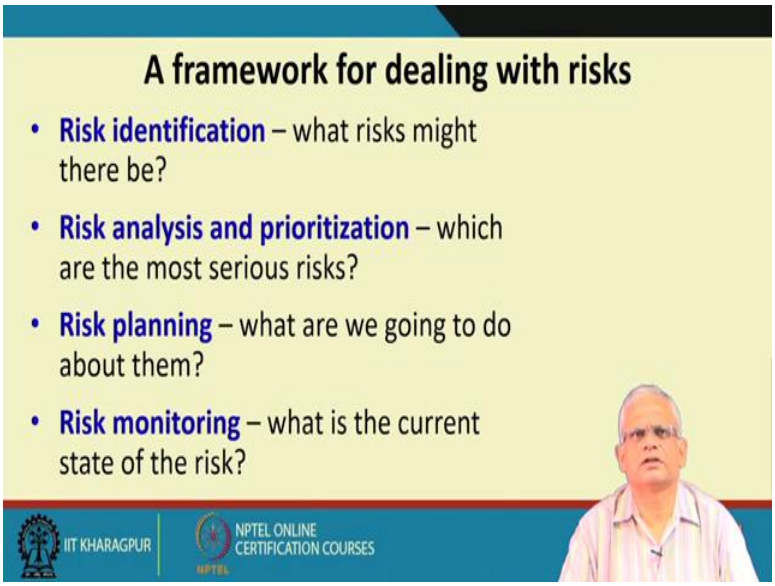


Now, let us look at the risk management approaches, there is hardly any project that does not have risk. Every project has several dozens of risks that it faces and it is a important job responsibility for the project manager to handle the risks.

And broadly two options are available to the project manager. One is the proactive approach, here the project manager anticipates sits down and thinks, what are the risks that may the project may be susceptible to, anticipate all possible risks and also plan what actions to take. If the risk actually becomes true and then as the risk becomes true, take the planned actions and manage the risks.

This is the one which is very popular among the managers the proactive approach; but then there is a reactive approach, where the project manager cannot even anticipate the risks and therefore, takes no action until the event occurs. Typically the project managers take the reactive approach that as the risk occurs because these risks are not predictable they cannot be foreseen by the project manager take no action until the unfavorable event actually occurs.

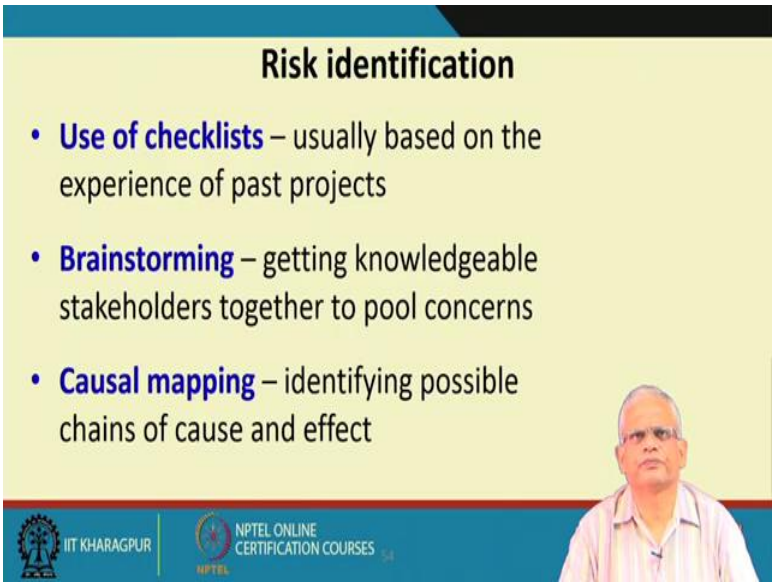
(Refer Slide Time: 16:35)



But in the proactive approach we said that in the reactive approach just handles the risk as they occur; but in the pro active approach how does the project manager go about dealing with risks? The first thing of course, is to anticipate the risks. Risk identification that is what risks can be there in for this project, notes down all the risks anticipates, finds out what technical risks can be there, what project risks like people leaving the project what will happen, schedule delayed what actions to take and so on.

So, the first step in dealing with a risk is to identify all possible risk at the start of the project. And once the risks have been identified need to analyze the risks, what will be the impact of the risk, which is a serious risk, need to prioritize all the risks; that which risks are the most damaging. And then the risk planning; what to do, when the risk actually becomes true and then as the project proceeds, risk monitoring is the risk probability becoming higher and so on, what is the current state of the risk.

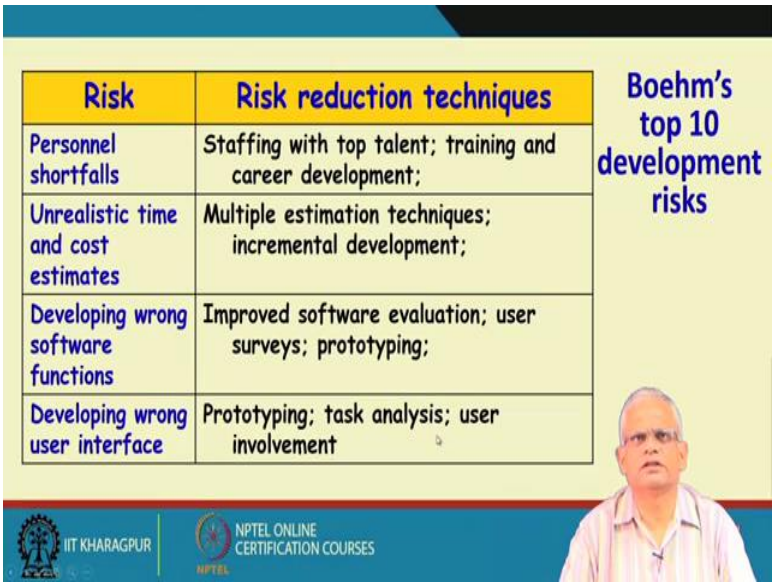
(Refer Slide Time: 18:27)



The first thing in this framework is Risk identification. How does the project manager identify the risk because these are future events and these are uncertain events; how does the project manager anticipate the risks? One is use checklist. Find out the common risks common type of risks that have occurred across different projects that he has worked or the projects that were running in the organization. And finds out what were the types of risks that the other projects faced and then checks out whether such risks are likely to be relevant for the project at hand.

The second way the risk can be identified is by brainstorming; get the team members, other stakeholders and then get their concerns and those at the risks. And the third type of risk identification is the causal mapping here identify the possible events that may occur and the affect that will of these events.

(Refer Slide Time: 20:07)



It is important for the project manager to know the different types of risks that fall on various projects and what are the possible risks reduction techniques that are suitable for these risks. Boehm has identified 10 most frequent risks that development project suffer; and therefore, this is quite in instructive for every project manager to identify what are the most frequent risks that the other projects have suffered and what are the best ways available to handle those risks.

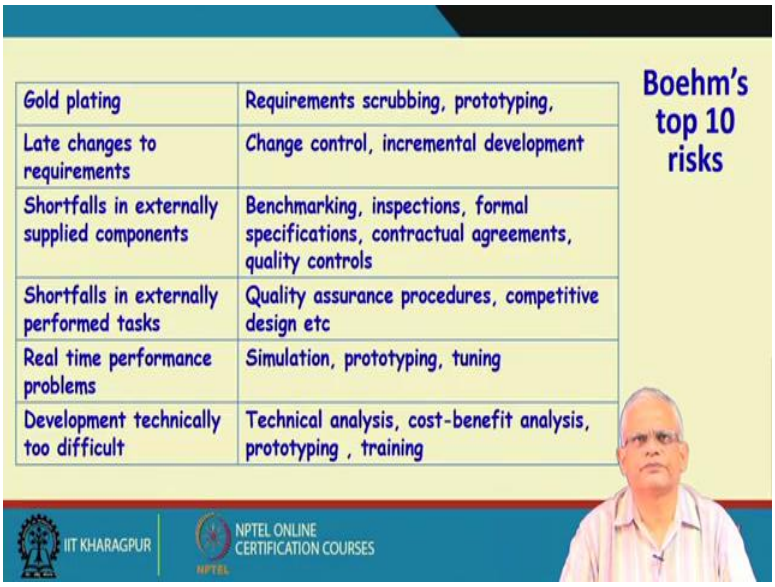
The first risk is of course, relates to the manpower; personal short fall, lack of competent manpower in the project. We have the project personal, but they are not able to perform, the project is getting delayed this is a major risk here. And Boehm suggest the corresponding reduction technique says that staff to start with the project get the top talent and also in the areas of the project provide training and career development.

Unrealistic time and cost estimates; the project manager could not really estimate the time and cost correctly and quoted a wrong figure to the customer. And now the customer says that, since you said that you will do within this budget and within this time, why you are not able to do it. To get more realistic, accurate estimate, the project manager needs to use multiple estimation techniques. And also one way to handle it by using incremental development because, making a long term estimate is very difficult it is error prone; but a short term estimate that is for one feature at a time is a much accurate way of estimating and that those two are the risk reduction techniques.

Developing the wrong software functions; the customer wanted something and finally, the development team develops some other functions, which the customer is not happy says that, we did not want it this way and that happens very frequently. And the risk reduction technique for this is improved software evaluation; user surveys trying to find out what exactly the customer needs; prototyping and getting it evaluated by the customer these are the risk reduction techniques.

Developing the wrong user interface, this is also a risk similar to the functionality; here the interface the customer may not like; here again prototyping getting it evaluated by the customer; task analysis do a proper task analysis where identify for performing a specific functionality or task, what are the input and output that are required; user involvement in the project these are the risk reduction techniques.

(Refer Slide Time: 24:17)



Gold plating; in gold plating the team members they think that some features may be appreciated by the customer, even though those are not really documented in the requirements. They start developing something thinking that, this work will be highly appreciated it will be very useful; but then it may so happened that the customer does not need them actually this is just a waste of time, it just delays the project, increases the cost. The risk reduction technique is requirement scrubbing, that is dropping any requirements that have been put as a later thought and are not actually required;

prototyping having the user evaluate a prototype version and say that whether it is required or not.

Late changes to requirements that is the customer may suggest some changes later; here change control if the change actually has to happen then we have to somehow have a procedure to effectively handle the change and that is called as a change control. Incremental development, so that the customer. Incrementally evaluates and the changes are not too much at the end, because the changes are troublesome; if they are to be done at the end once everything completes.

Shortfall in externally supplied components the projects do give out, contracts to other developers to other organizations and they may turn in inferior work products; here the risk handling is through benchmarking, inspection, formal specification, contractual agreement and quality control.

Short fall in externally performed tasks here it may not be a component, but a task which is performed externally; maybe let us say coding or may be testing and so on. Here the risk reduction technique is quality assurance procedures, competitive design, etcetera. Performance problems here the risk handling is through simulation, prototyping and tuning.

The development becomes technically too difficult; the risk reduction techniques are through technical analysis, cost benefit analysis, prototyping, showing it to the customer, evaluating by the team members, the prototype helps in developing the actual solution and then of course, training.

The project managers need to be aware of various important risks that the past projects are suffered and what were the risk reduction techniques. And these are the 10 most frequent risks according to Boehm and the corresponding risk reduction techniques he has identified; every project manager needs to know this. We are almost at the end of the time for the current lecture, we will stop here and continue in the next lecture.

Thank you.