

Design and Implementation of Human – Computer Interfaces
Prof. Dr. Samit Bhattacharya
Department of Computer Science & Engineering
Indian Institute of Technology, Guwahati

Lecture: 16
Prototype Evaluation I

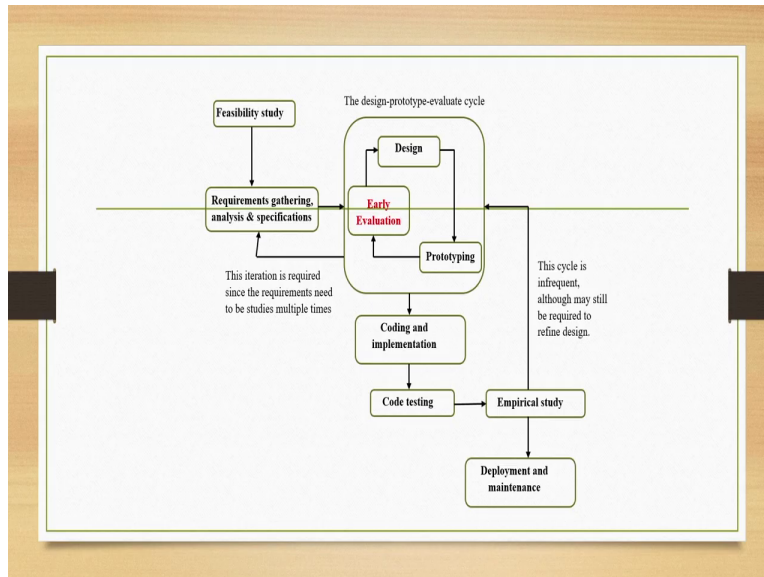
(Refer Slide Time: 00:54)



Hello and welcome to NPTEL MOOC's course on design and implementation of human computer interfaces lecture number 15. So, in the earlier lecture we talked about how to create prototypes. Prototypes are useful for testing design ideas as we have discussed elaborately during the lecture. Now when we are talking about prototype so, it is basically meant to test our ideas but testing means we need to evaluate it.

At the same time we have to ensure that whatever methodology we apply for evaluation should be first we should be able to get the results quickly otherwise that will defeat the purpose why so?

(Refer Slide Time: 01:41)



Let us try to understand this with respect to the software development life cycle for interactive systems. So, when we talked about developing a software we also learned about the interactive system development life cycle. If you may recollect we have talked about requirement gathering and after that we said that there is One Design prototype evaluate cycle which consists of three stages design stage, prototyping stage and evaluation stage.

Now here we are specifically focusing on the cycle that means once we get the design done maybe either based on our intuition or may be using the design guidelines as a starting point or based on our experience and so on. We express it in the form of prototype and then get it evaluated. So, that we get to know whether the design suffers from any problem. So, when you say evaluate it what we are trying to understand?

We are trying to understand whether the design suffers from usability problem because for interactive systems usability is our main concern. So, we evaluate the prototypes to learn about issues that may be there with the design. Then based on identification of those issues we go for refinement of the design again then again we go for prototyping and again we evaluate and this cycle goes on till we arrive at a design which no longer has significant amount of issues with respect to usability.

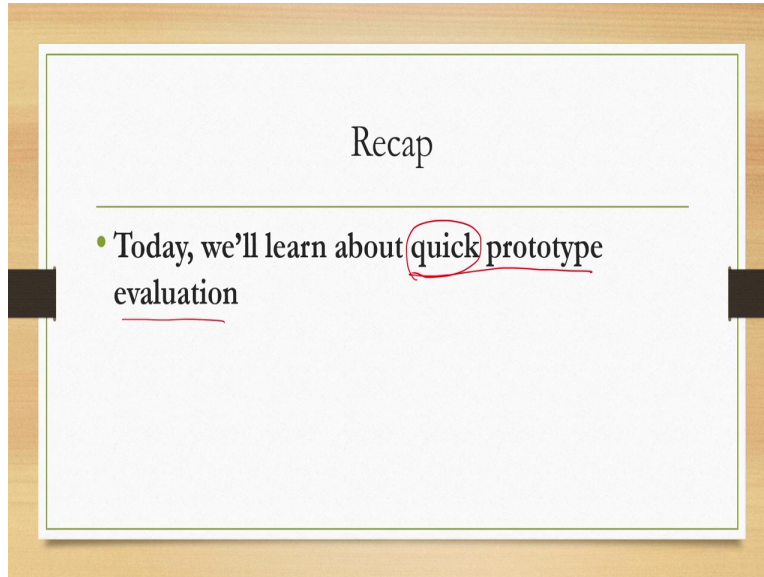
Now typically when we are trying to implement this life cycle typically the design prototype evaluate cycle takes place frequently and many times the iteration takes place. In other words we have to be ready to execute this cycle many times. Now if our evaluation takes time then when we need to do it many times repeatedly then overall cycle time till we arrive at a stable design increases which is detrimental to the overall turnaround time of the project.

So, our aim should be to have an evaluation method which gives us quick result. Now typically this evaluation is done at the early stages of design. So, specific and specialized evaluation methods are there which help us get quick results of evaluation of the prototype just for the sake of completion. So, once the stable design is arrived at then we go for system design from system design you go to coding and implementation stage this is followed by code testing.

So, after code testing what we get is an executable system that system we test further with end users to know about the usability of the final product. So, that is done in this stage empirical study. Once we are sure that no further usability issues are there or the usability issues are very few and not significant then we go for deployment and maintenance this is in brief the interactive system development life cycle.

Out of which we have already talked about requirement gathering design prototyping and in this lecture we are going to talk about how to quickly evaluate our prototypes.

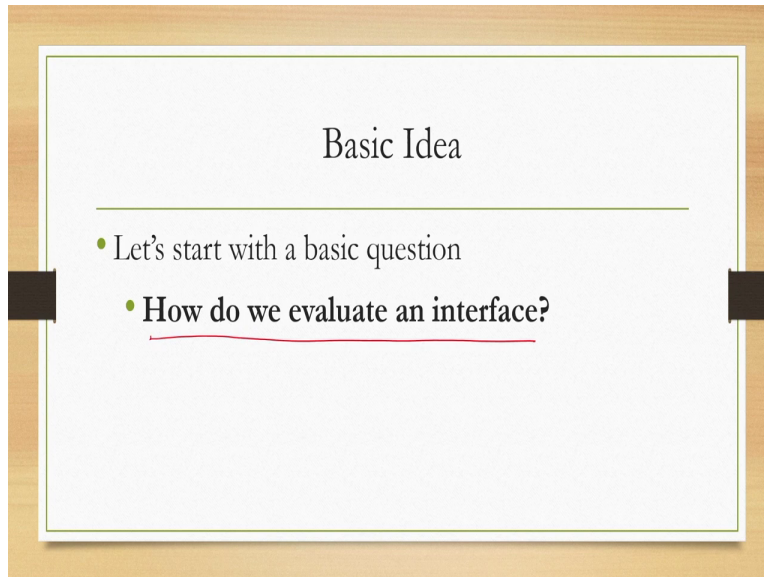
(Refer Slide Time: 05:37)



So, essentially the topic relates to quick prototype evaluation. So, the word quick is very significant here and we will see how or what methodology can ensure that the evaluation is done quickly. Now the quick is a relative term. So, just to put it into perspective to evaluate usability ideally we should go through rigorous user testing and that is the stage that we just mentioned empirical study stage where we do this rigorous testing.

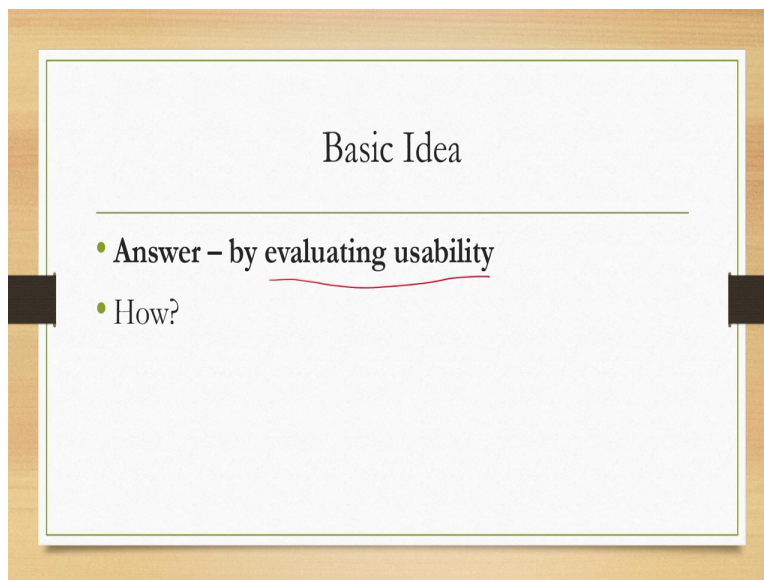
Now that rigorous testing takes lot of time and typically that testing is carried out only once or twice in the life cycle. Whereas the quick evaluation that we do in the early stages in the design prototype evaluate cycle can be performed many times and it produces results without much investment in manpower, effort and cost. So, with respect to the empirical study we are saying that our evaluation method should be quick.

(Refer Slide Time: 06:51)



So, let us now try to understand the evaluation methods that we can use to get quick results after testing the prototypes. So, now suppose I ask you a question a basic question how do we evaluate an interface what would be the answer to this question? This is a very basic question which we should be able to answer before we try to understand what we mean by quick evaluation.

(Refer Slide Time: 07:21)



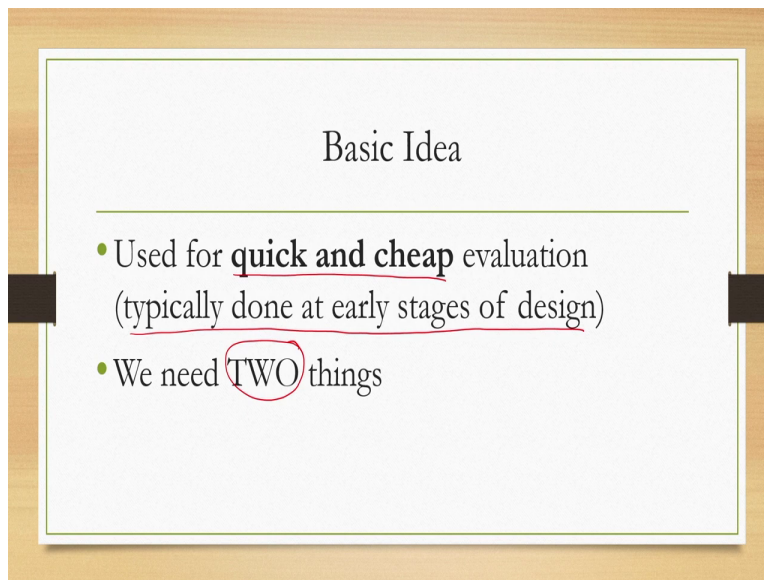
So, the answer is simple by evaluating usability in our earlier lectures we talked about the idea of usability there is an ISO standard definition of usability which we have discussed in details. Now that idea of usability needs to be evaluated or we need to check in our proposed design which is

expressed in the form of a prototype whether the usability idea is present or there are some deficiencies in the design. So, that usability is affected. So, how do we do that?

There are several methods available the most fundamental method of course is empirical study or empirical research method where we employ end users a group of end users collect usage data from them in a controlled setting analyze that data to conclude about the overall usability of a product. But that as I already mentioned earlier can involve cost and time overrun in the overall product life cycle if we do it repeatedly.

Because any empirical study involves lots of time lots of effort huge manpower may be resources so, it is a costly affair. Definitely empirical studies cannot give us quick result. So, we need to go for some alternatives. So, there are alternative ways to get quick results.

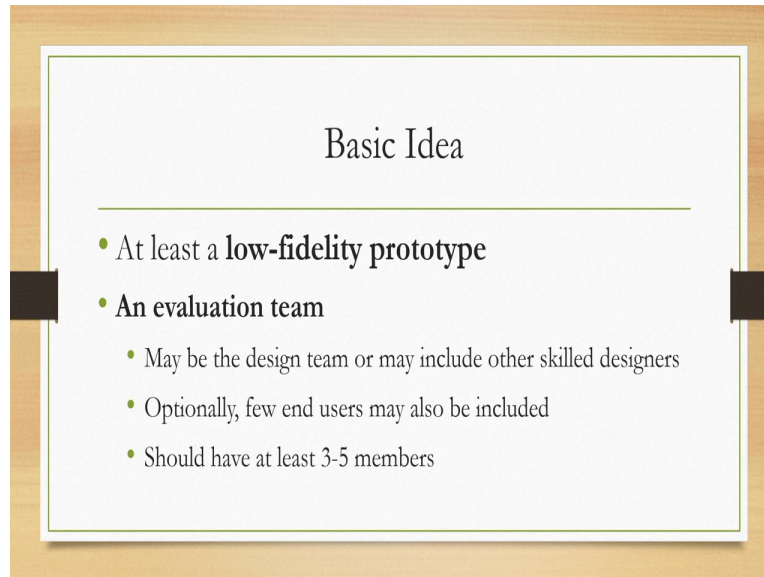
(Refer Slide Time: 09:12)



One such alternative method is called expert evaluation method. Now expert evaluation is used for quick and cheap evaluation of prototype. Evaluation of what? Evaluation for of prototype and evaluation for what evaluation for usability? So, expert evolution method is a quick and cheap method of evaluation of prototypes for usability and such method are typically applied at an early stage of design rather than after the product is developed.

In order to perform expert evaluation we need two things. So, there are two crucial components of this study method or the evaluation method or the testing method which we need to have before we go for expert evaluation.

(Refer Slide Time: 10:06)



One of those things is up prototype. Now we require at least a low fidelity prototype. Recollect our discussion on prototyping. So, we talked about two types of prototypes one is low fidelity prototype and the other one is high fidelity in between there is a medium Fidelity which is nothing but implementation of low fidelity ideas with the help of a computer. For expert evaluation to take place we require a prototype at least a low fidelity prototype should be available to let us test the system or let us test the design idea.

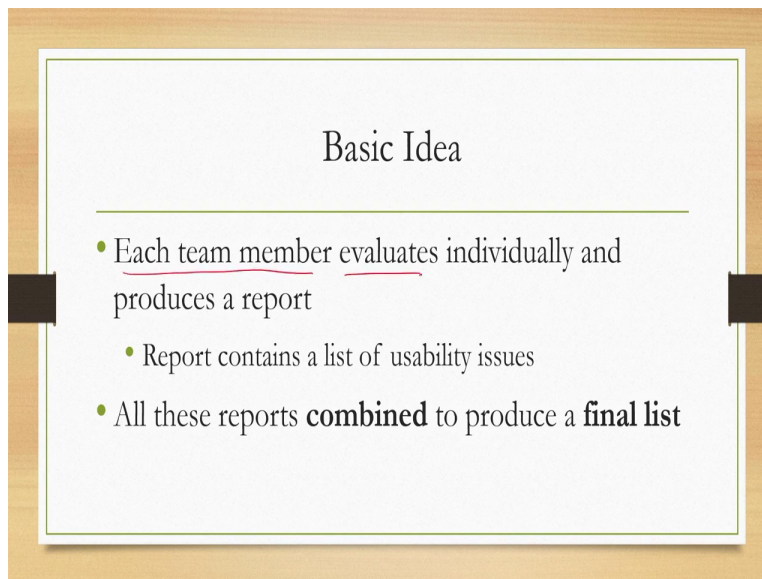
So, that is the first requirement along with that we also require an evaluation team. So, generally it is done by a team rather than a single person. So, we require an evaluation team. Now who can be team members that is a crucial question we should be aware of. Now the team may consist of the designers. So, design team can act as testing team or it may include other skilled designers. So, there may be other persons who are designers themselves.

Designer of the interface skill designer but they need not necessarily be part of the design of the current interface which we are testing. In addition and optionally few end users may also be included in the team if available. So, if we find that some end users are available and they can be

included we can include them in the testing team but that is optional not mandatory. The team should have at least three to five members that is also another requirement.

So, there are two things one is we need to have at least a low fidelity prototype and we need to have an evaluation team. In the team there can be designers there can be other designers there can be some end users and at least three to five members.

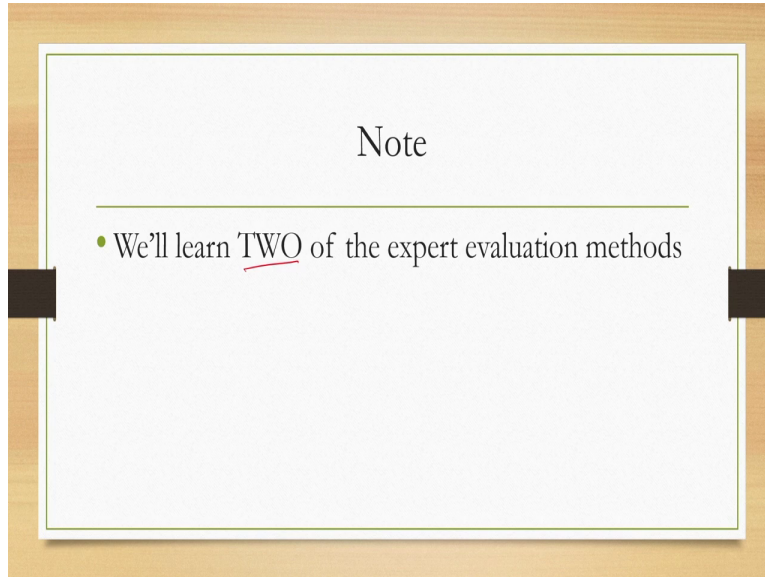
(Refer Slide Time: 12:34)



Now how the testing takes place each team member evaluates individually and produces a report it is not a group activity. So, there is a team. Now the Prototype is given to each team member with certain things along with the Prototype which we will discuss in subsequent part of this lecture. And each team member individually evaluates the Prototype and produces a report on various usability aspects of the system or the interface.

So, the report contains a list of usability issues that the evaluator found out. So, each member of the team produces such a report and all those reports are collected and combined to produce a final list of usability issues. Once that list is available that is the outcome of the evaluation stage and based on that final list we go for refinement of the design subsequently prototype the refined design again we evaluate and so on in this way the cycle takes place.

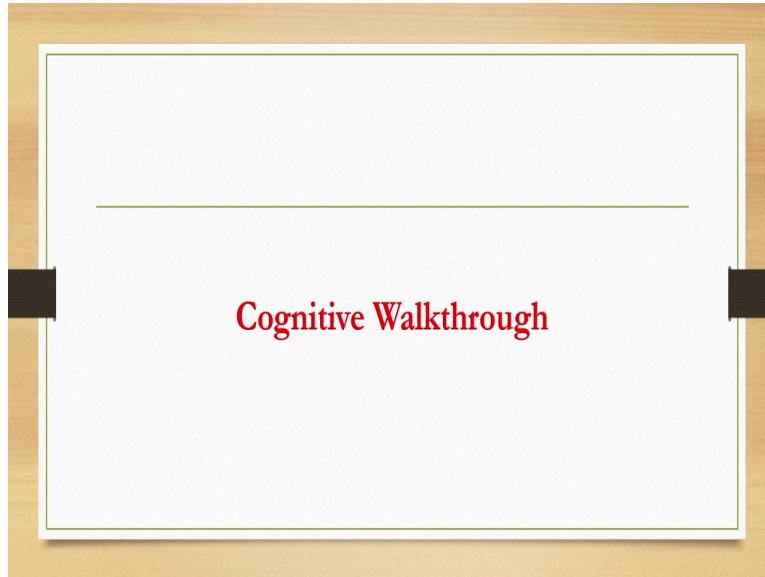
(Refer Slide Time: 13:52)



So, this broad idea is called expert evaluation. Now why it is called expert evaluation? Because we are relying on quote unquote experts to get the evaluation done rather than end users. So, usability as per definition if you recollect the definition as per definition usability is related to the end users but in these evaluation end users presence is optional as we just mentioned. So, in the team optionally we can include them we may not include them.

We are relying on the testing by skilled designers who are supposed to be experts in understanding or in the knowledge of user behaviour. We are assuming that those skill designers have sufficient knowledge of user behavior to understand from user's point of view the usability issues and they can produce relevant reports that is why this is called expert evaluation method. There are several ways in which expert evaluation can take place here in this lecture we are going to talk about two such methods.

(Refer Slide Time: 15:11)



Cognitive Walkthrough

These two are cognitive walkthrough method and heuristic evaluation method let us start with cognitive walkthrough method. What it is? How it is done and what it produces let us try to understand that.

(Refer Slide Time: 15:29)

- 
- ### Basic Idea
- An **Usability Inspection Method**
 - Requirements
 - At-least a low-fidelity prototype with support for several tasks (vertical prototype)
 - 3-5 member team (as mentioned earlier)

Broadly cognitive walkthrough method which is a type of expert evaluation method can be considered to be an usability inspection method. So, essentially this method refers to inspection of a system for identifying usability issues. What are the requirements for this method? Like the broad requirements we just discussed for any expert evaluation method here also the same

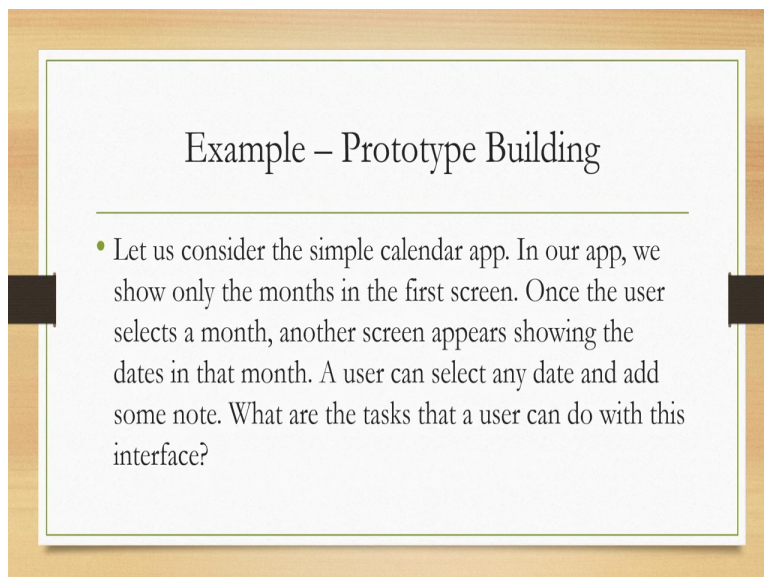
requirements are there. At least a low fidelity prototype with an additional requirement is that that the prototype should support several interface level tasks.

In other words we not only require a prototype but the Prototype should be a vertical prototype. Recollect our discussion on prototype where we said vertical prototypes are those that prototypes interface state at any instant of interaction as well as the interaction itself. So, essentially the prototype refers to a set of interfaces each interface refers to a particular stage of interaction and it also contains the mechanism to change interfaces in other words the mechanism to perform the interaction.

So, when we are talking of cognitive walkthrough we need to have such prototypes with us and importantly not only a vertical prototype but many vertical prototypes for the system that means we should have more than one tasks prototyped and available to us before we go for cognitive walkthrough method. At least three to five member team should be their evaluation team as we have already discussed along with the Prototype.

Now here in this team again just to recollect we can include end users if available we can include other skill designers or only the member of the design team.

(Refer Slide Time: 17:53)



The slide features a light beige background with a white rectangular area in the center. The title 'Example – Prototype Building' is centered at the top of the white area. Below the title is a horizontal line. A single bullet point follows, describing a simple calendar app. The text of the bullet point is: 'Let us consider the simple calendar app. In our app, we show only the months in the first screen. Once the user selects a month, another screen appears showing the dates in that month. A user can select any date and add some note. What are the tasks that a user can do with this interface?'

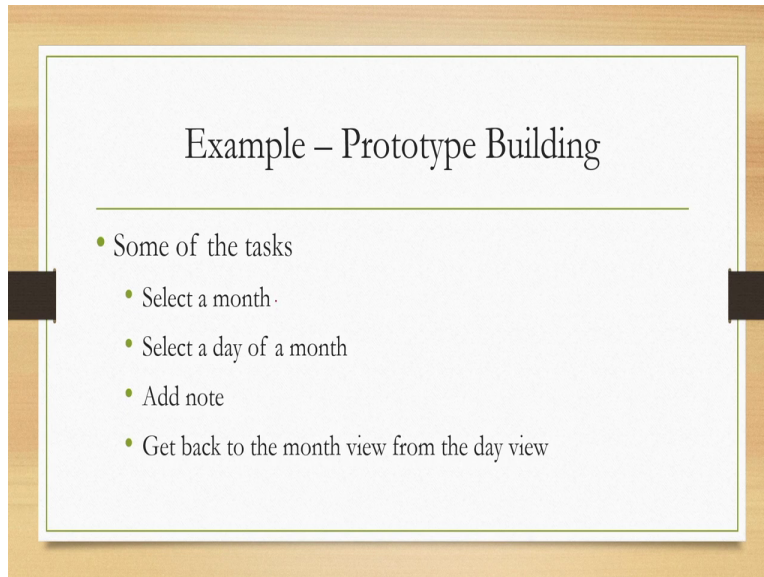
So, how it works those are the requirements based on that requirements how we can perform a cognitive walkthrough. Let us try to understand this in terms of one example. So, earlier we talked about a calendar application simple calendar application. So, let us now try to understand cognitive walkthrough with respect to an interface that we have designed for the calendar application.

Suppose we have a design for the calendar app in our app we show only the months in the first screen once the user selects a month by some means either by Mouse click or tap another screen appears showing the dates or the days in that in that month a user can select any day and add some note. So, that is the interface that we propose for the simple calendar app. It shows months at a time once a month is selected it shows days in that month.

And if we choose a date day then in that day we can keep some notables. So, that interface is available. Now based on this idea of the interface let us try to understand whether this interface suffers from any usability issue assume that we have prototyped it and we want to test it and we are applying cognitive walkthrough method for identifying usability issues with this interface. For that as we just mentioned cognitive work through requires tasks.

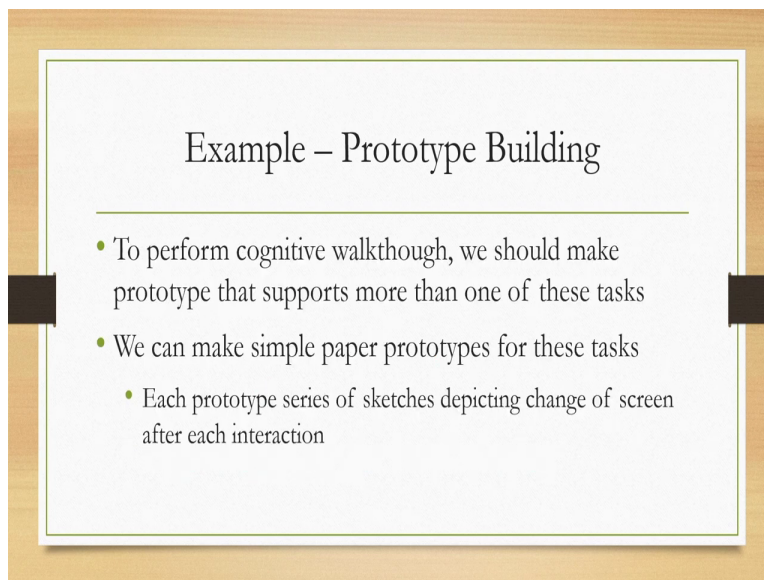
So, we assume prototype is available. So, what can be the tasks that a user can do with this interface. There can be many tasks but in prototype we of course cannot implement all the tasks.

(Refer Slide Time: 20:02)



So, few will select but let us first have a look at different types of tasks that can be done with this interface user can select a month select a day of a month add note to a particular day of a month can get back to the month view from the day view and vice versa. So, these are some of the tasks which are likely to be frequently performed by an end user of the Calendar app.

(Refer Slide Time: 20:26)

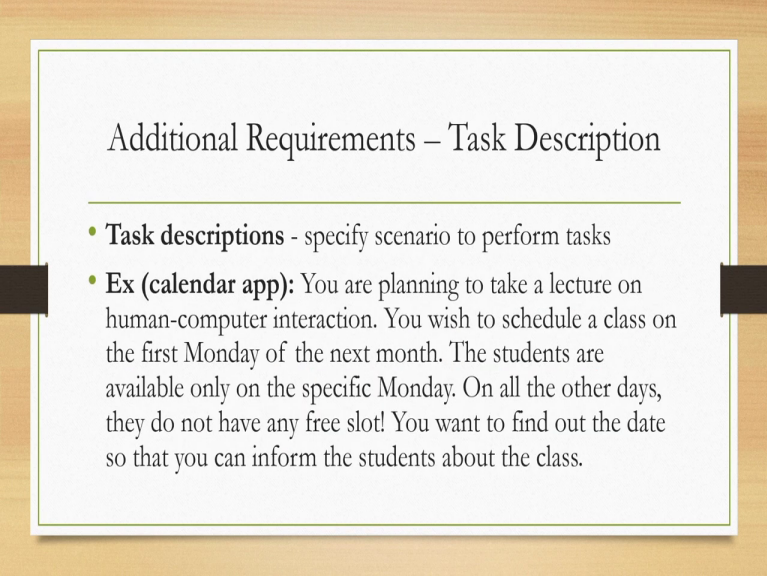


Now in order to perform cognitive walkthrough we need to have some prototypes ready in the Prototype we need to replicate some task scenario and the prototype should support more than one of these tasks. So, we can create prototypes for multiple tasks that are supported by the

interface. We can make simple paper prototypes for these tasks like we have discussed earlier we can create storyboards to create a vertical prototype.

So, each prototype is a series of sketches depicting change of screen after each interaction this is nothing but the idea of storyboarding where each intermediate sketch is called a keyframe.

(Refer Slide Time: 21:28)



Additional Requirements – Task Description

- **Task descriptions** - specify scenario to perform tasks
- **Ex (calendar app):** You are planning to take a lecture on human-computer interaction. You wish to schedule a class on the first Monday of the next month. The students are available only on the specific Monday. On all the other days, they do not have any free slot! You want to find out the date so that you can inform the students about the class.

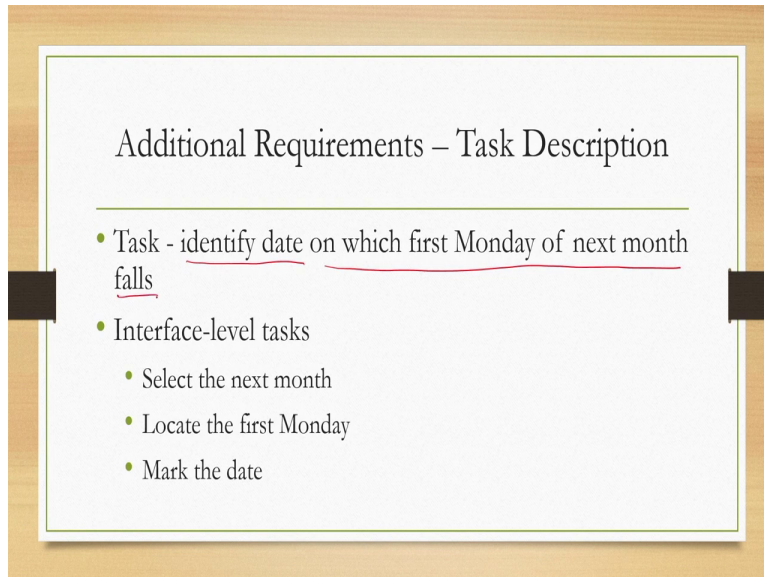
Now we first need to before we go for creating a prototype we first need to explicitly describe a task or rather specify the scenario to perform tasks. Now let us take one example suppose you are an user of the calendar app and if you recollect the Calendar app when you talked about the Calendar app we mentioned that it is to be used by either teacher or students in an academic environment. So, suppose you are a teacher or instructor.

So, you are planning to take a lecture on that subject human computer interaction you want to schedule a class on the first Monday of the next month that is what you want because you know that the students are available only on the specific Monday. So, every month first Monday within a semester the students are available other Mondays they are not available for some reason and on all the other days of the week they have some other works.

So, they do not have any free slot for you to take the lecture. Now you want to find out the date so, that you can inform the students about the class. So, you are given the Calendar app your task

is to find out the date on which the first Monday of the next month falls and once you are able to identify the date you can inform the students and additionally you can keep some note on that date in your Calendar app.

(Refer Slide Time: 23:24)



The slide is titled "Additional Requirements – Task Description" and is set against a light green background with a white central box. It contains a bulleted list of tasks:

- Task - identify date on which first Monday of next month falls
- Interface-level tasks
 - Select the next month
 - Locate the first Monday
 - Mark the date

So, what is your task as a user of the app your task is to identify or determine the date on which first Monday of the next month Falls. In order to perform this task we need to perform or rather you need to perform some interface level activities or tasks what are those tasks one is to select the next month next is locate the first Monday in the month view and the third is Mark the date by some means along with keeping a note on that particular date regarding the schedule of the class.

So, these are some activities that that you need to perform with the interface to know about the date. Now let us see if with the proposed design you perform these activities whether there will be any usability issues. We want to test for usability of the proposed design with respect to these tasks using cognitive walkthrough method. So, then in the method what happens.

(Refer Slide Time: 24:37)

Procedure

- The scenario is given to the evaluators
- They are asked to find out the date (by performing the interface-level tasks with the prototype)

This scenario is given to the evaluators that mean each member of the evaluation team this scenario is given then they are asked to find out the debt by performing the interface level tasks with the prototype. So, prototype is created to carry out these tasks that means you have a vertical prototype where screen changes take place by some means of interaction which you specify indicating the completion of the tasks.

Now, this prototype you give to the evaluators or individual evaluators in the team and ask them to carry out the tasks in particular carry out the interface level tasks to achieve the overall objective of identifying the debt.

(Refer Slide Time: 25:36)

Additional Requirements – Questions

- We also need to frame a set of questions (usability issues) beforehand
- Evaluators report on these issues while they perform the tasks

Now after giving the tasks to the evaluator so, you have identified the task scenario created prototype identified the evaluation team and you have given the tasks or whatever prototypes you have created to each member of the evaluation team to generate usability reports. After that you also need to do one more step we also need to frame not the word frame you also need to frame a set of questions which are related to usability issues beforehand.

This is essentially a way to guide the evaluators to identify usability issues. Evaluators are expected to report on these issues while they perform the tasks. So, as a designer of the test you are expected to come up with a set of questions each question pertains to some usability aspect of the interface and your expected to create a list of such questions this list is provided to each of the evaluators along with the prototype.

So, the evaluators are asked to perform the tasks and at the same time answer those questions which will bring out the usability issues as perceived by the evaluator. It may be noted here that each evaluator need not report identical findings. Each of them are free to report as per their understanding what are the usability issues that is why at the end we need to compile them together see whether there are duplicate findings or there are unique findings and we have to identify the unique findings and create the final list.

(Refer Slide Time: 27:34)

Additional Requirements – Questions

- Example questions (calendar app)
 - Are you able to locate the month you are looking for easily?
 - Is the interaction required to change from the month view to the day view apparent?
 - Did you find it difficult to locate the first Monday?
 - Was the date clearly visible along with the day?
 - Did you try to go back to the month view? Was the mechanism to go back clearly visible?

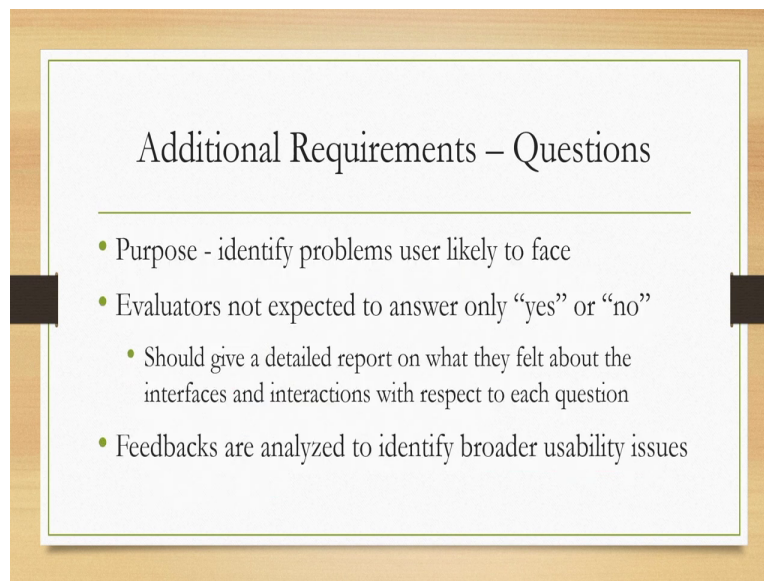
So, when we say questions what we mean by that let us try to understand with respect to the same example. So, in this case for this particular interface what can be the question or what can be a set of questions that pertains to usability concerns for the interface let us see few of those questions. One is you able to locate the month you are looking for easily this can be one question. Next is the interaction required to change from the month view to the day view apparent.

That means is it easily understandable how to perform the interaction to change from month view to day view did you find it difficult to locate the first Monday that can be yet another question. Was the date clearly visible along with the day. So, yet another question did you try to go back to the month view was the mechanism to go back clearly visible in fact these are two separate questions for brevity we have shown it together here.

So, these can be some of the questions, so, these questions if you notice carefully these questions pertain to some aspects of the interface which affects the usability of the product. So, these types of questions you are expected to frame and provide to the evaluators. Now the evaluators will perform the tasks with the interface and while performing they will try to answer these questions. As I have just mentioned earlier it is not necessary that every evaluator reports the same thing.

So, for example for this particular task with the specific interface one evaluator may find that the second question is the interaction required to change from the month view to the debut apparent one evaluator may find it not apparent or not very clear whereas two other evaluators may find them to be apparent or easy to locate or easy to perform. So, there can be variations in the reports produced by the evaluators.

(Refer Slide Time: 30:09)



There are few more things you should be aware of while trying to perform cognitive work through. So, the broad purpose of this method is to identify problems users are likely to face. So, when an evaluator is carrying out the tasks he or she has to assume that he or she is actually representing a user rather than a skill designer or an expert. And as a user he or she needs to answer the questions rather than as a skill designer.

So, it is very important that to get appropriate or reliable outcome after the test you choose your design team your evaluation team very carefully. Second important aspect that should be noted is that evaluators are not expected to answer only in yes or no. So, if we see the questions are you able to see this are you able to see that that type of questions are framed. Now evaluators can simply answer yes or no but that is not the purpose.

Along with yes or no they are supposed to give a detailed report on what they felt about the interfaces and interactions with respect to each question. So, only yes or no is not very

informative way of testing. Along with that evaluators are expected to give detailed explanation why they find it apparent or not apparent why they find something easily visible because of what it may be colour, it may be size, it may be placement.

Or why they do not find it easily visible whether it is because of contrast whether it is because of small size whether it is because of cluttering. So, that type of reasoning should be provided. Just to give you an example suppose one button is there to go back to month view from day view. Now one question is whether that button is easily recognizable. An evaluator may say yes or evaluator may say no but that is not what we want whenever somebody says yes somebody means the evaluator says yes why the evaluated thinks it is easily recognizable. Is it because the size is big is it because the colour is such that the contrast with the background is good. Is it because the placement is such that it is easily visible?

The possible reason with respect to the questions must be provided by the evaluator according to his or her understanding. Similarly if the evaluator says no the reason has to be mentioned why it is not clearly visible is it because of the background foreground contrast is it because of the size is it because of the placement and so on and so forth. So, answer with explanation is expected from each evaluator.

Now once all the evaluators submitted their report those are compiled by the lead evaluator or team lead and analyzed to identify broader unique usability issues which will form the final list of usability issues based on which the design may be refined or based on which we may decide not to refine because if the list is very small that means and points to insignificant issues that indicates that no further changes in the design is required. So, we may stop the cycle otherwise we may continue with the cycle.

(Refer Slide Time: 34:34)

Additional Requirements – Questions

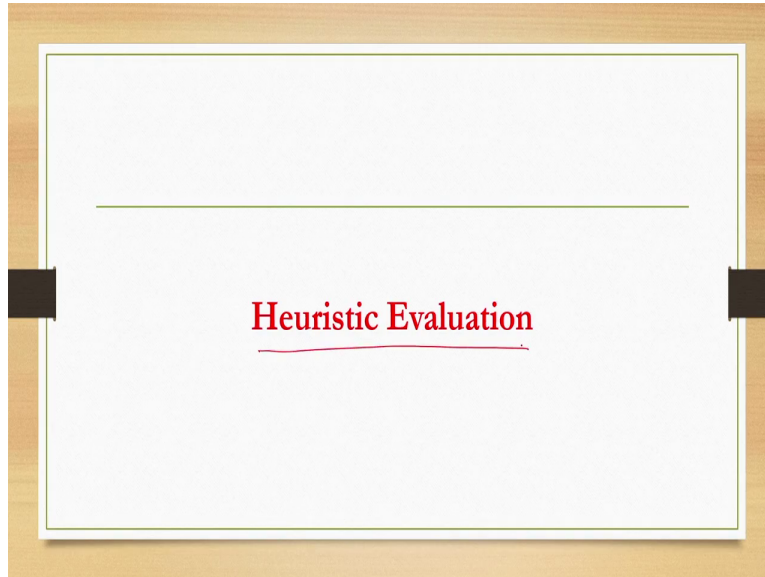
- Is the effect of the action same as that of the goal of the user at that point?
- Will a user see the control for a particular action?
- Will a user see that the control produces the desired effect?
- Will a user select a different control instead?
- Will a user understand the feedback provided by 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?

Some more examples of the questions for both for getting feedback as well as analyzing can be understood in terms of this example scenario. So, one such question can be is the effect of the action same as that of the goal of the user at that point this is at a broader level. Will a user see the control for a particular action. So, these are broad generic type of questions not referring to any specific system.

These questions can be adapted to specific scenarios as we have just seen, will a user see that the control produces the desired effect, will 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. So, whether error is taken care off. How a user who is familiar with other systems that perform similar tasks is going to react.

So, these questions are slightly different from the questions that we have seen earlier these are somewhat broader generic type of questions and based on this type of generic questions we can frame individual list of questionnaire for each task scenario as we have done in the example. So, that was about cognitive walkthrough which is one of the expert evaluation method. Let us try to understand another expert evaluation method that is called heuristic evaluation.

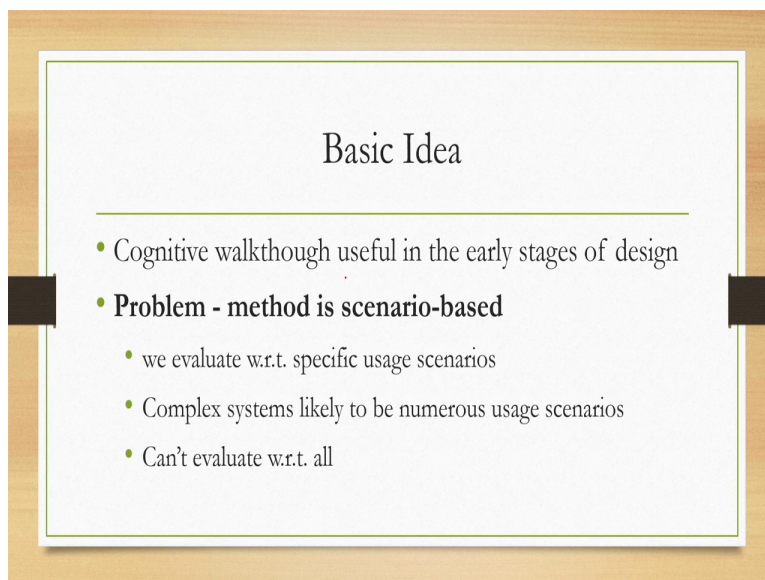
(Refer Slide Time: 36:23)



Heuristic Evaluation

So, we will try to understand what it is how it is done and what we get after the evaluation.

(Refer Slide Time: 36:33)

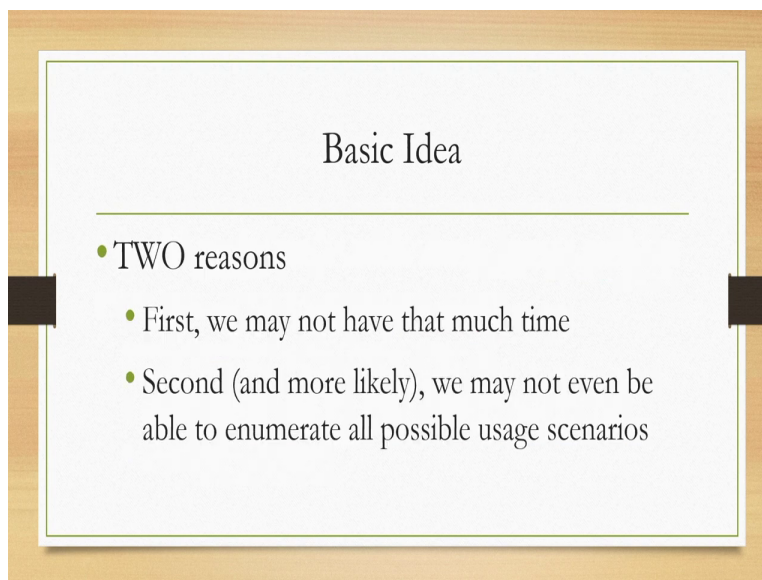
- 
- ### Basic Idea
- Cognitive walkthrough useful in the early stages of design
 - **Problem - method is scenario-based**
 - we evaluate w.r.t. specific usage scenarios
 - Complex systems likely to be numerous usage scenarios
 - Can't evaluate w.r.t. all

Now cognitive walkthrough which is one type of expert evaluation is useful definitely in the early stages of design. So, it can be used definitely for quick evaluation of prototypes because it does not require end users and it can be done with the design team skill designers to get the feedback about prototypes. But there is one problem the method is scenario based that means we evaluate with respect to specific usage scenarios.

Now for simple systems that is fine. For complex systems there are likely to be numerous usage scenarios and we definitely cannot evaluate with respect to all. So, if we are dealing with simple interfaces then usage scenarios are limited one or two and we can create prototypes for those usage scenarios as we have seen in the example and those prototypes can be used vertical prototypes which can be used to perform cognitive walkthrough.

But for more complex systems usage scenarios are likely to be numerous and in that case we cannot perform cognitive work through to test prototypes for each possible usage scenario. Now if we miss out some scenarios then we do not know whether there are usability issues related to those scenarios that knowledge will never gain.

(Refer Slide Time: 38:16)

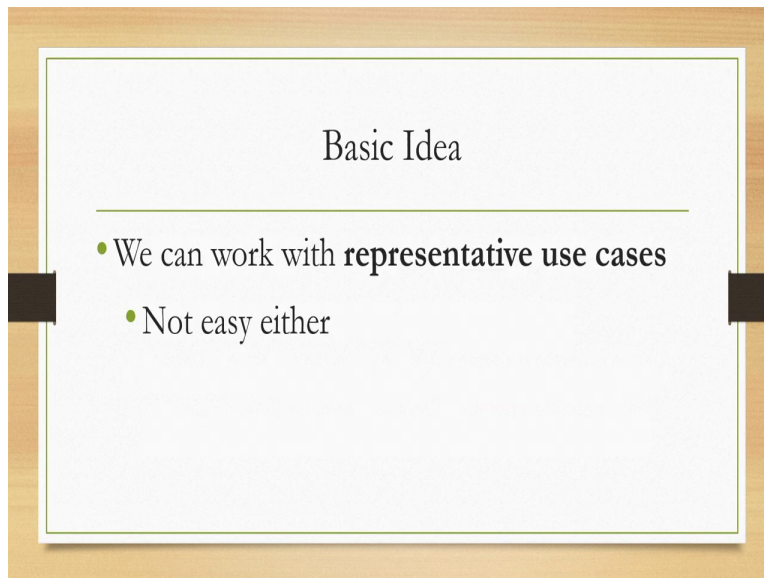


Now why we are unable to test for all usage scenarios in complex systems? So, there are primarily two reasons one is we may not have that much time. So, if there are large number of scenarios then even if we apply cognitive work through it will take long time to test each and every scenario and that much time will defeat the whole purpose of quick evaluation. Secondly which is more likely case we may not even be able to enumerate all possible you said scenarios.

So, in complex systems it is generally very difficult to identify all possible usage scenarios in advance. So, we will not be able to enable it and then if we are unable to enumerate we will not

be able to prototype also and even if we are able to enumerate and prototype time may not be there to perform cognitive walkthrough for all the possible usage scenarios.

(Refer Slide Time: 39:13)



There can be one way to address this concern that is instead of trying to enumerate and prototype all usage scenarios we can work with representative use cases. What is a representative use case. So, if there are large number of use cases it is not necessary that in real life situation all the use cases are frequently performed. Instead most likely very small subset of the use cases are frequently performed and our objective is to identify that small set of use cases which are frequently performed.

That small set is the representative use case or use cases that are likely to represent real world usage of the product. But this is easier said than done it is very difficult to identify in advance when the product is not ready and used very difficult to identify what constitutes the representative use cases. So, if we are able to identify then since the number is small we can prototype and go for cognitive evaluation but it is not easy to identify representative use cases.

So, we have to go for all possible use cases which is time consuming and which may not be possible either.

(Refer Slide Time: 40:40)

Basic Idea

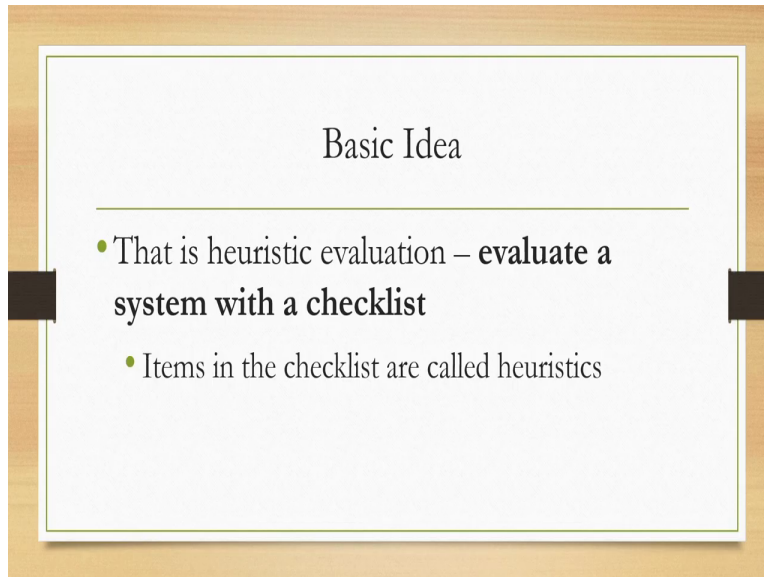
- We may go for comprehensive evaluation of the whole system
 - We no longer need to create scenarios and ask evaluators to perform tasks
 - Instead, we can ask evaluators to **tick** on a checklist of features of the system as a whole

In order to address this concern that is there with cognitive walkthrough method we can use another approach. Now in case of cognitive walkthrough we are relying on task scenario. So, this is Task Centric approach instead what we can do is instead of trying to evaluate usability with respect to tasks in a Task Centric approach what we can try to do is, we can discard the task Centric approach we no longer require tasks to be identified and usability is identified based on the performance in the tasks.

What we can do we can go for comprehensive evaluation of the whole system without bothering about task scenarios or tasks to be performed with the system. So, we no longer need to create scenarios and ask evaluators to perform the tasks. Instead what we can do we can ask evaluators to tick or select on a checklist of features of the system as a whole. So, the idea is very simple. In cognitive walkthrough what we had is some task scenarios.

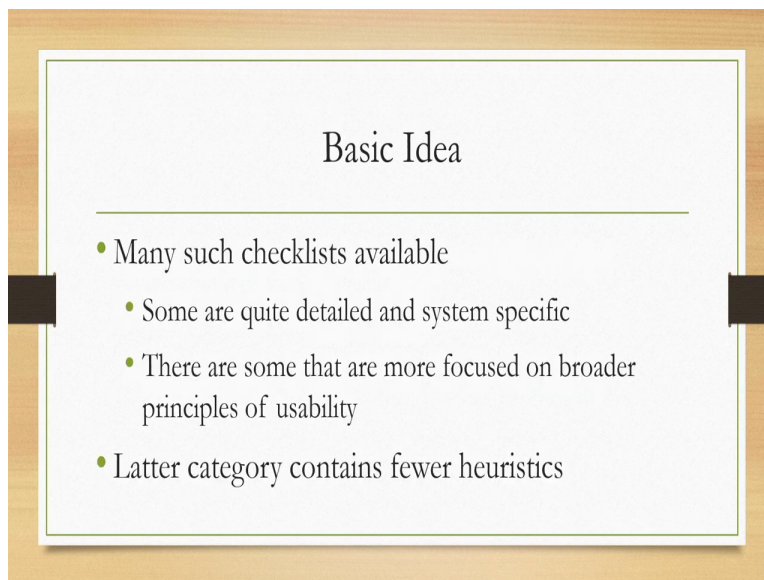
Now here we are not having those task scenarios instead we are having a checklist, checklist of system features that are identified in advance which are related to usability of the interface. So, an evaluator will be asked to take the interface prototype and tick in the checklist whichever of the features the prototype supports that is the basic idea.

(Refer Slide Time: 42:23)



Now when we follow this idea this is called heuristic evaluation. The checklist is nothing but a set of heuristics that we assume to represent usability of the interface or usability aspects of the interface. So, the items in the checklist are called heuristics and these heuristics are used to evaluate the overall system irrespective of tasks.

(Refer Slide Time: 42:53)

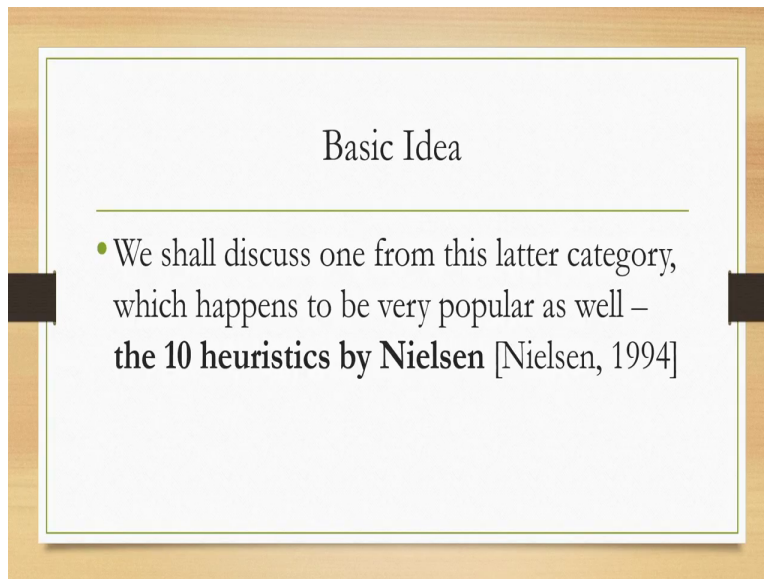


Like design guidelines which we have discussed earlier for heuristic evaluation also many such checklists are available some are quite detailed and system specific similar to design guidelines but there are some that are more focused on broader principles of usability. So, like design

guidelines we have detailed checklist which are likely to be system specific and broader checklists where the heuristics in the checklist refer to broader usability aspects of the system.

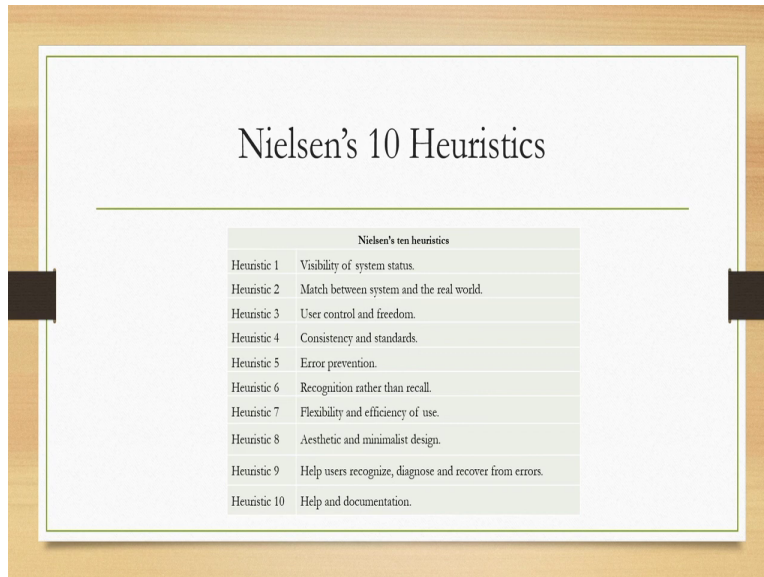
So, evidently in that case the number of heuristics will be much less compared to detailed checklists.

(Refer Slide Time: 43:38)



In order to understand the idea of heuristic evaluation let us see one such checklist or heuristics which is a broader set of heuristics. This is a very popular checklist as well the 10 heuristics by Nielsen proposed in 1994.

(Refer Slide Time: 44:01)



So, there are 10 such heuristics or items in the checklist. First heuristic says visibility of system status whether status of the system at any instant of interaction is clearly visible to the user. Second heuristic it talks about match between system and the real world. Whether the actions and elements present on the interface have matches with our day to day experience in the outside world then heuristic 3 talks about user control and freedom how the control and freedom perceived by the user.

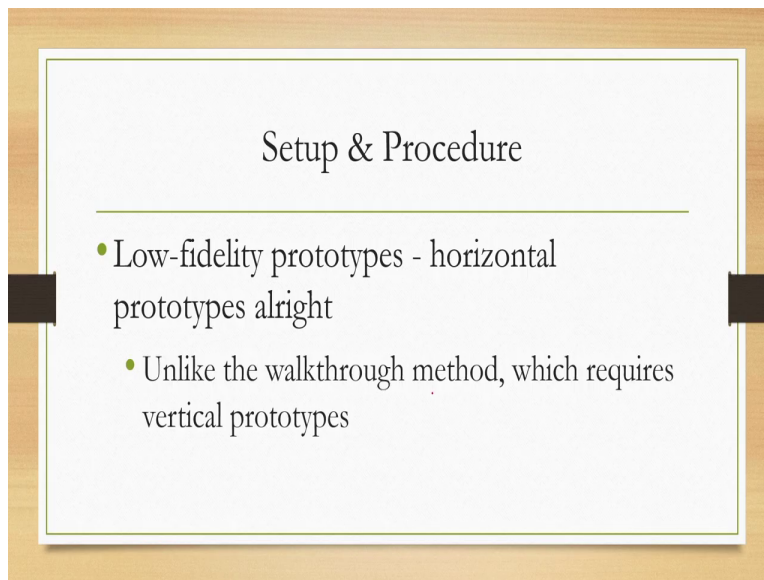
Fourth heuristic it talks about consistency and standards. So, whether the design follows consistency and follow standards that is the fourth heuristics. Fifth heuristics is regarding error prevention whether the design that we are evaluating supports or Aids error prevention. Heuristic 6 talks about recognition rather than recall that means whether it supports the user to recognize items on the screen rather than forcing the user to recall from memory before he or she can use the interface.

Heuristic 7 talks about flexibility and efficiency of use whether these are supported by the design. The eighth heuristic talks about aesthetic and minimalist design whether the design that we are evaluating is aesthetically pleasing and it contains minimum number of elements as well as minimum number of interactions to achieve tasks goals. The ninth heuristic talks about whether the system helps users recognize diagnose and recover from errors and the tenth at the final heuristic talks about whether the system supports help and documentation.

So, these heuristics as you can see lists some aspects of the design which are related to usability for example recognition rather than recall. So, see this can be related to the Golden Rule of Snyderman where it was mentioned that the 7 plus minus 2 rule was mentioned that means if the user is forced to recall sequences or meaning of items if the number of recalls or the amount of recall is large then of course that will exceed our short term memory capacity and that will be difficult for the user to perform.

So, eventually the interaction will not be usable whereas if user is able to recognize actions and items by the very design of the interface then the memory constraint is taken care of. So, a design that supports recognition rather than recall is more usable than the opposite thing. So, what the evaluator needs to do evaluator we will get this list and a prototype.

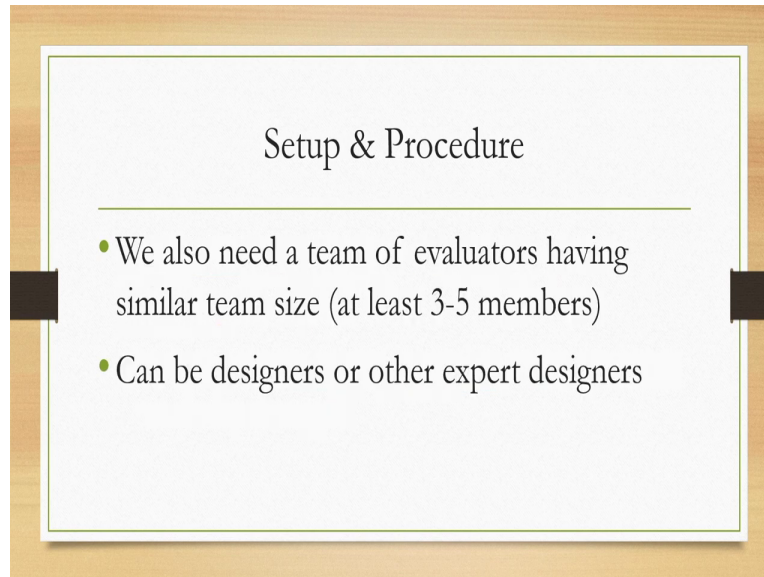
(Refer Slide Time: 47:55)



Now unlike cognitive walkthrough in case of heuristic evaluation we require low fidelity prototype which can be a horizontal prototype because here we are not relying on tasks. So, execution sequence is not important rather the elements present are more important. However vertical prototypes can also be used to see the nature of interaction as well with respect to the checklist. But even if we have a vertical prototype we do not need to specify any specific tasks scenario.

So, for vertical prototype task scenarios need not be specified anything is fine but typically heuristic evaluation is used for horizontal prototypes. This is in contrast to cognitive walkthrough method where we rely mostly on vertical prototypes.

(Refer Slide Time: 48:51)



And like the other method we also need a team of evaluators. Now the team should contain at least between three to five members that is the same as before. Now these team members can be the designers or other expert designers or even end users along with the expert designers. So, this point is very important. So, the evaluation team should not be only end users at this stage there should be few skilled designers as experts in the team so, that we can get some expert feedback along with simple Yes, No type feedback.

Now here as we said evaluation process is slightly different. So, we no longer require the task scenario each evaluator checks the design or the prototype with respect to the heuristics and report their findings. Now reporting here again is not simple Yes, No type like in cognitive walkthrough. So, even if suppose one item says that whether the design supports error prevention if the evaluator simply say yes then that will not be very revealing rather the evaluator should say why he or she thinks that the design supports error prevention.

What are the features present on the system or the design that supports error prevention that detailing is required like in case of cognitive walkthrough. And once all the evaluators submitted

their reports those reports are combined to determine heuristics that are violated this is very important. So, where the end result is identification of heuristics that are violated by the design as reported by the majority or all the evaluators.

So, it may happen that one evaluator said it is violated but majority side it is not violated then we will consider it to be not violated. So, in majority things that particular heuristic is violated we will consider it to be violated and those violated heuristics are considered while refining the interface or while refining the interface design. So, that is in a nutshell what is heuristic evaluation.

With that we have come to the end of this lecture here we learned about quick evaluation technique for evaluating design ideas which are prototyped. Just to quickly recap here we talked about expert evaluation technique. Now in expert evaluation we require a prototype and an evaluation team consisting of at least three to five members. Now the team members can be designed team members or other skilled designers.

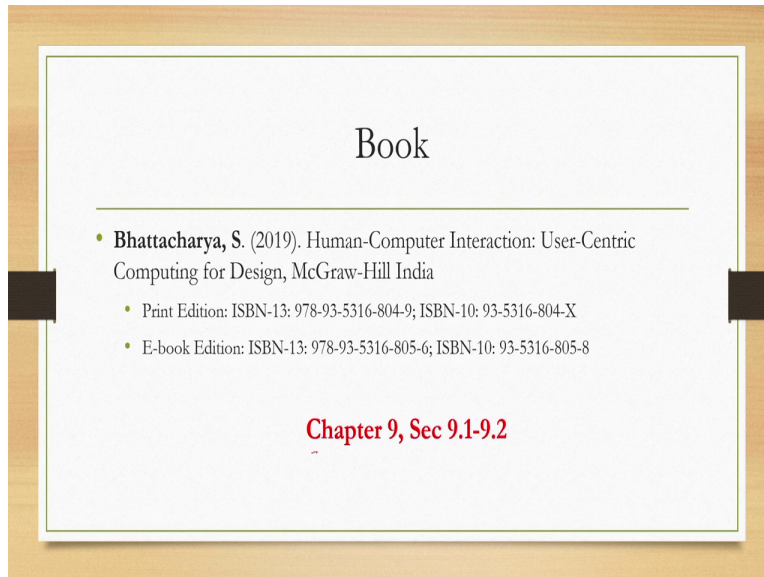
And additionally and optionally in this team you can include end users if available. We have talked about two expert evaluation techniques one is cognitive walkthrough and one is heuristic evaluation. These two techniques are different in terms of their approach to the evaluation process. In cognitive walkthrough what we do is we perform the evaluation based on tasks task scenarios accordingly vertical prototypes are created.

Prototypes along with a set of questions are provided to each evaluators. Evaluators carry out the task and answers the questions in detail and submit and at the end all the reports collected from the evaluators are combined to identify usability issues. In contrast in heuristic evaluation we do not have any questions or a task instead what we have is a checklist or set of heuristics and a prototype typically horizontal prototype.

So, each evaluator is given a horizontal prototype and the checklist they report their findings based on the checklist detailed report like cognitive walkthrough at the end those are combined to identify which of the heuristics are reported to be violated by the design and accordingly the

design is refined that is the basic idea. So, this is one way of performing quick evaluation of the prototypes. In the next lecture we will talk about another way of weak evaluation technique involving end users.

(Refer Slide Time: 54:01)



That is all for this lecture whatever material I have covered can be found in this book you requested to refer to chapter 9 section 9.1 to 9.2 both the sections you can refer to, to learn about these methods that we have covered in this lecture. Hope you have enjoyed the topic and looking forward to meet you in the next lecture, thank you and goodbye.