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

Lecture – 41
Project Monitoring and Control (Contd.)

Good morning to all of you. Last class we have discussed about Project Review Technique. Today, we will discuss about Source Code Review.

(Refer Slide Time: 00:21)

We will see two types of code review that is code walkthrough and code inspection. Also, we will discuss something about a relatively new technique called as cleanroom technique.

(Refer Slide Time: 00:36)

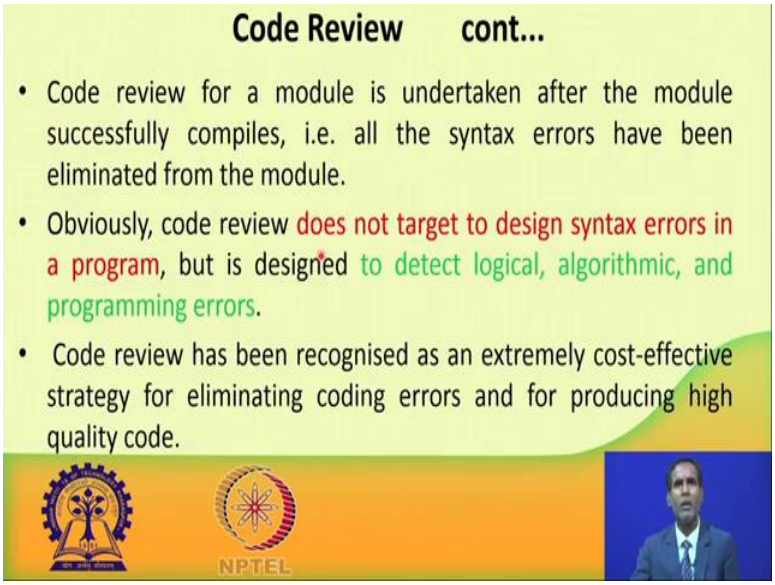
So, the yesterday we have discussed that all the interim and final work products they are usually the candidates for a review, and particularly the following work products they are suitable candidates for review. Those are like requirements specification documents or SRS document, user interface specification and design documents. Architec different design documents are such as architectural design, high-level design, detailed design documents, also source codes obvious, this can be also reviewed.

Then various test plans on the test design test cases, they can also be reviewed. Also, the various project management plans, the configuration management plan etcetera, they can also be reviewed. So, right now we will discuss about how the source codes they can be reviewed.

(Refer Slide Time: 01:25)

We have also seen that yesterday review is a very effective technique to remove the detail defeat detects from the source code. And in fact, a review has been acknowledged to be more cost effective in removing the defects as compared to testing. Over the years various review techniques have become extremely popular and they have been generalized to be used with other work products also.

(Refer Slide Time: 01:52)

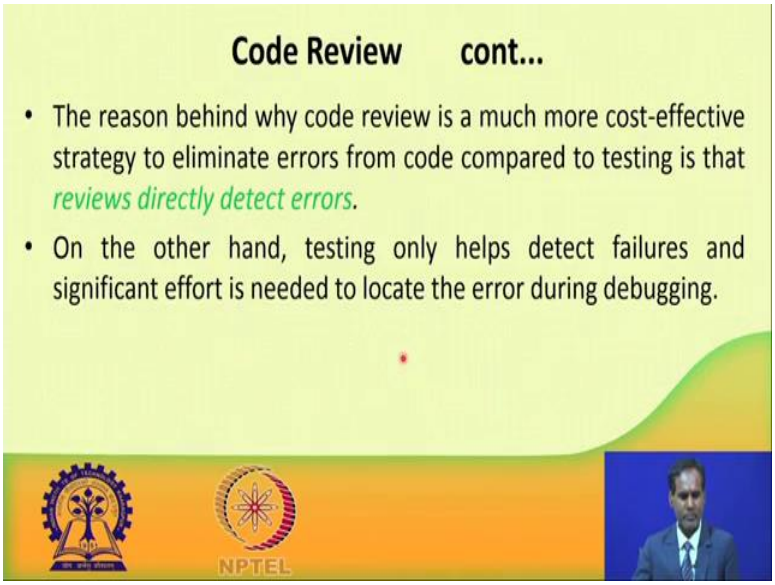


So, now let us see when the code review should start. The code review for a particular module it is undertaken after the module is successfully compiled that is all the syntax

errors that have been alimented eliminated from the module. Obviously, you see that code review does not target to design the syntax errors. Please remember, code review it does not target to design the syntax errors in a program. So, it does not target to syntax or design syntax errors in a program. It is designed or it is targeted to detect logical errors, algorithmic errors and programming errors. That should be kept in mind.

The code review has been recognized as an extremely cost effective strategy for eliminating the coding errors and for producing high quality code. So, here it will be a cost effective, much cost effective than this a what testing techniques while detect eliminating the or and detecting the errors.

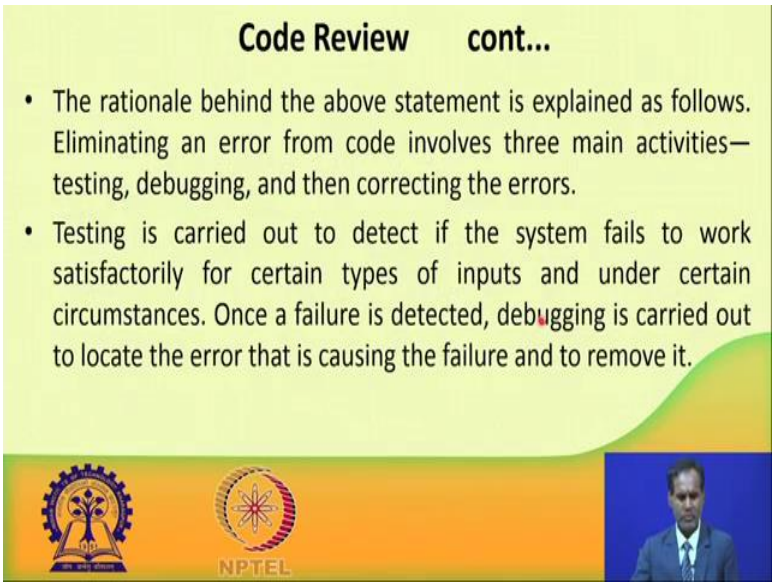
(Refer Slide Time: 03:00)



Now, let us see what is the reason why code review is so much cost effective than the testing technique. The reason behind a why code review is a much more cost effective strategy is to eliminate the errors from code compared to testing is that review directly detect the errors. Let us see what is happening in testing, why in a review it is so much cost effective, because review is that directly detect errors while testing they only help detect the failures and the significant errors.

By using the testing technique you can only detect the failures and the significant, and you can only detect the failures. But in case of this review you can directly what detect the errors. So, in and here in case of testing you have to spend a lot of effort to locate the errors during debugging.

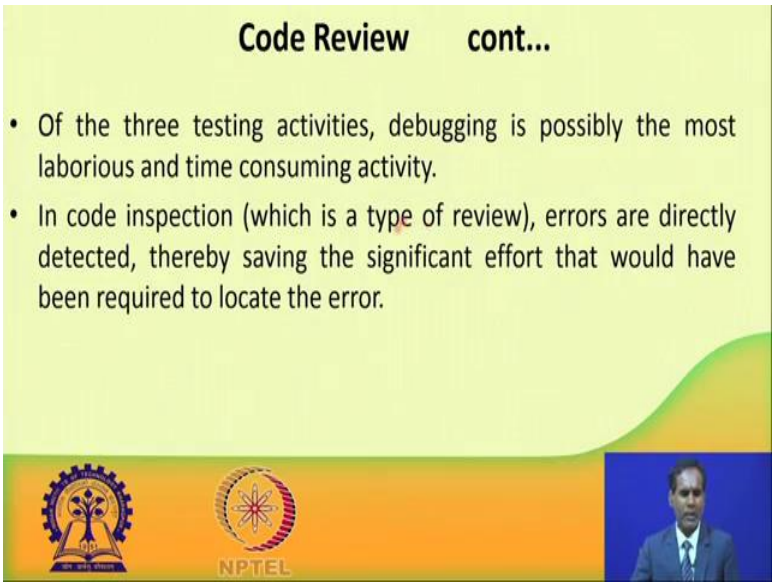
(Refer Slide Time: 03:55)



The rationale behind the above statement is explained as follows. Why it is cost effective? What is the rationale behind this? You see the eliminating an error from code it involves 3 main activities, one testing, then debugging and then correcting the errors. So, a 3 activities are there where in case of eliminating errors from the code.

Testing is carried out to detect if the system fails to work. So, normally testing is carried out why? It is carried out to detect if the system fails to works for certain types of inputs and under certain types of circumstances, and when a failure is detected once a failure is detected, then debugging is carried out to locate the error that is causing the failure and then you can remove it. But in a what a code review you can directly detect the error, so that is why this is much more cost effective than the testing process.

(Refer Slide Time: 04:57)



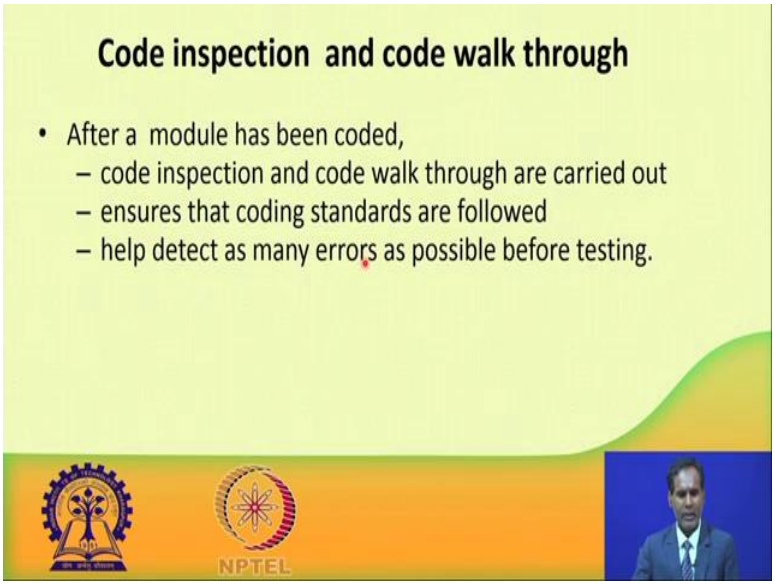
So, of this as I have already told you there are 3 activities are called what carried out to due while you are eliminating the errors that is testing debugging and correcting the errors. So, out of these 3 testing activities, debugging is the most lubricous and time consuming activity.

So, we will see that in code inspection which is a particular type of review, we will see that there are two types of review, one is code walkthrough, another is code inspection. In code inspection what is happening the errors are directly detected. So, here in what testing what it is only help you in detect the errors which is again a time consuming and lubricous activity, but in code inspection the errors are directly detected thus saving the significant effort that is required to locate the error. So, that is why a review is much more cost effective than the testing technique.

(Refer Slide Time: 05:55)

Normally, there will be two types of code review techniques, one code walkthrough and the other is code inspection.

(Refer Slide Time: 06:05)

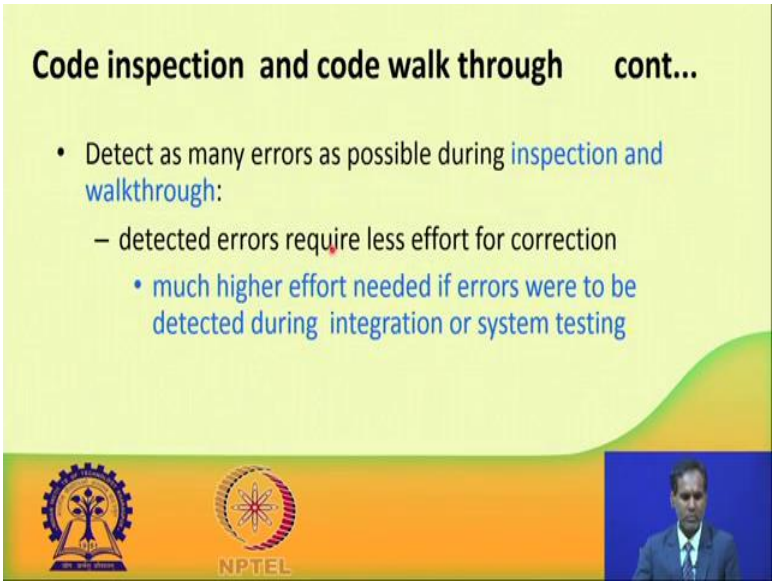


Let us see about this things. So, as I have already told you that when this what code inspection or code walkthrough can start after it is what compiled, after a module has been coded, then code inspection and code walkthrough can be carried out.

So, code inspection and code walkthrough they ensure that the proper coding standards that have been followed like the standard errors you have used and what proper variables

etcetera that you have used it ensures the various coding standards that the programmer has used in the software that he is developing, it. They also help to detect as many errors as possible before testing. So, before this system testing or integration testing the this code review techniques that is code inspection and code walkthrough they help detect as many errors as possible before the system testing and the integration testing starts.

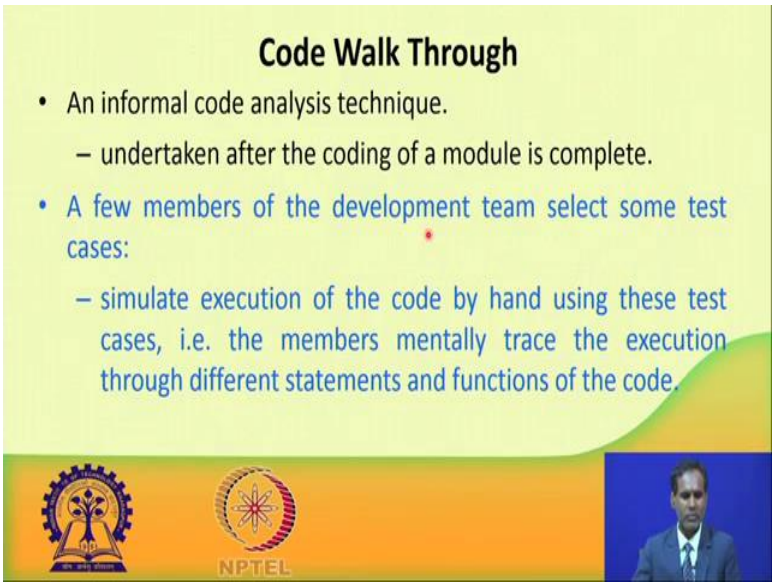
(Refer Slide Time: 07:06)



So, this techniques they detect as many errors as possible during a inspection and code require, ok, the code inspection and code review, they detect as many errors as possible during this process. They also, and these detected errors require less effort for correction. So, after this detection we must correct these errors. So, the errors detected during inspection and walkthrough they require less effort for correction, otherwise it would require much higher effort if these errors was would be detected during the integration and system testing.

So, we will summarize that the errors detected during a code inspection and walkthrough, they require less effort for correction, in comparison to the errors if they could have been detected during what integration or the system testing level.

(Refer Slide Time: 07:58)

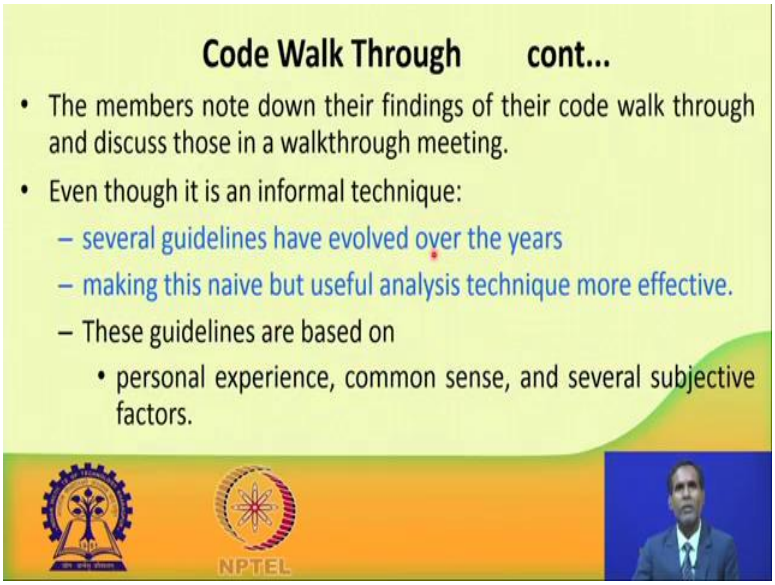


We will say first the code walkthrough. Here, this is basically an informal code analysis technique. So, code walkthrough is considered as an informal code analysis technique. It is not a formal code analysis technique, it is a informal code analysis technique which is undertaken after the coding of a module is complete. So, after one module or one unit its coding is over, then you can go for this code walkthrough which is an informal code analysis technique. Now, let us see how this code walkthrough process system works.

So, a few members of the development team select some test cases. So, a small what committee maybe formed and a few members of the development team and they select some test cases, and those test cases then they simulate execution of the code by using those test cases. So, they first select some test cases, then simulate the execution of the code. How? Not by computer, by hand, by mentally, by hand using these test cases that is these members, this development team members they mentally trace the execution through or different statements and a functions of the code. So, this is how this code walkthrough works.

So, that is why its name is code walkthrough. You are, just as if you are working through the a code you are executing the code by hand or you are mentally testing the execution that is why this is known as a code walkthrough.

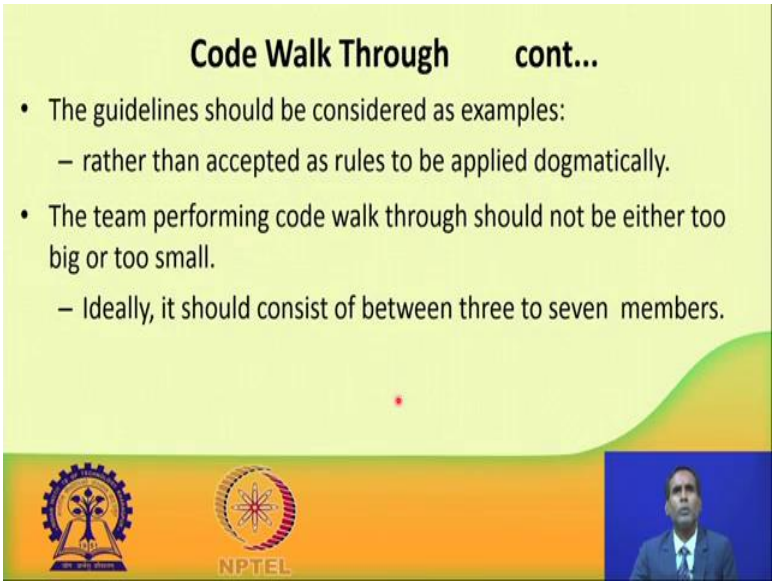
(Refer Slide Time: 09:27)



The members the while they are executing the code by hand or by mentally, they note down their findings of their code walkthrough and discuss those in a walkthrough meeting. So, in that discussion they will say what kind of what defects that they have found or what could be the cause. So, those details they discuss in the walkthrough meeting.

So, even though this code walkthrough is a in is an informal technique, but several guidelines have already evolved over the previous years, so which makes a or which make this naive approach, which make this naive, but this is a very useful analysis technique and this is a more effective technique. So, these guidelines those have been evolved over the years, they are based on what? They are based on the personal experience of the programmers, they are common sense and several subjective factors.

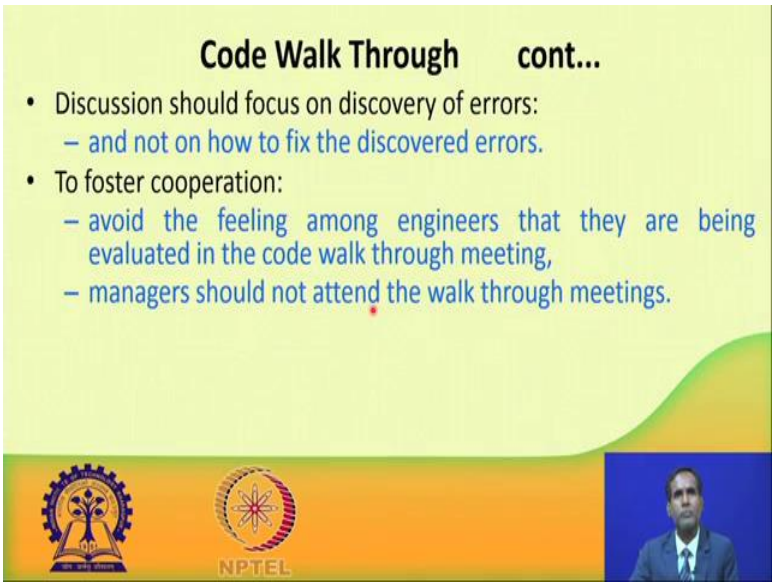
(Refer Slide Time: 10:24)



The guidelines should be considered as example please say. So, as I have already told you during a or in these guidelines have evolved over the years, these guidelines are not actually very a hard and fast that you have must have to follow. So, rather these guidelines should be considered as examples, rather than accepted as rules to be applied dogmatically. These are not that hard and fast you have to follow exactly, they can be considered as only examples.

The team performing the code walkthrough should not be either too big or too small. So, let us say a how many what members should be there in their code walkthrough committee. It should not be a either a too big or too large. Ideally, it should consist of between 3 to 7, 7 members.

(Refer Slide Time: 11:13)



Now, let us see in this code walkthrough meeting a what should be focused, the member should focus on what. Discussion in the code walkthrough meeting should be focus on only the discovery of errors, they should not discuss how to fix the discussion discovered errors.

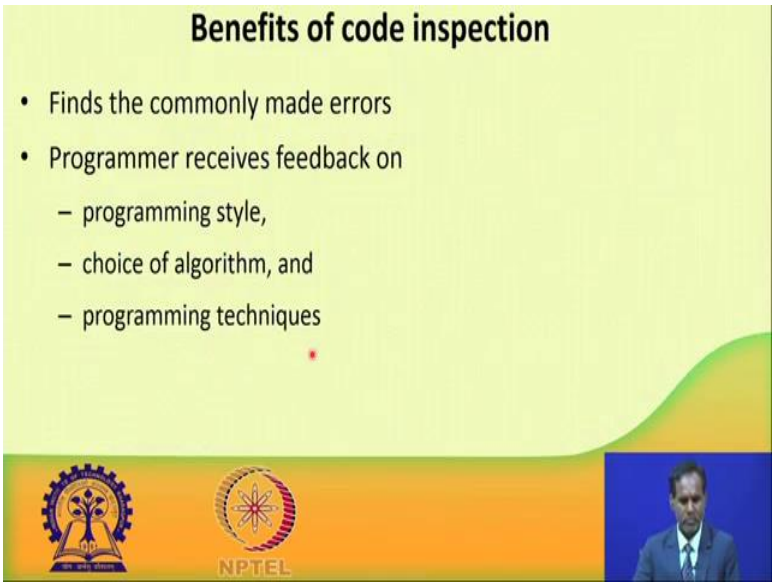
The objective is, the focus is only on discovering the errors, but not on how to fix the discovered errors. So, to faster cooperation and the members should avoid on filling that they are being avoid, they are being evaluated in the code walkthrough through meeting. You know they should that fear not be there that somebody is evaluating them in the meeting, so that is why managers should not attend the walkthrough meetings, only the developers or the programmers, their peers they may attend the walkthrough meeting. So, that this fear should not be there, that they are being evaluated in the walkthrough meeting, that is why managers or draft able managers they should not attend the walkthrough meetings.

(Refer Slide Time: 12:14)

Now, let us see, we have seen about code walkthrough what is code inspection. In contrast to code walkthrough on code inspection aims mainly at the discovery of the commonly made errors that usually creep into code due to programmer mistakes and over sites. I am repeating again.

So, in code walkthrough what is happening? That programmers they execute the what test cases by hand or by mentally trace the execution of the code but here in code inspection the objective is discovering ops of commonly made errors, because the human the developers are human beings and they may make some common mistakes.

(Refer Slide Time: 14:23)



So, the objective the aim of code inspection is to discover those commonly made errors that usually creep into the code. Why? Due to the programmer's mistakes and their oversights. This code walkthrough also it aims at checking adherence to coding standards for the proper coding standards like a proper header files global variables etcetera you have used properly or not. So, it also what checks that.

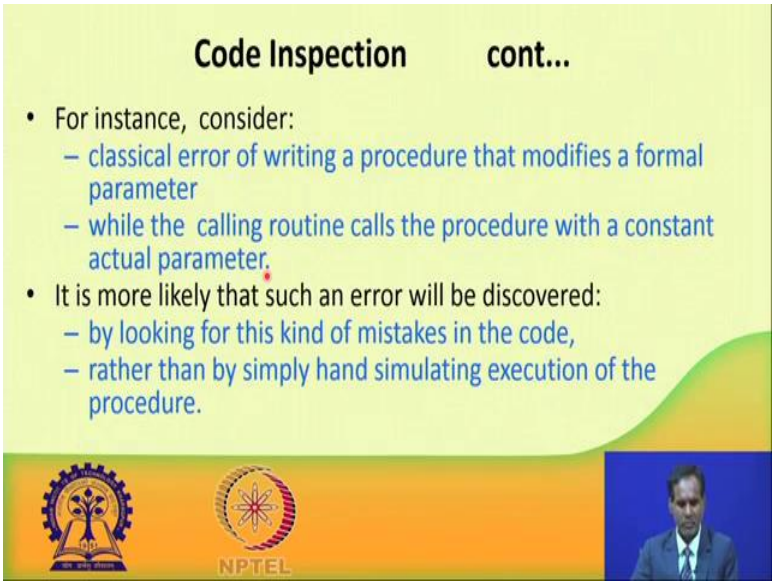
So, another objective of code war, another objective of code inspection is that to check adherence to the coding standards. So, now, let us see what is for performed during code inspection. During code inspection the code is examined for what? For the presence of certain really certain common kinds of errors.

So, in code inspection, the code is examined, it is checked for the presence of certain common kinds of errors that the programmers might commit the mistakes due, they might do, they might commint those errors due to some oversight issue. So, but in case of code walkthrough what is happening? You are the programmer or the team members they perform some handy hand simulation of the code execution, that is done in code walkthrough. But in code inspection, these code is examined for the presence of some commonly made errors.

What are the benefits of code inspection? I have already told you in code inspection the objective is to find the commonly made errors, so several commonly made errors which are committed by the programmers they are detected here. The programmer also

receives; so, let us see what are the side benefits, the side benefits of code inspection is that the programmer receives a feedback on his programming style, the choice of algorithm or the algorithm you have used and the pre the programming techniques he has used, you can get a feedback on these things, ok. So, these are the side benefits of code inspection.

(Refer Slide Time: 15:06)



And now let us take a simple example of code inspection. So, for instance let us consider the classical error of writing a procedure that modifies a formal parameter, you have written a procedure and that is modifying a formal parameter, when while the calling routine calls the procedure with a constant actual parameters. So, there is a you are calling a routine and the calling routine calls a procedure with a constant actual parameter and you are trying to modify a formal parameter.

This is obviously an error, and this error can be detected by this code inspection. So, it is more likely that such an error will be discovered by looking for this kind of mistakes in the code, you while you will inspect the code probably it will come to the eye of the what developer or so.

So, by looking for this kind of mistakes in the code, this types of errors can be discovered, rather than by simply hand simulating execution of the code. So, for this kind of errors code inspection is the better one, rather than the code walkthrough where you are simply hand simulating the execution of the procedure.

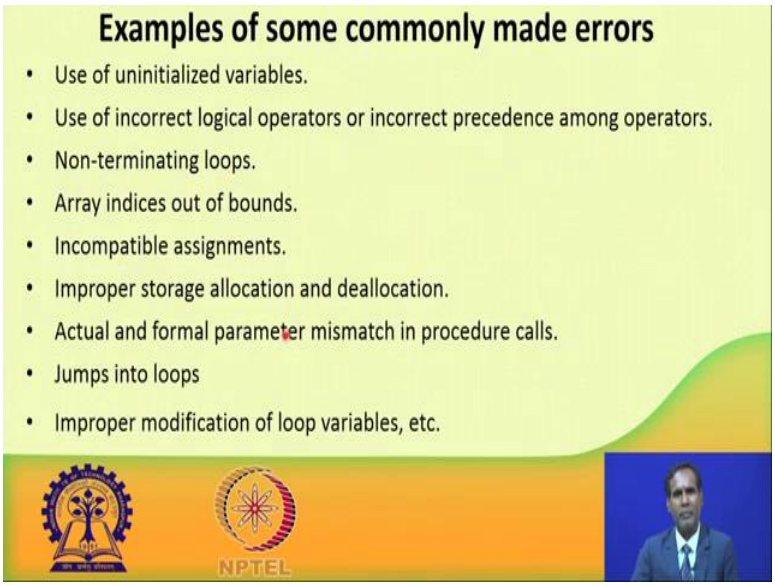
(Refer Slide Time: 16:21)

So, a good software development companies, they collect statistics of errors committed by their engineers, ok. From their previous projects what kind of errors the engineers have committed, in similar kinds of projects, they collect all those information, they collect all those statistics and record in a file. So, the good software development companies, they collect the statistics of errors which are committed by their engineers and then they identify the types of errors most frequently committed.

So, out of all those errors those are statistically recorded then they identify which kind of errors the programmers, they are most frequently they are committing. Then, what they do? They prepare a list, they prepare a list of a common errors made by the programmers and this list of common errors can be used during the code inspection. So, they can prepare just a checklist and this checklist can be prepared from the previous experience of the programmers.

so then this checklist can be given and to the what programmers while doing code in inspection for the other types of projects, then this list checklist of the common errors it can be a used during code inspection for searching for possible types of errors. So, now, let us see, let us look out about some commonly made errors which are there or which the a programmers might commit during the development of a software.

(Refer Slide Time: 17:56)



So, these are some of the examples of some commonly made errors. Like use of un initialized variable. The programmer has used a variable, but he has forgotten to initialize it. So, that can be searched during code inspection. Similarly, use of incorrect logical operators or incorrect president precedence among operators, the programmer due to what oversight he has used some incorrect logical operators or incorrect precedence among the operators he has used that also can be detected during code inspection.

And non-terminating loop here that, the programmer suppose has written a loop like a while do, while or for, but that is not terminating he has forgotten to give a what put the termination, the condition for termination. So, then the loop will be a what leading to what infinite or the for loop will be will be leading to for infinite loop. So, those kind of non-terminating loops. These errors also can be detected by your code inspection.

Similarly, array indices out of bounds. You have written an array maybe a 100 element array, but you are searching an array or an array which position is more than 100. So, array indices which are out of bounds these errors also can be detected. Then incompatible assignments, you know that in assignment step statements left hand side of the left hand side must correspond to the right hand side. If the left hand side is what integer and right hand side is a floating point variable then this becomes incompatible assignments. So, those type types of errors also can be detected by code inspection.

Similarly, improper storage allocation and de-allocation. You know that memory allocation and de-allocation like malloc and dealloc etcetera, so if you have improperly done this storage allocation and de-allocation or some mistakes, you have put some errors by mistake have put that also can be detected during code inspection.

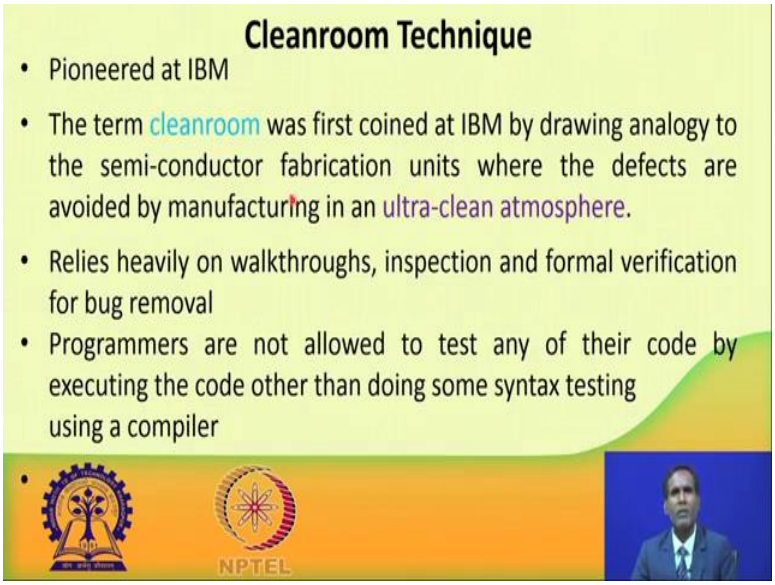
Similarly, actual and formal parameter mismatch in the procedural calls. While you are writing a procedure call like a some the number of factual parameters should match with the number of formal parameters. The types of the form actual parameter should match with the type of the formal parameters.

So, if there is any mismatch among them then also that means, any mismatch between the a number of actual parameters to the formal parameter, number of formal parameters or the types of actual parameters with the types of formal parameters. So, then these kinds of mismatch also can be easily find out, can easily be found out during this code inspection and similarly jumps into loops. You have use some go to statements or so, and those are eventually leading to the loops. So, jumps into loops which is not desirable. So, this is also a commonly what made data error, what, one of the commonly made errors by the programmers. You can use these code inspection techniques to detect those things.

Then, improper modification of loop variables you have used a loop variable inside your loop and you should not actually a modify, so you are, if you are improperly modifying a loop variable. Then, what will happen? This will be an error and this error also can be detected by using code inspection. So, these are some of the commonly made errors that a programmers might do due to some common mistakes or due to their oversight. So, these types of errors can easily be detected using code inspection.

So, in the for these types of things it is better to go for code inspection rather than going for this code walkthrough. Similar examples, so you can as a programmer you can also experience, you might have experience. So, all such commonly types of errors can be detected by using code inspection techniques.

(Refer Slide Time: 21:48)



We will take up a little bit of a new technique called as cleanroom Technique. So, this cleanroom technique is basically pioneered at IBM and let us see why they have use this term cleanroom. How this was coined? The term cleanroom was first coined at IBM, ok. This term first coined at IBM. How? They have drawn some analogy with what they have drawn, they have drawn an analogy to the semi-conductor fabrication units.

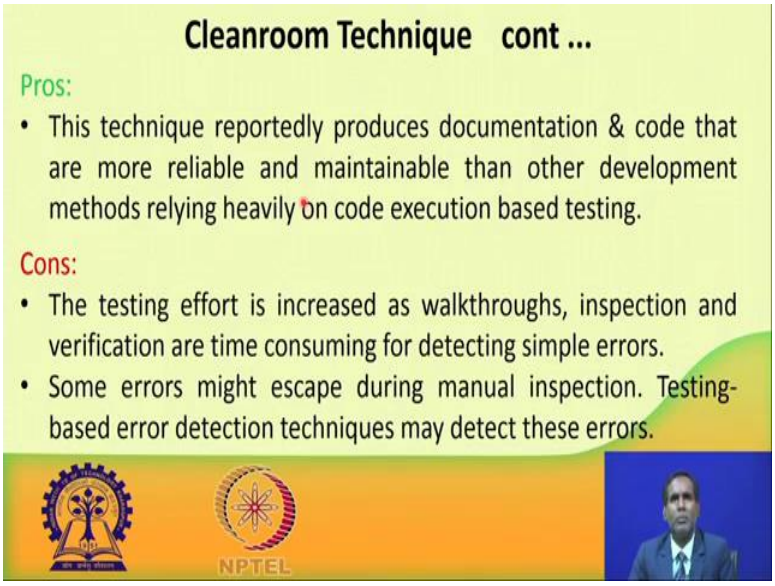
So, in the IBM they were trying to establish an analogy to the semiconductor fabrication units, where the defects are avoided. How? By manufacturing in an ultra-clean atmosphere. They are trying to not allow the defects. They are trying to avoid, they are trying to what avoid the defects. How? They are thinking that if the what semi-conductor fabrication it can be manufactured in an ultra-clean atmosphere then probably the defects will not arise at all. The defects can be avoided from the beginning. So, that is why this name clean they came, and they have seen from this and they have an, they have made an analogy with the same with the software development and then that is why they have given the term cleanroom testing or this cleanroom technique.

So, this technique cleanroom till, this technique that the cleanroom technique it relies heavily on these various review techniques we have seen, ok. So, this cleanroom technique relies heavily on code walkthroughs, code inspection and formal verification for bug emoval. So, here the programmers are not allowed to test any of their code by executing the code, ok. The programmers are just name suggest they will try to avoid the

defects. So, here programmers will not be allowed to test any of their code by executing the code. They have to use what walkthroughs, inspections or formal verification for bug removal for what detecting the bugs and removing the bugs. They cannot use. What? They are not allowed to test any of their code by what executing the code.

They may do only some exceptions like the accepting doing some syntax testing testing they may due to using a compiler, but normally they are not allowed to what test their code by a code execution. They can only perform walkthrough, code walkthrough inspection and formal verification for bug identification and bug removal. This is what is the objective of cleanroom testing or cleanroom technique.

(Refer Slide Time: 24:35)



Now, let us quickly see what is the advantage of cleanroom technique. This cleanroom technique reportedly produces documentation and code that are more reliable and a maintainable. So, this technique produces reportedly produces various documents and code. Those are more reliable, also they are maintainable than other development methods which are reliable relying heavily on code execution based testing.

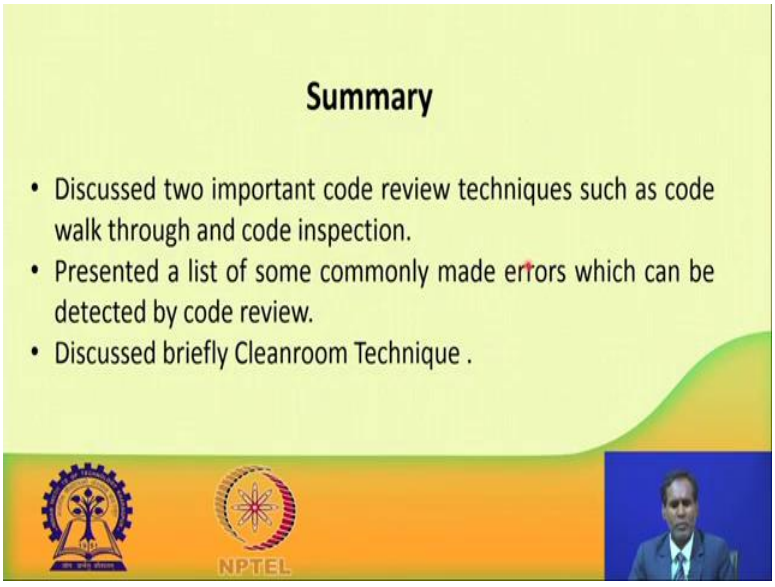
So, the technique as it produces a various documents and code, and these are more reliable and maintainable in comparison to other methods which are based on execution based testing, but the disadvantage is that the testing effort is increased as walkthroughs, inspection and verifications are time consuming for detecting this simple errors.

You know that code walkthrough; so, how many inputs you will take to for hand simulation? You it will take a much long amount of time. You have to inspect the code line by line. Again, it is a time consuming process. Also, formal verification, that is also another time consuming process. So, that is why the testing effort is increased to a large extent as all these techniques, walkthroughs, inspections and verification are very much time consuming for detecting this simple errors and the cleanroom technique is based on only these what code review un-formal execution techniques.

Another problem is that some errors might test still escape doing manual inspection. You see all these walkthrough or inspection etcetera they are manual techniques. So, some errors might escape from the eyes of the team members or the programmers during manual inspection, because after all a we are a manual human beings we may commit mistakes. So, some of the errors might escape during manual in inspection. So, those errors, those escaped errors can be maybe detected using the error base detection techniques.

So, these errors which can be expect what escaped during this manual inspection, they can be detected easily, these escaped errors maybe detected easily if you will use some testing based error detection techniques. So, that is another disadvantage of cleanroom testing.

(Refer Slide Time: 26:58)



So, today we have discussed two important source code review techniques. Those are code walkthrough and code inspection. We have also presented a list of some commonly made errors which can be detected by what a code review, particularly by a code inspection. We have also discussed briefly about cleanroom technique which is which heavily relies on code structured walkthrough and or code walkthrough, code inspection and formal verification technique.

(Refer Slide Time: 27:32)

We have taken the materials from these books, ok We have taken the materials from these books.

Thank you very much.