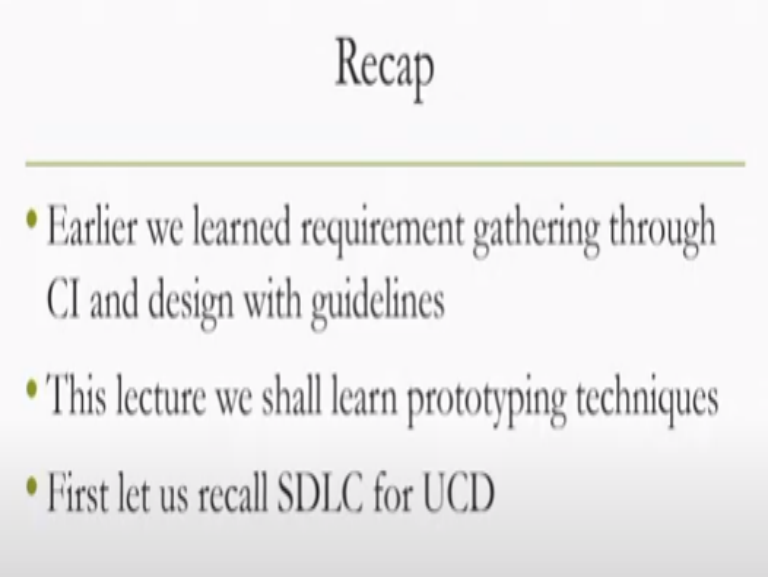
Design & Implementation of Human – Computer Interfaces
Prof. Dr. Samit Bhattacharya
Department of Computer Science & Engineering
Indian Institute of Technology, Guwahati

Module No # 04
Lecture No # 15
Prototyping

Hello and welcome to the NPTEL MOOCS course on design and implementation of human computer interfaces lecture number 14 on prototyping. In the previous lectures we have covered several topics.

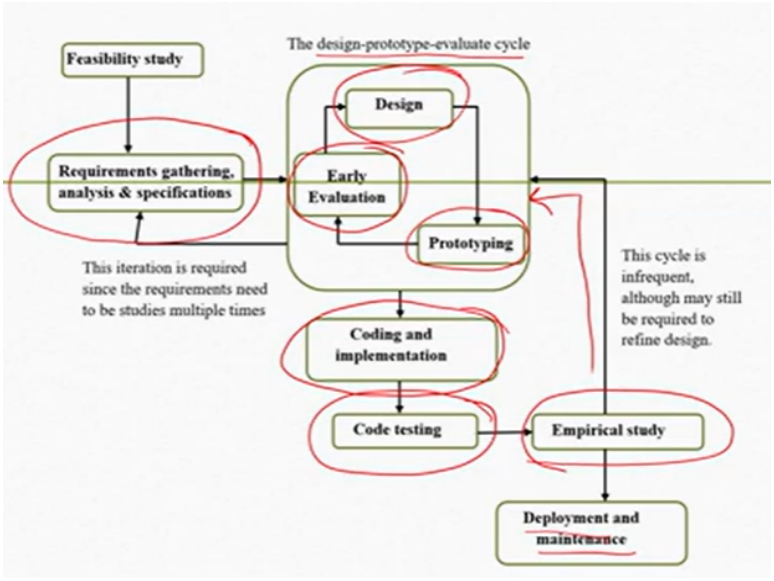
(Refer Slide Time: 01:00)



The topics are related to the different stages of the software development life cycle for interactive systems. Earlier we have seen how to gather usability requirements using contextual inquiry. We have also seen how to convert the requirements into system design into design of interfaces where we made use of guidelines and we have also covered few guidelines for the purpose. In this lecture we are going to talk about another stage in the development life cycle namely the stage in which we create a prototype of the design that we got after the design stage.

But before we proceed further as usual we will first have a quick relook at the stages of a software development lifecycle for interactive system development.

(Refer Slide Time: 02:17)



If you can recollect we have several stages as shown in this life cycle we have requirement gathering analysis and specification stage we covered this stage in details. Where we learned about different ways to capture end user requirements and we went through one particular approach that is contextual inquiry. Now here end user means the users who are going to use the system rather than the clients or customers and the requirements here primarily refers to usability requirements.

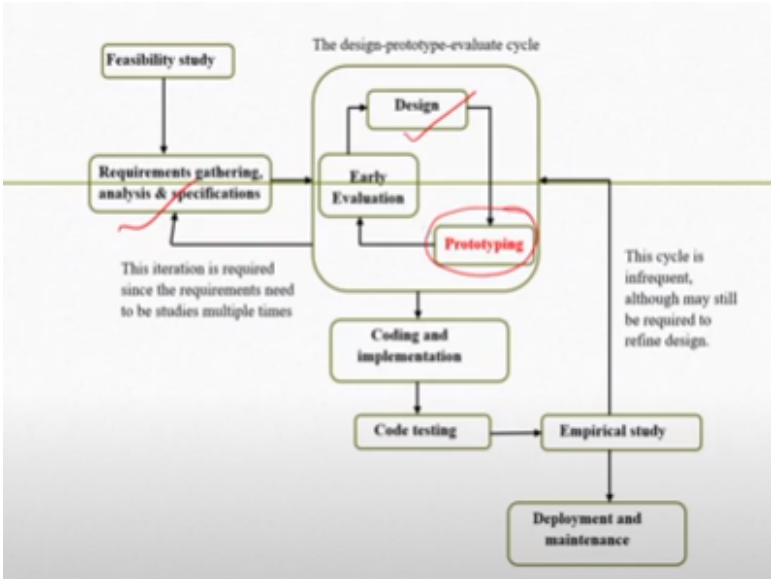
Then we have this design prototype evaluate cycle in which there are 3 stages design stage, prototyping stage and evaluation stage evaluation of the prototype. So here design refers to 2 types of designs one is the design of the interface where usability is the primary concern other one is the design of the system where execution efficiency is the primary concern. Now we learned how to create a design of an interface? We talked about some design guidelines as a starting point for designing our interface.

Now that design needs to be tested for that we need to make prototypes and get it tested through some early evaluation methods based on the test results we can update that design and this goes on in a cycle. Once the interface design is finalized we go for the system design where we build the design for the code. Now there we basically focus on modularity and maintainability of the code for quick implementation by a team.

After the system is designed we go for coding and implementation which is followed by code testing. So the code testing primarily deals with testing the execution ability as well as the efficiency of the code in terms of resource usage. Once this phase is over we get a working system which is supposed to be executable and resource efficient. However we still have not addressed through these testing's the overall usability of the end product.

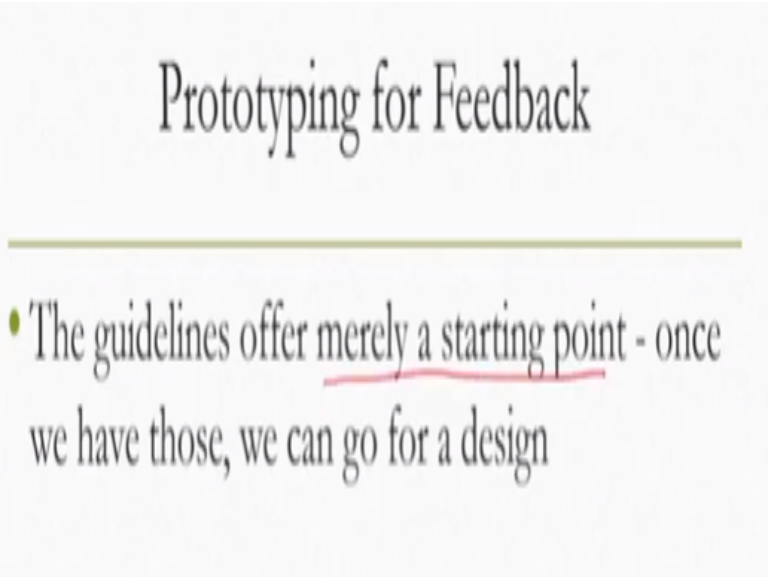
For that we have this next stage that is empirical study also called empirical research here with end users we test the product for usability issues if any. And if some issues are found out then we may need to go back to the earlier stages to refine and revise the design and subsequently change the implementation. Once we get a system which is usable as well as executable we go for deployment and the subsequent stage of maintenance. So this is in a nutshell the stages and the overall life cycle for interactive system development.

(Refer Slide Time: 05:44)



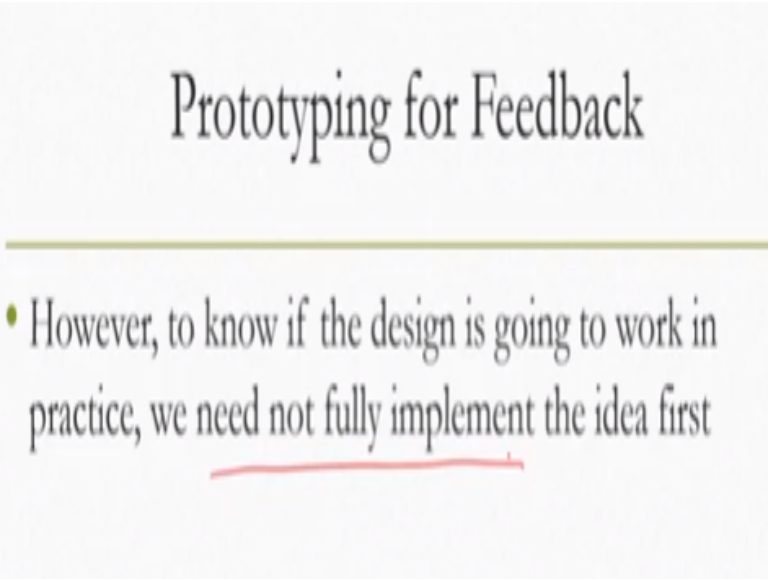
Among these stages we have already covered requirement gathering stage; we have covered interface design stage. We are now going to cover the next stage that is prototyping so essentially expressing the design that we have come up with for testing purpose. In the earlier lectures we have seen that we can make use of design guidelines to go for interface design now that provides a starting point.

(Refer Slide Time: 06:22)



However they are just that merely a starting point with this we can go for a design but we still do not know whether that design is going to be usable or not. So essentially what we need to do is? Basically we need to express the design for evaluation some sort of evaluation need not be very rigorous but some sort of evaluation.

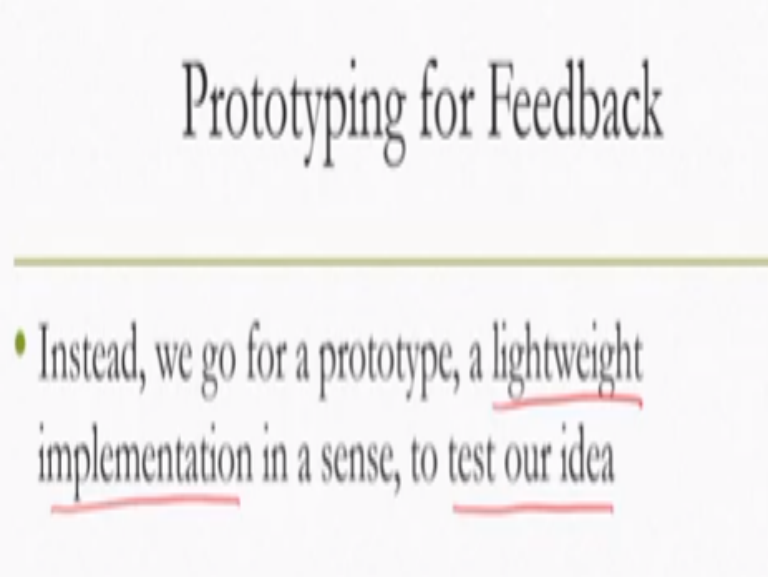
(Refer Slide Time: 06:42)



Now to express the design our intuition may tell us that go for full implementation of the system. Which is what most of us anyway think intuitively that ok this is the design idea let us go and implement it and then test it to know whether it is going to work in practice but that need not be

the case. To know if the design is going to work in practice we need not fully implement the idea first that is not necessary.

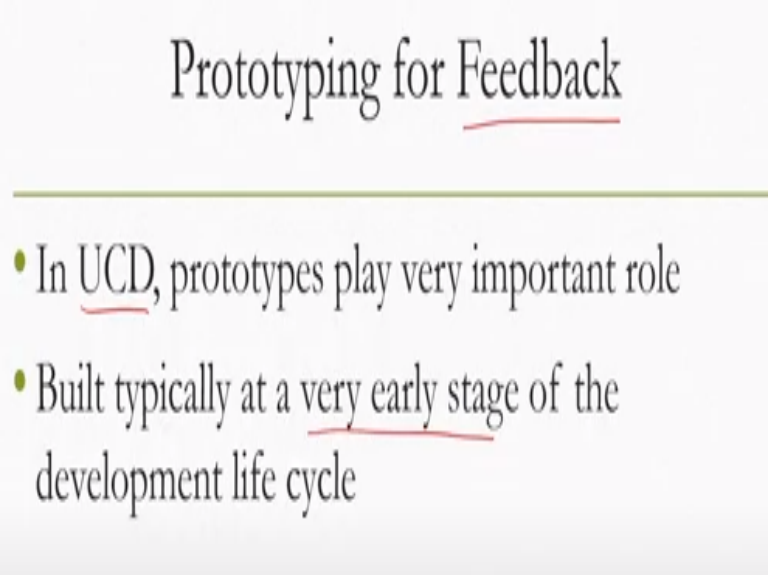
(Refer Slide Time: 07:19)



Instead what we can do we can go for building a prototype which is a lightweight implementation in a sense to test our idea. So once we have come out with a design idea to test the idea it is not necessary to go for full-fledged implementation of the idea in this case interactive software interface. Instead what we can do we can much reduce our effort and cost of development by going for lightweight implementation.

We will see what we mean by lightweight implementation in subsequent part of this lecture but what is required is a lightweight implementation of the design idea. So that it can be tested to know whether the design idea is going to work in practice if not then we need to revise our idea. And that lightweight implementation is nothing but a prototype for the design idea.

(Refer Slide Time: 08:26)

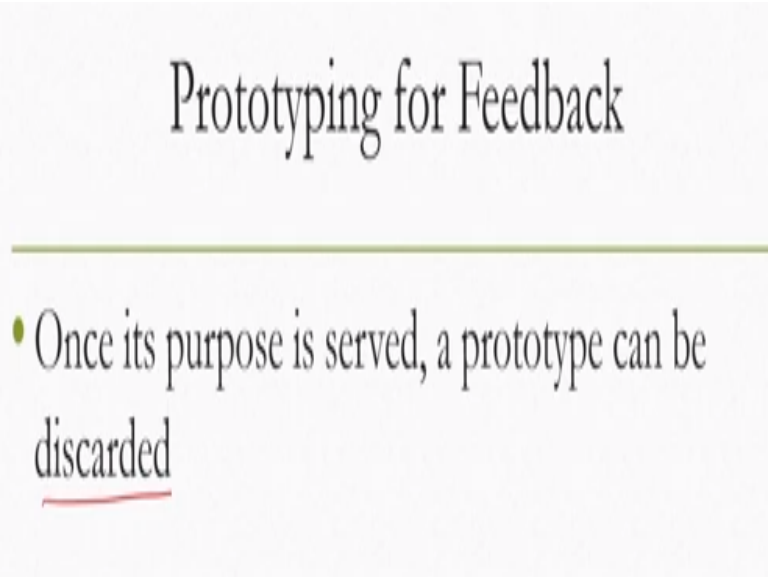


So prototyping can be used for getting feedback on our design idea whether this is going to work in practice or not. In user centric or user center design approach prototyping plays a very important role these are built typically at a very early stage of the development life cycle. So unless we take recourse to prototypes what we have to do is basically go for full-fledged design and full-fledged implementation which involves lots of effort and cost.

Now once everything is done completely and then we test it and then get to know that there are issues revising, those designs or implementations adds to the cost significantly. So we need to avoid that for that we need to make use of prototypes which are lightweight implementation reduces cost to a very significant extent. Cost as well as effort of development and allows us to test without going for full-fledged implementation in that way it helps us manage the overall effort and cost of the project.

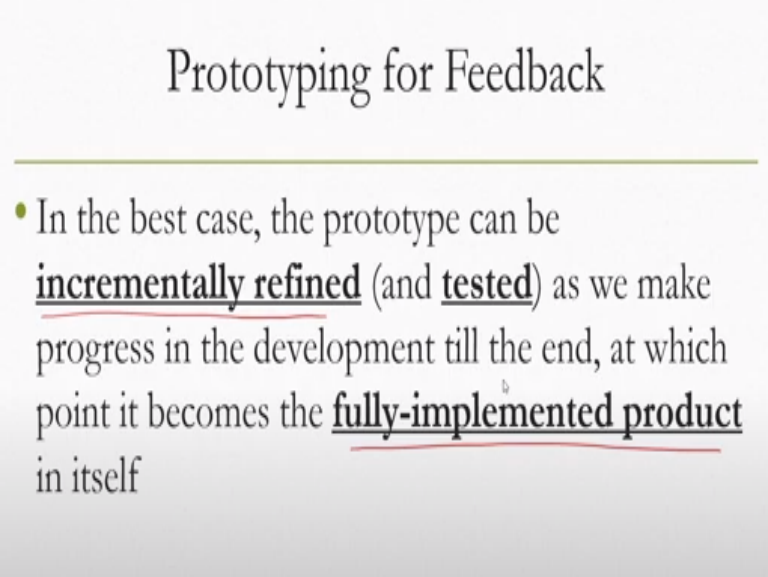
Also typically prototypes are built at a very early stage of the development so that we get feedback quickly and without spending too much time and effort as well as money. So these prototypes are significant part of any user centered design approach.

(Refer Slide Time: 10:05)



Now once the purpose of a prototype is served that means we get some feedback on the design idea by evaluating the prototypes we can discard the prototype that is what is commonly done. So the prototype is discarded.

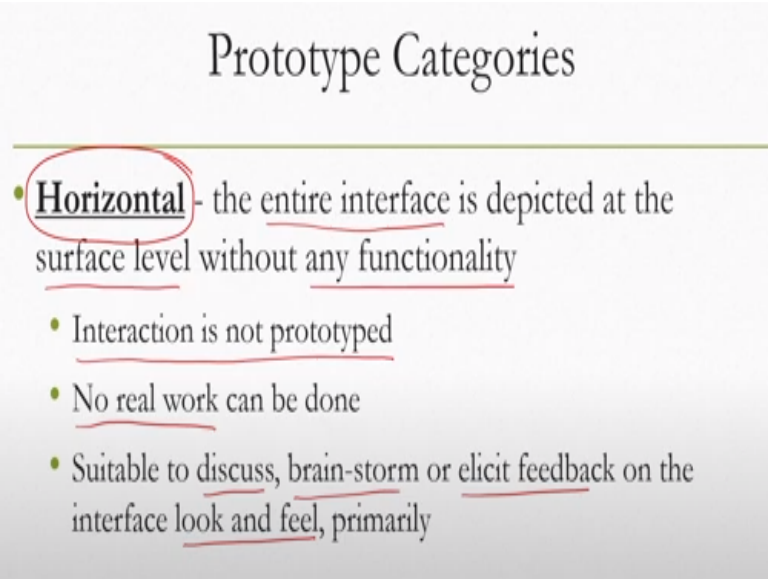
(Refer Slide Time: 10:25)



But that need not be the only possible way of utilizing prototype. On the other hand we can reuse the prototype we can incrementally refine the prototype as we make progress in the development till the end at which point it becomes the fully implemented product in itself. So what we can do? Either we can discard the prototype after we test the design idea or we can keep on incrementally refine the prototype. And test it and again refine it till we reach the end of the development cycle.

At which point of time the prototype itself becomes the final product that is another extreme of the use of prototype. Now there are different ways a prototype can be made so based on functionality that a prototype serves we can divide them into 2 broad categories.

(Refer Slide Time: 11:35)

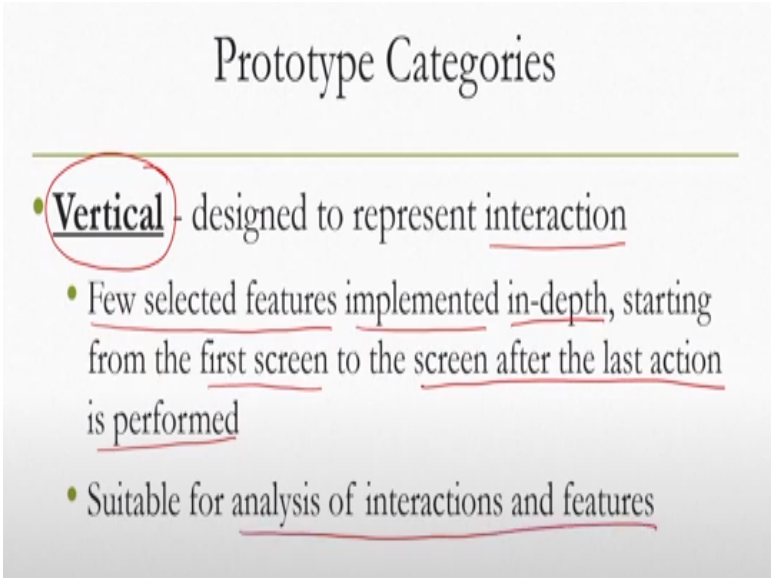


One is horizontal prototypes what these prototypes do is basically they allow us to express the entire interface at the surface level but without any functionality. So in horizontal prototype what we try to do we basically express the entire interface at the very surface level without adding any functionality to the prototype. So we cannot do any real work rather we can just test the look and feel.

So interactions are not prototype in horizontal prototypes since we are not implementing any functionality so we cannot do any real work in that sense that is we cannot interactively perform any task with such a prototype. So then if we are unable to do anything then what purpose does it serve? It serves significant purpose of course these are primarily suitable to discuss brainstorm or elicit feedback on the interface look and feel primarily.

So the primary purpose is to get feedback or to test the look and feel of the proposed design that is what we can achieve with horizontal prototypes.

(Refer Slide Time: 13:14)



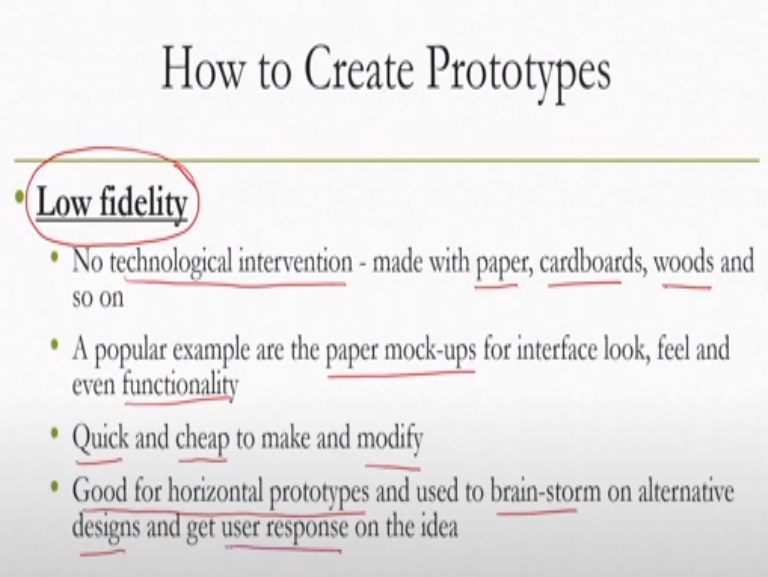
Now there is another category of prototypes that is vertical prototypes as opposed to horizontal prototype in these prototypes we can represent interactions. So vertical prototypes are designed to represent interaction horizontal prototypes are designed to represent interfaces primarily. Whereas vertical prototypes are primarily designed to represent interaction since interaction is represented it can also represent interfaces so it can represent both.

However that does not mean that with vertical prototypes we implement every functionality; of the system that can be achieved through interaction. Instead few selected features or functionalities are implemented in depth that is from starting to the end point. Starting from the first screen when the interaction starts to the screen after, the last action is performed. So, essentially the idea is that you choose a set of activities or interactions that can be performed with the proposed system.

Now this set has to be chosen carefully typically representative use cases should be chosen. Secondly the use cases that are chosen are to be implemented in full that means from the starting screen the screen when the interaction starts to the screen when the interaction ends. And all intermediate screens in between as well as the way to interact all these things have to be implemented in a vertical prototype. Clearly these prototypes are suitable for analysis of interactions and features of a proposed system.

So these are the 2 categories of prototypes which we can think of based on the functionalities that they support. There can be another way to categorize prototypes as well.

(Refer Slide Time: 15:42)



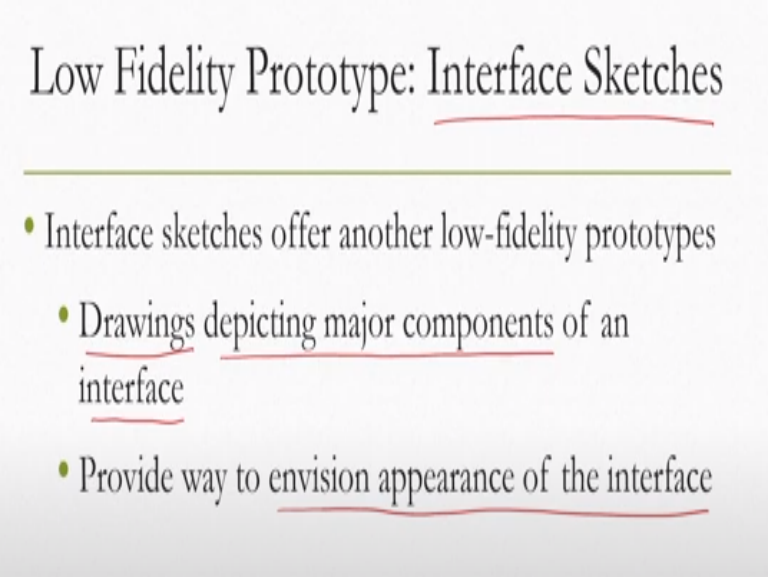
That is based on how we can create the prototypes how prototypes are implemented. So one such category is low fidelity so how to create prototypes there are different ways to create prototypes and broadly there are 3 categories of prototypes according to the way they are created. One category is low fidelity prototype in this case we do not require any technological intervention to create a prototype we do not require any computer software or anything.

Instead we can make use of objects that are easily available such as paper, cardboards, woods, clay and so on to create a prototype for a product. A very popular example of, such a prototype are the paper mock-ups for interface look fill and even functionality. So when we are trying to create a horizontal prototype we can use paper mock-ups which are a low cost way of implementing the idea.

Since we are not relying on technology and technological expertise is not required these types or these prototypes are quick to make and cheap to create. They are also quick and cheap to modify as well because in this design prototype evaluate cycle we need to revise our design and accordingly recreate the prototypes. So we should have a prototype which is not only easy to create but also easy to modify and low fidelity prototypes are best fit for that purpose.

By the very nature of its way of implementation these prototypes are good for horizontal prototypes and generally used for brainstorming on alternative designs and get user responses on the idea so that is the anyway the purpose of horizontal prototypes.

(Refer Slide Time: 17:58)

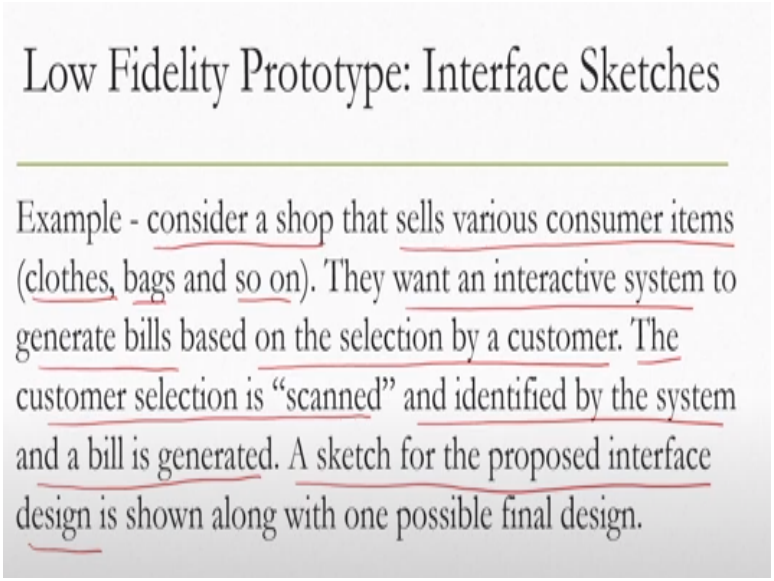


One special and popular low fidelity prototype is interface sketches. Now this can be considered to be slightly different from the other prototypes low fidelity prototypes that we have mentioned in the previous part of the lecture. So this actually offers another, low fidelity prototypes which are nothing but drawings as the name suggests name suggests drawings depicting major components of an interface. So again we are trying to basically create interface look and feel

So we are trying to create prototypes for interface look and feel and interface sketches allow us to do that. Again good for horizontal prototype because it, allow us to envision appearance of the interface. However when we are talking of interface sketches it need not be the case that they are made with simple papers or simple woods or clays or something so they can be typically drawn on a paper.

In that case that is low cost but they can also be created using computers so some technological expertise may be required but in principle interface sketches are considered to be low fidelity prototype.

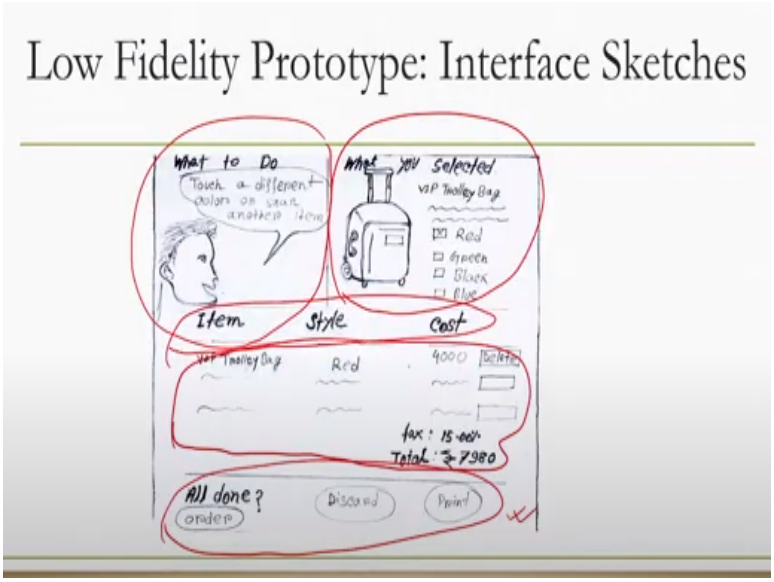
(Refer Slide Time: 19:17)



let us try to understand this with an example consider a shop now the shop sells various consumer items such as clothes, bags and so on. The shop owner wants an interactive system to generate bills based on the selection by a customer that is the requirement that has come from the owner of the shop. The customer selection is scanned and identified by the system and the bill is generated.

So that is a functional requirement which again has come from the client which in this case is the shop owner. Now the shop owner wants to have a system which allows the customer to perform such actions and then accordingly the system responds by generating a bill. A sketch for the proposed interface design can be made so that whatever design ideas the developers come up with can be shown to the client in this case the shop owner and his or her feedback can be taken.

(Refer Slide Time: 20:41)

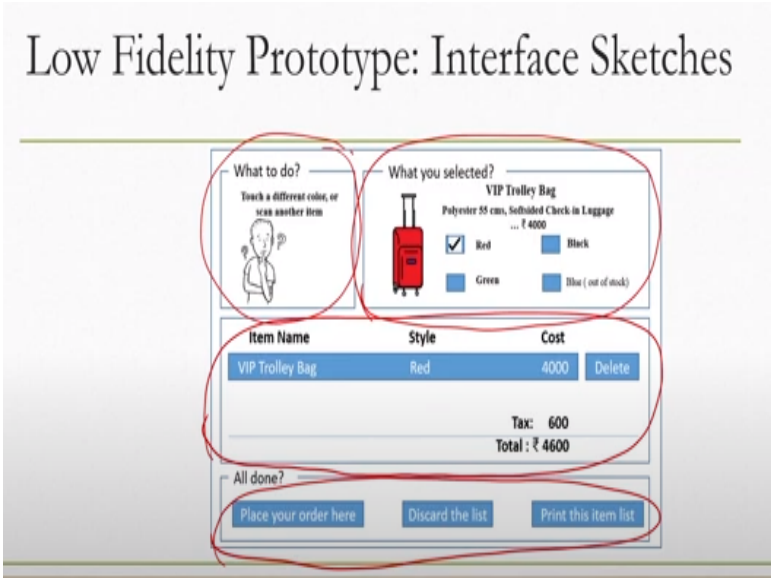


Let us see one such interface sketch here as you can see this is one idea of the interface how it should look like so it is created on a piece of paper. So it is a low cost implementation of the prototype of the interface only the appearance rather than any real function. Here there are according to this design there are several components one top component, then middle component, top left and top right there are 2 components. Then middle component, then lower component and a central component.

So these are the components that can be part of the interface as proposed by the design team and this design can be taken for feedback. And this prototype can be taken to the client or the shop owner for feedback on the design idea. So the top left corner component talks about certain things top right corner component talks about the selected product.

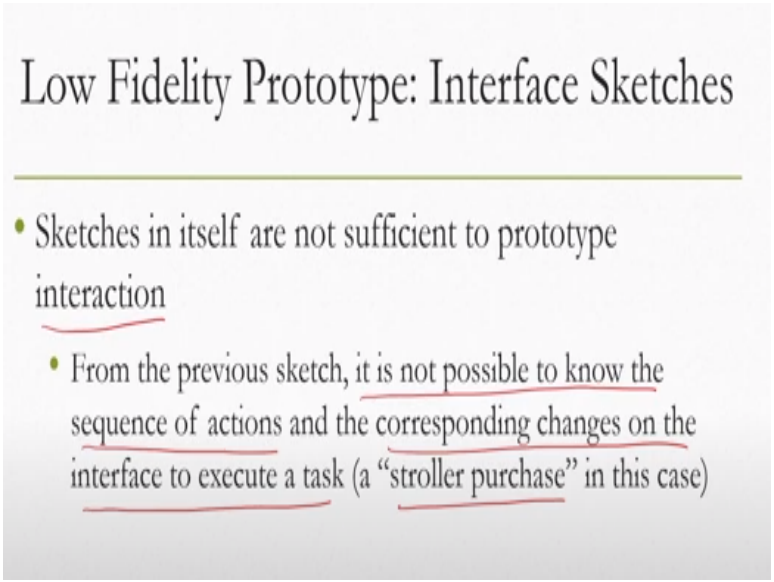
Central component with the menu and the details shows the billing information and the lower component here shows the options now the same sketch can be done in a different way. So here when we are using paper and hand draws the sketch clearly the quality is not very high. We can do the same thing with the help of a computer or sketching software.

(Refer Slide Time: 22:29)



As shown in this next version of the prototype the same thing but now; it is created with sketching software. Again the same components top left, top right, central component with menus and lower components. But now it looks much better and probably it will be easier to collect feedback if such a sketch is used. So here we are not only relying on low cost non-technological approaches to create the prototype. We may use some computers or technology to create a good quality prototype although that is still low fidelity prototype.

(Refer Slide Time: 23:23)

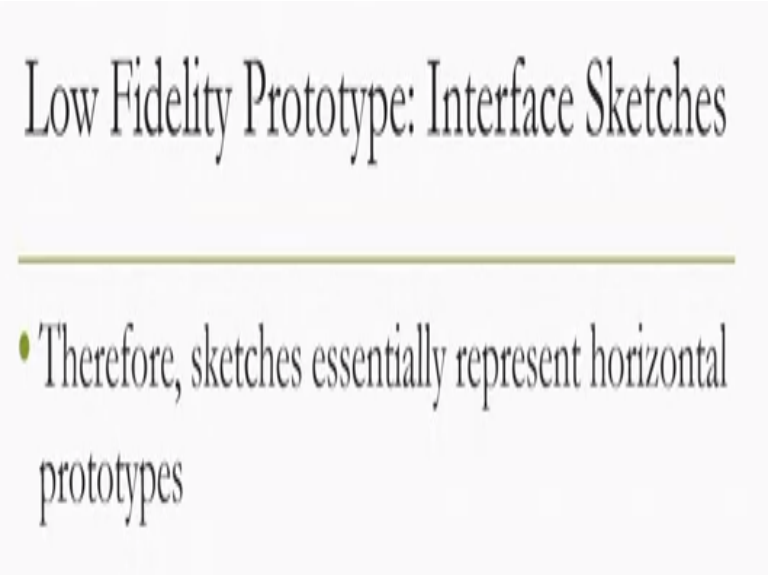


Now sketches in itself are not sufficient to prototype interaction because they are primarily used for horizontal prototyping. Primarily to get feedback on the look and feel of the interface as

probably we have seen in the previous example. In the example if we create such a prototype of the interface which is the sketch it is not possible to know the sequence of actions and the corresponding changes on the interface to execute a task such as a stroller purchase.

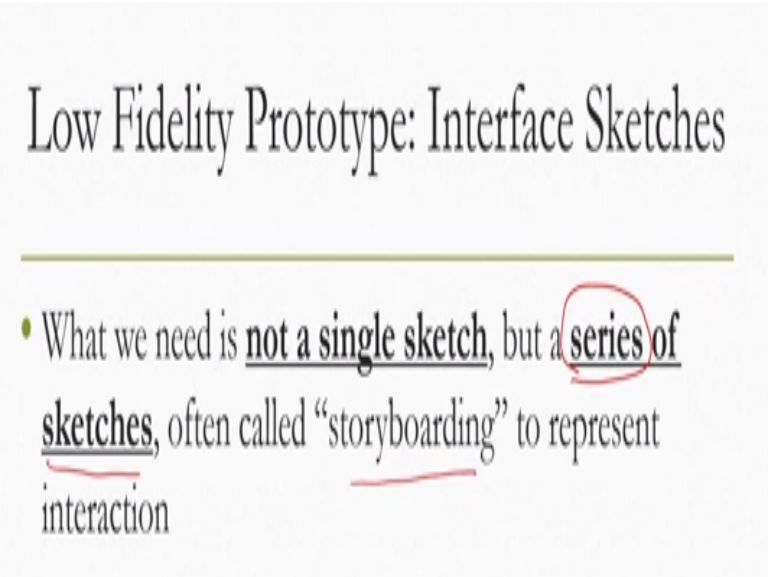
If the customer purchases stroller using the interface what are the actions and how the screen changes that type of information is not possible to get using simple interface sketches for that we need to do something more.

(Refer Slide Time: 24:21)



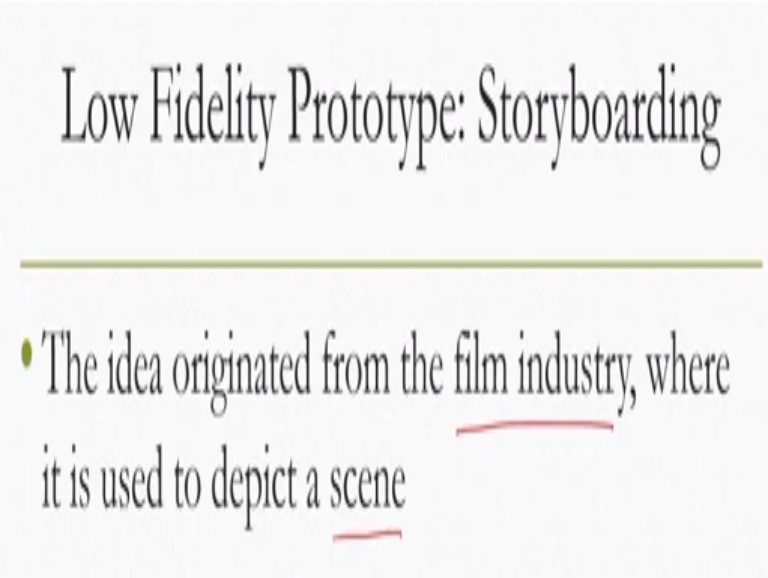
So sketches essentially represents horizontal prototypes now using the same idea and extending the idea a bit further we can actually make use of the idea to go for vertical prototypes as well. For that what we need to do is not to create a single sketch rather a series of such sketches.

(Refer Slide Time: 24:32)



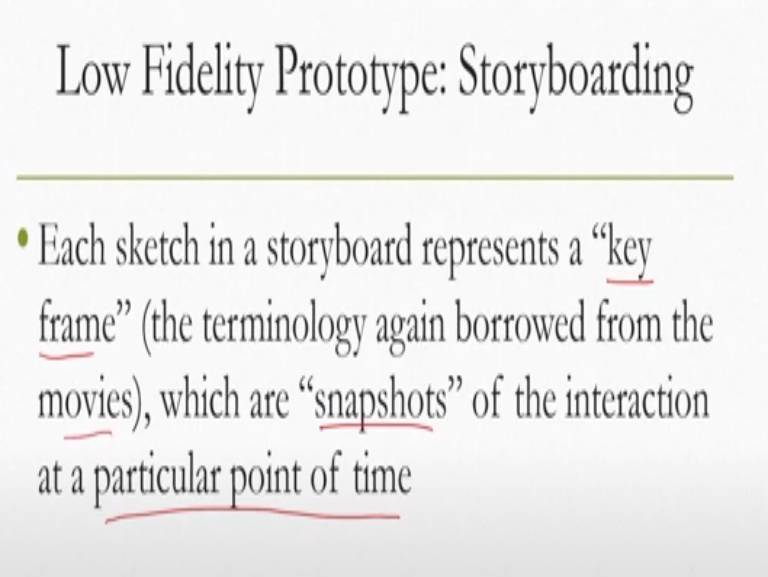
For that what we need to do is not to create a single sketch rather a series of such sketches for the interfaces at different stages of interaction. Now when we create such a series of sketches where each sketch represents an instance of the interface during the course of interaction for achieving a specific goal that series of sketches is called storyboarding. Now in other words storyboarding idea can be used to represent interaction or to go a little bit further we can create story boards to have a vertical prototype.

(Refer Slide Time: 25:18)



The idea originated from the film industry where such a storyboards are used to depict a scene. So before a scene is sought its sequence is depicted using storyboards so the same idea is used here.

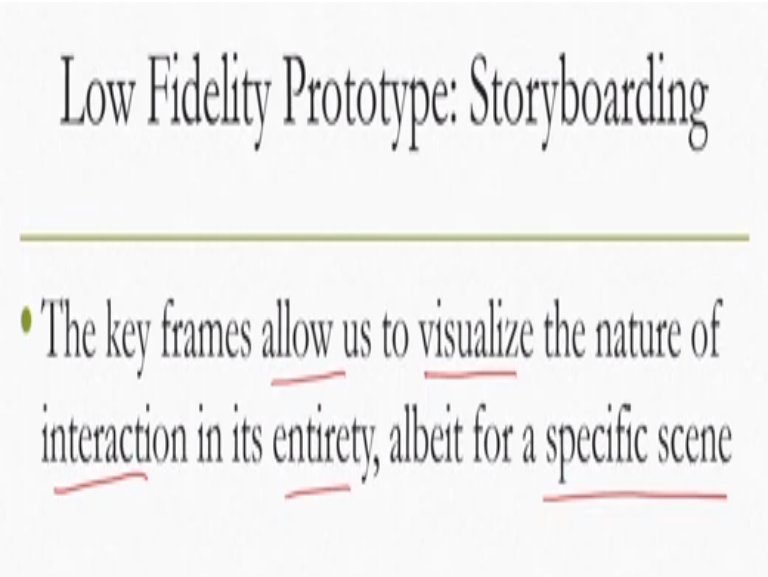
(Refer Slide Time: 25:40)



Remember that storyboard refers to a series of sketches now each intermediate sketch in the storyboard is called a key frame again the term is borrowed from the movies. Now a key frame can be considered to be a snapshot of the interaction at a particular point of time at any specific instance of interaction or a particular point of time during the course of the interaction.

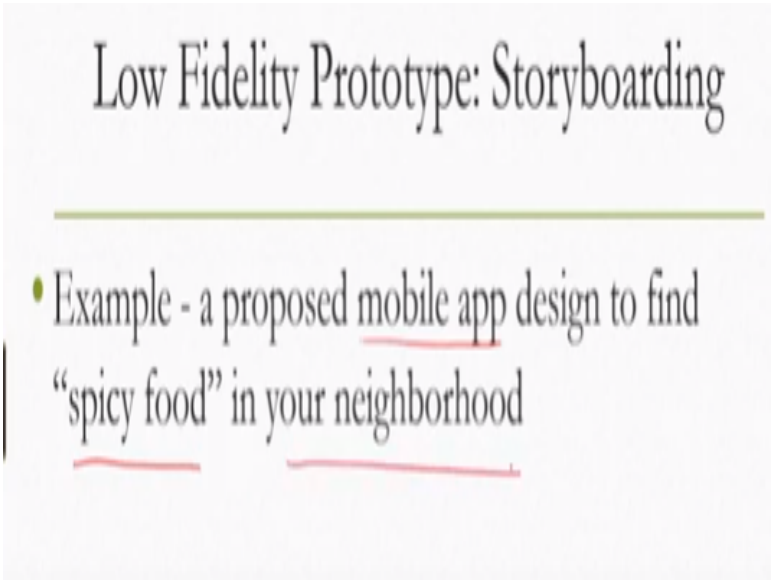
So when we are using the idea of interface sketches and extending it to create vertical prototype what we need to do is we have to create a storyboard depicting the interaction. Now in a storyboard there are key frames or intermediate interfaces which results due to the interaction during the course of the entire interaction.

(Refer Slide Time: 26:35)



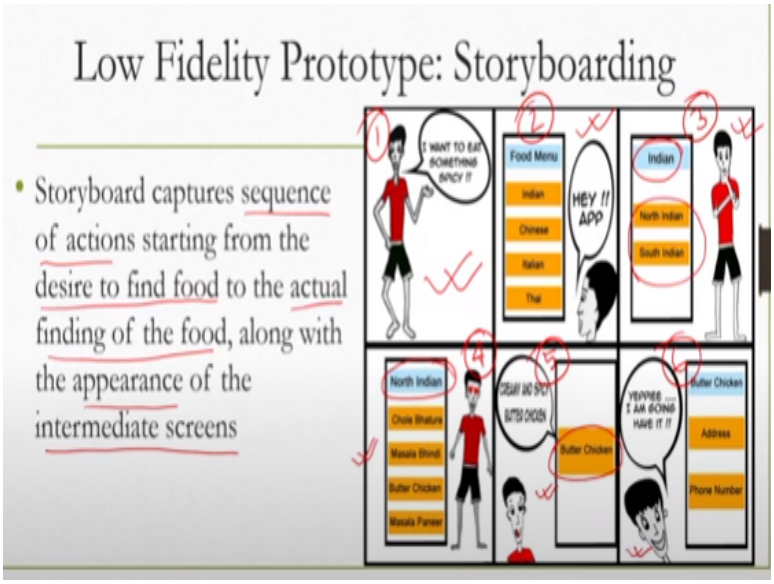
So the key frames allow us to visualize the nature of interactions in its entirety for a specific scene. So this is nothing but what we have discussed earlier as vertical prototype we want to visualize interactions involving change in screens or interfaces and that we can achieve with a storyboard having key frames. Where, each key frame represents an intermediate interface which results during the course of interaction at any given point of time.

(Refer Slide Time: 27:13)



Let us try to understand this idea with an example suppose we are trying to design a mobile app to find spicy food in our neighborhood that is the design idea.

(Refer Slide Time: 27:29)



So in a storyboard we try to capture the sequence of actions starting from the desire to find food to the actual finding of food along with the appearance of the intermediate screens. So that should be captured in a storyboard as shown in this series of screens. First screen shows the desire to find food that is the starting point of the interaction then opening the app to do the same that is second screen where it shows some food menu and whatever is available.

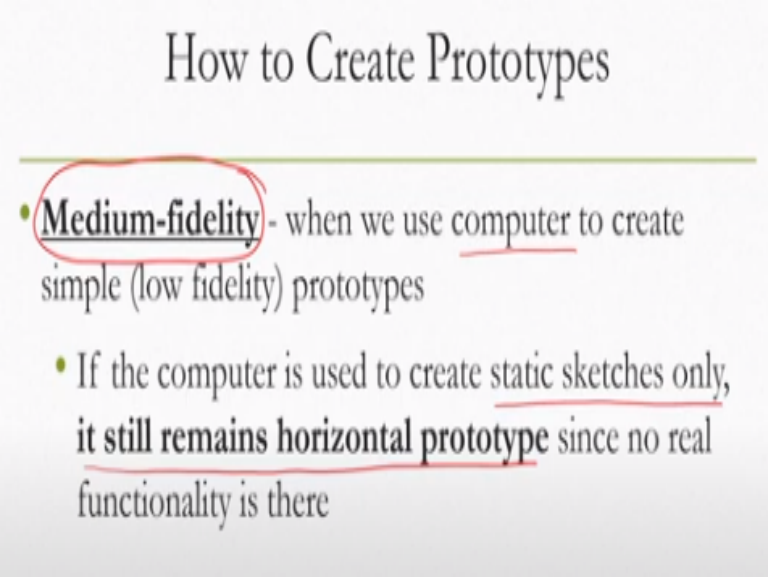
Then in the third screen we go for specific food items say in this case Indian then what should be on they are under this Indian food menu north India and south Indian. 4th screen another category is selected north Indian food and under which some menu options are available that is depicted here. Fifth screen shows that we have zeroed in on a specific menu item with some description.

And finally in the last screen we can depict what to do to order that food so we have 6 sketches one, 2, 3, 4, 5 and 6 together these 6 interface sketches constitute the storyboard. Each of these sketches is a key frame so there are 6 key frames which together make up the storyboard. And each of these key frames represents an instance of interaction and the corresponding interface at that point of time.

This is how we can create a storyboard and here as you can see by specifying the sequence of key frames we can actually specify the interaction that the interaction for selecting a food should happen in this way. So that is how we can create a prototype for the interaction using the idea of

storyboarding and just want to emphasize again that storyboarding is a low fidelity prototyping technique.

(Refer Slide Time: 30:18)



So apart from low fidelity prototypes we can have another way of creating a prototype that is medium fidelity by definition medium fidelity prototypes are those which are created using computers. Now if the computer is used to create static sketches only it still remains horizontal prototype since no real functionality is there. So earlier we talked of low fidelity prototype where we are using non-technology based solutions such as paper mock-ups, use of clay, use of woods to create prototypes.

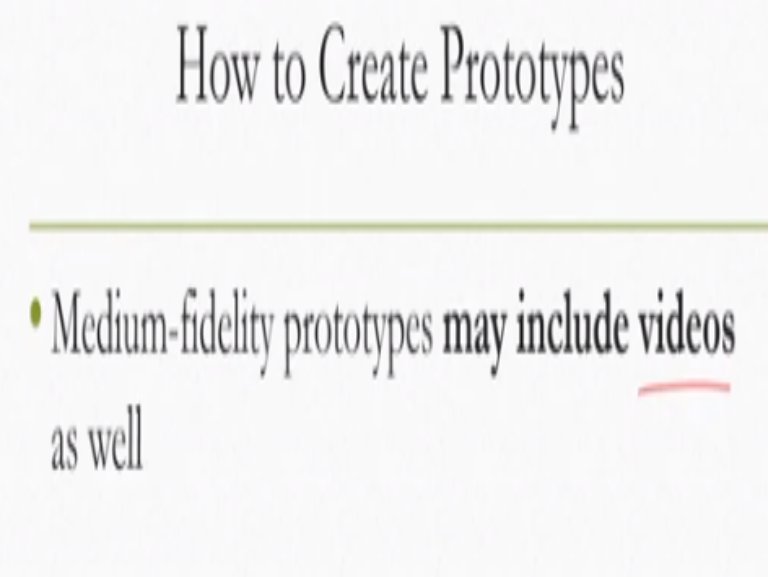
In interface sketches we have seen that we can do it in either of the 2 ways one is simply drawing the sketch on a paper hand-drawn. Other one is using a computer tool sketching tool to create it on created using a computer in the latter case the quality is likely to be much better than creating something on pen and paper unless the person who is creating the sketch is a skilled painter.

Now when we are creating a prototype using, say a paper and pen that is low fidelity. When we are using the same prototype using a computer we can call it medium fidelity rather than low fidelity because here we are having some technological intervention. So although in principle everything is low fidelity because we are unable to do any specific functionality using sophisticated mechanisms still since we are using computer we can call it medium fidelity to differentiate it from the low fidelity prototypes.

So interface using sketches created using pen and paper falls under low fidelity prototype. Interface sketches using created using computer falls under medium fidelity prototype. Similarly storyboards having key frames created, with pen and paper falls under low fidelity prototype. Whereas storyboards having key frames created using computer tools fall under medium fidelity prototype. Of course this is how we implement the prototype it does not change the nature of the prototype that is whether it is horizontal or vertical.

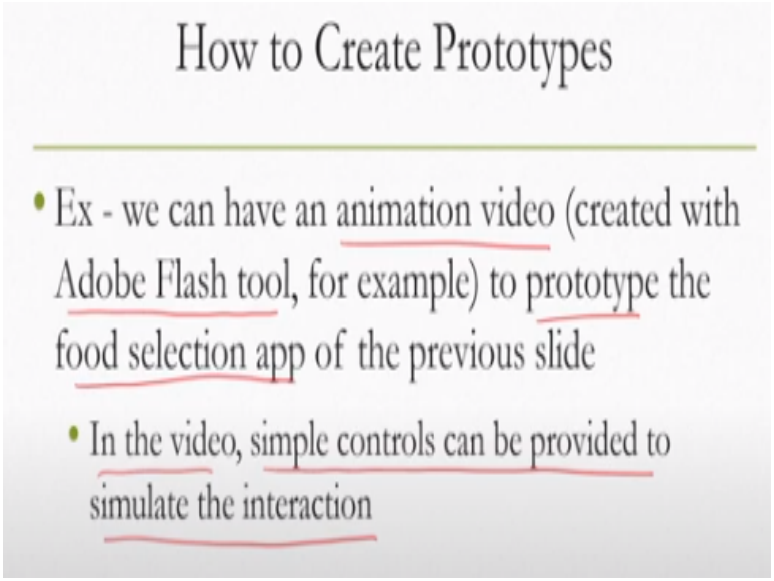
With medium fidelity we can have horizontal prototypes if there is no functionality support provided. Whereas if we are using computer to create a storyboard with support for functionality or interaction then that medium fidelity can be a vertical prototype. However with medium fidelity we can go even further than simply replicating low fidelity ideas.

(Refer Slide Time: 33:32)



Medium fidelity prototypes can include videos as well so instead of creating simply sketches or story boards we can create a full-fledged video showing the transitions with a medium fidelity prototype.

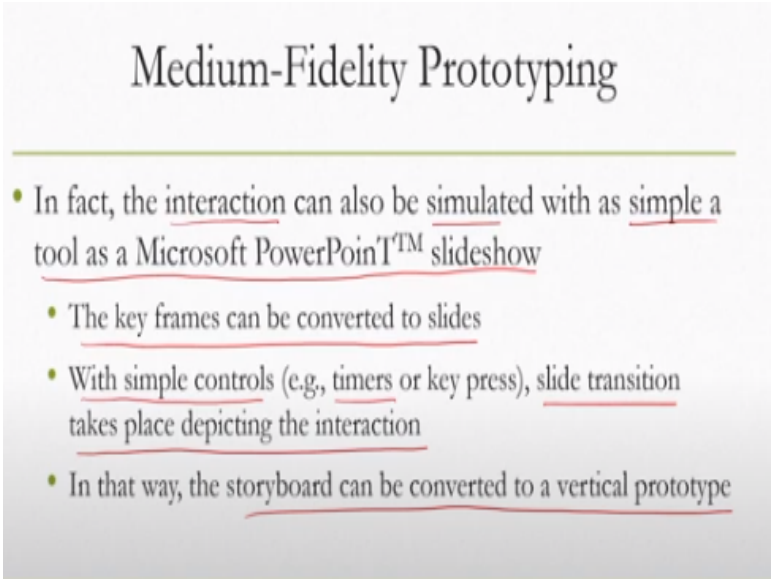
(Refer Slide Time: 33:48)



For example we can have an animation video created with some software tools such as adobe flash tool or similar tools. And that animation video can be used to prototype the food selection app of the previous example in the video simple controls can be provided to simulate the interaction. For example simply pressing a keyboard button say enter button or simply clicking on the mouse button these can be simple controls utilized to simulate change of interfaces.

So we can create an animation video and enable it with simple controls to move forward to simulate a vertical prototype but that is not the only way to have a medium fidelity prototype.

(Refer Slide Time: 34:46)

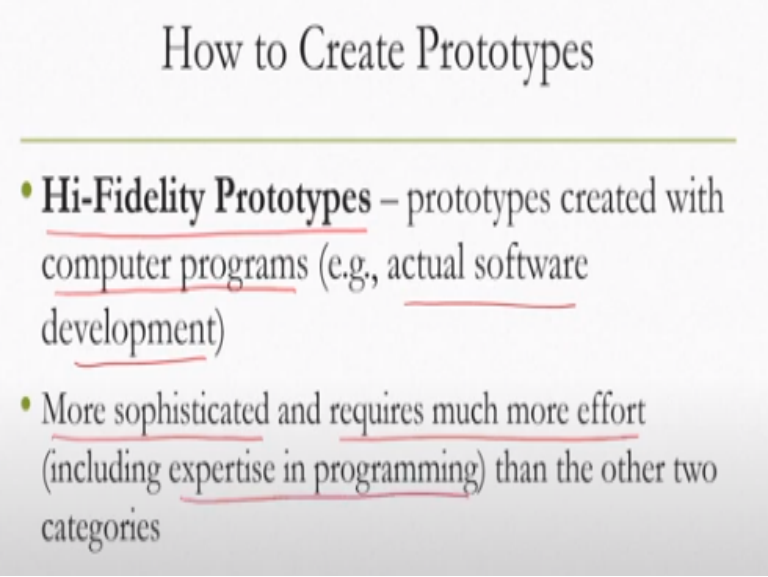


When we are talking of vertical prototype the interaction can also be simulated with as simple a tool as a Microsoft PowerPoint slideshow. So a simple slideshow with timer can be used in place of other controls to basically simulate a vertical prototype. So in this case the key frames can be converted to slides if you know already how to create slides in ppt Microsoft PowerPoint then it will be easier to understand.

So we can create slides which represent the key frames with simple controls such as a timer or key phrase slight transition takes place depicting the interaction and in that way the storyboard can be converted to a particle prototype a medium fidelity vertical prototype. So when we are referring to the term medium fidelity we are essentially referring to the idea of low fidelity but implementing those, idea with the help of computer.

Rather than non-technological means of implementation such as papers, pens or such similar things but the ideas remain more or less similar such as interface sketches, storyboards, key frames all these things remain the same.

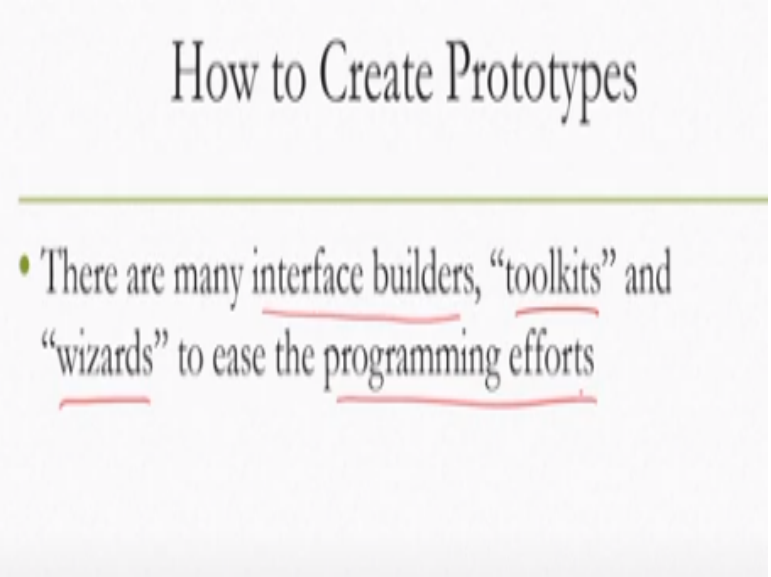
(Refer Slide Time: 36:10)



The other major category is high fidelity prototypes unlike in low fidelity or medium fidelity prototypes in high fidelity prototypes we rely on computer programs so the prototypes are created by writing programs. So this indirectly means actual software development activities involved in creating the prototype. Since, we are using programs to build the prototype so these are more sophisticated as compared to low or medium fidelity prototype. But at the same time

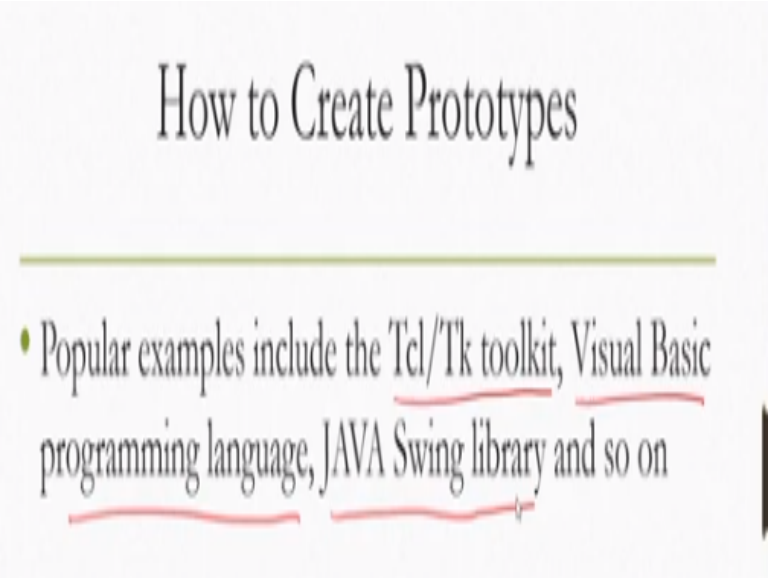
obviously they require much more effort including expertise in programming than the other 2 prototyping methods.

(Refer Slide Time: 37:05)



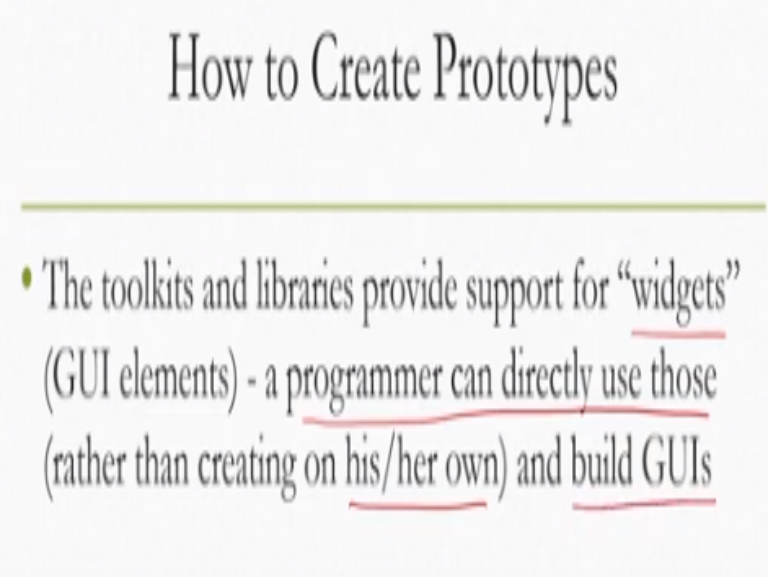
To create high fidelity prototype it is not necessary to always start writing programs from scratch there are supports provided in the form of interface builders toolkits and wizards to ease the programming efforts such supports are already there.

(Refer Slide Time: 37:27)



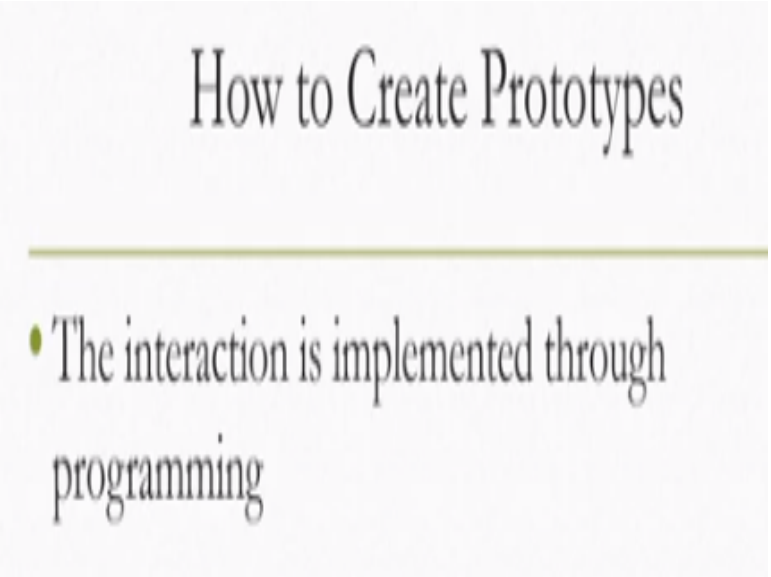
Examples of such supports include the Tcl Tk toolkit for creating GUI'S visual basic programming language again for creating GUI'S java swing library for creating GUI'S and so on.

(Refer Slide Time: 37:46)



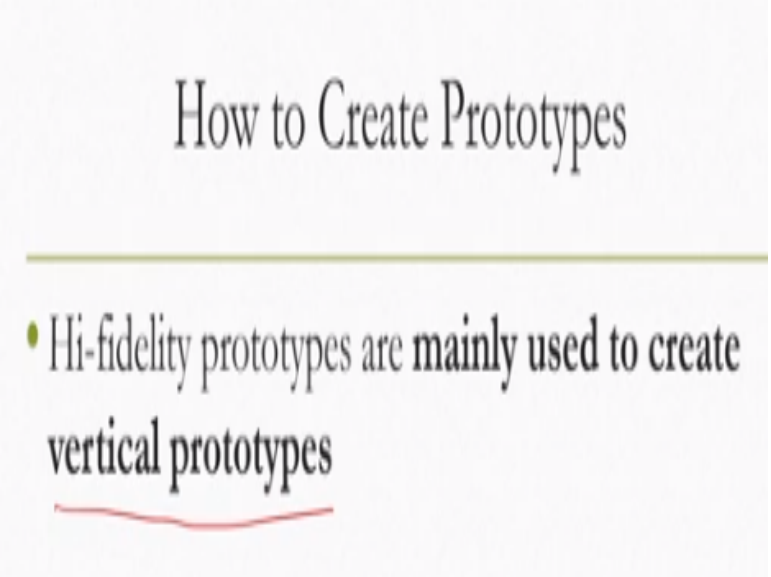
Now the toolkits and libraries that are already available provide support for widgets or GUI elements graphical user interface elements such toolkits and libraries allow us to easily create and manipulate such widgets in a prototype. So a programmer can directly use those widgets from the tools and libraries available rather than creating them from scratch on his or her own. And using those widgets it is much more, easier to build GUI's quickly without too much effort.

(Refer Slide Time: 38:32)



That is one part of the story that is to create the interface other part is to implement the interaction that is done through programming. Essentially by writing codes to listen to user inputs and then accordingly take actions to change the interface all these activities are done by writing programs. So tools and libraries provide support for both creating GUI'S as well as implementing interactions.

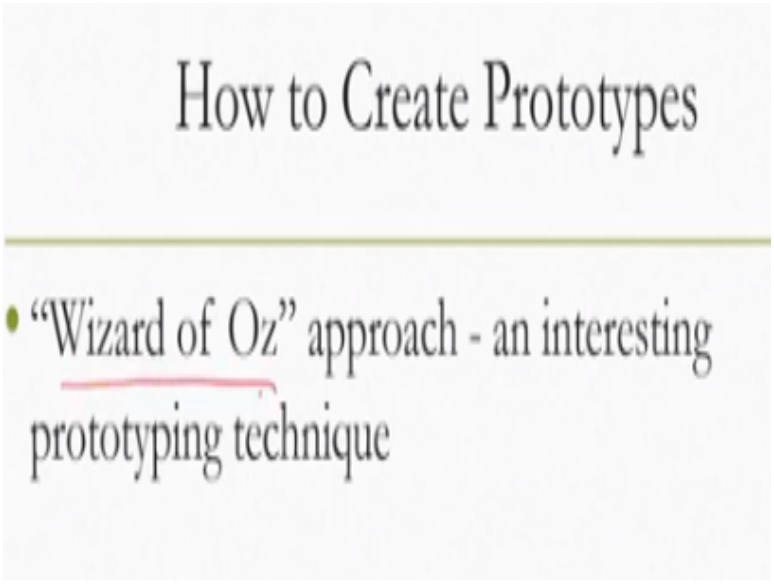
(Refer Slide Time: 39:05)



For implementing interactions tools and libraries again provide support for event listeners so that user inputs can be easily captured and then necessary actions can be taken. Now since it involves lots of effort and it provides sophisticated prototypes high fidelity prototypes are typically used

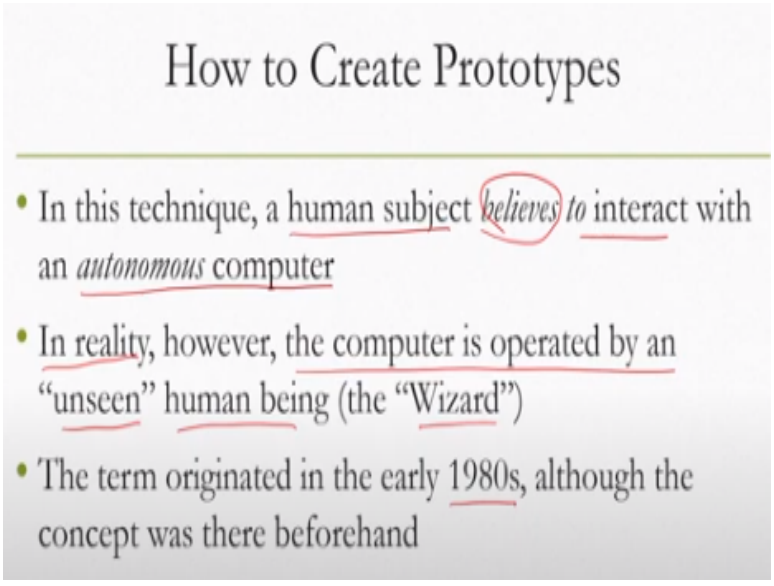
for creating vertical prototypes where sophistication is required. For horizontal prototypes such effort may not be worth of it so for purely horizontal prototypes it is preferable to go for low fidelity or medium fidelity prototyping approaches.

(Refer Slide Time: 39:50)



One interesting prototyping approach is called wizard of Oz approach this is actually a very interesting approach let us quickly try to understand it.

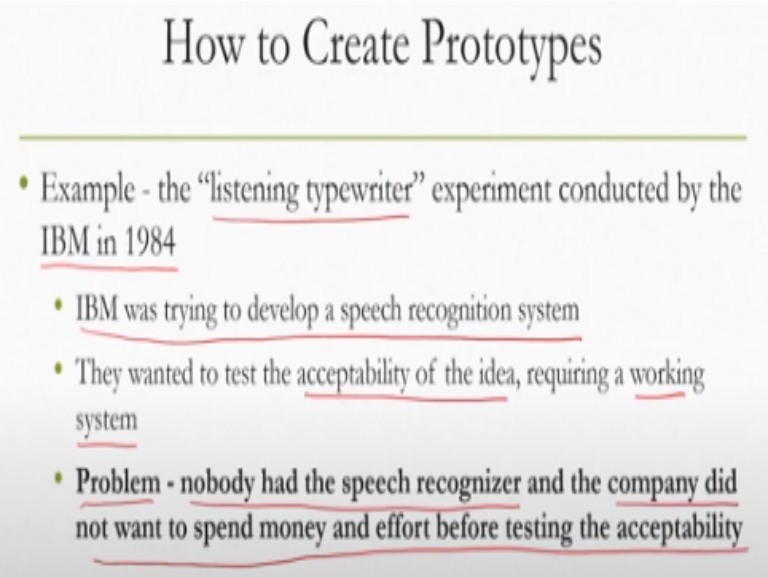
(Refer Slide Time: 40:02)



In this technique a human subject believes to interact with an autonomous computer note the emphasis on the word believe. So, human subject believes that he or she is interacting with a

computer in reality that is not the case. What happens is that the computer is operated by an unseen human being who is referred to as the wizard. Now the term originated wizard of oz this phrase originated in the early 1980s but the concept was there from before but in the early 1980s this was put into practice.

(Refer Slide Time: 40:46)



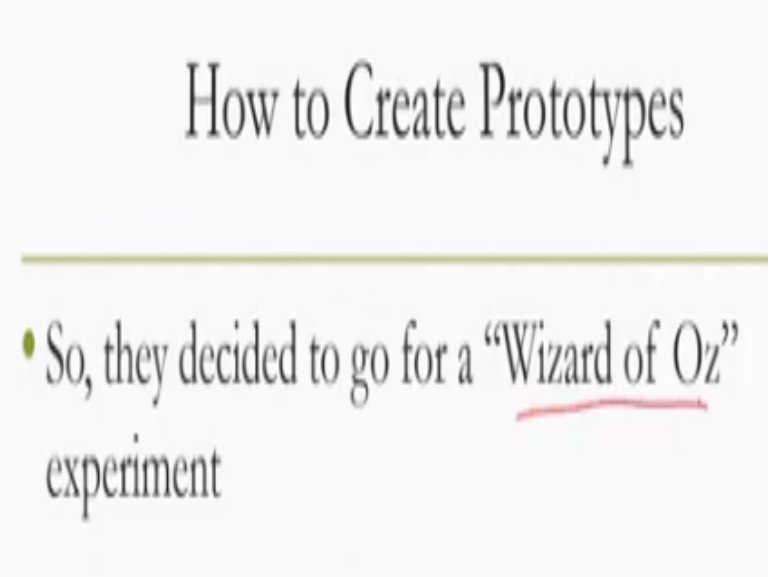
An example of application of this concept is the testing of the listening typewriter product by IBM way back in 1984 what happened that time is that IBM was trying to develop a speech recognition system. Now at that time speech recognition was a very futuristic technique although nowadays we take it for granted but in 1984 it was a futuristic technique nobody knew how to implement it successfully.

But at that time IBM was trying to estimate whether if such a system is made whether that will be a sellable product an acceptable product. So IBM was trying to test something which is not yet there and they applied this wizard of oz approach. So, they wanted to test the acceptability of the idea requiring a working system otherwise how do you test the acceptability if you do not have a system. But the problem is at that time nobody had the speech recognizer and the company IBM did not want to spend money and effort before testing the acceptability.

So nobody knew how long it will take or whether it will ever be possible to develop a speech recognizer. So to test it we first need to develop it but then that is risky uncertain and involves lots of money. IBM was not interested in spending the money before they know that if they put

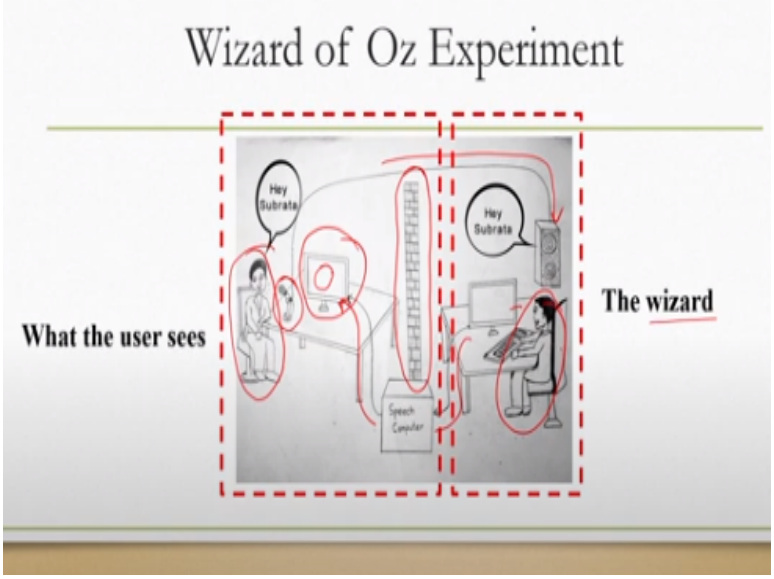
in effort then actually they will come up with a sellable product. So they want to test something which cannot be made at that time.

(Refer Slide Time: 42:30)



So they decided to go for a wizard of oz experiment.

(Refer Slide Time: 42:38)



So what was there in the experiment in this figure the experimental setup is shown in this figure on the left side sits the user. In front of the user there is a microphone user utters something and get some response on the screen. So user thinks that there is only this computer and the

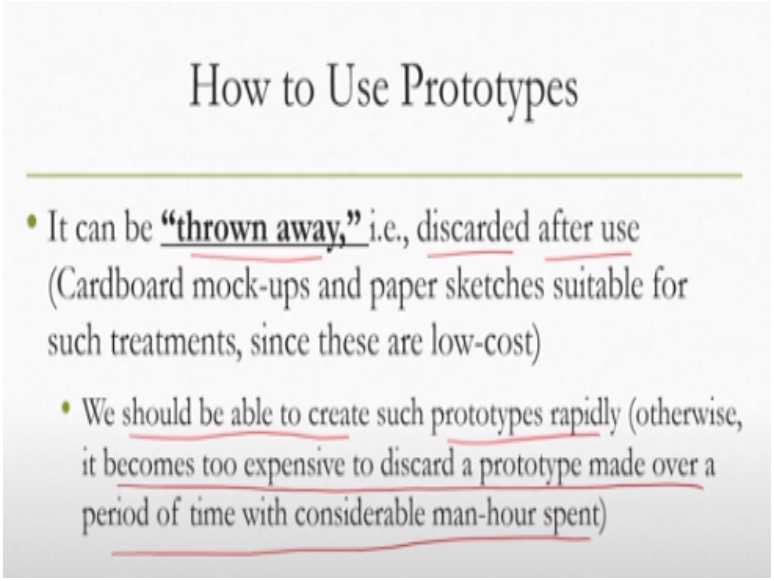
microphone but there is a wall in between behind the wall there are seats the wizard or a human operator of the computer.

So the microphone output is connected to the computer of the wizard and to the speaker near the wizard which is not audible to the user. So whatever the user utters the human wizard gets to here and based on that the human wizard types a response which is transmitted to the user's computer. And it appears on the computer here and the user thinks that because he uttered the word the computer understood it and then gave the response because the user is unable to see the wizard.

Or however of his presence because the user is unable to see the wizard or however of his presence in that way the user thinks that there is a computer which understand the speech and then responds according to the content of the speech. So based on this understanding the user decides whether this system is good or bad so in this setup this feedback can be collected about the acceptability of the system.

So this overall approach is called wizard of oz where we are not relying on actual technology fully rather we are relying on a human in the loop for creating the prototype. So that is an interesting way of creating prototype for futuristic technologies which are yet to be developed or which are yet to be in a matured stage. So we have learned about different ways to categorize prototypes.

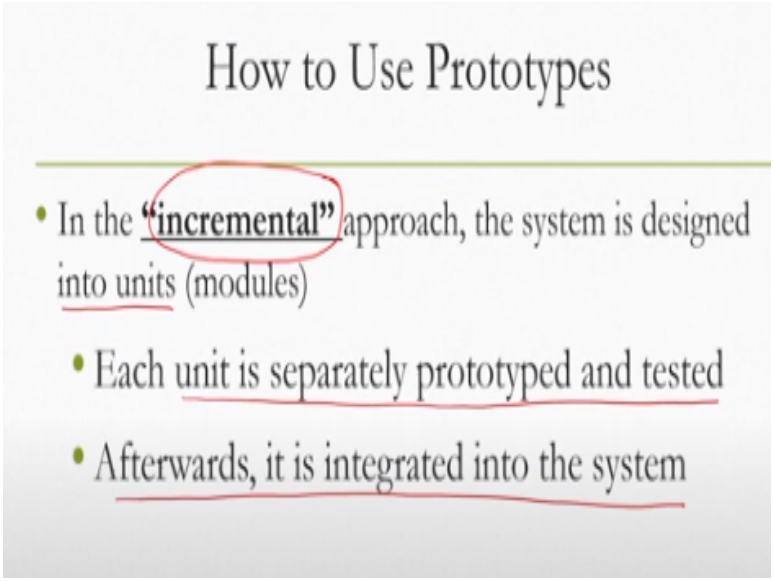
(Refer Slide Time: 44:57)



Another important issue is how to use the prototypes again there are broadly 3 ways in which we can make use of the prototypes. It can be thrown away that means the prototype is discarded after use. For example if we are creating a cardboard mock-up or a paper mock-up or a wooden mock-up or a clay model or sketches drawn on paper we need to throw it out. Because they are not anywhere computerized but throwing them out is also not a problem because they are low cost and takes very little effort to create.

So the idea of thrown away prototypes is that we should be able to create such prototypes rapidly and with low cost. Otherwise it becomes too expensive to discard a prototype made over a period of time with considerable man hours spent. So when we meet these criteria then we can throw away the prototypes and these are one way of utilizing after getting feedback we throw it away.

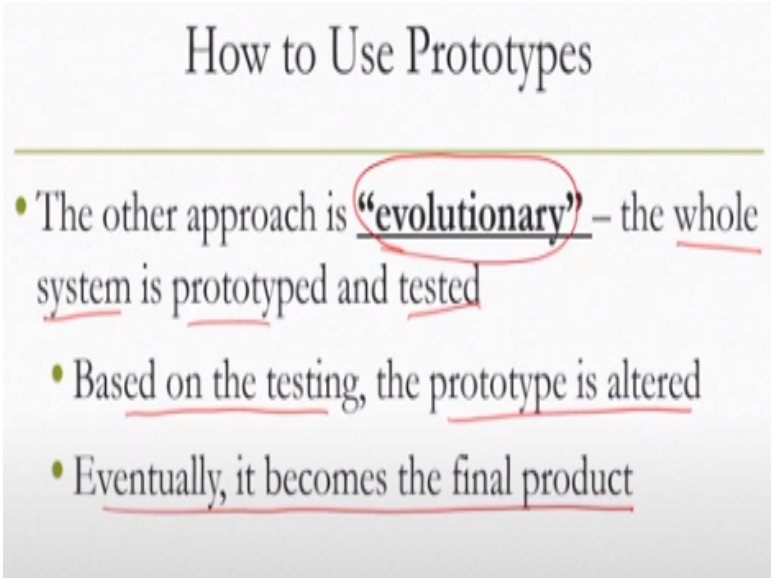
(Refer Slide Time: 46:02)



Another way of using prototype is the incremental approach what this approach says here the system is designed into units. So, overall system that we are trying to test is divided into units or modules. Each unit is separately prototyped and tested afterwards it is integrated into the system. So the entire system is divided into units each unit is prototyped, tested, feedback collected revised all these things done.

And then they are integrated together into the whole system so that is an incremental approach. Here we are not discarding but we are refining the prototypes of individual modules or units of the system. And then finally once all are tested and refined we are integrating them into the whole system that is the incremental approach.

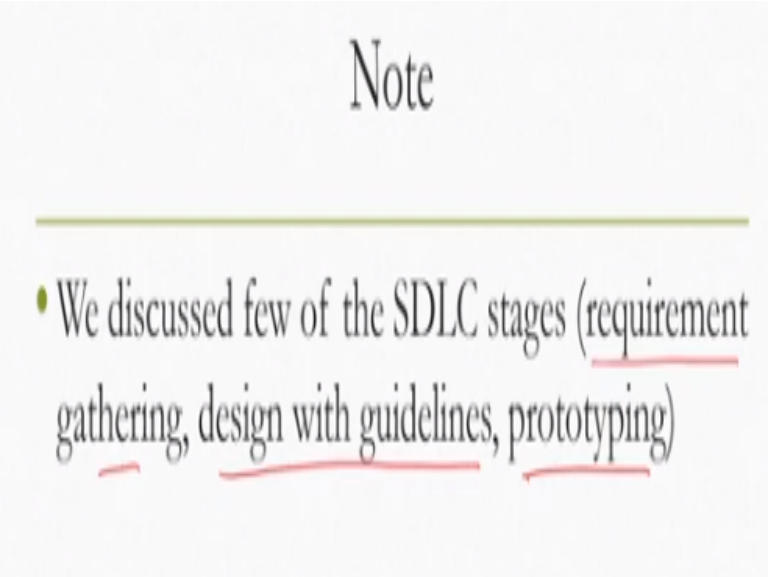
(Refer Slide Time: 46:58)



Third approach is called evolutionary for utilizing prototypes we can go for evolutionary approach as well. Here also we are not throwing away the prototypes but unlike incremental approach here we are not dividing the system into modules and separately prototyping each module. Instead what we are doing we are creating a prototype for the whole system together and then we are testing it.

Based on the testing the prototype of the system is altered and this goes on in a cycle and eventually it becomes the final product. So the evolutionary approach is what immediately comes to our mind when we think of prototypes to create the full system, test it, refine the system, test it again and goes on in a cycle till we reach the final product so that is called evolutionary approach. But that is not the only approach other approaches as we have discussed include thrown away approach and incremental approach we can choose any of these approaches.

(Refer Slide Time: 48:07)



So with that we come to the end of our discussion on prototypes so here we have learned various things about prototypes. What is prototype, how it is useful so we have seen that prototypes can be vertical or horizontal based on what we implement as prototype. We have also seen that depending on the use of the material and technology we can categorize them as low fidelity medium fidelity or high fidelity prototypes.

And learned about few of the prototyping techniques such as interface sketches, storyboarding high fidelity of course provides us the most sophisticated prototypes because they rely on programs and purely technology oriented. Also we have seen how to make use of the prototypes once they are built. We can either throw them away after use that is after getting the feedback on the design idea which is typically the case with low fidelity low cost prototypes.

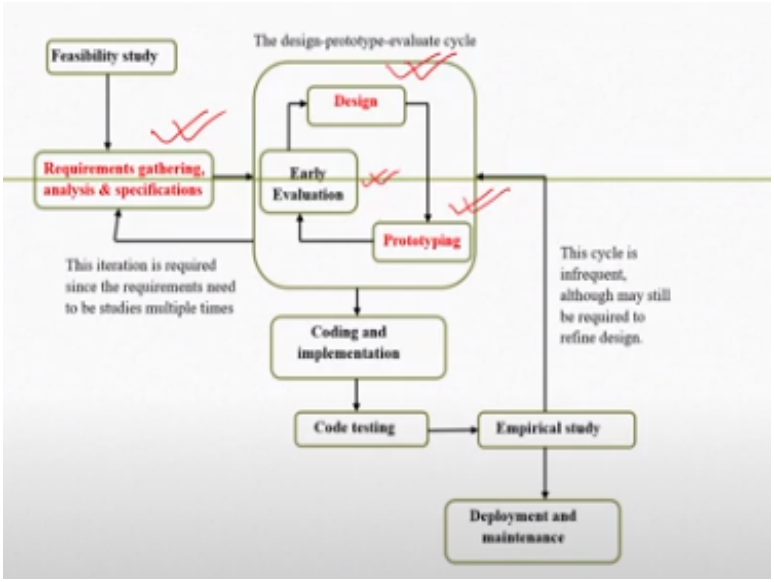
Or we can keep on using them till they convert to the final product so now this approach can be done in either of the 2 ways either we can go for incremental approach or we can go for evolutionary approach. In incremental approach we divide the whole system into units prototype each unit test each unit separately. And finally integrate them together to get the whole product.

In evolutionary we create the prototype of the whole system itself at the beginning then incrementally refine them after every testing cycle and finally it leads to the final product. So that is in a nutshell we also have learned about wizard of oz approach. So that is in a nutshell

what is prototype and how we can make use of prototypes in the interactive system development lifecycle.

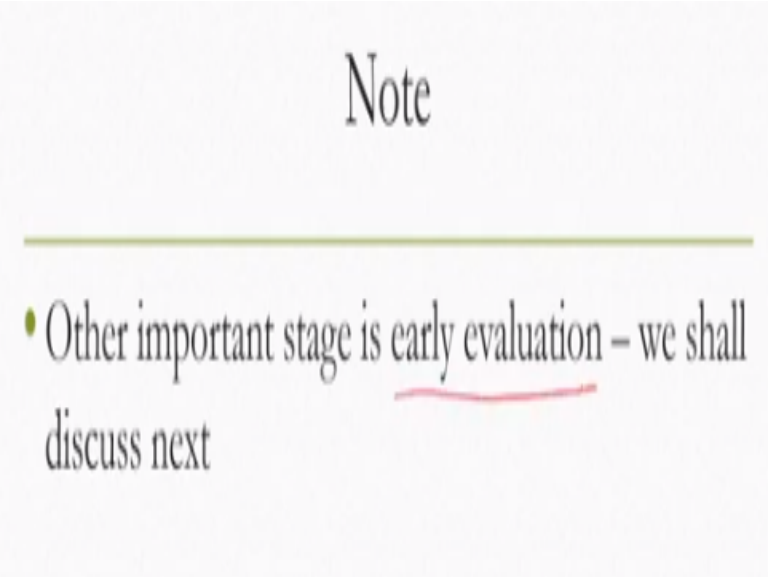
So with that we have finished our discussion on several of the SDLC stages namely requirement gathering, interface design stage with guidelines and with this lecture on prototyping stage as well.

(Refer Slide Time: 50:33)



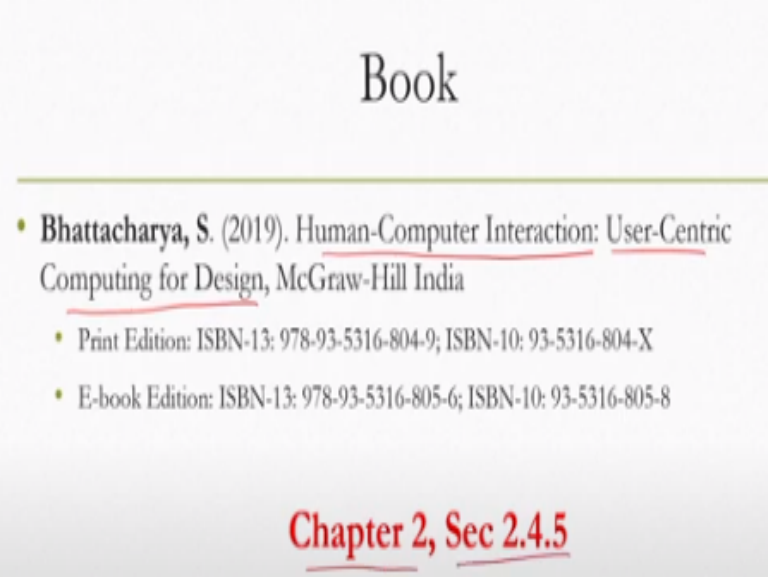
So if we see the life cycle again so we have already covered this stage requirement gathering stage, design stage. Note here that when we talk of design stage we are referring to 2 types of design one is interface design one is system design so we have covered so far interface design stage.And with this lecture we have covered prototyping stage in subsequent lectures we will take up the other stages namely the quick evaluation of prototype and system design stage. So that will be the subject matter of our next lecture.

(Refer Slide Time: 51:06)



That is how to evaluate the prototypes that is the other important stage in the design prototype evaluates cycle which is typically early evaluation of the design idea. So that will cover in the next lecture whatever we have covered in this lecture.

(Refer Slide Time: 51:23)



Can be found in this book Human Computer Interaction User Centric Computing for Design you can refer to chapter 2 section 2.4.5. With that we have come to the end of this lecture I hope you enjoyed the lecture looking forward to see you again in the next lecture thank you and goodbye you.