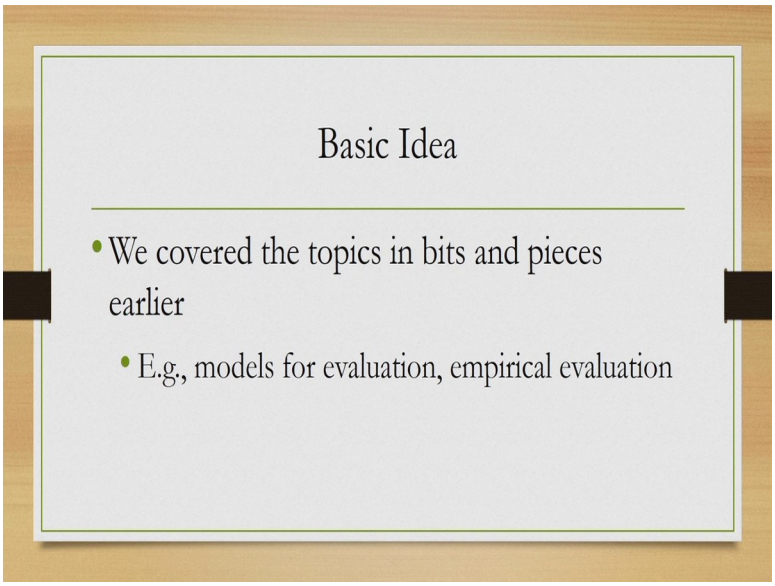
User-Centric Computing for Human-Computer Interaction
Prof. Samit Bhattacharya
Department of Computer Science and Engineering
Indian Institute of Technology, Guwahati

Lecture - 29
Introduction to user-centric design evaluation and expert evaluation technique

Hello and welcome to lecture number 29 in the course user-centric computing for human-computer interaction. So, in the previous 28 lectures, we have learned about many concepts related to this course. So, if we quickly have a recollection of what you have learned. So, we started with a discussion on overall interactive system design process what are the stages involved, then we discussed about the importance of computation in the design of interactive systems and we have discussed one computational framework.

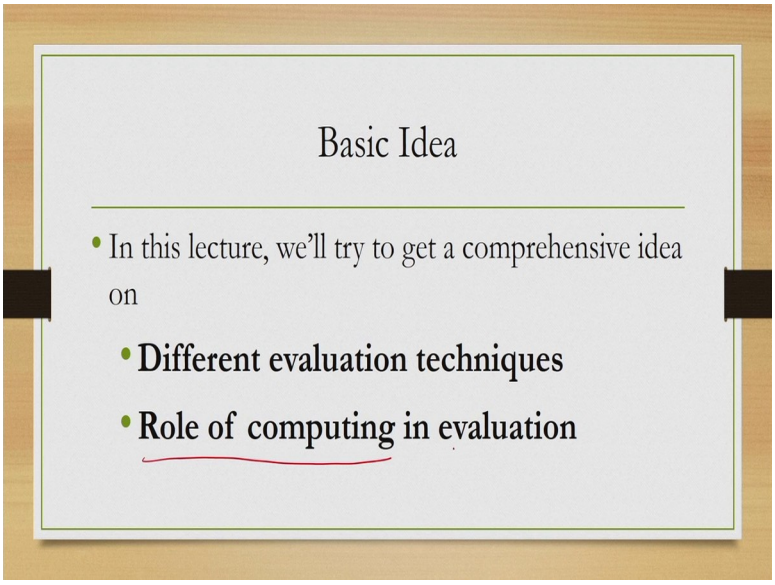
In the subsequent lectures, we discussed about user models particularly computational user models which is a very important component of the framework and then we have discussed in details about the empirical research method. So, one more important concept remains to be discussed which we are going to discuss starting from today that is how to evaluate interactive systems.

(Refer Slide Time: 01:49)



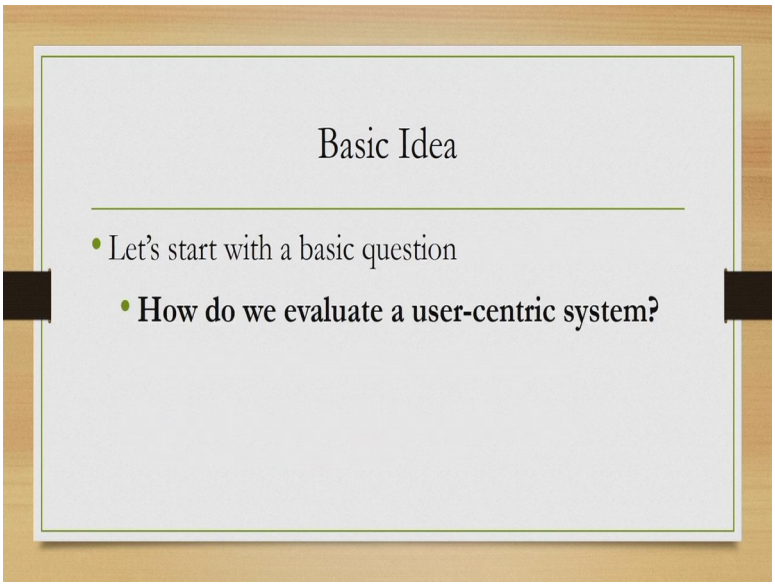
So, when we talk of evaluation; if you recollect the previous lectures, we talked of this concept in bits and pieces earlier as well when we talked of the use of models for evaluation or in the context of empirical studies.

(Refer Slide Time: 02:11)



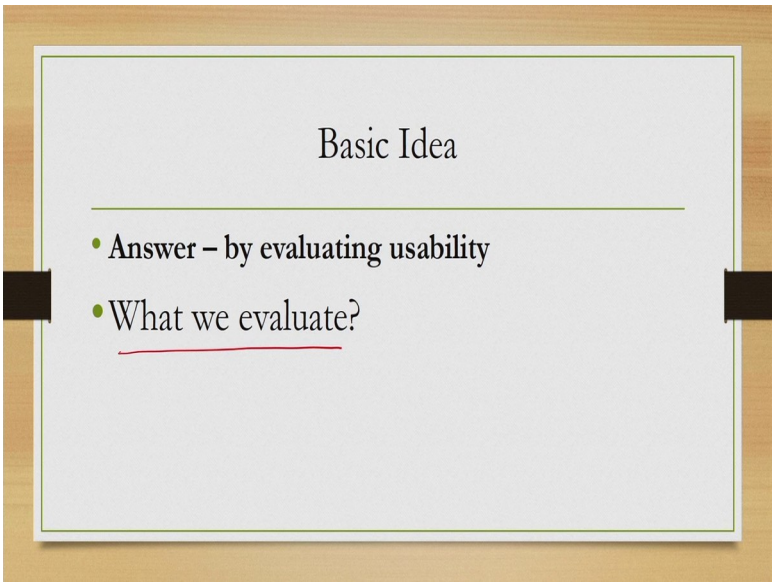
Now, in this lecture we will try to go a little deeper. We will try to have a comprehensive understanding of the concept of evaluation in the context of user centric design and we will try to also discuss what role computing plays in evaluation of user centric systems.

(Refer Slide Time: 02:37)



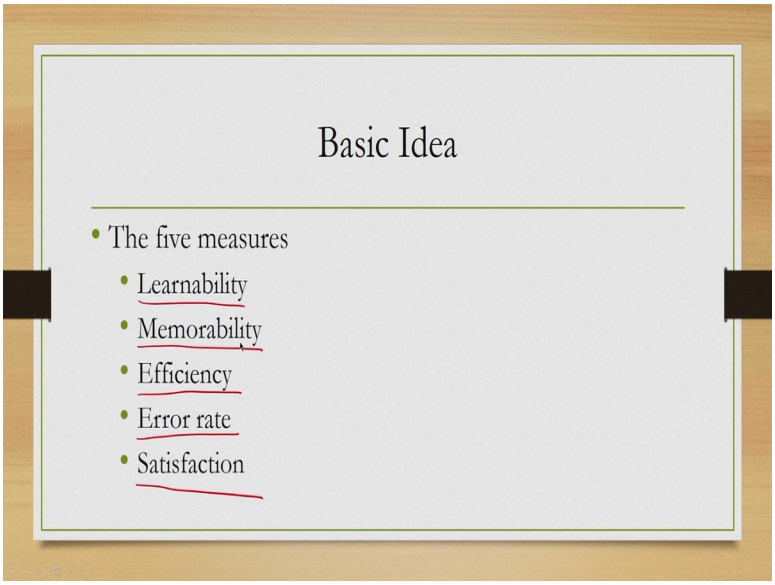
So, in order to start, we will start with a very basic question how do we evaluate a user centric system. So, if I ask you this question, what will be your answer? The most straightforward answer is of course, by evaluating usability of the system, but the next question is what we evaluate?

(Refer Slide Time: 03:05)



When we talked of evaluating usability, what are the things that we evaluate in usability?

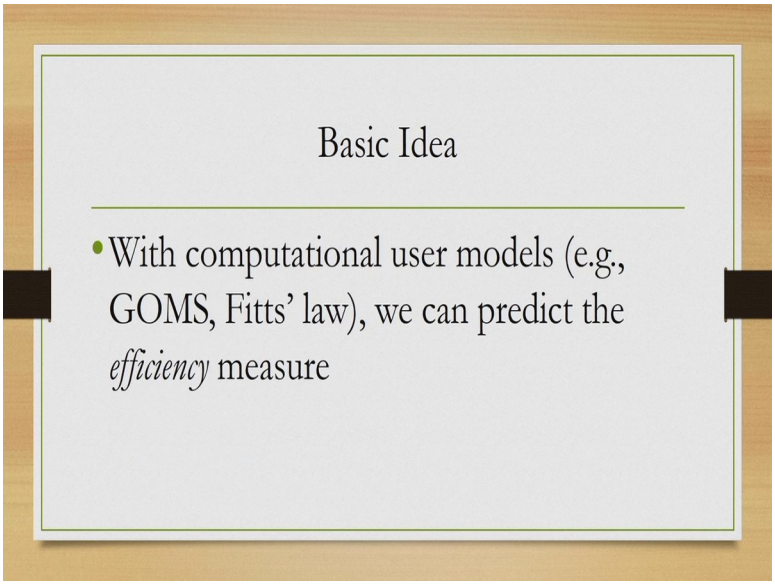
(Refer Slide Time: 03:15)



Now, in one of the earlier lectures; if you may recollect, we mentioned five measures of usability. Those are learnability, memorability, efficiency, error rate at user satisfaction.

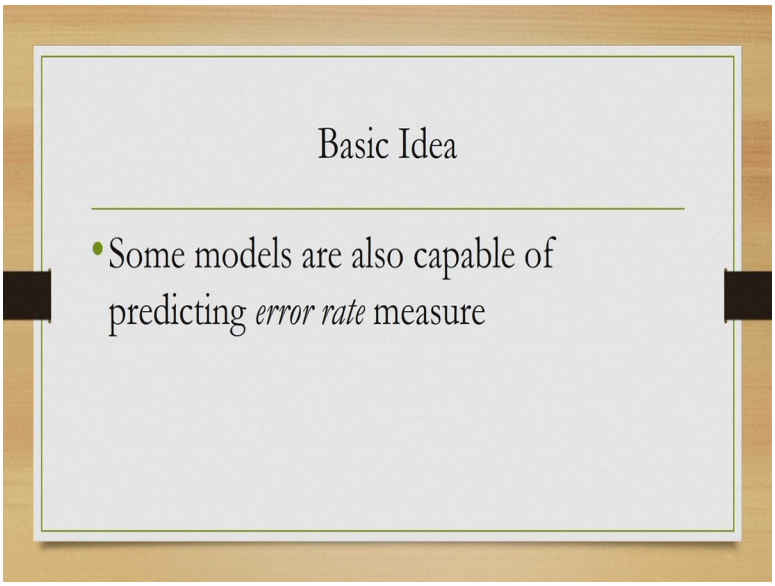
So, when we say that we want to evaluate a system in terms of usability, what we indirectly refers to is to measure these components of usability.

(Refer Slide Time: 03:45)



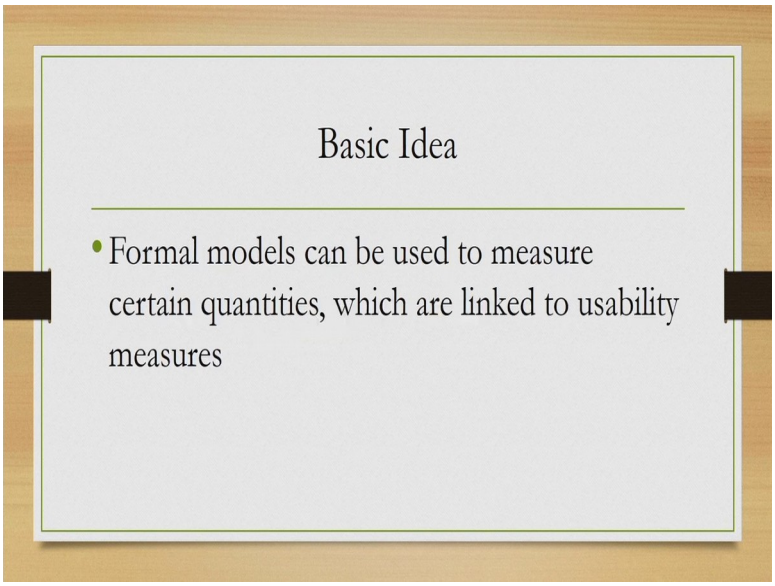
The computational models that we have discussed earlier can be used for the purpose as we have seen during the discussion on those models. Model such as GOMS or the Fitts' law or the Hick Hyman law can be used to predict performance measures that are related to the efficiency measure of usability.

(Refer Slide Time: 04:13)



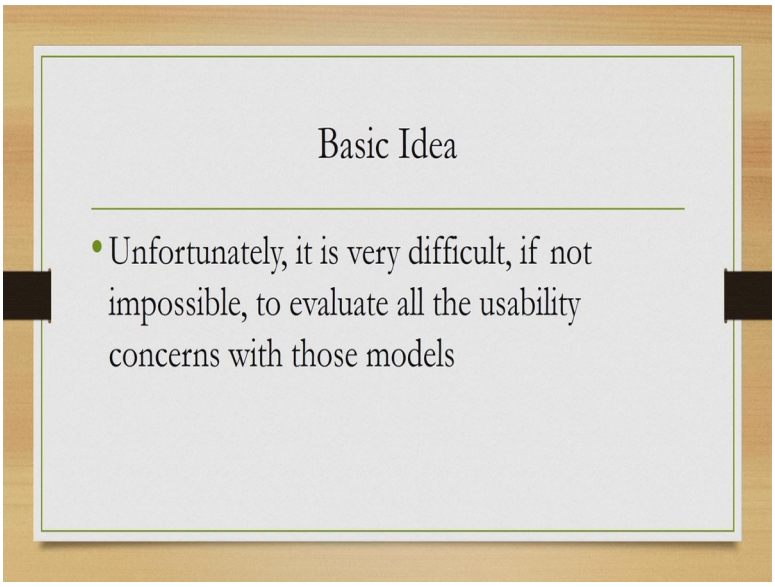
Some models are also there which can be used to measure error rate.

(Refer Slide Time: 04:21)



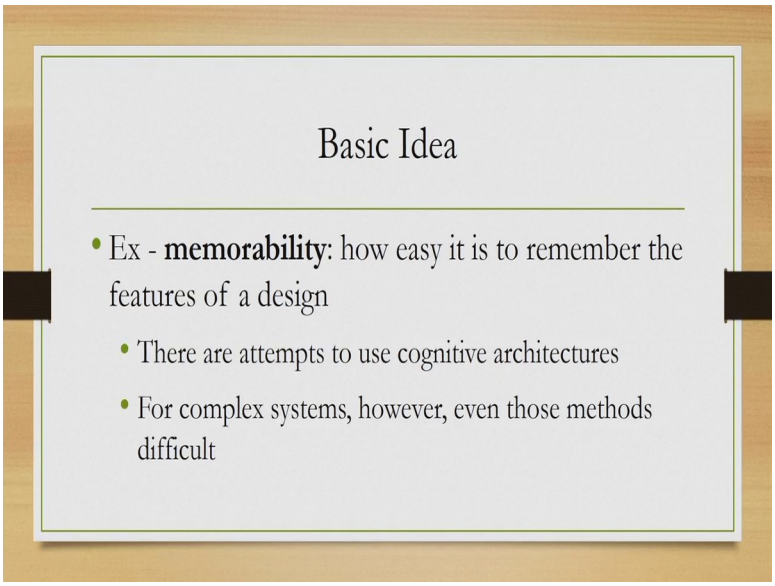
However, the formal models that we have discussed can also be used to measure certain quantities which can indirectly be linked to the usability measures. These again we have seen during the discussion on the use of formal models.

(Refer Slide Time: 04:43)



But the point is that it is very difficult or maybe impossible to evaluate all the usability concerns with these models. So, the models can be utilized to measure few things such as efficiency, error rate, but there are certain components of usability which may not be possible to be measured with this formal models.

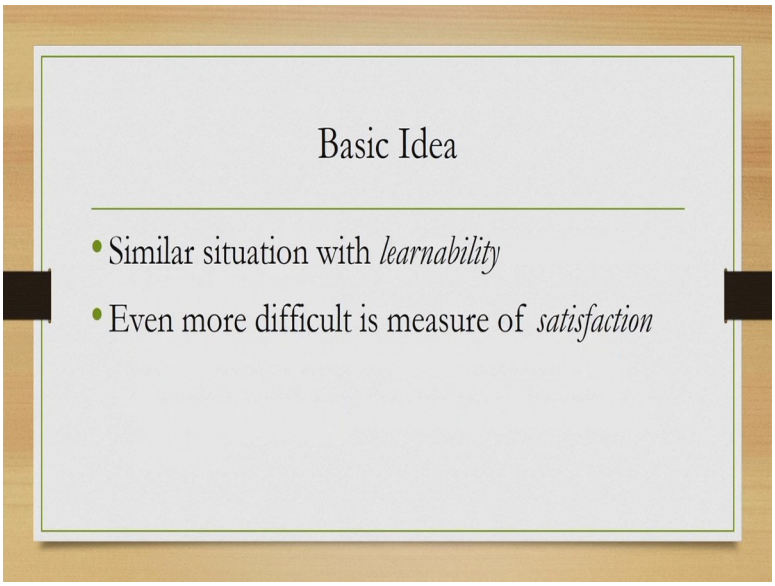
(Refer Slide Time: 05:19)



For example if you are asked to measure memorability, how can that be measured with formal model? So, what is memorability? It essentially refers to the ease with which we can remember the features of a design that is the definition which is of course, somewhat vague cannot be quantified easily and there lies the problem how to measure it.

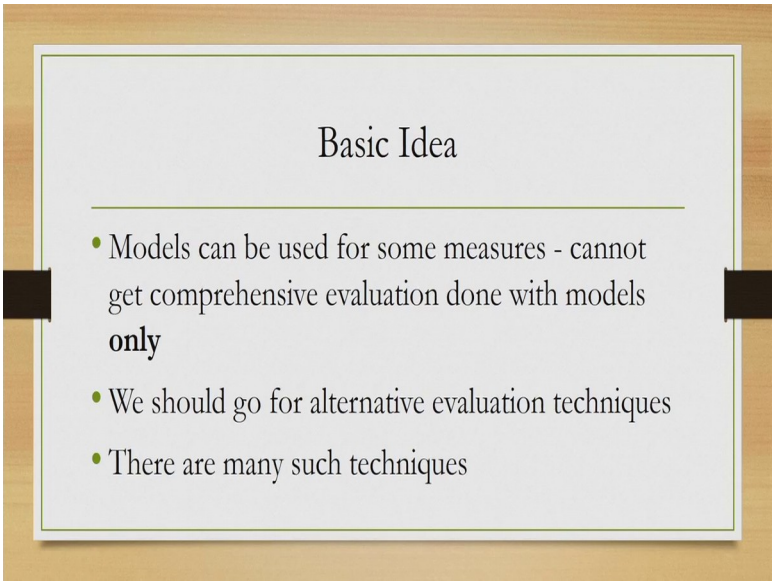
Now, there had been attempts to measure it using cognitive architecture. Now, these cognitive architectures, we have not discussed earlier. This is an advanced form of user models and they are quite complex, but even with cognitive architectures; it is possible to measure memorability of very simple systems. For complex systems even such advanced techniques such as the cognitive architectures will fail.

(Refer Slide Time: 06:19)



Similarly, there is an issue with measurement of learnability or satisfaction. So, in a nutshell what we can say is that we can use models whichever we have learned earlier. Two major say the error rate or the efficiency, but these models may not be sufficient or suitable to measure memorability, learnability, subjective satisfaction.

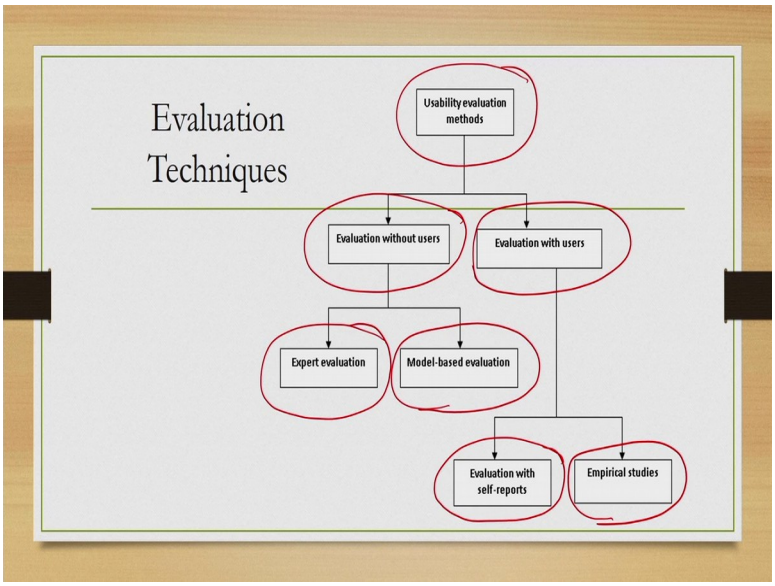
(Refer Slide Time: 07:03)



So, some components of usability can be measured with models; some may not be measured with models only. So, then what we need? We have to go for alternative techniques to evaluate those systems and evaluate those measures. So, given a system

only with models we may be able to evaluate its efficiency or error rate, but we may not be able to evaluate memorability, learnability, satisfaction. So, overall usability measurement or overall usability evaluation may not be possible only with models and we have to go for alternative evaluation techniques.

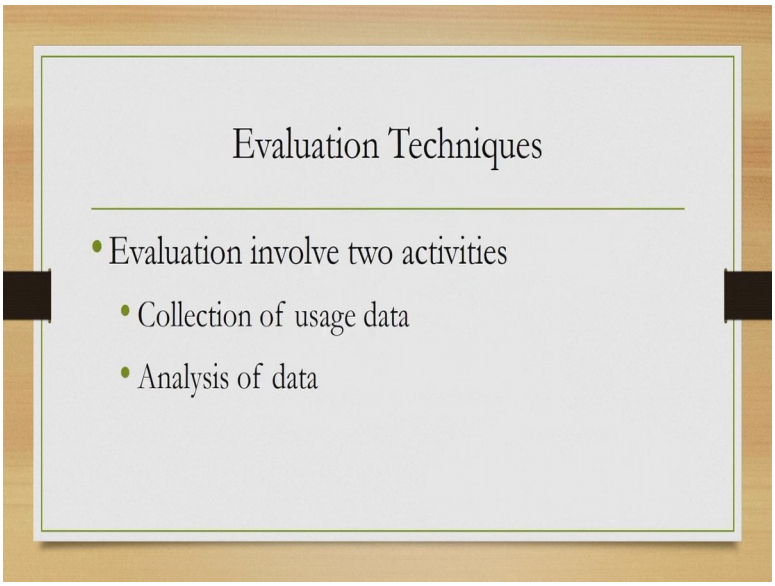
(Refer Slide Time: 07:51)



There are in fact, many such techniques available, we may categorize all these techniques in the form of a hierarchy which is shown here. So, all the evaluation methods may be divided into two broad categories; evaluation without users and evaluation with users. Now under evaluation without users again, we have two broad categories. One is evaluation that are done by experts or expert evaluation and the other category of evaluation methods comes under model based evaluation.

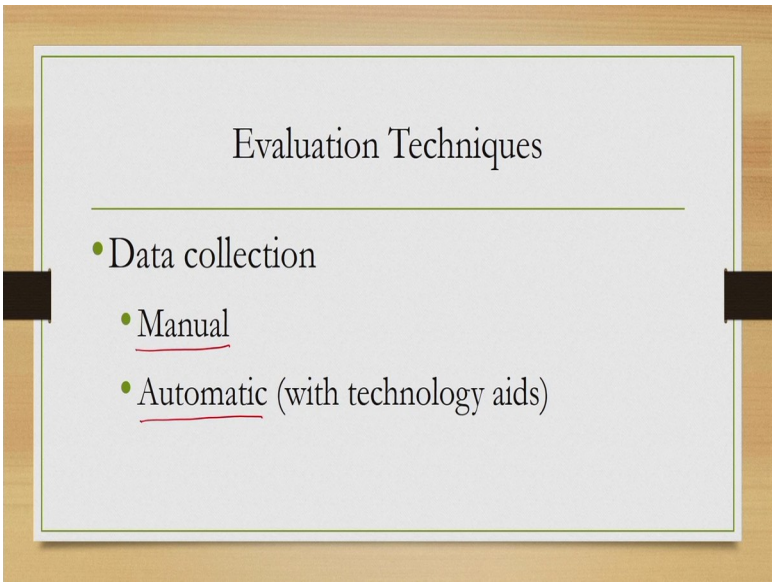
The category evaluation with users the techniques that belong to this category can again be divided into two groups; one is evaluation with self reports and other one is empirical studies. So, broadly all the evaluation methods we can categorize into this hierarchical form.

(Refer Slide Time: 09:03)



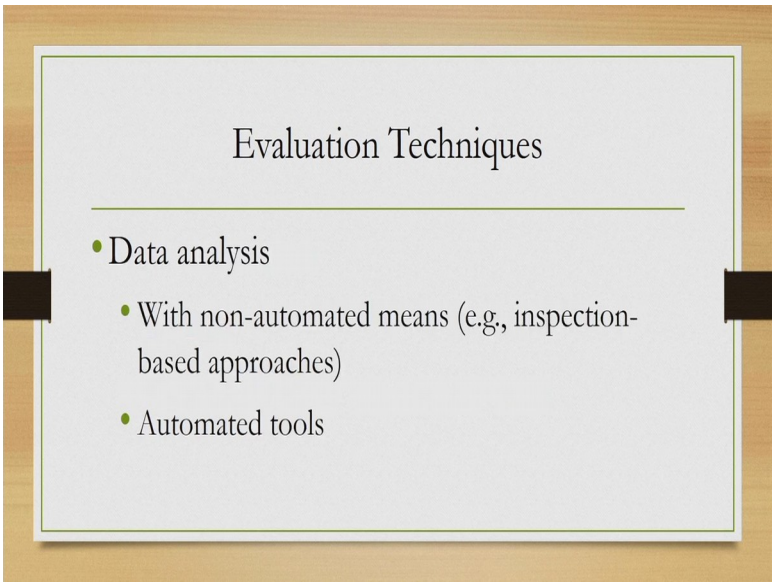
There is another way to view this. Before we discuss that let us try to look at another aspect of evaluation that is how we perform the evaluation. It involves two stages or two activities, one is collection of data other one is analysis of data.

(Refer Slide Time: 09:27)



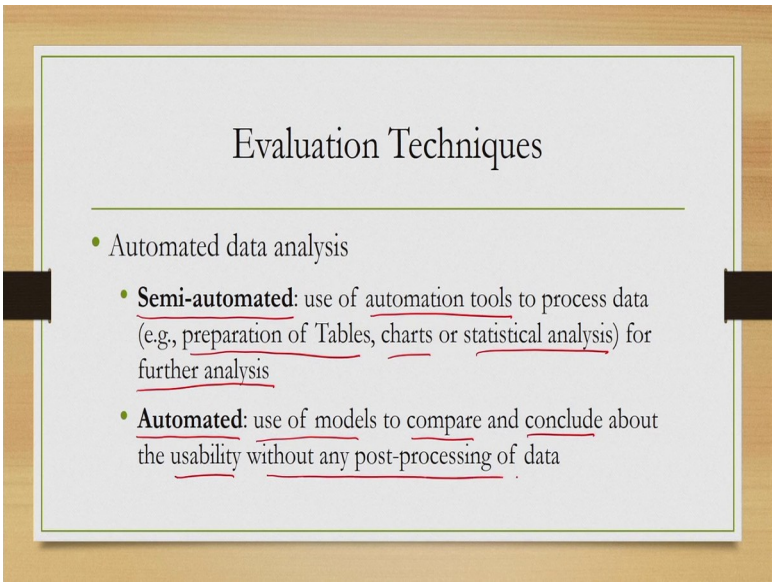
Now, if you may recollect in a previous lecture, we mentioned that the data collection can be performed either manually through intervention of a human observer. And also it can be performed automatically with the use of technology aids such as computers, recorders, sensors and so on.

(Refer Slide Time: 09:49)



Similarly, analysis of data can also be performed in different ways. We can have non automated means of analysis for example, inspection based approaches or we can have tools to perform the analysis automatically.

(Refer Slide Time: 10:17)



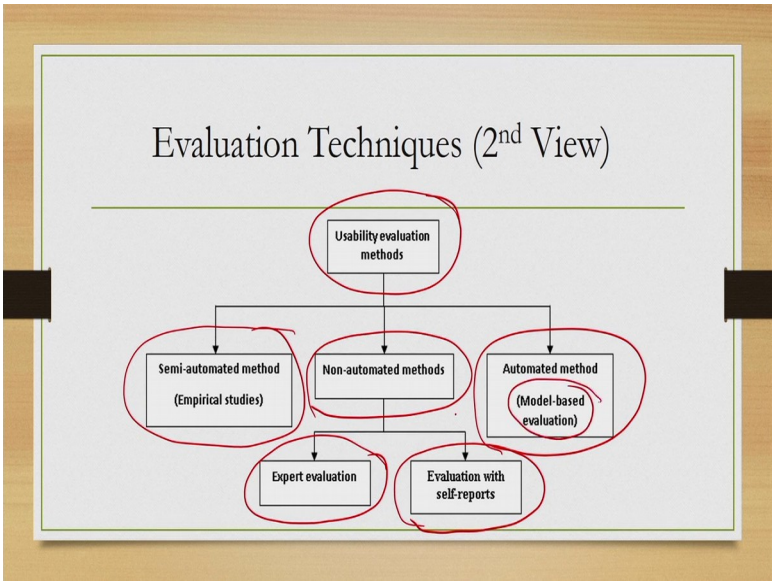
Now these automated analysis tools and the use of tools to perform data analysis automatically can again be divided into two subcategories. So, automated data analysis can be done in either of the two ways; one is semi-automated approach, other one is automated approach. So, in semi-automated approach, to use automation tools for

processing of data primarily. For example, preparation of tables, charts, statistical analysis and these help us in further analysis of the data which we need to do sometimes manually.

However, in case of purely automated data analysis approach, we use models to compare and conclude about the usability without any post processing of data. So, to summarize any evaluation involves two activities data collection and data analysis. Data collection we can do it either with manual intervention or with the use of automated tools. Data analysis also we can perform in an automated way or in an automated way. In automated data analysis approach, there are two categories.

In one category which is called semi automated analysis, we use tools automation tools to primarily process the data rather than finally, analyze it and in fully automated approach, we use models to analyze and conclude about the data without any post processing.

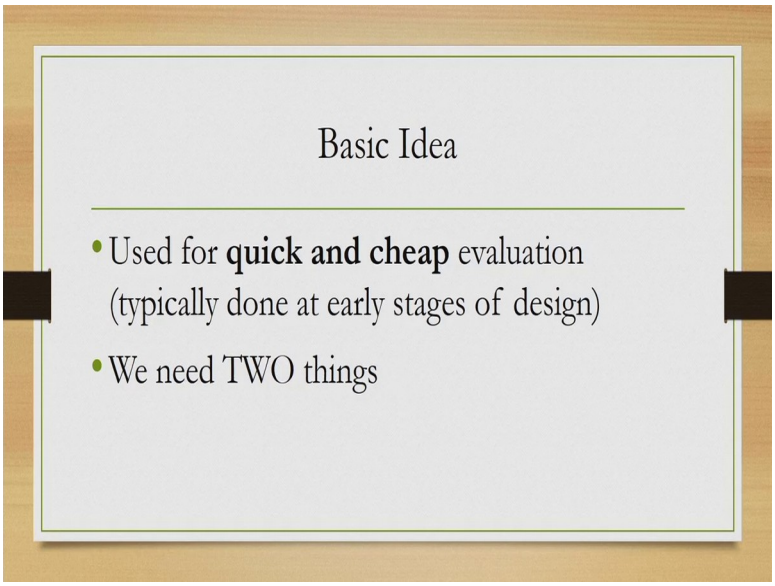
(Refer Slide Time: 12:23)



So, based on the way we collect and analyze data, we can have a different hierarchy of usability evaluation methods than the one we have discussed earlier. In this hierarchy, we have three broad categories one is semi automatic method; one is non automated method, one is automated method. The empirical studies that we have discussed earlier comes under semi automated method.

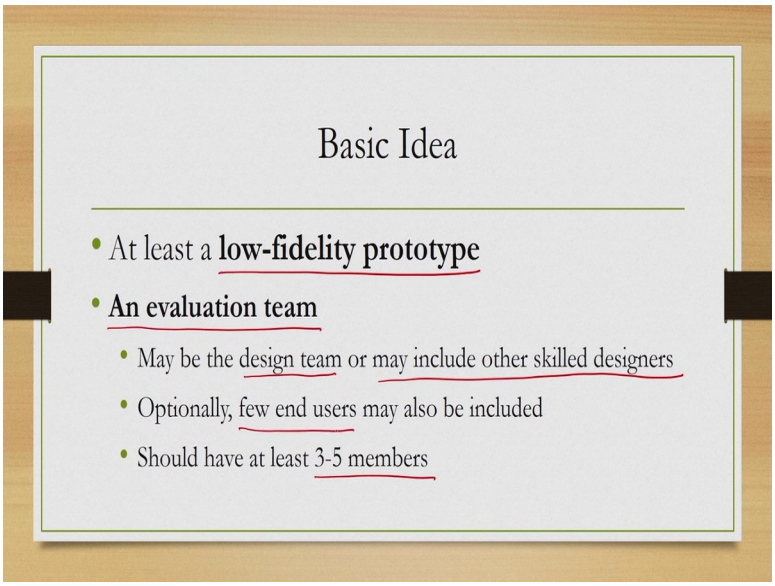
Now, under non automated methods there can be two categories of evaluation methods one is expert evaluation, other one is evaluation with self reports. And under automated method, we have the model based evaluation. Now, let us try to understand these techniques one by one. So, we will start with expert evaluation what do we mean by this.

(Refer Slide Time: 13:31)



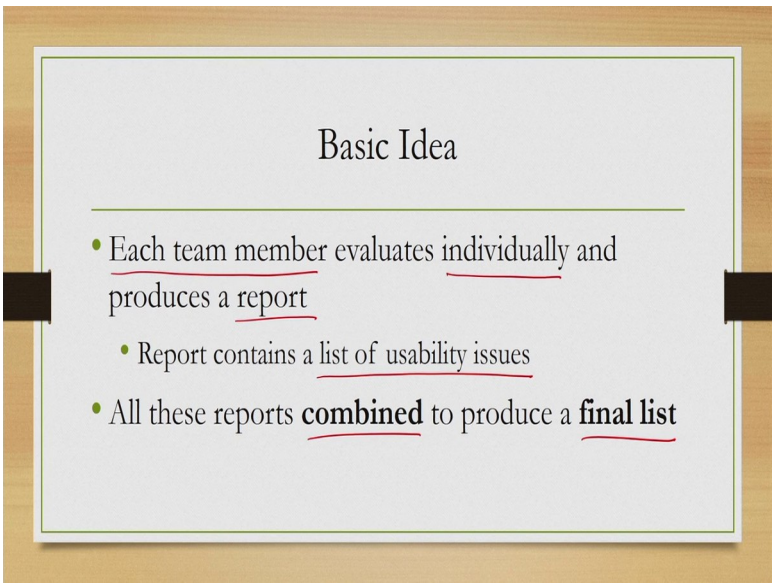
So, when we are talking of expert evaluation, it is essentially referring to an evaluation method that is used for quick and cheap evaluation. And typically this is done at early stages of design when lots of iteration takes place. In order to perform expert evaluation, we need two things what are those two things.

(Refer Slide Time: 14:05)



The first thing is a prototype. So, we require at least a low fidelity prototype and the second thing that we need is an evaluation team which should have at least 3 to 5 members. Now, who are these members? They may be from the design team or may include some other skill designers who are not part of the design of the particular product optionally we may also include few end users in the team.

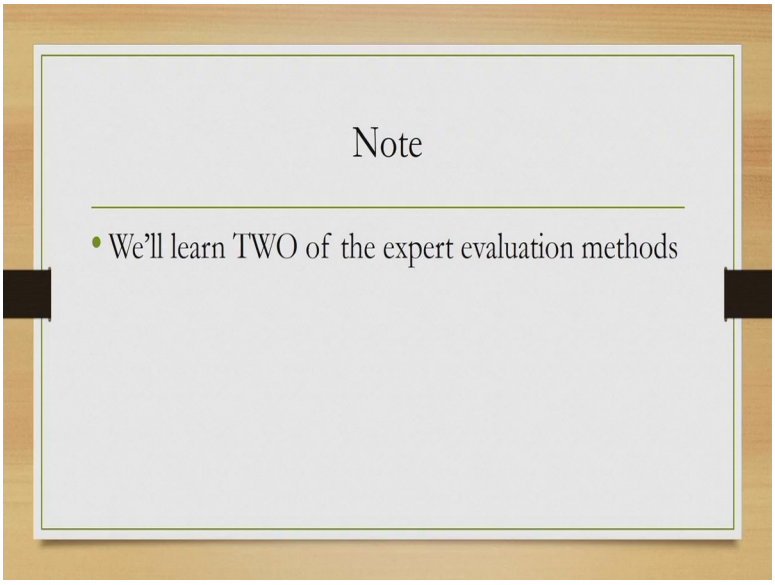
(Refer Slide Time: 14:41)



So, what they do? So, each team member evaluates individually and produces a report. At the end of the evaluation, all these reports are combined to produce a final list. What

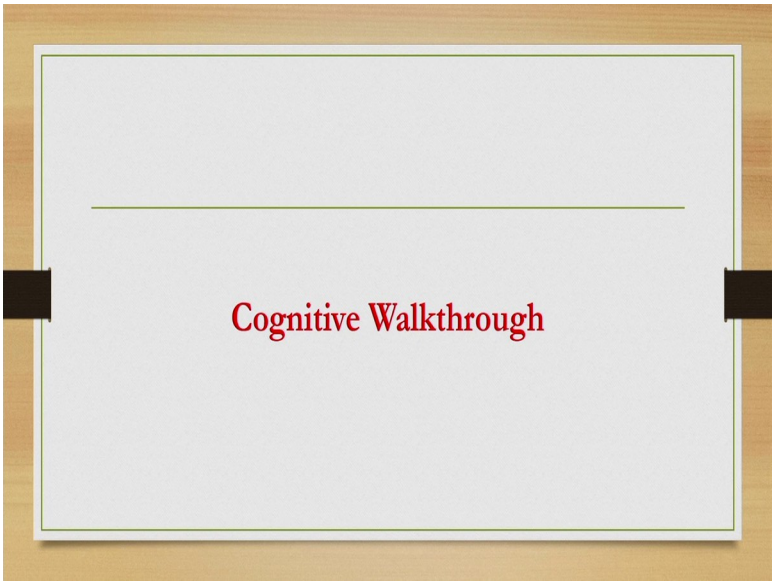
this list contains? Now each report contains a list of usability issues and at the end when we combine some issues may be duplicated across the reports. Those are removed and at the end, it contains a list of unique usability issues identified by the evaluators.

(Refer Slide Time: 15:15)



Now, there are many expert evaluation techniques, we will learn two of those namely the cognitive walkthrough and the heuristic evaluation.

(Refer Slide Time: 15:27)



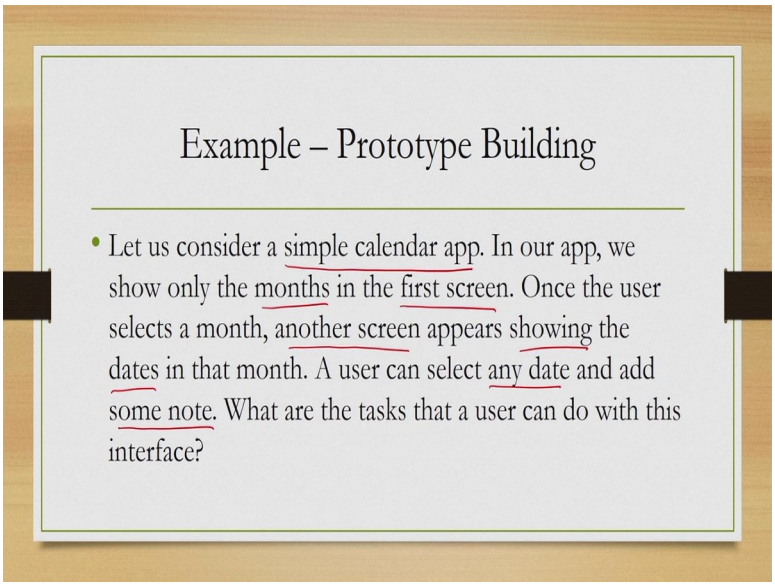
Let us start with cognitive walkthrough.

(Refer Slide Time: 15:29)

Now, this is an usability inspection method and its requirements are the same as that of any expert evaluation method that is we need at least a low fidelity prototype and team consisting of 3 to 5 members which may include; may include the designers, some other skilled designers or endusers.

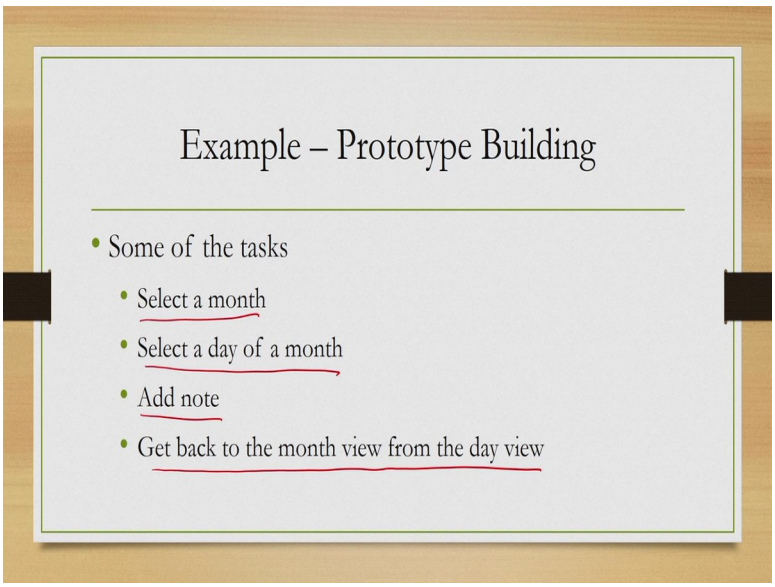
Now, in case of cognitive walkthrough the prototype that we require should support several tasks not a single tasks and it should be developed as vertical prototype. Let us try to understand how to create this prototype and what we meant by support for several tasks and vertical prototype in this context of cognitive walkthrough.

(Refer Slide Time: 16:29)



Let us consider an example suppose we want to develop a simple calendar app. In our app, we show only the months in the first screen and once the user selects a month, another screen appears showing the dates of the month. Now a user can select any date and add some note. So, what are the tasks that a user can do this interface? There can be several tasks.

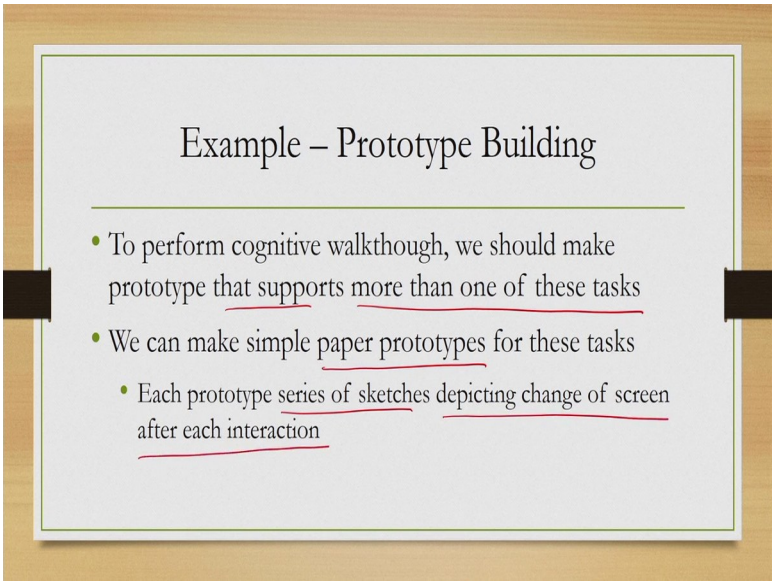
(Refer Slide Time: 17:11)



Let us see a few of those one task can be select a month, another task can be select a day of a month, third task can be add a note, fourth task can be get back to the month view

from the day view there may be other tasks as well. Now, when we are talking of building a prototype for performing a cognitive walkthrough, what we need to do?

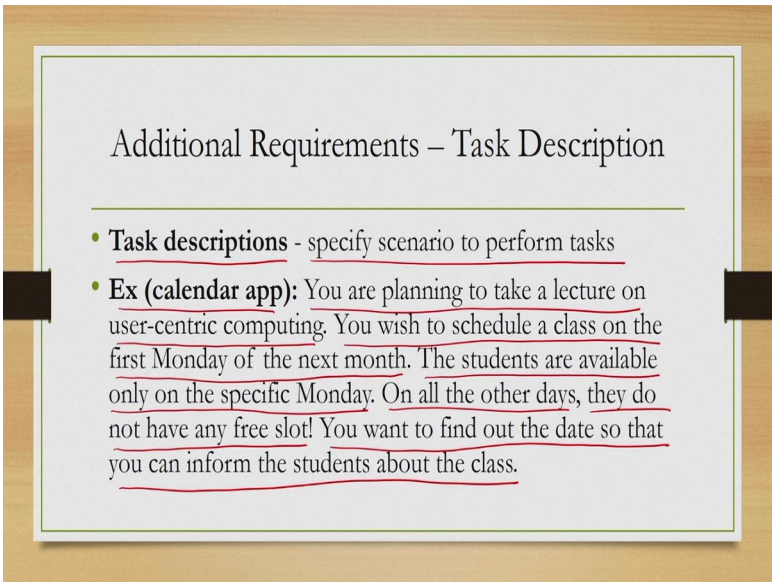
(Refer Slide Time: 17:37)



We need to build a prototype that supports more than one of these tasks. However, it is not necessary to go for very sophisticated prototypes, we can work with simple paper prototypes as well. That is why we called it low fidelity and each prototype can simply be a series of sketches depicting the change of screen after each interaction.

So, when we say that we want to build a prototype to perform the task of selecting a month or a date. So, for this task we can actually create a series of sketches depicting the change of screen when we perform the task. But since all the screens are depicted so, we are calling it a vertical prototype.

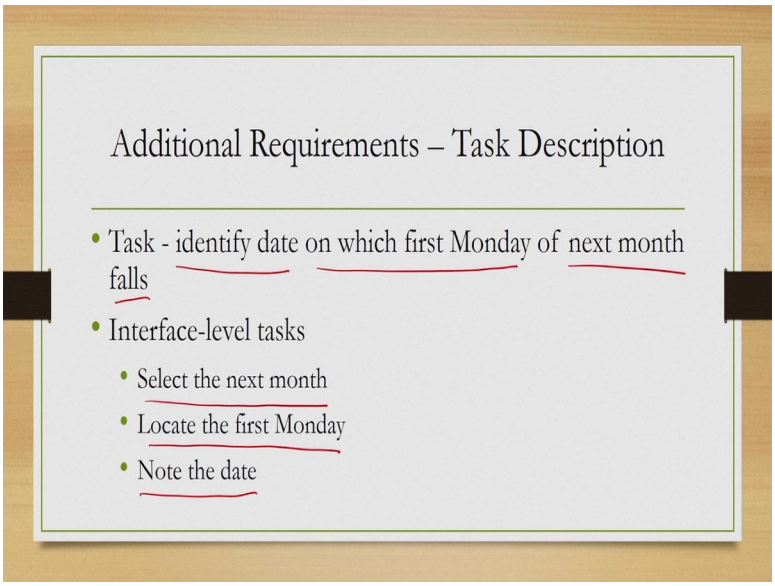
(Refer Slide Time: 18:35)



So, along with those that is requirement for a low fidelity vertical prototype for several tasks and design team, what else we require for performing a cognitive walkthrough? One is task description as we have just seen before; what it means is specify the scenario to perform tasks.

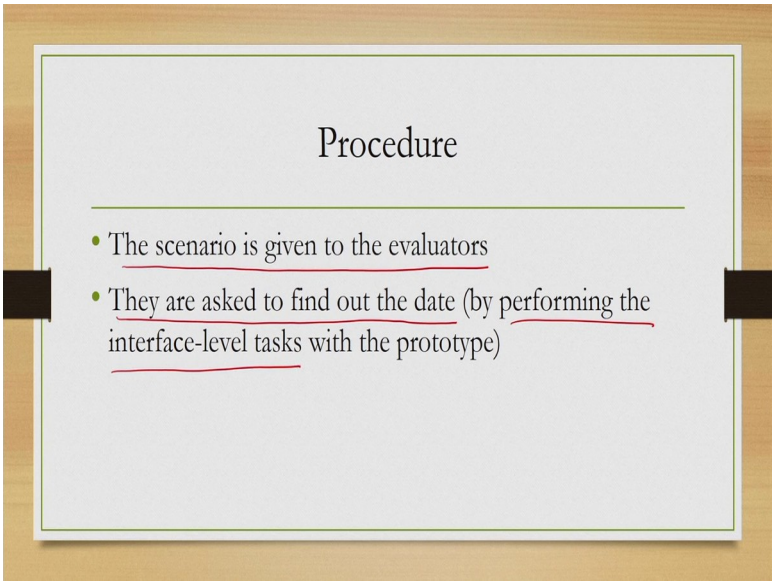
So, for the calendar app example that we have just discussed, what can be a scenario? One can be something like this that you are planning to take a lecture on user centric computing. You wish to schedule a class on the first Monday of the next month. The students are available only on the specific Mondays. On all the other days they do not have any free slot. So, you want to find out the date so, that you can inform the students about the class. This is the usage scenario or task scenario.

(Refer Slide Time: 19:49)



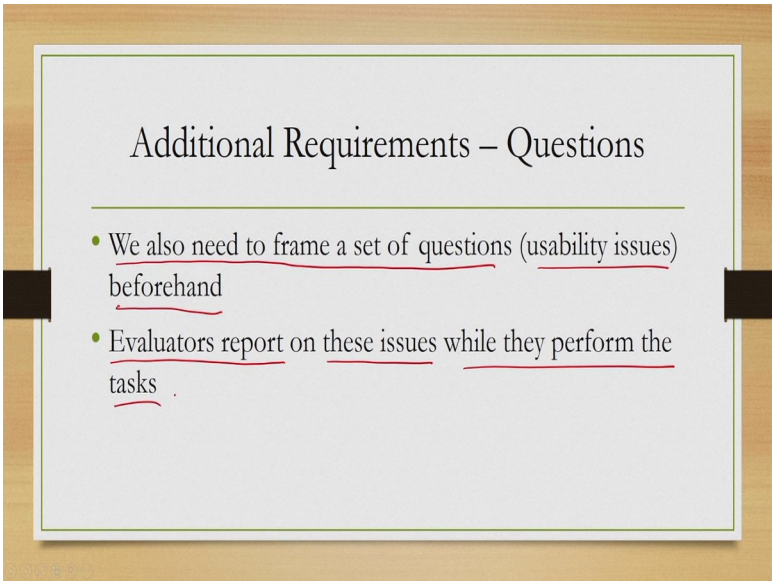
So, in order to achieve this, what we need to do? We need to identify the date on which first Monday of the next month falls. What are the sub tasks involved or the tasks that we need to perform on the interface involved in order to achieve this task? Select the next month, locate the first Monday and note the date. So, if we can perform these tasks on the interface, then we will be able to achieve the overall task of identifying the particular date.

(Refer Slide Time: 20:29)



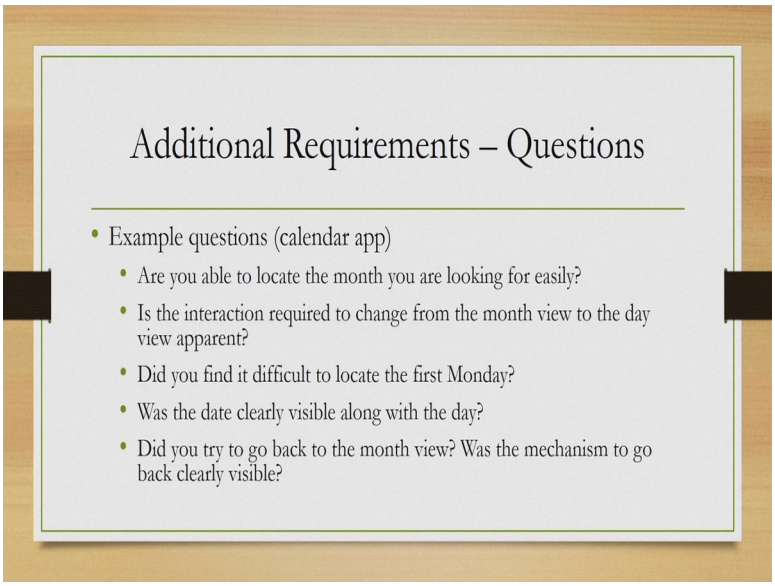
So, in cognitive walkthrough, what we do this scenario is given to all the evaluators. Remember that we should have more than one scenarios, but in this example we are showing only one scenario, in practice there should be more than one scenario. Now each scenario is given to the evaluators and they are asked to find out the date by performing the interface level tasks with the prototype that is given to them.

(Refer Slide Time: 21:17)



So, each evaluator is asked to perform the task with the prototype. There is one more requirement in order to perform a cognitive walkthrough. We also need to frame a set of questions beforehand. So, these questions are related to the usability issues and the evaluators are given these questions and they are ask to report on these issues while they perform the tasks.

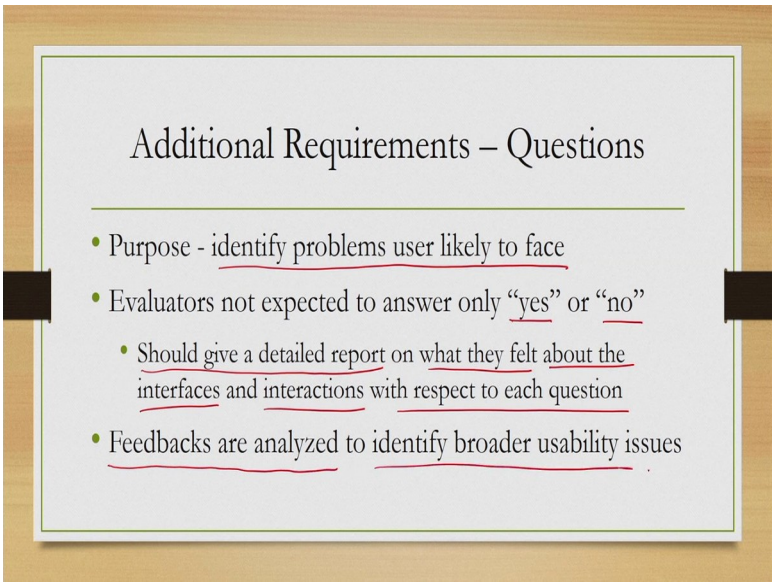
(Refer Slide Time: 21:49)



Let us see some example for our calendar app, what can be the questions, are you able to locate the month you are looking for easily. There can be another question which is the interaction required to change from the month view to the day view apparent. Third question can be did you find it difficult to locate the first Monday. There can be a fourth question which is was the date clearly visible along with the day.

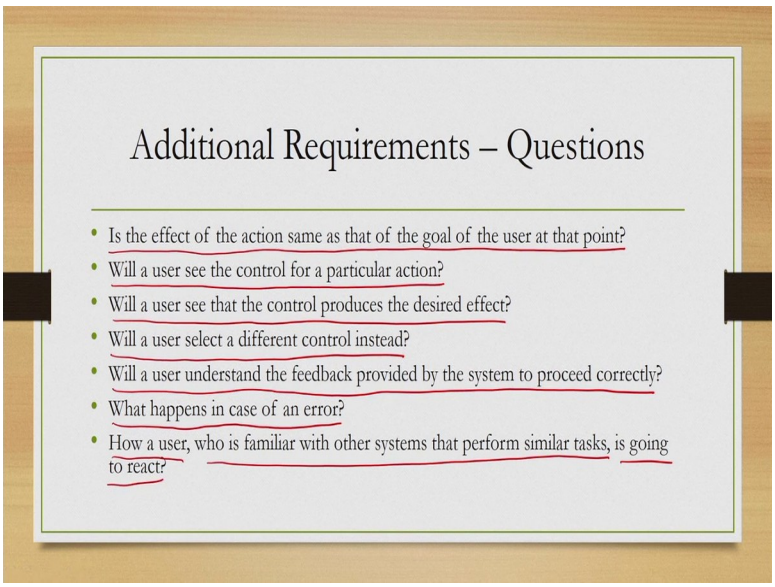
Even a fifth question did you try to go back to the month view was the mechanism to go back clearly visible. There can be many more questions; these are only some sample questions that can be given to the evaluators. So, each evaluator will be given these questions and they will be ask to right their opinion on each of these questions while they are performing the tasks with the prototype.

(Refer Slide Time: 22:59)



What is the purpose of these questions? As I mentioned before to identify problems that the user is likely to face. So, one very important thing is that the evaluator should not answer only in terms of yes or no. They must give a detailed report on what they felt about the interface and the interactions with respect to each question. So, only a binary answer yes or no will not be sufficient. Once the feedbacks are received from all the participants, from all the evaluators; they are compiled together and analyzed to identify broader usability issues.

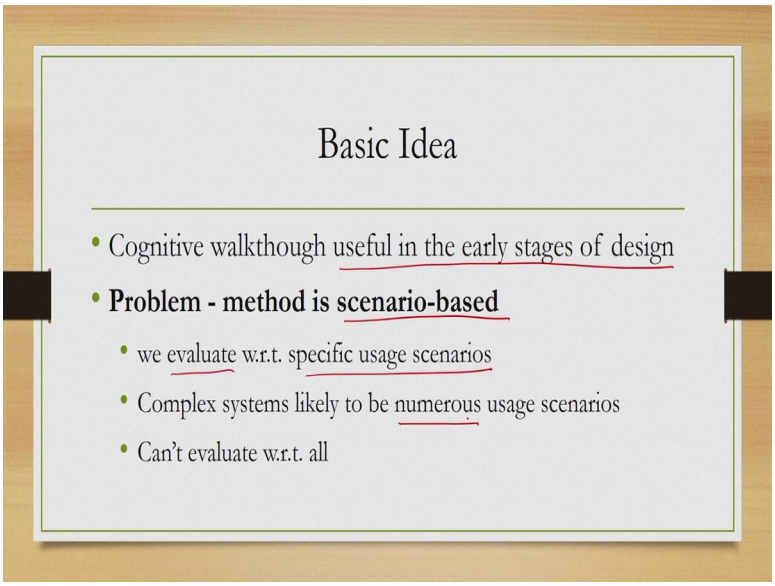
(Refer Slide Time: 23:49)



What are those broader issues? Few are shown here as an analyst you can try to answer these questions based on the feedback received that is the effect of the action same as that of the goal of the user at that point, will a user see the control for a particular action, will a user see that the control produces the desired effect, will a user select a different control instead, will a user understand the feedback provided by the system to proceed correctly, what happens in case of an error, how a user who is familiar with other systems that perform similar tasks is going to react.

These are some of the broader issues that can be answered based on the feedback received from the evaluators by analyzing their answers to each of the questions.

(Refer Slide Time: 24:57)



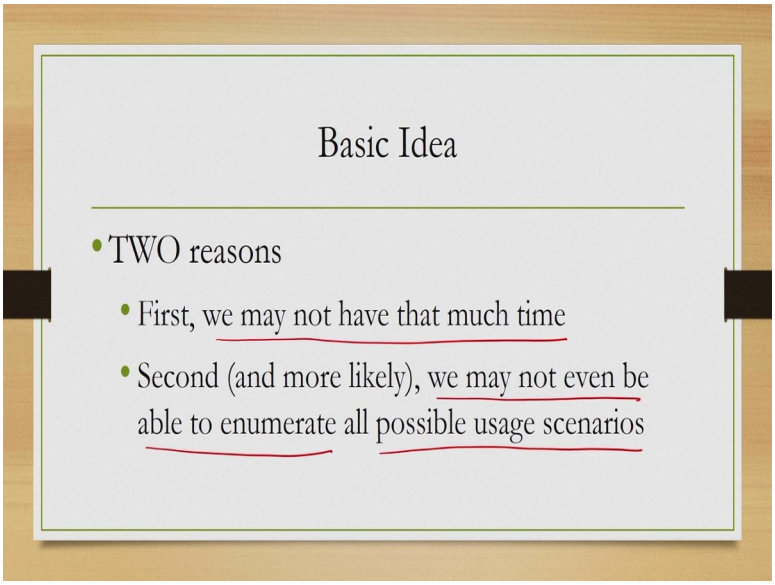
So, that is broadly what we perform in a cognitive walkthrough just to recap. So, we start with a low fidelity vertical prototype for more than one scenarios preferably and evaluation team comprising of 3 to 5 members. We frame the scenarios and based on those scenarios, we create the prototypes. Each evaluation team member is given the prototypes along with the prototypes, they are given the task scenario and they are asked to perform the tasks.

Along with that they are given set of questions and they are asked to give their opinion or feedback for each of those questions while they perform the tasks. These questions are meant to elicit response on the performance of the interface and interaction with respect to user and they are meant to identify broader usability issues.

So, as you can see here the issues are identified based on inspection of the interaction; inspection of the overall interface and interaction design. There is another evaluation technique which is called heuristic evaluation. In case of cognitive walkthrough it is useful in the early stages of the design.

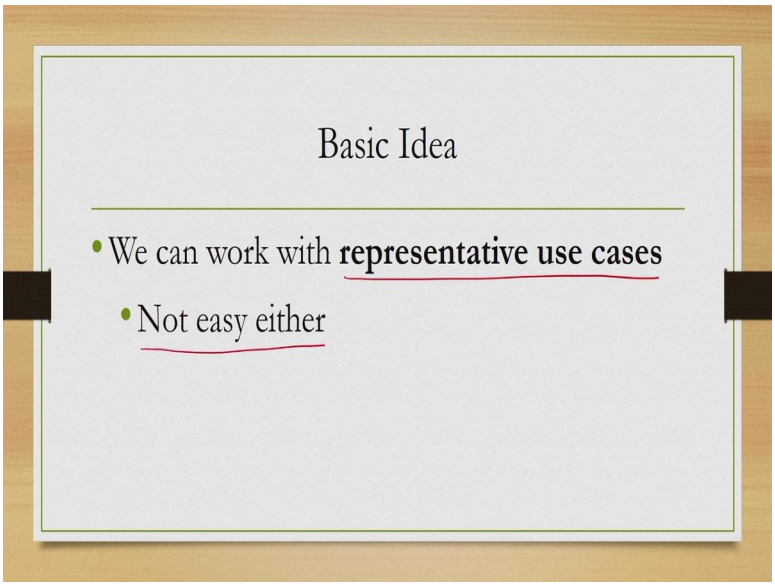
Now, the problem with this method is that it is scenario based. So, we evaluate with respect to specific usage scenarios. In case of complex systems that is a problem because complex systems are likely to have numerous scenarios and we cannot evaluate with respect to all.

(Refer Slide Time: 27:01)



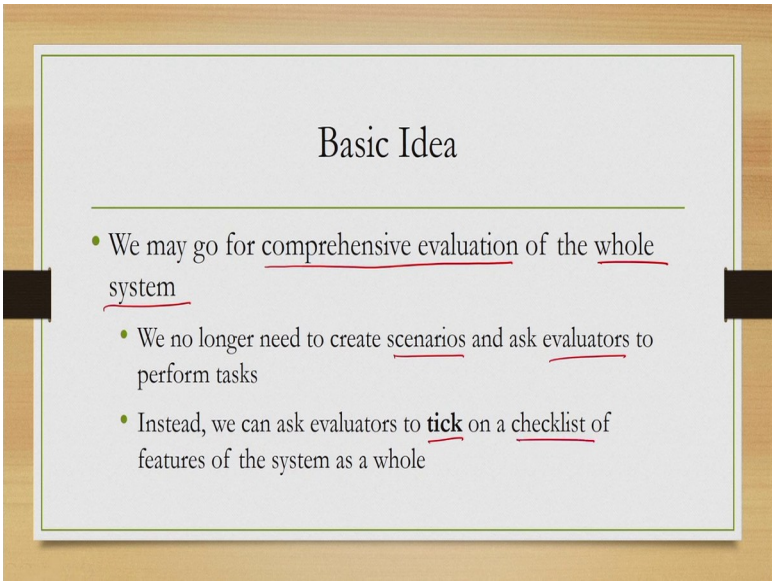
There are two reasons for that. First of all we may not have sufficient time to do that. Secondly, even if we have time we actually may not be able to enumerate all possible usage scenarios. As an example if you are using any advanced text editor to type messages such as Microsoft Word, you may try to enumerate all the possible usage scenarios to understand the difficulty involved.

(Refer Slide Time: 27:41)



What we can do instead of trying to enumerate all possible usage scenarios? We may try to figure out representative use cases and corresponding scenarios. However, as we have discussed in one of our previous lectures. This is also not easy either identifying the representative use cases.

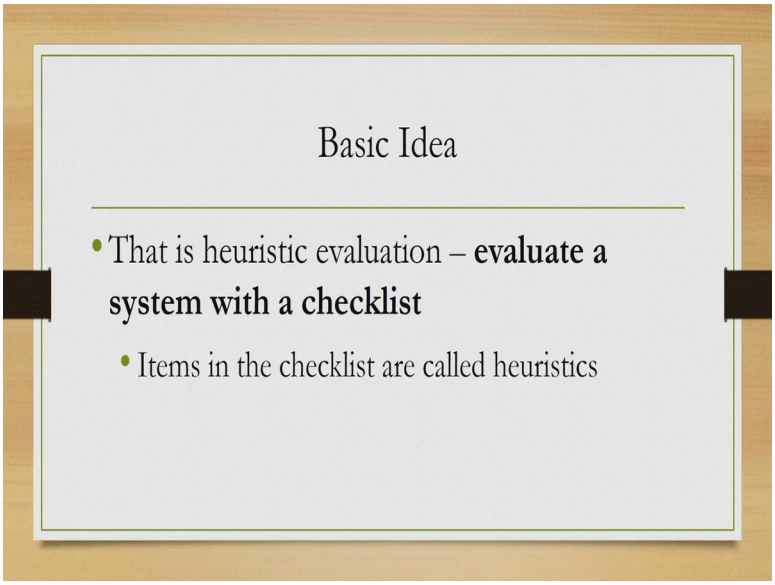
(Refer Slide Time: 28:07)



In such situations, what we can do? We can actually go for comprehensive evaluation of the whole system. So, we no longer need to create scenarios and ask evaluators to perform tasks instead we can ask evaluators to tick on a checklist of features that the

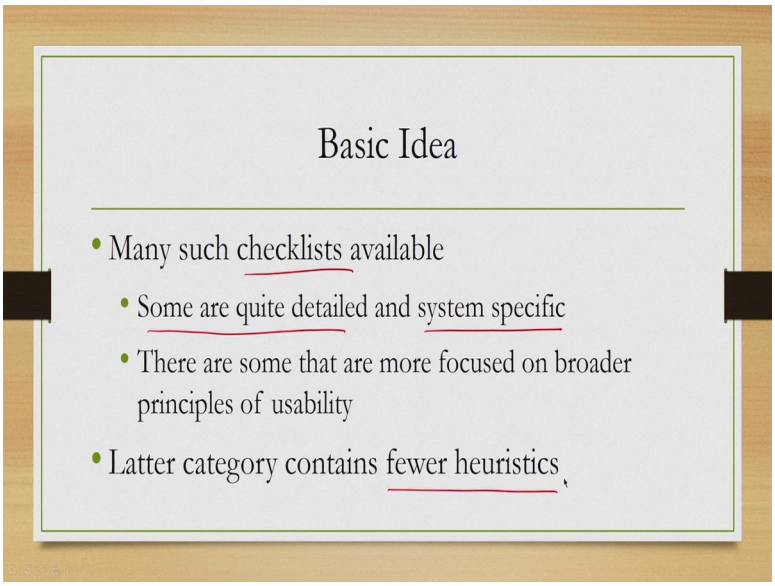
whole system supports. So, this is inspection based method. We are actually inspecting all the features and trying to figure out if there is any issue or not.

(Refer Slide Time: 28:43)



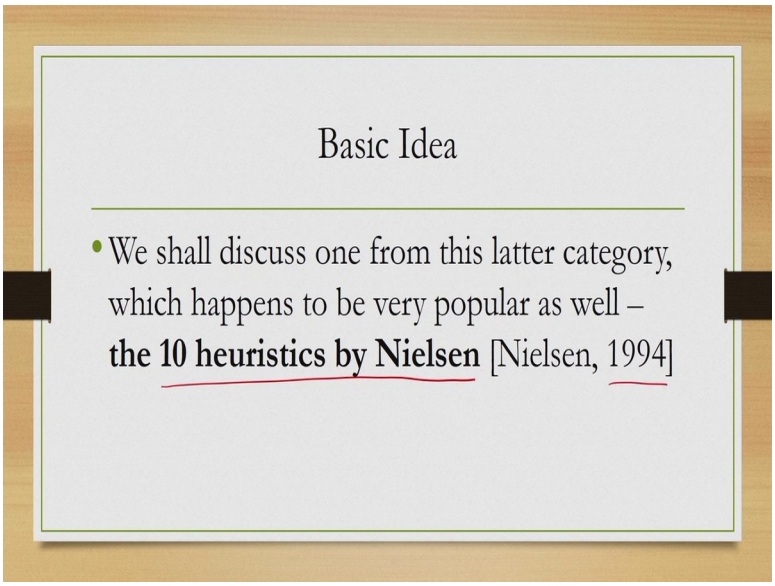
Now, this approach is called heuristic evaluation. We evaluate a system with a checklist and the items in the checklist are called heuristics.

(Refer Slide Time: 29:03)



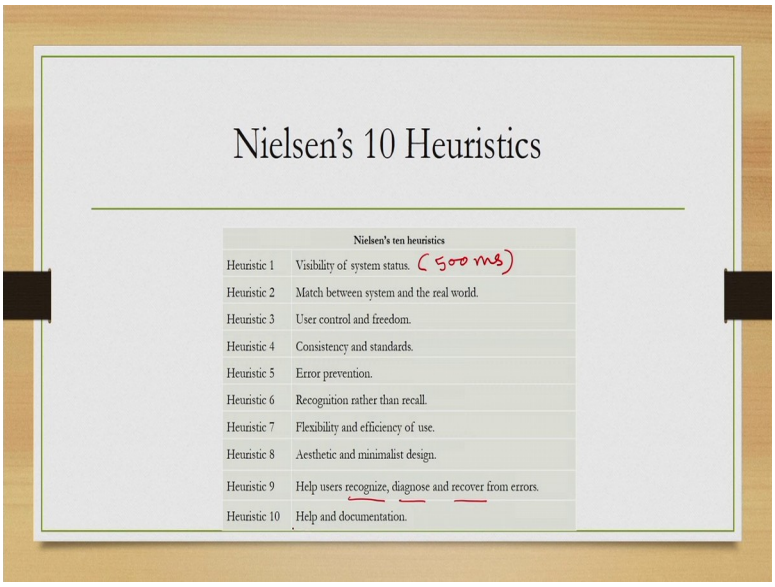
There are many such checklists available; some are quite detailed and system specific and there are some that are more focused on broader principles of usability and clearly because of that reason the corresponding checklists are having fewer heuristics.

(Refer Slide Time: 29:31)



In this lecture will discuss, one from this latter category of heuristics which focuses on broader issues which is called the 10 heuristics by Nielsen or Nielsen's 19 heuristics which was proposed by Nielsen in 1994.

(Refer Slide Time: 29:49)



Let us have a look at those 10 heuristics as shown in this table. So, heuristic 1 says the system status should always visible to the user what it means is that the user should always get a feeling of what is going on in the system. So, essentially it refers to some sort of feedback for system activities which should be given to the user for whatever

tasks he or she performs. Not only that it also tells that the feedback should be given within a reasonable time. It should not be too late so, that it is difficult to relate the feedback with the activity a time is mentioned which is 500 milli seconds. So, essentially what it tells is that give feedback within 500 milli second, then it will be more effective.

In heuristic 2, what it tells is that there should be a match between the system and the real world. So, whatever we get to see in the real world should be used to the extent possible to design things in the system. So, that there is some consistency between our experience in the real world and our experience in the virtual world. If you may recollect earlier we talked of guidelines, there we talked of external consistency; consistency between the system and the real world. This heuristic essentially points to the same.

The third is user control and freedom. So, this third heuristic refers to the; refers to the fact that the design should have undo and redo facilities because users may make mistakes and they should be able to come out of those mistakes without going through elaborate dialogues. So, if they can come out quickly then, they feel that they are in control and their freedom to do whatever they wish.

Heuristic 4 is on consistency and standards that is throughout the system you should follow some consistency and it should follow standards. So, that every time they user need not learn new things essentially it refers to internal consistency in the system. The concern of designs that helps the user to prevent errors is taken care of in heuristic 5 that we should go for designs that prevents occurrence of errors.

Heuristics 6 is interesting what it tells is that whatever you do to design the system should help the user recognize the purpose rather than forcing the user to recollect the purpose. So, in other words by looking at the interface, we should be able to recognize what it does or by looking at the interface objects we should be able to recognize what it is meant for, what this objects are meant for, how to use them rather than the requirement of remembering the usage or purpose of those objects or the overall interface.

A simple example is the design of a pattern. So, if I look at a button, I immediately recognize that it is meant for clicking. So, that shape or the way it is designed makes it apparent that is recognition. Now, if I use a very abstract symbol to represent a button and then force the user to remember that this symbol means button which in turns means you have to click on it, then that is called recall. Clearly recall is a bad idea. So, if the

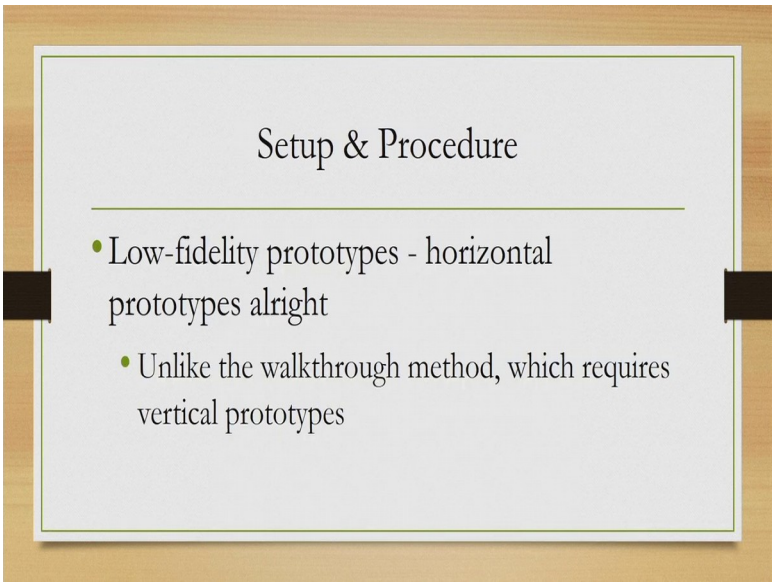
more we force the user to recall the less, the visibility of the interface is. So, whatever we design our objective should be to help the user recognize rather than force the user to recall.

The 7th heuristic talks about flexibility and efficiency of use. Essentially it refers to the fact that whenever you design we should keep in mind different categories of users namely the expert users, now is users intermittent users. In the 5th heuristic the 8 heuristic refers to a simple design which looks aesthetically pleasing and minimalist that is it a words redundancy. It is expected that such designs help in satisfying the users.

In the 9th heuristic, the fact that human makes mistakes irrespective of how well the design is has been taken care of. It says that help users recognize diagnose and recover from errors. So, it assumes that there will be errors irrespective of how well the design is, but the design should support the user to recognize diagnose and recover from errors. So, we should not give some very cryptic error messages to indicate the occurrence of an error instead it should produce some understandable messages preferably in natural language and produce some friendly advice on how to recover from the error and what are the likely consequences.

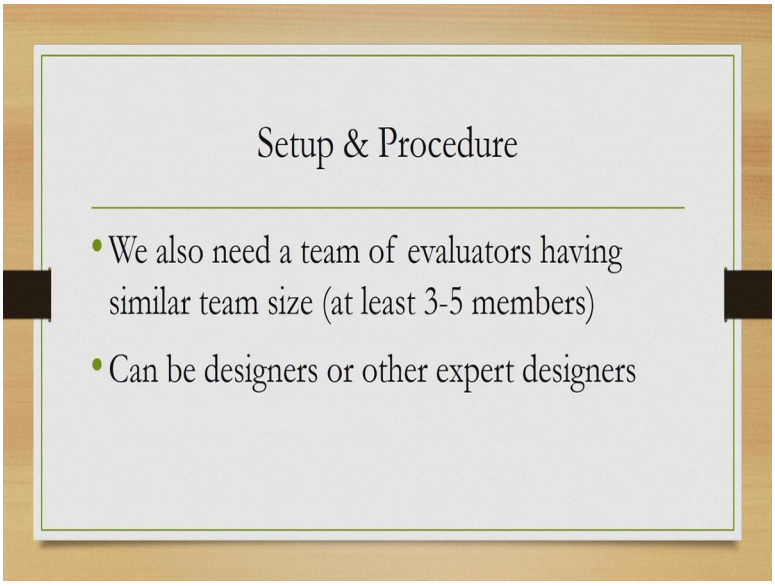
Finally in heuristic 10, it is emphasized that any design or any system should have suitably designed help and documentation where the object should be easily identifiable and searchable. Together the set of ten heuristics has been designed to ensure that usability concerns are taken care of in a design.

(Refer Slide Time: 37:13)



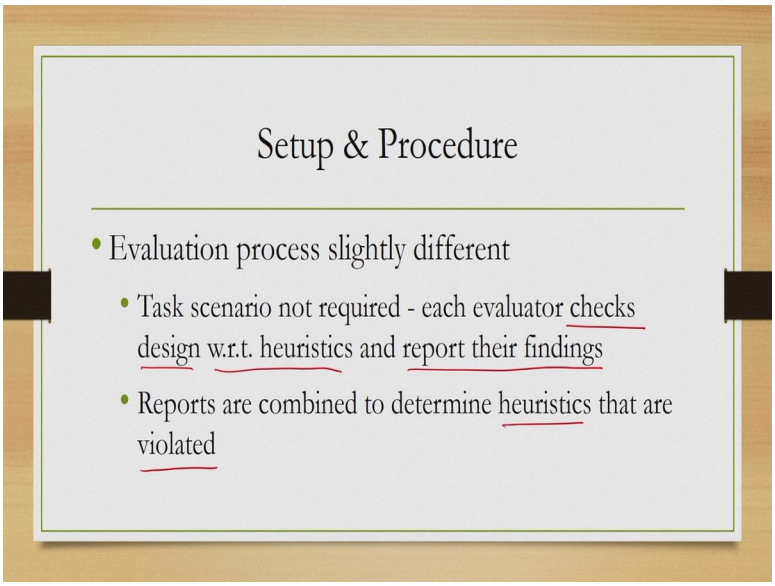
So, in order to perform heuristic evaluation what we do? Again we require a low fidelity prototype, but in this case we do not require vertical prototypes; horizontal prototype is fine. This is in contrast to the walkthrough method where we required a vertical prototype.

(Refer Slide Time: 37:37)



The second requirement is same that is we required team of evaluators having at least 3 to 5 members. And this team can have designers can have other skill designers or may also include one or two users.

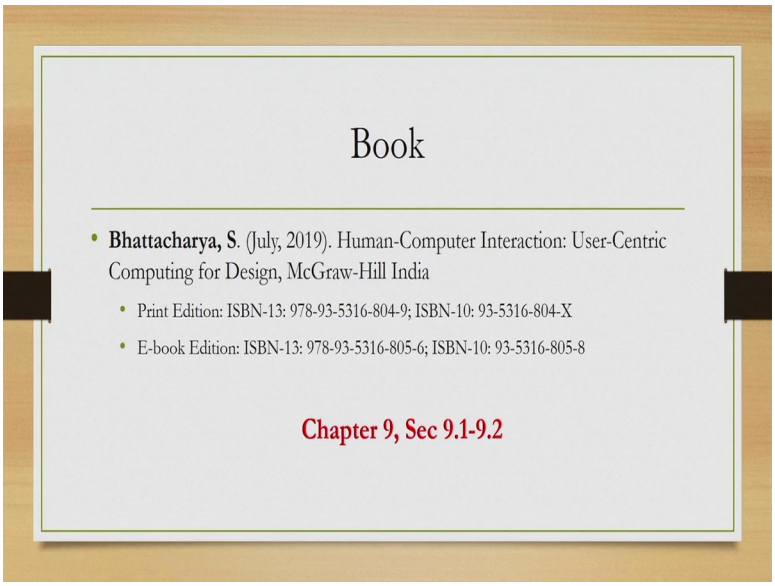
(Refer Slide Time: 38:03)



The evaluation process; however, is slightly different. Earlier we ask the users to perform the tasks specified in the scenario here we do not do that because we do not have any task scenario. So, what we tell each evaluator is to check the design with respect to the heuristics and report their findings. Now, these reports are combined to determine which heuristics are getting violated in the design.

So, in summary what we do is we start with a set of heuristics or checklist. We have a team of evaluators 3 to 5 member steam is sufficient and low fidelity horizontal prototype of the system. Now each evaluated is given the prototype and the heuristics. They are asked to produce a report based on the heuristics and those reports are combined at the end to find out the heuristics that are getting violated due to the design. So, these are some of the evaluation techniques that we have discussed today. Few more techniques will be discussed in the next lecture.

(Refer Slide Time: 39:33)



Whatever we have discussed today can be found in this book. You are advised to refer to chapter 9, Section 9.1 to 9.2 to get more details on these techniques that we have covered.

Thank you and goodbye.