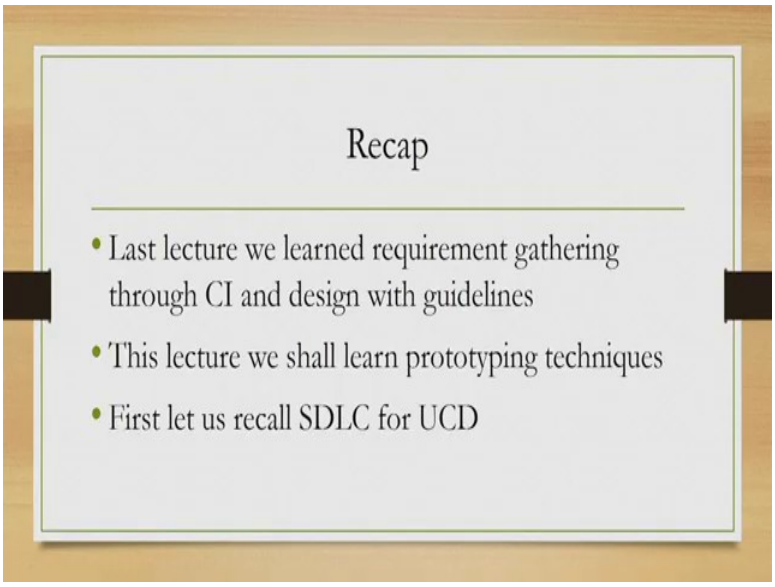
User-Centric Computing for Human-Computer Interaction
Prof. Samit Bhattacharya
Department of Computer Science & Engineering
Indian Institute of Technology, Guwahati

Lecture – 07
Components of SDLC – Prototyping

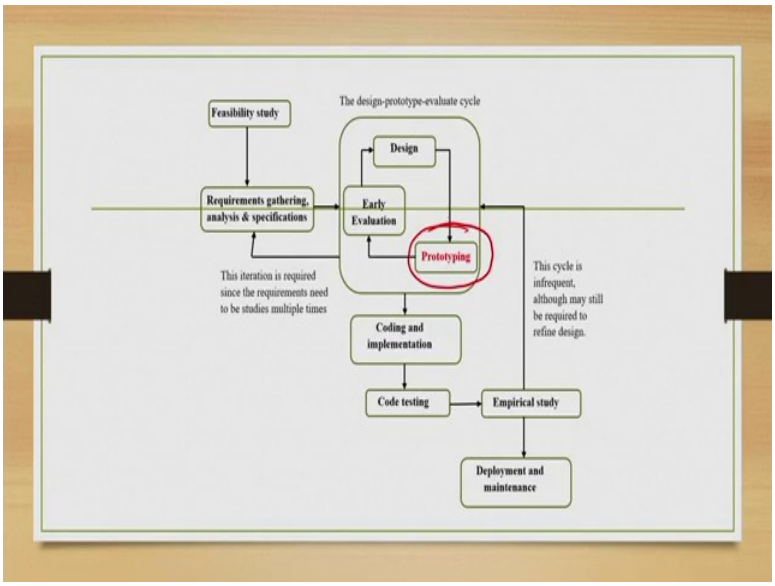
Hello and welcome to the course, User-Centric Computing for Human Computer interaction, the 7th lecture in the course. Now, before we proceed into the content of this lecture, let us briefly recollect, what we have learned in the previous lectures. So, we are discussing the interactive system development lifecycle and its stages.

(Refer Slide Time: 00:53)



So, earlier we have seen the iterative lifecycle and two of its stages, namely contextual inquiry for requirement gathering and design guidelines for coming up with early designs. Now, today what we are going to discuss is another of the stages, but before we go into that details, in case you have forgotten, let us recollect the iterative design lifecycle.

(Refer Slide Time: 01:29)



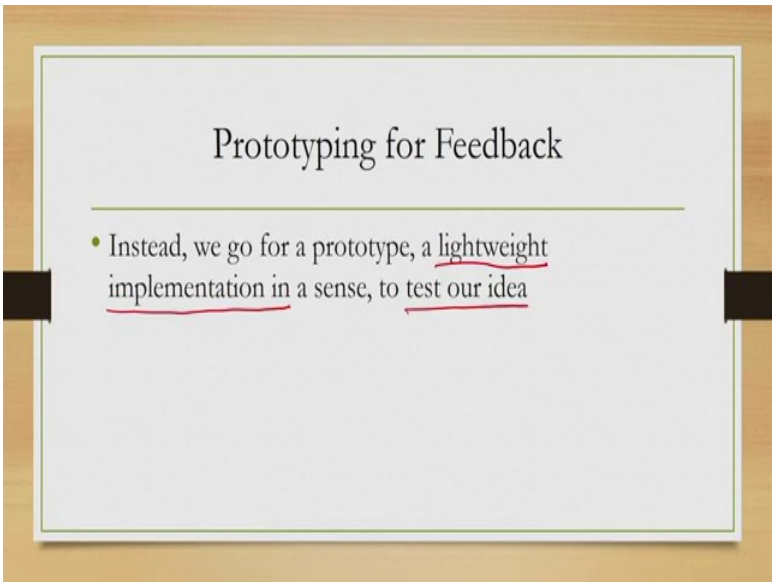
So, this is the life cycle, as you can see, the central iteration part is one of the most important part and here, we have already discussed about the two stages namely; the requirement gathering stage and the design guideline stage. There are few more stages which makes it different from other software developments, out of which we are going to discuss one more stage that is the prototyping stage.

This stage is very crucial in the development of an iterative system. In order to make the system usable, we need to continuously get feedback from the users during the development stage, and prototyping is one way of achieving that in a very efficient way.

So, when we talked about guidelines, the guidelines offer some starting point with the guidelines we can come up with a design, but we still do not know whether the design is going to be accepted, going to be usable; in order to do that we need to go for evaluation of the design. Now, there may be a common thinking that evaluation requires a full scale implementation of the design, but that may not be true always.

So, sometimes we can actually come up with a very lightweight implementation and get it evaluated, tested by some means to know whether this design is going to be usable to the end users. And, those lightweight implementations are technically called prototypes, prototypes of the end product.

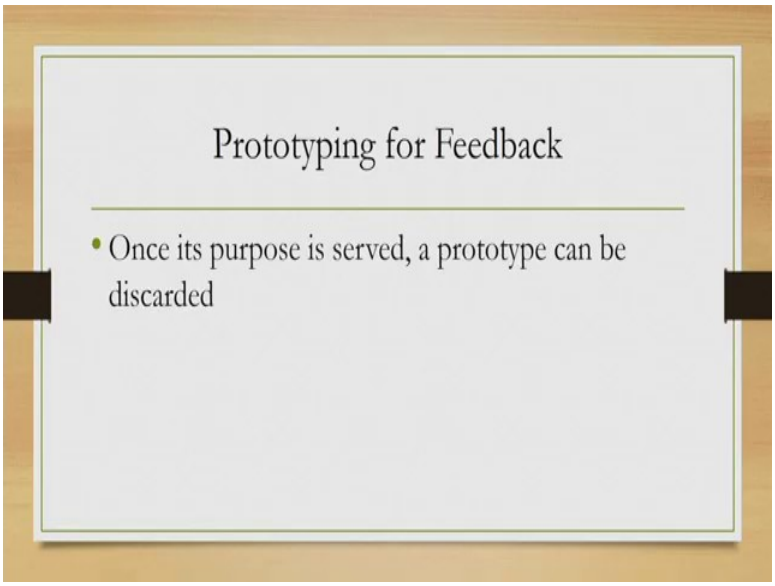
(Refer Slide Time: 03:05)



So, the crucial thing to remember is that prototypes are essentially lightweight implementations of the final product and these are required to test our idea. So, this is important and we should always remember this. Now, in user centric design as I already mentioned prototypes play a very important role, because it allows us to collect user feedback to evaluate our early ideas, design ideas, incorporating the feedbacks, without actually going for full scale implementation which is anyway as you know time and time consuming and requires a lot of effort and entails lots of monetary expenditure as well.

So, when we are talking about prototype, we actually mean that some early implementation of our design. So, in the early stages of the design, we go for implementation of prototypes and get it evaluated rather than going for a full scale system building exercise.

(Refer Slide Time: 04:18)



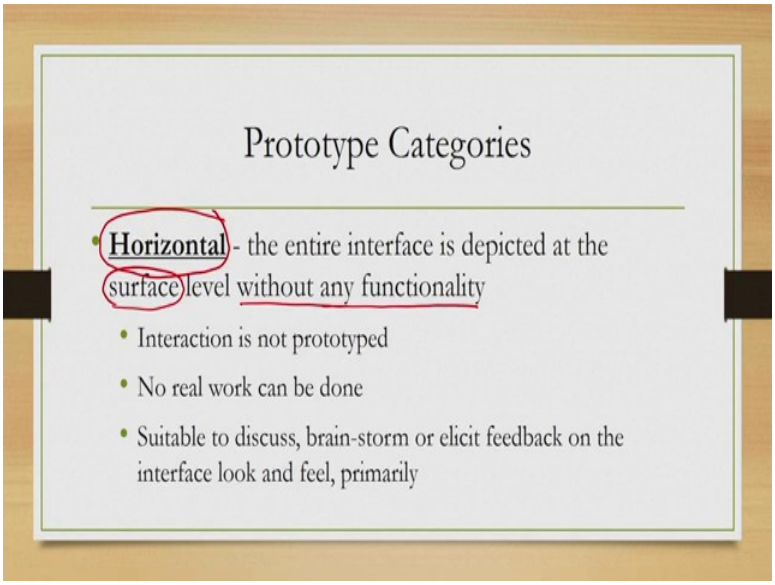
So, what to do with the prototype? Once it is built, once its purpose is served that is; once we are able to get user feedback on our design idea, using the prototype what we should do with it.

(Refer Slide Time: 04:32)

So, there are broadly two ways, one is we can keep on incrementally refining the prototype till it becomes a final product or fully implemented product, also we can actually discard it, after its purpose is served. So, both are possible and we will see in details later, the advantages and disadvantages of each of those. Now, before we talk

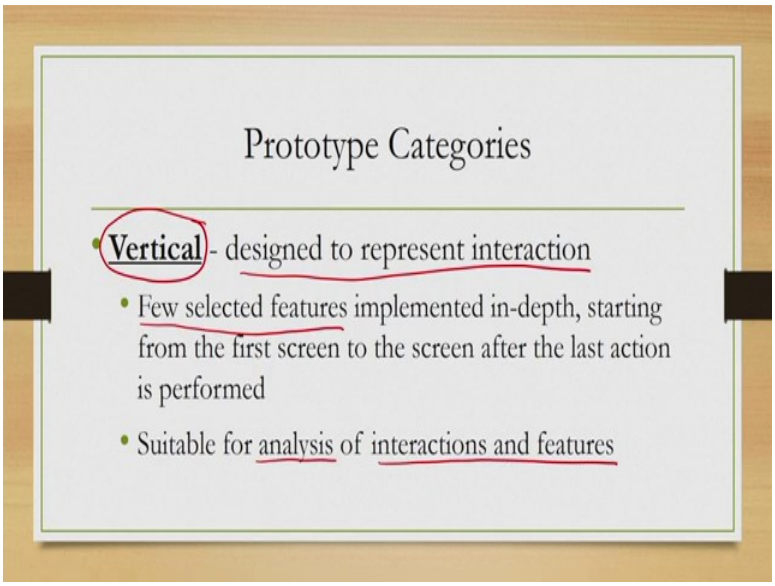
about how to create prototypes, let us see what kind of prototypes are there. Broadly, there are two categories; one is the horizontal prototype.

(Refer Slide Time: 05:06)



So, horizontal prototypes essentially referred to the implementation of an entire interface at a surface level. Essentially, there is no functionality implemented without any functionality and it is implemented at the surface level. So, when we are talking of a horizontal prototype, it is primarily meant to elicit feedback on the interface look and feel and we cannot actually implement with horizontal prototype any functionality, no real work can be done with this type of prototypes.

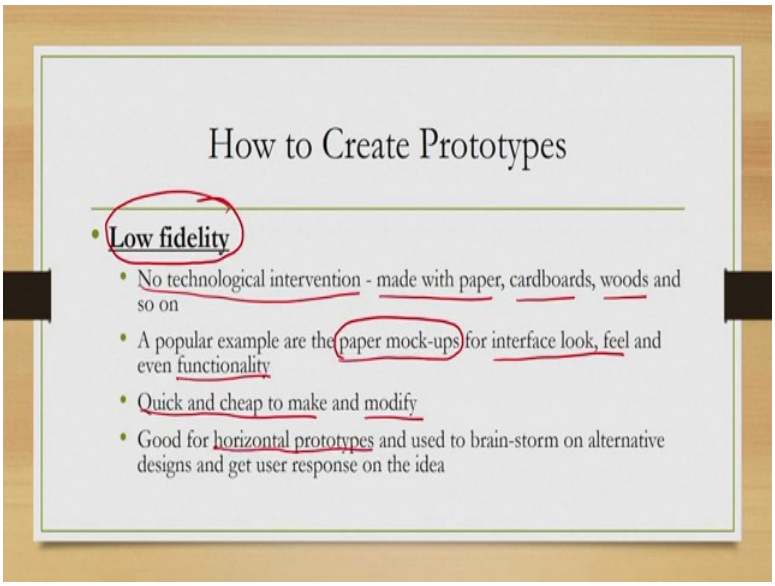
(Refer Slide Time: 05:46)



As compared to horizontal prototypes, we have also another class of prototypes known as vertical prototypes. So, in this prototype we can actually implement various features functionalities. So, essentially they are designed to represent the vertical prototypes are designed to represent interactions.

So, we can actually implement few selected features in the prototype and we can use that to analyze the interaction and different features; particularly, in terms of their usability whether they are usable or not. So, horizontal prototypes are surface level implementations, vertical prototypes or functional level implementations.

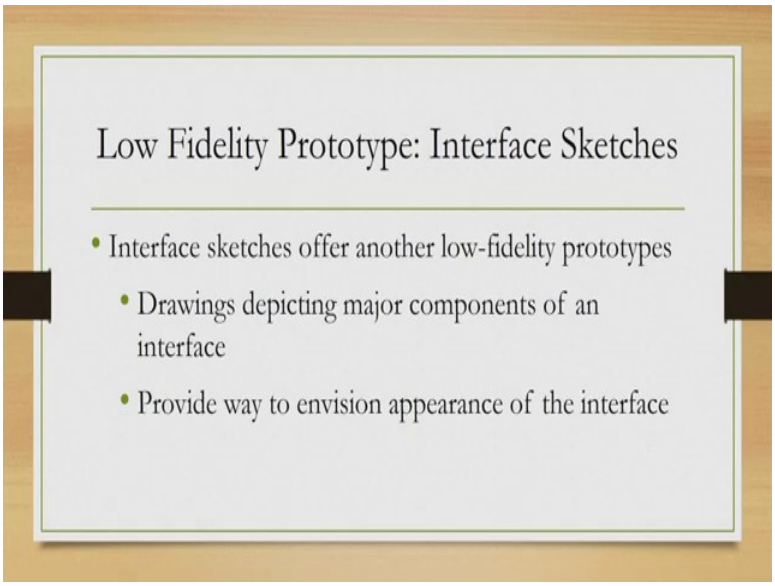
(Refer Slide Time: 06:36)



Now, once we are aware of these two types, let us briefly discuss how we can create these different types of implementations. So, the first category is low fidelity. So, when we are talking of a low fidelity implementation of a prototype essentially, what we refer to is that we are creating a prototype without the use of technology.

So, they are very quick and cheap to make and modify. There are no technological intervention for example, prototypes made with paper, cardboard, woods, these are termed as low fidelity, the popular example is a paper mockup, which you probably have seen that many places are probably you yourself have created to visualize your idea.

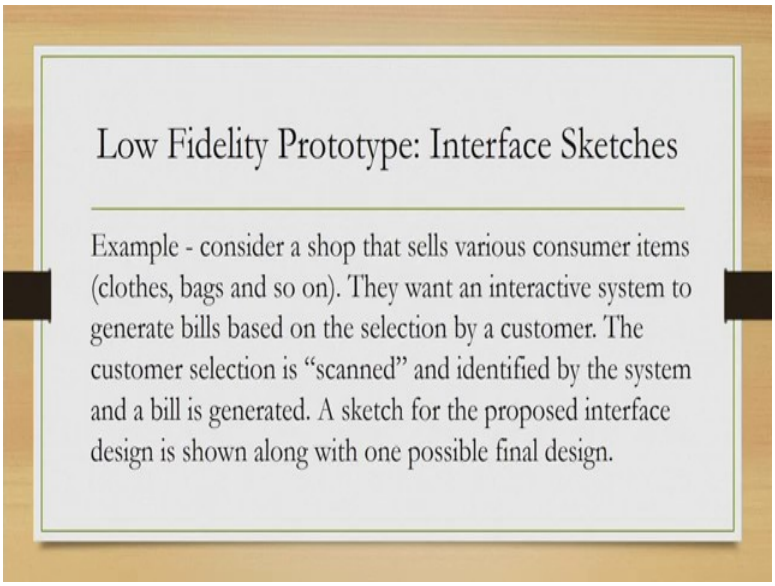
(Refer Slide Time: 07:52)



And they are good for implementing interface look, and feel and in a proper way if they are implemented, then using this low fidelity we can actually implement functionality as well, but mostly it is used for horizontal prototypes surface level implementation of the idea. Now, there are many low fidelity prototypes as I said, mostly used to represent surface level look and feel and appearance.

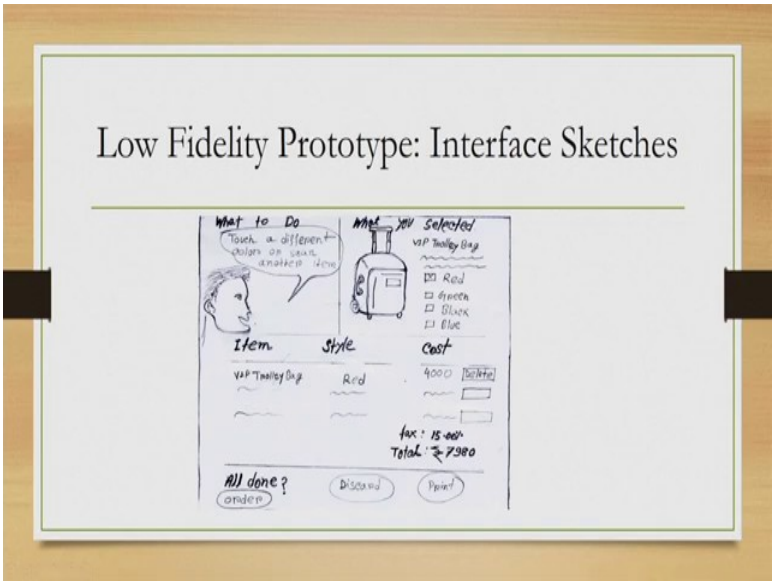
So, one very popular application of these prototypes are basically, to create interface sketches. So, these sketches are essentially used to convey the appearance of the interface at a particular instant of time and their purpose is to depict the major components of an interface rather than interaction. In order to understand this sketches, let us consider one example.

(Refer Slide Time: 08:39)



Suppose, there is a shop that sells various consumer items; clothes, bags and so on. Now, they want an interactive system to generate bills based on the selections by your customer or online purchase by your customer. The customer selection is scanned and identified by the system and the bill is generated, sketch for the proposed interface we can draw, which is shown in the next slide.

(Refer Slide Time: 09:06)



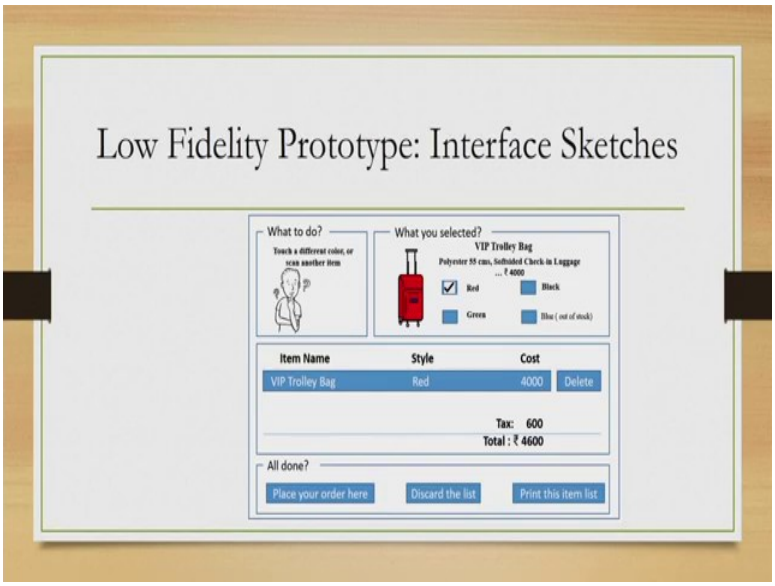
Now, let us see the sketch as you can see here. This is of course, one of the many possible ways to create the interface. So, here along with the items major items

checkboxes and other components some notes are also mentioned that what the user is supposed to do with this. Of course, they may or may not go, this note may or may not go in the final design, but this is for brainstorming purpose that, this is one of the possibility.

So, essentially the idea is that the customer selects some items and the item code is scanned and the bill is generated for the item. This is a very common application probably, you have seen it many shopping malls, shopping centers, shopping stores and we are asked to design an interface for the application.

So, here in one of these design, this sketch is generated where on the top right corner, you can see that the item is, the picture of the item is shown, then the color is mentioned, then the description of the item is there in the middle part followed by the price and then other options.

(Refer Slide Time: 10:23)

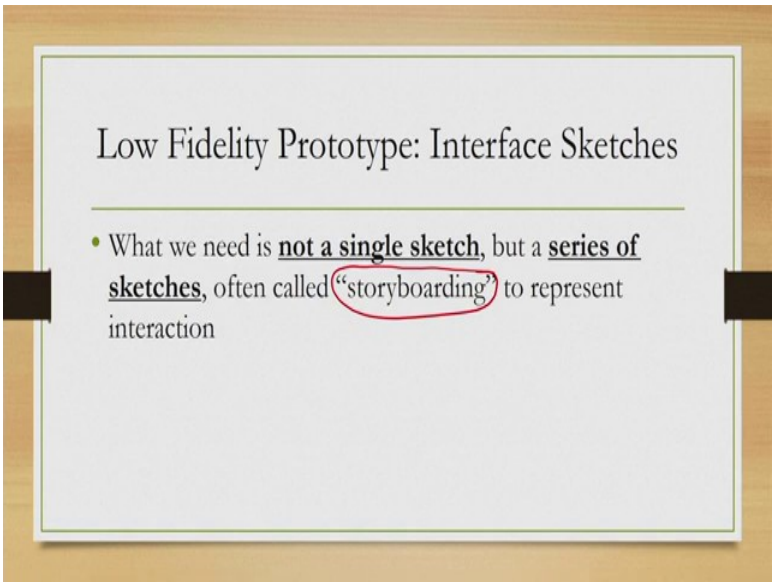


So, from this particular sketch, we can actually go for a final interface design which may look like this. As I said, so the left top corner is actually meant to guide the user of this interface what to do and other parts like top right corner, middle part, lower part are for showing different options to the user of this product of this interface.

So, from the sketch we can come to this design, but as we have already mentioned sketch is used for horizontal prototyping mostly. They can basically prototype surface level

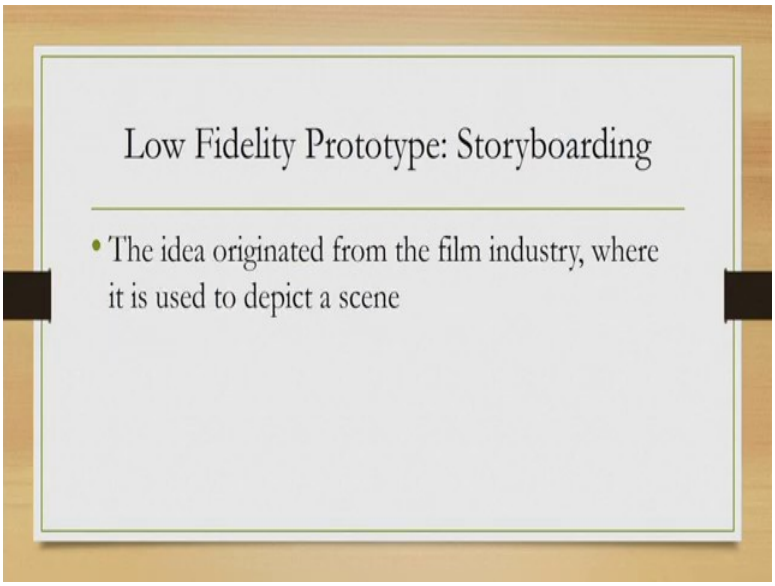
features, surface level elements, surface level organization of the interface. Now, sometimes with low fidelity prototype, we can also develop prototype for functionality. So, we can also implement functionalities, but probably it is obvious to you that with one sketch that is not possible.

(Refer Slide Time: 11:23)



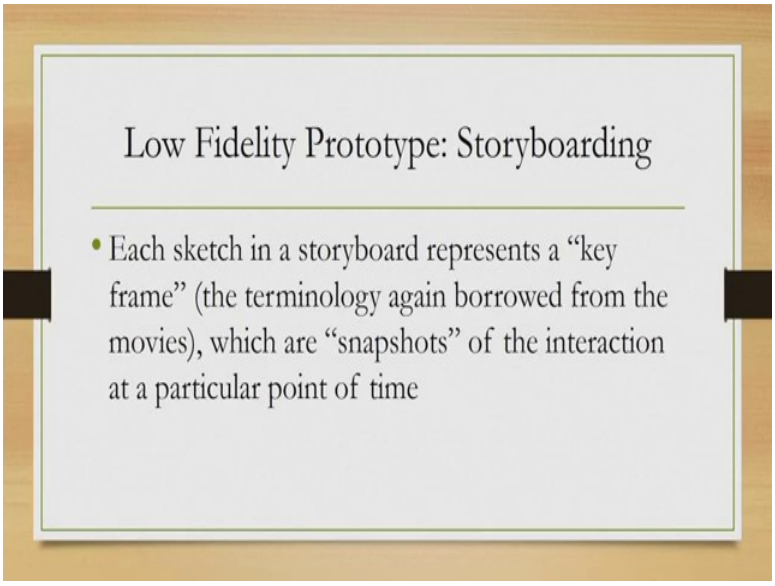
So, what we require is a series of sketches and that brings us to the other important type of prototypes, low fidelity prototypes, which is known as storyboarding. The term storyboarding actually refers to the implementation of a low fidelity prototype, where we have a series of sketches to depict particular functionality from the starting point to the end point.

(Refer Slide Time: 11:43)



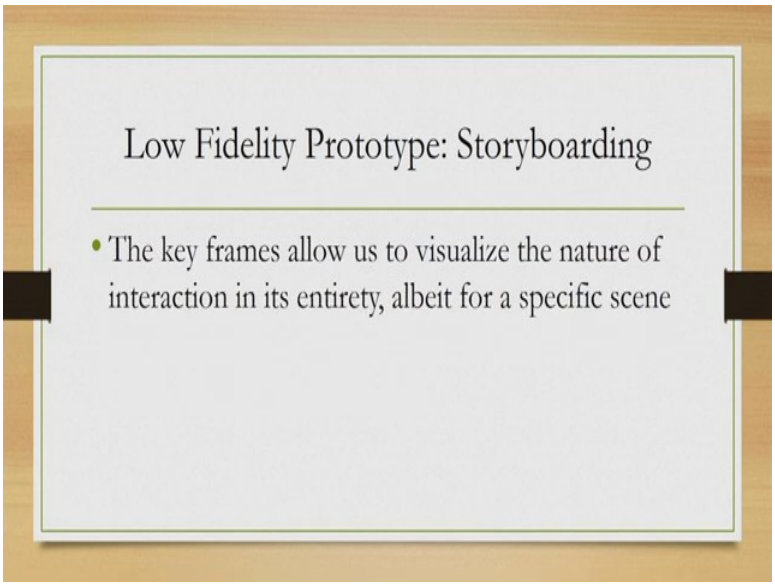
Now, the idea originated from the movie industry, film industry where this term is heavily used and each sketch in a storyboarding implementation is known as a key frame. And again, this term has been borrowed from the movie industry, where this key frame is a common concept.

(Refer Slide Time: 11:49)



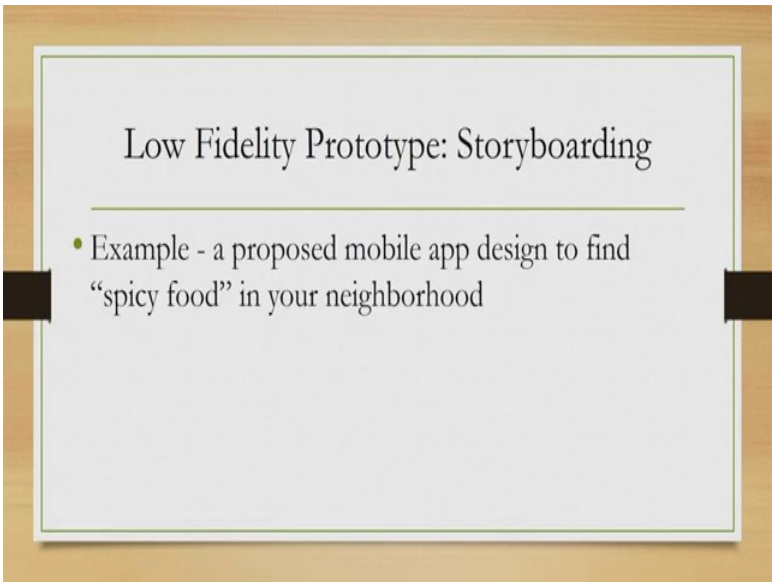
Essentially, it means a snapshot of the interaction at a particular instant of time.

(Refer Slide Time: 12:09)



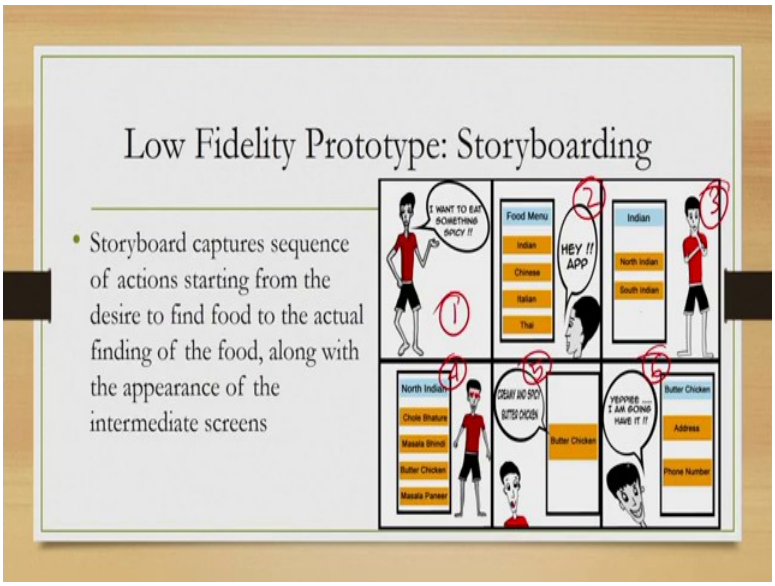
So, in the key frames, we can actually visualize the way the interaction takes place from a starting point, where we start performing a task to an end point, where the last sub task to accomplish the task has been performed.

(Refer Slide Time: 12:27)



As an example, consider the development of an app, you are asked to propose a mobile app that will help a user find out spicy food in his or her neighborhood.

(Refer Slide Time: 12:39)



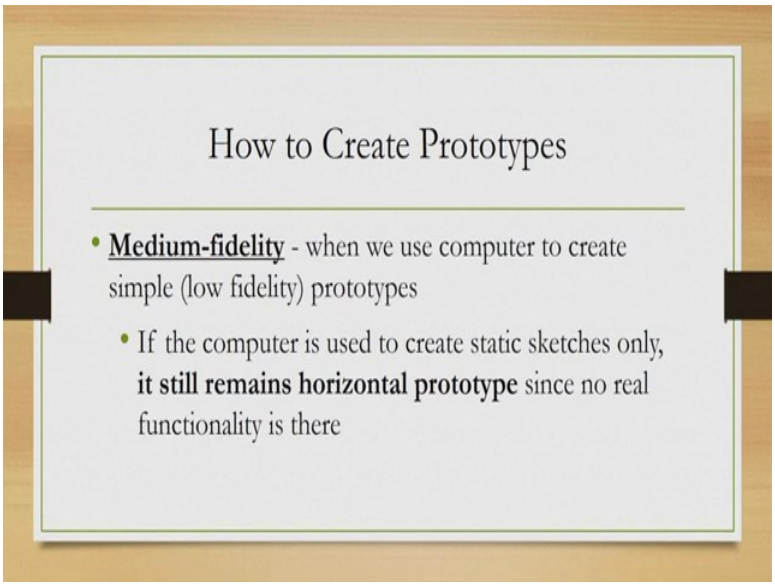
Now, look at this storyboard. So, here it consists of six sketches, each of these boxes represent one sketch; sketch number 1, then we can say 2, then 3 and so on, 4, 5 and 6. Each sketch depicts, what the interface is supposed to do at a particular instant of time.

So, first there is a menu item, list of menu items, the kind of food that are available, then category of the food and in this way if somebody wants to choose a spicy food, so he or she has to follow the menu options till he or she gets the food that is desired and this comment boxes may or may not be there of course, it is just to brainstorm the idea.

So, here in this example as you can see, it consists of six sketches, each sketch depict the interface date of the interface at a particular instant of time and together they depict. The entire sequence of interactions that are required to choose a particular food item and this is one of the possible designs for the app. Now, storyboarding or interface sketches can be created without the help of any technology.

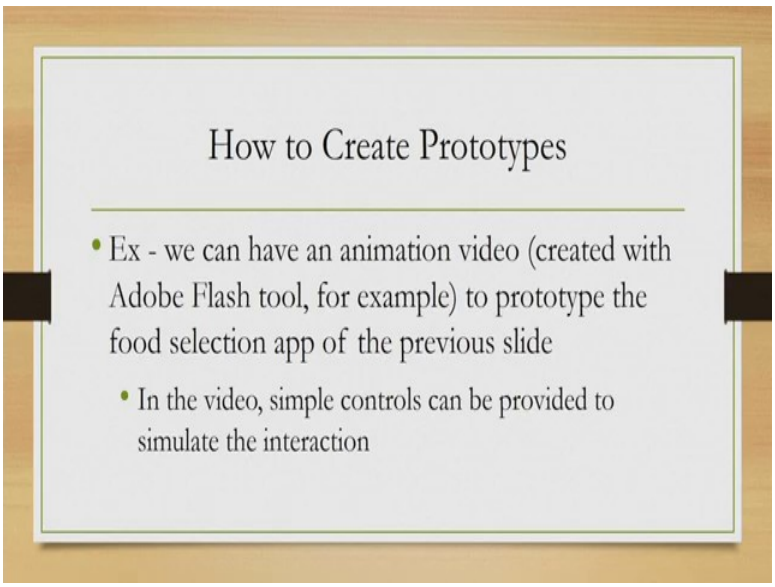
So, we can basically do this on a paper sketch, all these things arrange a sequence of sketches on paper to create a storyboard and so on.

(Refer Slide Time: 14:12)



So, when we are not using any technology then that is called low fidelity as we have already seen. Now, if we use some technology to create this sketches or storyboards or mock ups or such prototypes, then it becomes medium fidelity. Although, medium fidelity the term medium fidelity actually refers to the way it is created rather than the category. So, even if we are using a computer to create a drawing of an interface as an interface sketch it still remains horizontal prototype, because we are unable to do any interaction with it.

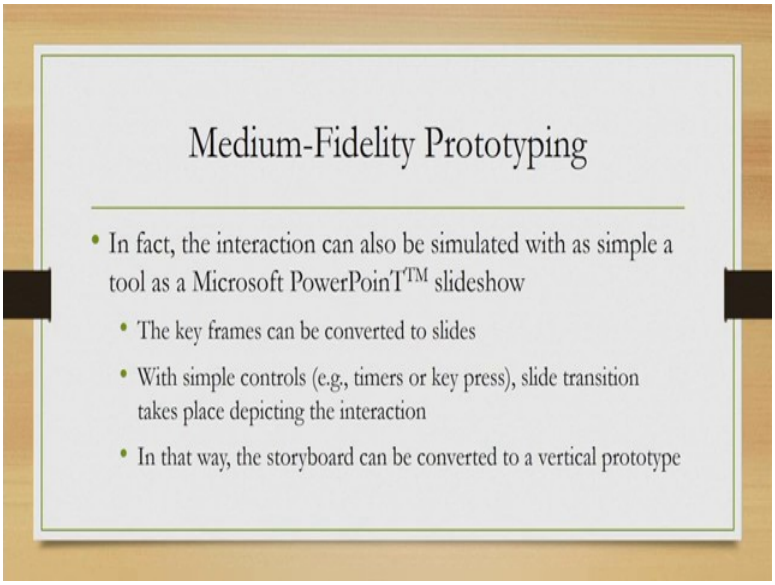
(Refer Slide Time: 14:50)



But using technology we can actually create vertical prototypes by artificially inserting some controls between the sketches when we have a storyboard implemented with a computer.

Say for example, you have created an animation video of the sketches appearing one after another and the transition between these key frames are controlled with the press of a button, which we have integrated in the animation video. So, then it becomes a vertical prototype as if the interaction takes place on the real app.

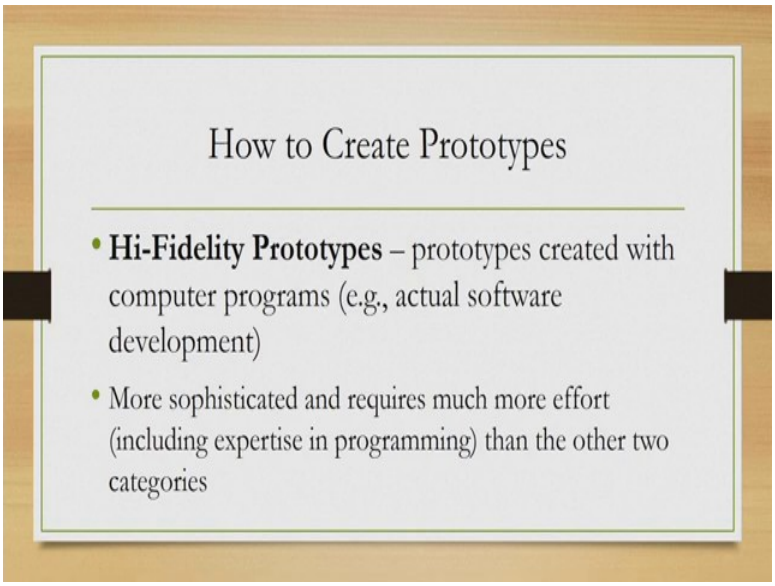
(Refer Slide Time: 15:30)



Similar things can be done with say Microsoft PowerPoint slides also. The slides transitions can be controlled with the press of a button or a key and then it may give you a feeling of a vertical prototype.

So, when we are using technology to represent a prototype or to create a prototype, it is easier to basically implement vertical prototypes with the help of technology, even if we are sticking to the basic concepts of storyboarding and sketches. And, we still do not need to actually implement the system, actually implement say the app or the other programs that are required to run the app.

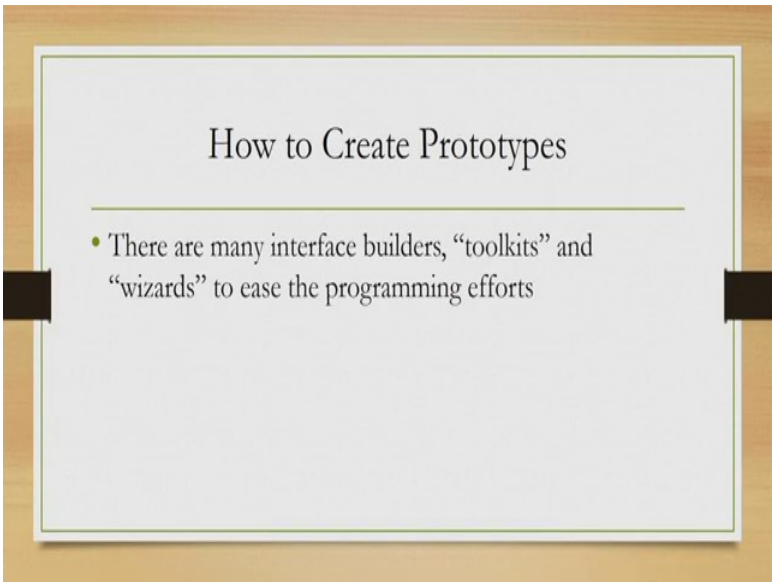
(Refer Slide Time: 16:14)



Finally, we have high fidelity prototypes. So, earlier low fidelity or medium fidelity prototypes do not actually require you to implement the program, implement the actual computer programs. Here, in high fidelity prototypes we typically refer to the actual programs with this term high fidelity prototype.

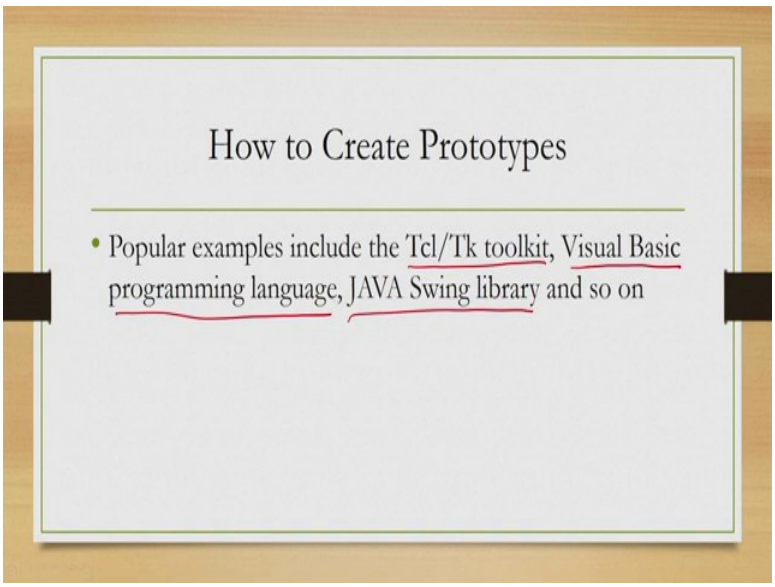
So, we are actually building the software system and getting it evaluated. So, the prototype itself is the software. Clearly, it is more sophisticated and it requires much more expertise than creation of a low or medium fidelity prototype. So, essentially you need to be a programmer to be able to implement a high fidelity prototype.

(Refer Slide Time: 17:04)



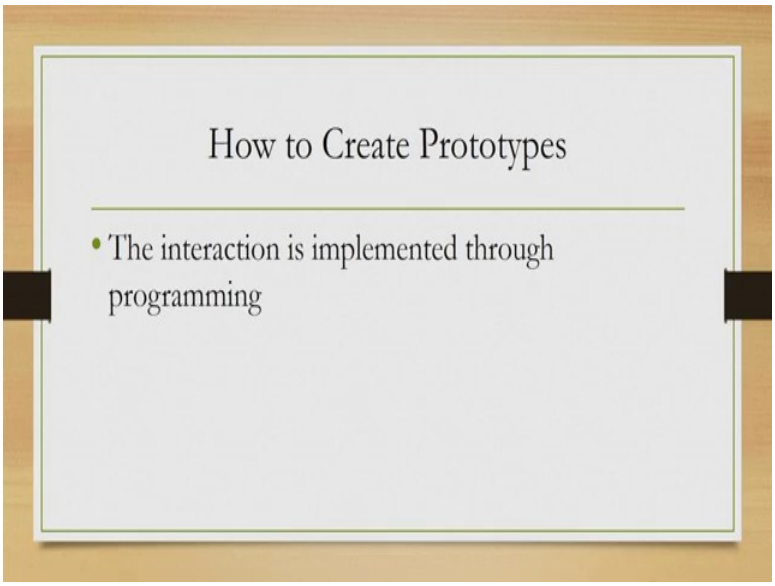
But there are of course, tools and supports available to help a programmer build a prototype particularly, in the form of tool kits or wizards to easier programming effort.

(Refer Slide Time: 17:16)



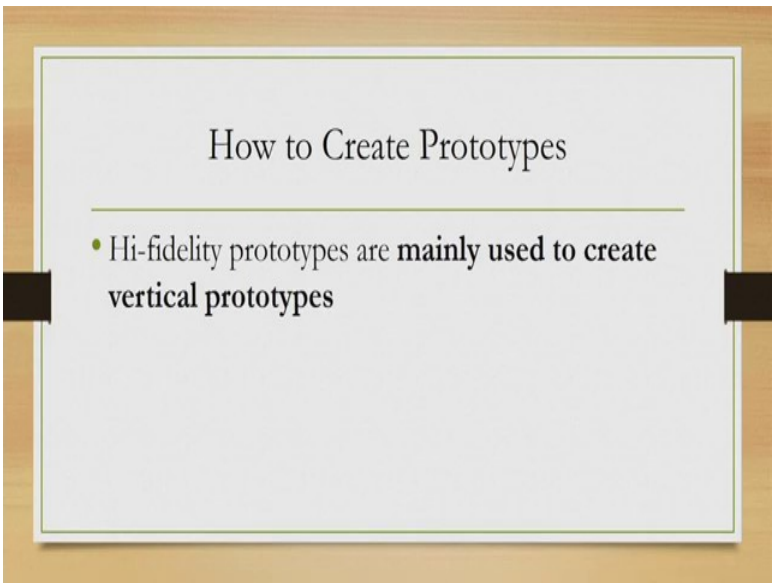
Many search tool kits are available like Tcl, Tk toolkit, Visual Basic programming language, JAVA Swing Library and so on. So, these are essentially used to help you create interfaces and control interfaces the way you want. They are very helpful in creating high fidelity prototype with much less effort and in much less time, if you have basic knowledge of these toolkits and programming.

(Refer Slide Time: 17:46)



Now, the other important concept or the other important thing that we should know is what happens with the prototype.

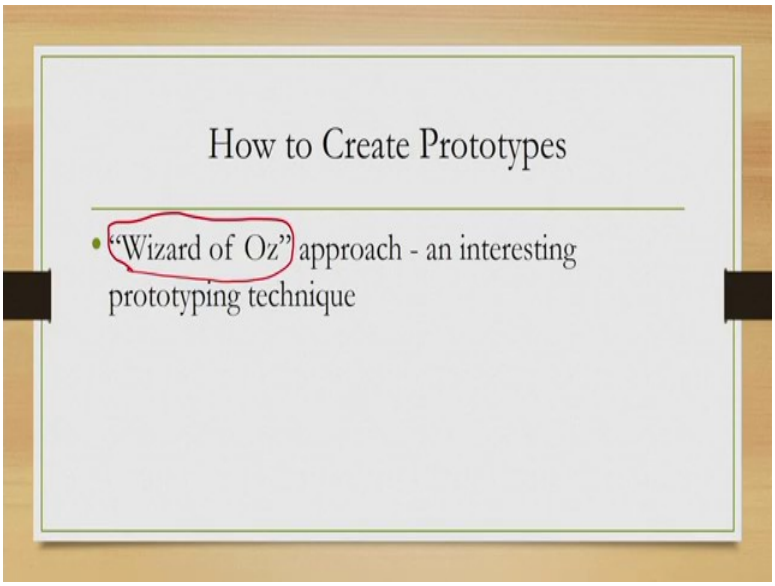
(Refer Slide Time: 17:53)



So, high fidelity prototypes are mainly used to create vertical prototypes. As I said, in low fidelity or medium fidelity, if we are talking of only sketches or paper mockups or such prototypes, then they are mostly horizontal prototypes. They are useful for creating surface level features.

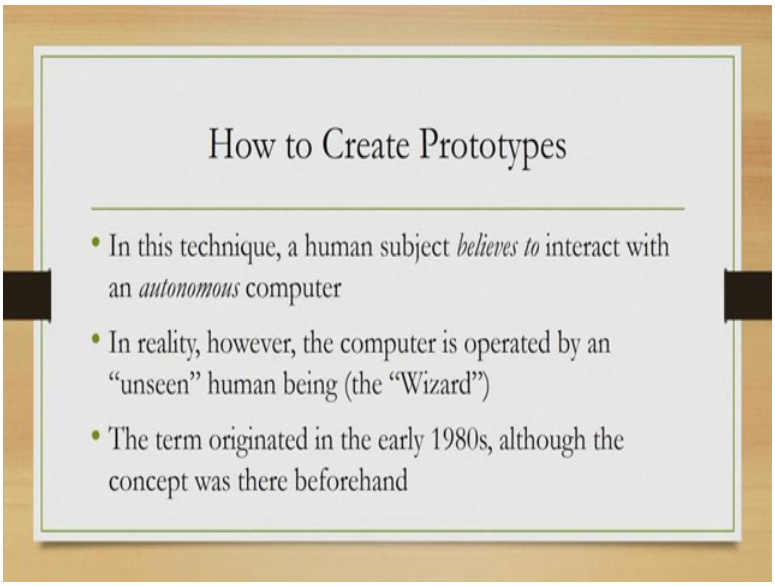
When we are talking of storyboarding we are gradually moving towards implementing features that a user is expected to encounter or expected to have in the final product and in high fidelity prototype our objective is to implement features, because programs allow us to do that.

(Refer Slide Time: 18:38)



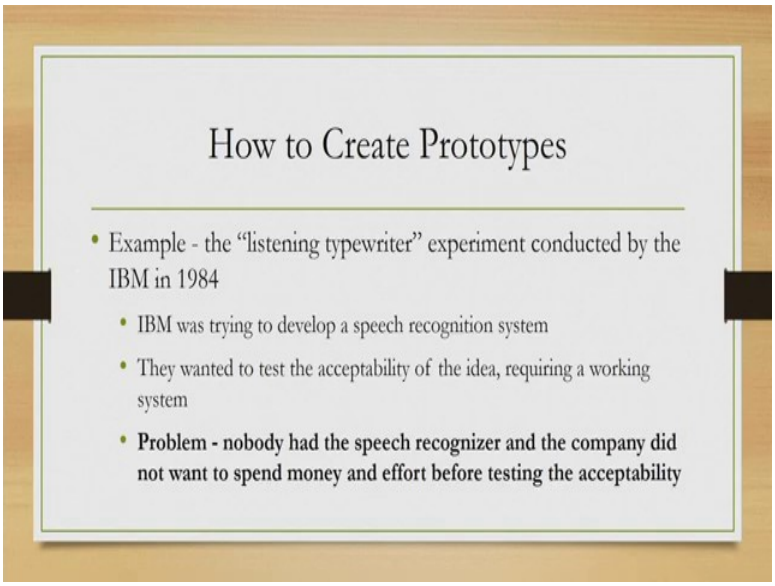
Now, apart from these three broad ways to create prototype low, high and medium fidelity, there is an interesting way no known as the wizard of approach of creation of prototype. So, these actually, this particular approach does not belong to any of the three as such. It is difficult to place this approach into high, low or medium ways to create prototype.

(Refer Slide Time: 19:08)



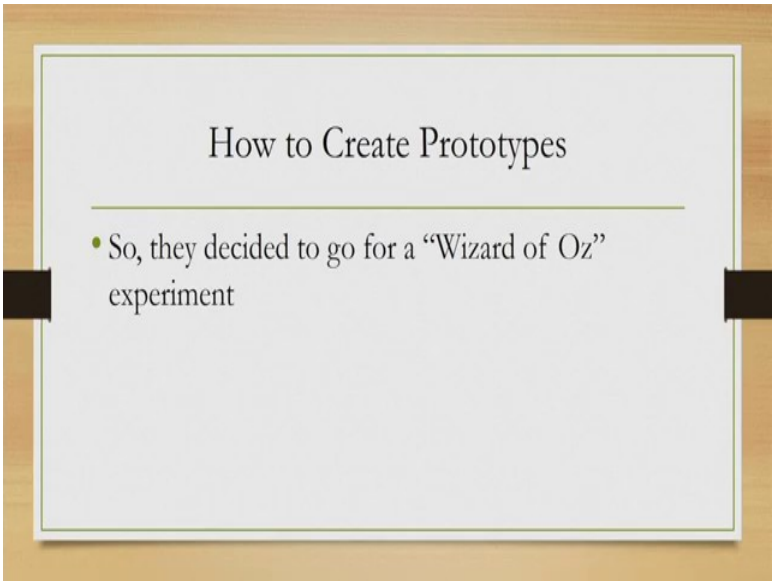
So, in this technique a human subject or typically the user believes that he or she is interacting with a interactive system with an autonomous computer having an interface. But, in reality what happens is that, there is no actual computer instead, there is one person who is hidden from the view of the user, who is behaving in a way such that the user gets the feeling of interacting with an actual computer. So, the hidden person is called the wizard and although the term originated in the 1980s, the basic idea was that they are different early in the 1980s, it started finding applications in the development of interactive systems.

(Refer Slide Time: 19:59)



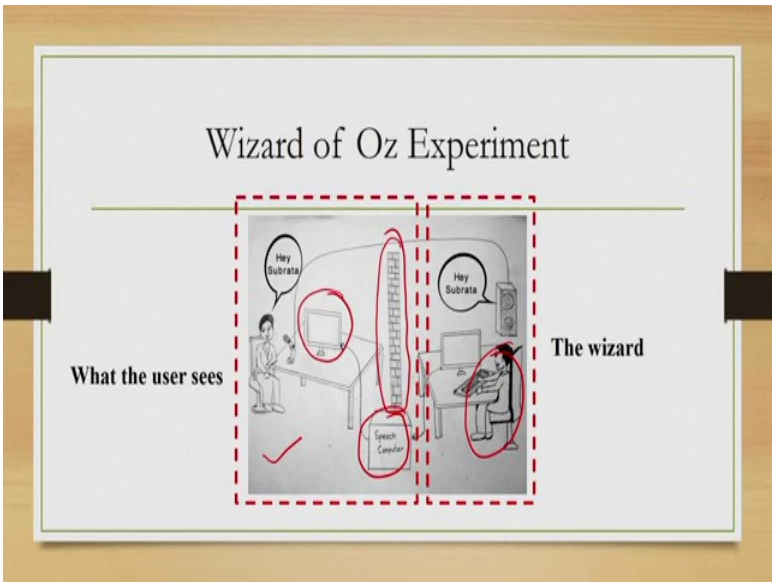
And one of the most well known application of Wizard of Oz approach is the listening typewriter experiment, conducted by IBM in 1984. So, at that time IBM was trying to develop a speech enabled keyboard that is a speech recognition system where you will speak out and the system will recognize what you are saying and accordingly it will display the text. So, essentially speech to text conversion system or speech recognition system. Now, at that time of course, there is no speech recognition system, but the IBM wanted to test their idea although, the technology was not available.

(Refer Slide Time: 20:43)



So, what they adopted is that Wizard of Oz approach of prototyping this system.

(Refer Slide Time: 20:50)



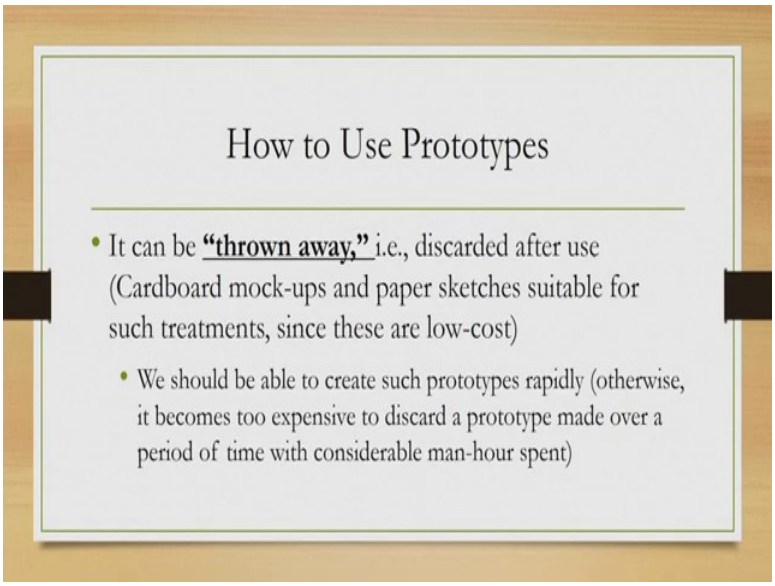
So, in this approach, as you can see in the figure, this is the setup used where on the left side, there is the user and the user in front of the user there is a microphone to provide input and there is a terminal where the user can get to see the output. Now, this terminal was connected to a computer system behind a wall. So, there is a wall.

Now, behind this wall one operator was sitting, who was operating this computer system? So, what happens is that whenever the user is saying something, the operator gets to hear that through the speaker system, which is connected to the microphone of the user and immediately, the operator types that and the typed text is displayed on the terminal of the user.

So, here the user feels that whatever he said is actually getting recognized by the computer, this is the computer system visible to the user, user feels that this computer converts whatever he says to text and displays it on the screen. But, in reality what is happening is that a type, there is an operator who is actually typing it and that is getting displayed on the screen. So, this operator is the wizard and who is, whose presence is not known to the user.

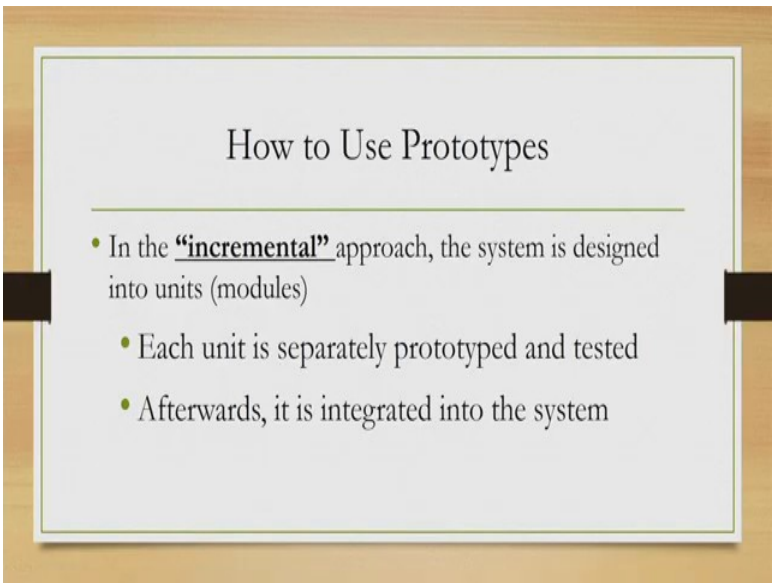
So, user feels the system is doing it, but in reality and human operator is doing that. So, essentially the idea of this experiment is that if you do not have a technology, but you can still think of ways to prototype your system and get it evaluated.

(Refer Slide Time: 22:36)



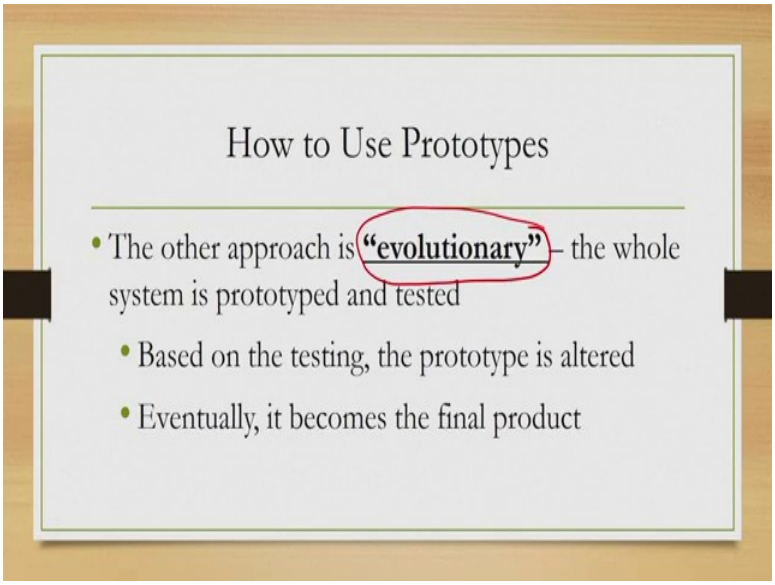
So, the other concern is what to do with the prototypes It can be thrown away. So, if we are talking of low fidelity paper mock ups or sketches on paper. Of course, they are meant for brainstorming in a group you sit down, sit together, brainstorm and then you throw it away, because they cannot be incorporated in the final product, which is software. The precondition of creating through away prototypes is of course, as we can intuitively understand that they should be easy to build and cheap. They should not be very expensive otherwise, throwing them away will be problematic.

(Refer Slide Time: 23:16)



Next comes the incremental approach, here the system is divided into units, for each unit there is a prototype. The prototype is tested and afterwards this unit level prototype is integrated into the system. So, essentially separate modules or units are prototype tested and integrated together to build the system incrementally and the third approach is evolutionary.

(Refer Slide Time: 23:42)



So, in this approach the whole system, we start prototyping the whole system, at the beginning maybe with one feature then we get it tested, then add one more feature, get it tested and so on. So, the there is a single system unlike the incremental approach which keeps on getting modified till we evolve to the final system. So, that is in summary, what are the types of prototypes? How to build them and what we can do with it?

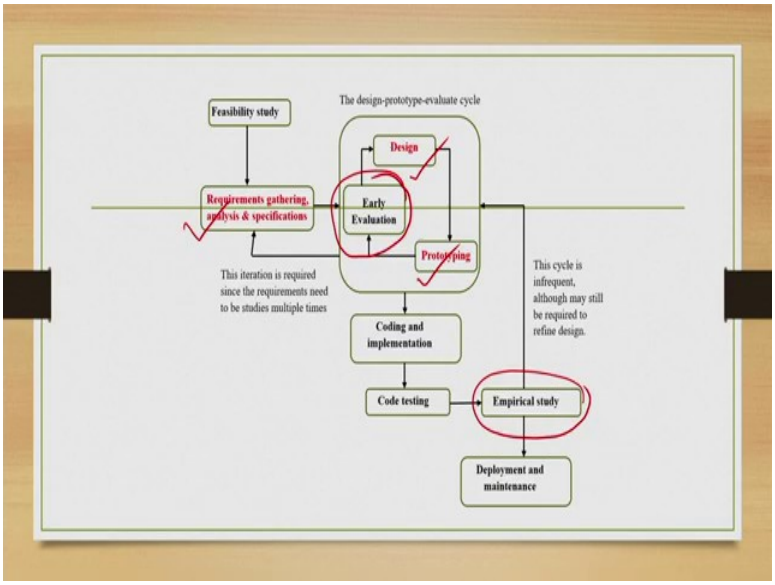
So, just to recollect so, there are broadly two category of prototypes; one is horizontal where only surface level features our prototype and other one is vertical where we try to create interactions, try to implement features of the system for evaluation. We can build prototypes in four ways; low fidelity where we do not use technology, medium fidelity where we use technological interventions such as the use of computers or animation packages or powerpoints and so on.

And high fidelity where we actually go for implementing the product, using programming languages, using programming tool kits and wizards plus we have this Wizard of Oz approach, where we actually can prototype something for which

technology is still not available, but we want to test the idea. So, there we can actually create a setup for implementing a Wizard of Oz prototype.

And the prototypes either we can throw away, if they are cheap and quick easy to build or we can keep on testing them at unit level of the whole system and then integrate them to design the whole system that is the incremental approach. Or, we can follow an evolutionary approach, where we can build the whole system from the very beginning and keep on enhancing it, keep on adding features to it, at the same time testing it to evolve to the final product.

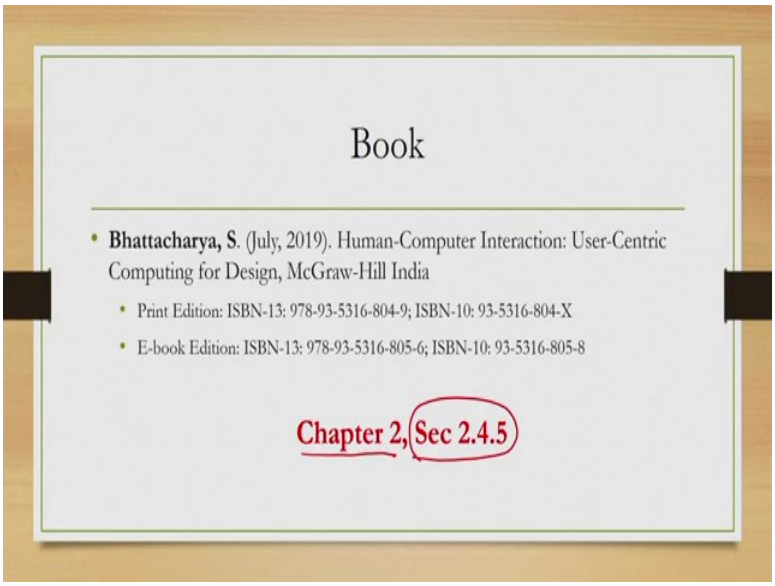
(Refer Slide Time: 25:55)



So far we have discussed a few of the major components of the software development lifecycle. To recollect three components, we have discussed in the previous lectures and today's lecture.

So, they are the requirement analysis, stage through contextual inquiry, design with guidelines and prototyping. There are two more which are unique in this interactive system development lifecycle; one is the part of central cycle that is the early evolution of prototypes, there are different techniques available to do that. So, we will discuss that in a subsequent lecture and empirical study where we go for user studies, which is a very complex process and which we will discuss in a future lecture.

(Refer Slide Time: 26:49)



The materials that we have discussed today are taken from section 2.4.5 of chapter 2 of this book. So, we are advice to go through the sixth chapter of the book to know in more details about prototypes as well as the relevant references, which you can use for further studies.

Thank you and goodbye.