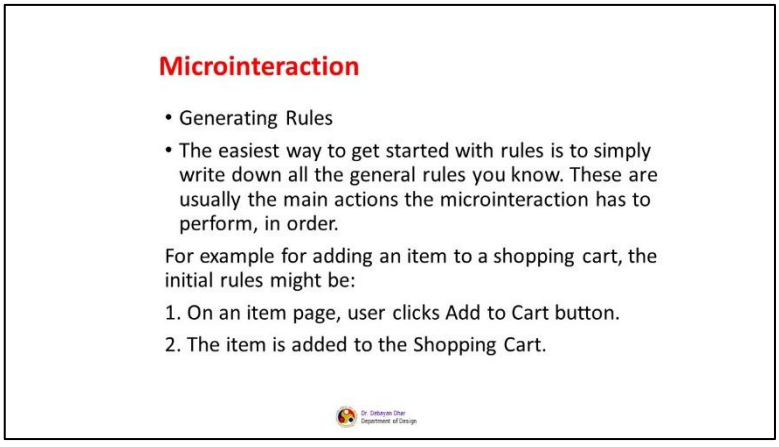
Usability Engineering
Prof. Debayan Dhar
Department of Design
Indian Institute of Technology, Guwahati

Module - 10
Lecture - 33
Conceptualization and Prototyping II

Welcome to module 10, lecture number 33. We will continue discussion on design for micro interactions in this session as well.

(Refer Slide Time: 00:46)

(Refer Slide Time: 00:48)



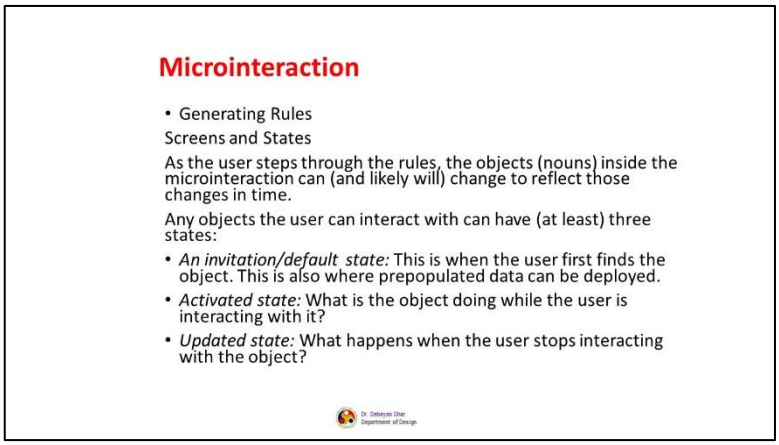
We have been discussing about a rule and we will continue discussion on how we can generate rules. Now the easiest way to get started with rules is to simply write down all the general rules you know, rules means how does that function executes in terms of the

decision making that happens. Now, these are usually the main actions in the micro interaction domain that we are discussing about and these actions has to be performed in an order.

Now take an example for adding an item to a shopping cart, the initial rules might be you know you open an item page and while you are there in the item page the user clicks Add to Cart button. He selects a product and then clicks Add to Cart button and the item is then added to the Shopping Cart. So, what you see is an activity and behind that activity there is decision making wherein the user inputs is being recognized by the system as events for deciding.

Or these events are being registered and then the rule starts actuating so that the data points can be activated, data points can be registered and then the entire systemic perspective can be activated.

(Refer Slide Time: 02:18)



Now, while we talk about generating rules, the screens that your user see provides a vital important piece of information for them to understand and assess the state of the system. Now, as the user steps through the rules, the objects inside the micro interactions which we often call as nouns can change; that means, the state change, in order to reflect the changes that are happening in time.

Now, these changes do communicate about their state to your user, these are vital sources and piece of information. So, any objects the user can interact with can have at least three states through which we perceive or understand your user perceive or understand the

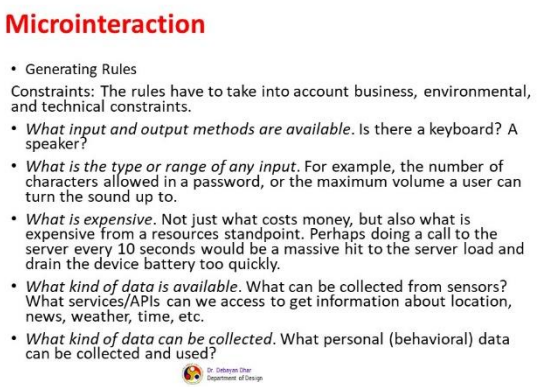
information. And these states are an invitation or the default stage; that means, this is when the user first finds the object and this is also where pre-populated data can be deployed.

The second one is the activated state. Now, this is the state what is the object doing the user is interacting with right. So, the second state is called the activated state. What is the object doing while the user is interacting with it? So, during the process of interaction the object that is being used to activate a particular situation to activate a particular event on action, how does that happen?

What information is being transferred to the user to communicate that state and the third one is the updated state; that means, what happens when the user stops interacting with the object. So, once that activity or function is performed, what happens after that? What is the state that is being observed by the user so that he or she can comprehend about the situation? That ok this is I have completed my activity, now probably this is going to happen or now I need to move on to a next stage of action.

These are vital pieces of information that the user needs to understand in order to ensure that the micro interaction that he is performing can be completed and the goal can be reached.

(Refer Slide Time: 04:56)



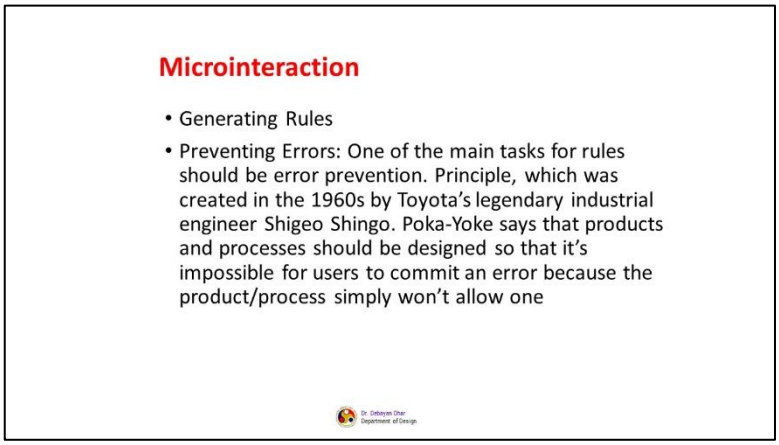
Now, there are a lot of constraints while designing for the rules as well. And constraints mean what? It means the rules have to consider the business, environmental, and technical constraints. There are a lot of constraints in the situations for example, if you look at the questions that I have listed here in the slide you would be able to understand that in more close detail.

What input and output methods are available? See, so based on this modality your entire structure of defining the rules would depend. So, is there a keyboard? A speaker? What is the type or range of any input? For example, the number of characters allowed in a password or maximum volume a user can turn the sound up to, these are constraints right.

And these influences the way you are designing your rule. What is expensive? Not just what costs money, but also what is expensive from a resource stand point of view. Perhaps you know doing a call to the server every 10 seconds, I mean this is an example that I am quoting would be a massive hit to the server load and drain the device battery too quickly.

Then what kind of data is available? What can be collected from sensors? What services or APIs can we access to get information about location, news, weather, time, etcetera? What kind of data can be collected now these are very very crucial pieces of information that defines how these rules are being designed or generated, what personal behavioral data can be collected and used. These are some of the ways constraints that influences how your rules are being structured, your rules are being defined and modified over a period of time.

(Refer Slide Time: 07:08)

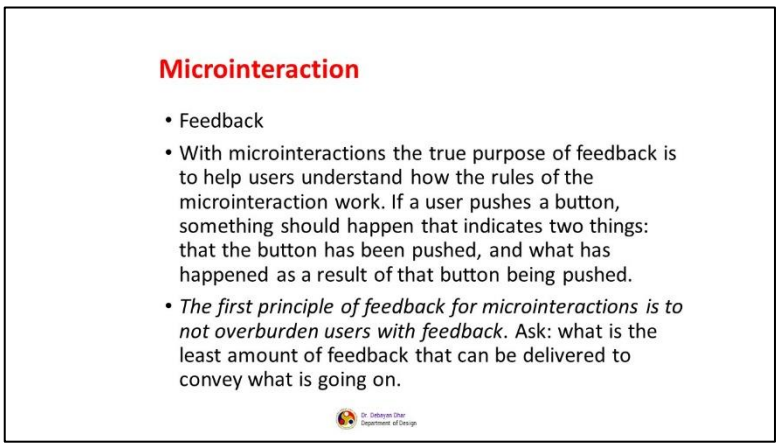


Now, one of the important aspects, during generation of rules while you are designing the rules or defining the structure of the rules is to ensure that errors are prevented. See, one of the main tasks for rules should be error prevention. So, principle which was created in 1960s, by Toyota's legendary industrial engineer Shigeo Shingo, you know Poka-Yoke says that products and processes should be designed so that it is impossible for users to commit an error, because the product or process simply will not allow one.

Now, in order to ensure that errors are being handled to ensure, see errors need to be handled in a way that your users will not be able to commit an error, this is a far-fetched dream or a ideal situation that we are talking about. While in usability, we also consider that eventually there might be some situations, wherein the users will have or commit some errors.

Now, in those situations what is important for you as a designer to undertake is to design for ways through which he can recover for errors. So, in the next module where we will be finally, discussing about the various usability evaluation ways, techniques, tools and also some experiment evolution techniques. We will learn about that in a more detailed way.

(Refer Slide Time: 08:49)



After rules, we now we will discuss about feedback. Now, what is feedback? See when micro interactions, with micro interactions the true purpose of feedback is to help users understand how the rules of the system or the micro interaction work. Now, this means what? This means there has to be some communication of information from you; that means, from the interface, from the visible information that is available in front of him in terms of the screen, so that he can understand how the rules are being governing his activity, his communication dialogue with the system.

So, take an example of this, if a user pushes a button, something should happen that should indicate two things: one that the button has been pushed, the user must get a feedback. That the button, has been pushed and what has happened as a result of that button been pushed right, these are very very vital source of information. The ability of your user to

recognize, the state of the system to recognize what he or she is doing with the system depends on how you design these feedback mechanisms efficiently.

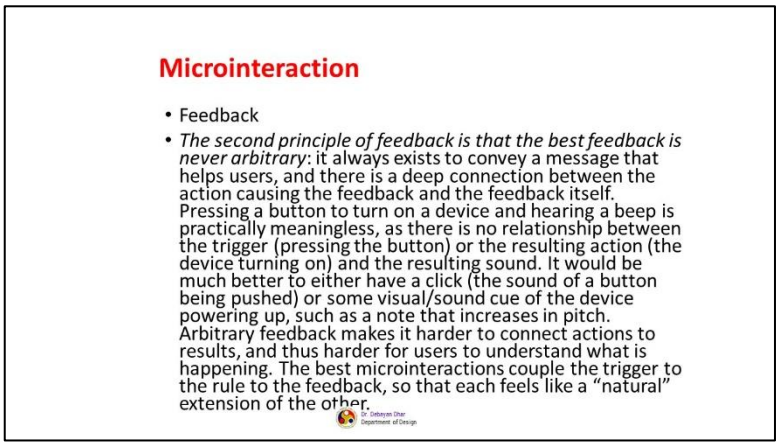
Now, why do I mean, what do I mean by when I use the word efficiently? I mean at that instant of communication which is happening the interaction that is happening there should be effective communication of information about the state, about this state of the system to the user so that the user can understand ok, whatever activity I am doing there is a response from the system.

And this response tells me that my input has been registered. You know that is why feedback is so important. The first principle of feedback for micro interactions is not to overburden users with feedback. So, this is also very crucial, you must understand and ask what is the least amount of feedback that can be delivered to convey what is going on. See you have a limited space in the screen, it can be a mobile screen it can be a laptop or desktop screen, it can be a tablet screen or it can be anything projected screen.

What is important here for us is to know that we have a limited space. And in this limited space we have to convey this information to our user. So, we cannot just let on bombard him with a lot many amounts of information, because that would eventually lead on to cognitive overload. I am using the word overload, because cognitive load as you know to some extent is helpful for execution of tasks and decision making. But then, if it is overload it would hamper the decision-making process.

Remember what we discussed during Hicks law, how different sources of information, how different options that you provide may increase certain point of time and the certain moment the cognitive burden on the user. So, therefore, the first principle while designing this feedback during your micro interaction phase is to ensure that you do not overburden your users, numbers of feedbacks it should be just and appropriate.

(Refer Slide Time: 12:40)

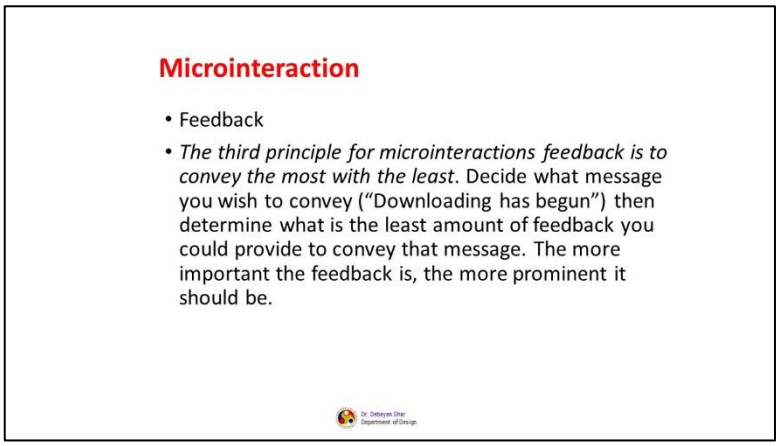


The second principle of feedback is that the best feedback is never arbitrary. Now, what does that mean? It means that it always exists to convey a message that help users, and there is a deep connection between the action causing the feedback and the feedback itself. You know take us let us take an example, say pressing a button to turn on a device and hearing a beep is practically meaningless, is not it.

As there is no relationship between the trigger which is pressing the button and the resulting action which is the device turning on and the resulting sound, there is no relationship between them, if you put a beep. So, it would be much better to either have a click the sound of a button instead of a beep or some visual or sound based cues of the device powering up, such as a node that increases in pitch.

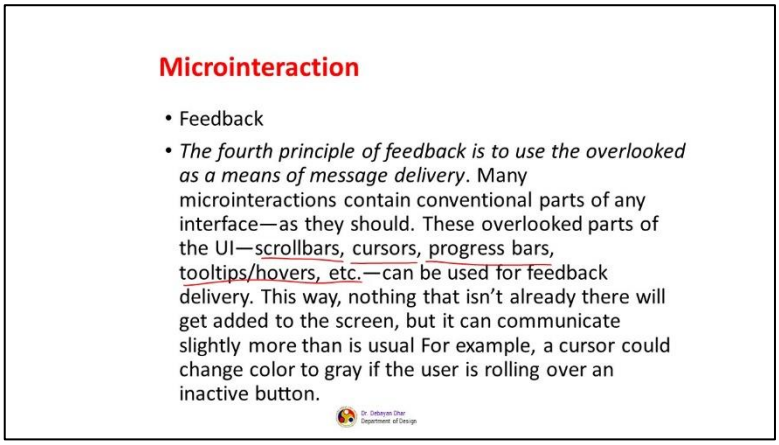
So, this is where we mean by arbitrary feedback. So, arbitrary feedback makes it harder for your user to connect actions to results and thus harder for your users to understand what is happening. The best micro interactions couple the trigger to the rule of the feedback so that each feels like a natural extension of the other.

(Refer Slide Time: 14:15)



The third principle of feedback is to convey the most with the least. That means, decide what message you wish to convey. For example, downloading has begun, then determine what is the least amount of feedback you could provide to convey that message. It can be visual, it can be sound based, the more important the feedback is the more prominent the visual cues your ability to convey the message should be more prominent.

(Refer Slide Time: 14:53)

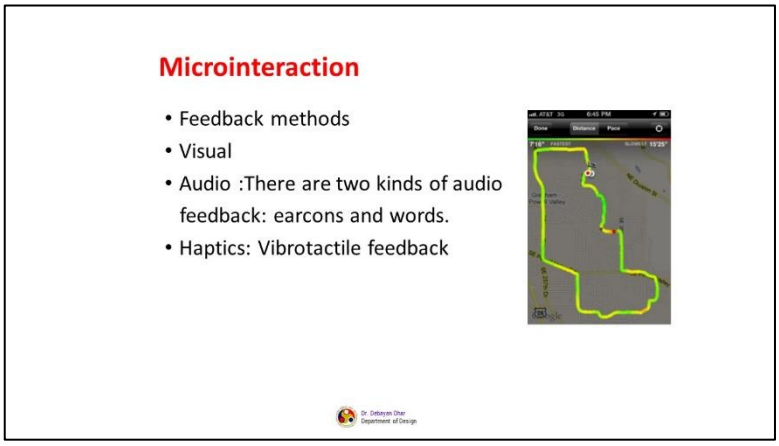


The fourth principle of feedback is to use the overlooked as a means of message delivery. What do we mean by overlooked? By overlooked we mean some of the features which are often taken for granted. For example, you know many micro interactions contain conventional parts of interface like these overlooked parts are like the parts of the user interface, such as the scrollbars, the cursors, the progress bars, the tooltips, hovers etcetera.

And these important interface features can be used to communicate feedback delivery, this way nothing that is not already there will be will get added to the screen, but it can communicate slightly more than is usual. For example, a cursor could change color to grey if the user is rolling over an active button you know. So, you are not adding something new in order to communicate a message,

What else you are doing? You are using the already overlooked elements of the UI like this, the scrollbars, the cursors, the progress bars to communicate the feedback. That is what in the fourth principle of feedback we talked about, overlooked items in the UI which can be used as a means to deliver the message or to deliver the message, the feedback message.

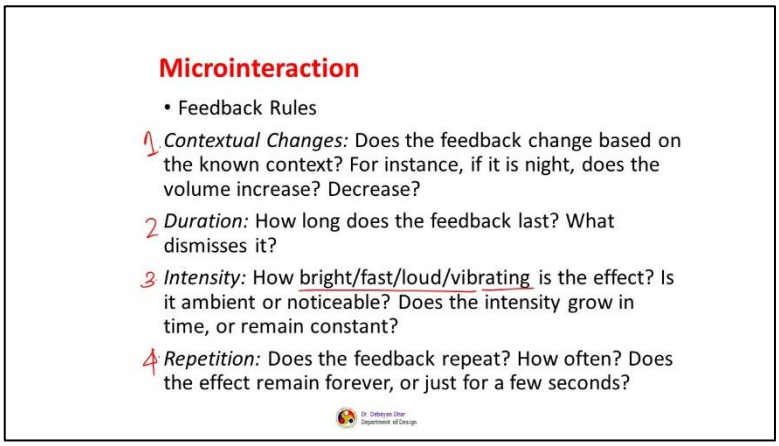
(Refer Slide Time: 16:30)



Now, what are the various feedback methods? See there are ideally three feedback methods. It can be visual, like you can see in the slide. It can be visual feedback, it can be audio feedback. In audio feedback we have two different modes. One is called earcons and the other one is words. Now, you have heard about icons, icons are visible in nature you can see through eye, earcons means they are through sound. So, by through ear you can perceive this sound and therefore, they are called earcons.

Words means words that are spoken words, which you can understand which you can hear and then comprehend the meaning. Haptics means vibrotactile feedback. So, there are motors which vibrates and that vibrating sound as well as the vibration, nature of the vibration communicates that message through as a feedback means right.

(Refer Slide Time: 17:31)



Now, there are some of the rules that govern how do you design this feedback and these are contextual changes, duration, intensity and reputation; contextual changes, duration, intensity and reputation. What do you mean by contextual changes? Contextual changes mean context means where the work is being performed. Does the feedback change base on the known context?

For instance, if it is night, does the volume increase or decrease? Does the screen display changes according to it? Duration: how long does the feedback last? What dismisses it? These are very important crucial pieces of information that you need to decide while designing the feedback rules. Intensity: how bright, see these are characteristics of the feedback modes right. Be it visual or tactile right or through sound, bright fast loud vibrating is the effect.

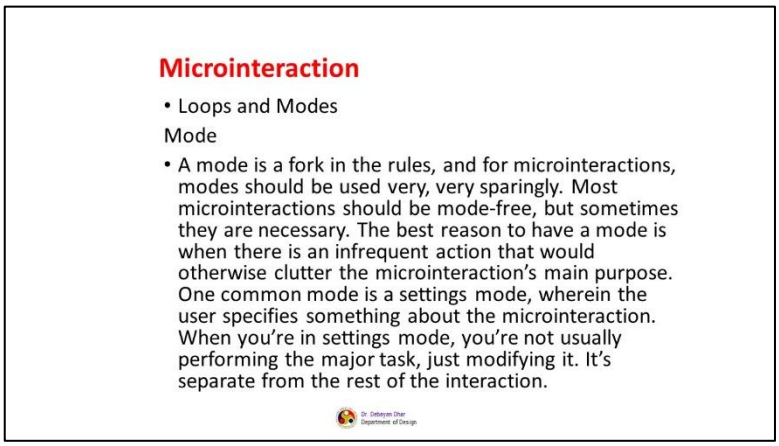
How bright fast loud or vibrating is the effect? Is it ambient or noticeable? Does the intensity grow in time or remain constant? Then reputation: does the feedback repeat? How often? If it repeats, how often? Does the effect remain forever or just for a few seconds? So, see why we are discussing all these things, we are discussing all these things to ensure that the level of designing those interactions plays a very very crucial role in ensuring, whether the user is going to enjoy the moment or it is going to be a disgrace for him.

I am using the word disgrace because the mental model and the conceptual model of the interaction does not match at all. And you will see majority of the time it does not matches because the concept of micro interactions or the basic structure and the basic rules of micro interactions have not been followed. By following these, rules you ensure that the

engagement between the user and the system turns out to be a delightful one, because every time the user starts interacting with the system there is a response.

There is a response that communicates the state of the system, which allows the user to comprehend that whatever he wants the system to do there is some kind of change in that state and something is happening based on which he is waiting to gather the outcome of it.

(Refer Slide Time: 20:39)



The last part of micro interaction is the loops and modes. So, when we talk about mode you might have understood or heard about of more in a different way, we talk about in which mode you are working, in which mode you are selecting a particular feature. So, a mode is a fork in the rules, it is kind of fork and for micro interactions mode should be used very very sparingly, not every time you are going to use a mode.

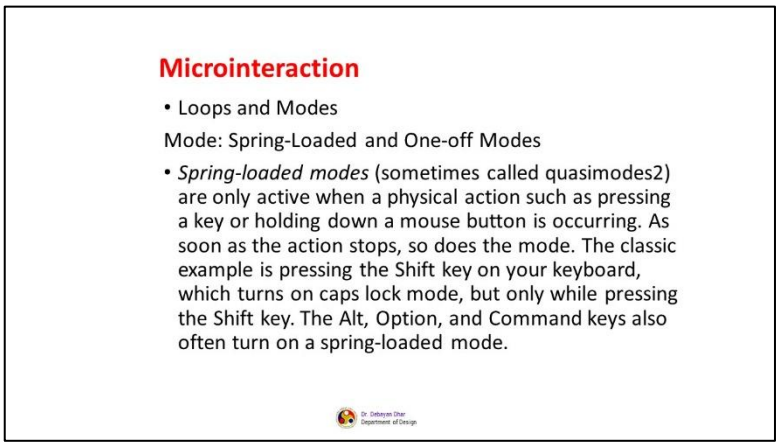
Most micro interactions should be mode free, but sometimes they are necessary, the best reason to have a mode is when there is a infrequent action. That means, there are some actions which are not very frequent in nature and that would otherwise clutter the micro interactions main purpose. So, one common mode is this settings mode, which you see in your mobile device or in your laptop or in your system, wherein the user specifies something about the micro interactions.

So, this is not a frequent mode. So, just remember when was the last time or when was the time when you have used the settings mode. The settings mode is a overall structure, overall a phenomena which allows you to decide how the micro interactions are going to

perform. This is a meta rule kind of stuff, that allows the system to understand your choices and preferences.

So, when you are in settings mode, you are not usually performing the major task. You are just modifying, manipulating. It is separate from the rest of the interaction, you it is a kind of a meta state where you are telling the system that I prefer this, this, this kind of ways through which the system should respond right, that is called a mode.

(Refer Slide Time: 22:37)

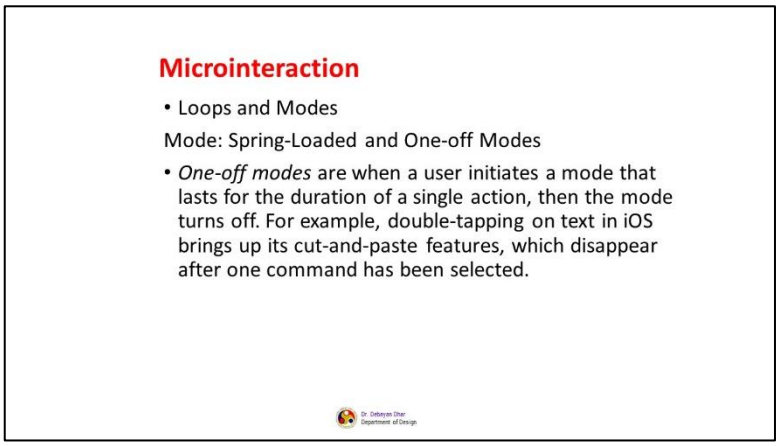


Now, there are two types of modes, one is called spring loaded and the one-off modes. What do we mean by spring loaded modes? It is also called as quasi modes too sometime. Now, these spring-loaded modes are only active when a physical action such as pressing a key or holding down a mouse button is occurring. As soon as the action stops, so does the mode also stop.

The classic example is the shift key on your keyboard, the moment you press the shift key and press some alternate keys you would see the action being activated, being performed as you and as and when you release that key the mode stops. So, the classic example is the shift key on the keyboard which turns on the caps lock mode, but only while pressing the shift key, this is called spring loaded modes.

So, the Alt, Option, and the Command keys also often turn on a spring-loaded mode. This is called forced one, you force its activated then as it is released the mode turns off. That is an example of a spring-loaded modes.

(Refer Slide Time: 23:54)

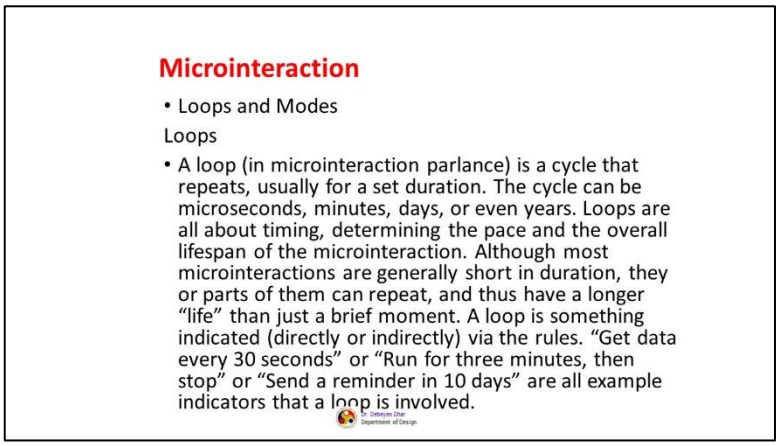


The other type of modes that we have is called the one-off modes. Now, what are they? One off mode is when a user initiates a mode that lasts for the duration of a single action, then the mode turns off. For example, double tapping on a text in your iOS operating systems. What it does is that it brings up its cut and paste features which disappear after one command has been selected.

The same thing happens with your screen in your mobile phone, you continue touching a particular text for a long period of time, you can see the copy and paste and the copy function appearing. You can select that one and then use the text copied in the clip file, in the clipboard and then you can paste it somewhere else right, these are one off modes.

So, the two different types of modes that we have discussed is the spring-loaded modes, which get only activated when a particular action is performed and it gets quickly wiped out the moment that action is taken off. The second one is one off mode which is based on the task completion. So, once that task is complete distriggered, the task is completed it is done.

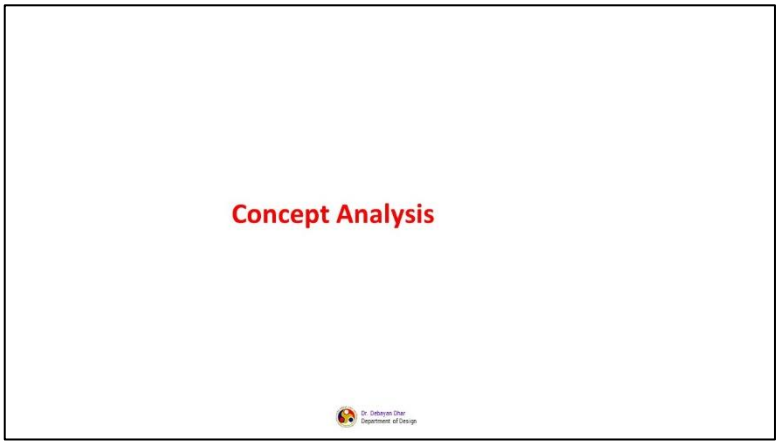
(Refer Slide Time: 25:13)



The next part and the final one is the loops. Now, a loop is a cycle that repeats, you know in the concept of micro introduction parlance. We call a loop a something that repeats, it is usually for a set duration. So, the cycle can be microseconds, minutes, days or even years. Now, loops are all about timing, determining the pace and the overall life span of the micro interaction. Although, now most micro interactions are generally short in duration they are or parts of them can repeat and thus can have a longer life than just being a brief moment.

So, a loop is something indicated directly or indirectly via the rules. Some example, "get data every 30 seconds" or "run for 3 minutes, then stop" and then "send the reminder in 10 days" these are all examples or indicators that a loop is involved.

(Refer Slide Time: 26:24)



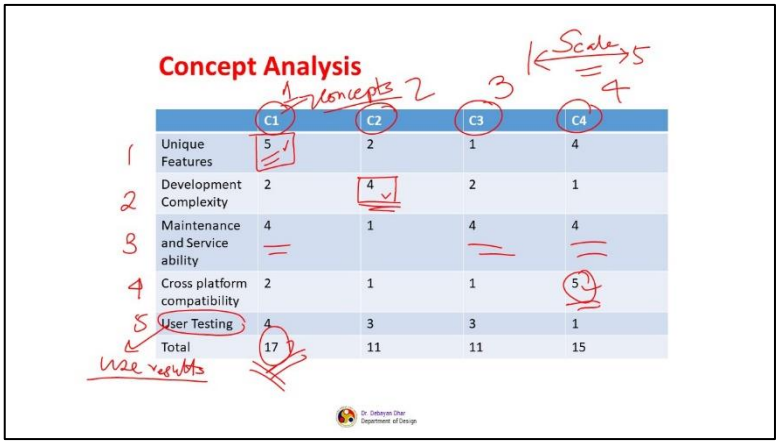
So, this brings us to the end of how we design for micro interactions. We are now in a stage where we have explained or discussed about various ideation techniques. And now, we see we started with the various ideation techniques in order to come up with large quantities of ideas. Then we also discussed about micro interaction techniques because that would allow you to go deeper, take up one idea and go deeper into designing that idea into a more detailed structure.

So, when you have these kinds of detailed concepts, few concepts now is the time where we need to select one ok. We need to select one concept and move forward, but this is not an easy task, there are many ways through which you can do it. There are some groups that say that ok, let us user test all these concepts and based on the outcome or the results that we get from our users' preferences we will decide which concept should we go ahead with. This is one of the most known technique that is used by designers.

There are other ways through which it can be done. Other ways include that ok testing is done then the users comment commentaries of individual features or functions of each individual concepts are being recorded. And then the best features of all the concepts are combined extracted and printed a new concept is, the concept is created that is also an accepted one ok.

Now, in cases where you do not have users or you do not have the resource to go to your users and go for a quick testing, in that case you know the team itself identifies some parameters and then test the concepts out. So, we will discuss about that in a very quick way.

(Refer Slide Time: 28:35)



What you see in this slide is a table that enlists some of the parameters these are random parameters which I have listed here which are of importance to the team. For example, based on your requirement based on your design brief, I am sure you are now in a position to understand which are the important characteristics or which are the important parameters that should govern the design of your ideas. And this is also partly influenced from your market study, by this time you are pretty sure about the main features that should be used to govern or analyze your concepts.

So, according to me I have just randomly put down these features what you see on the table, like whether the features are absolutely unique or novel, whether the concepts would be complex in developments. So, complex procedures, complex coding technological constraints are there or not. Maintenance and serviceability: so, once that product is developed what is the complexity or how easy it is to maintain the system and service it right.

Cross platform compatibility how well it is structured in a way that it can communicate across platforms and devices in order to ensure that multimodal communication happens um. And finally, we can also talk about have data from user testing or we can also think from the perspective of the scenario and the user that we have decided and then we can randomly give scores to them.

So, what we have, what you see here in the table is this that these are the concepts ok. So, these are the concepts that are being stacked, so concept 1, concept 2, concept 3, concept 4. So, these are the concepts that are being stacked against each other and now what you have are the characteristics which you want to use to evaluate them. Now, when I say that them, it means the design team. I am not going to the users right now rather I am focusing on the design team.

Now, you have these factors which you want to use to assess the concepts. What you do is that you ask your design team to evaluate each of the concepts taking a step back, not to be biased and then ensure that how would the rate each of the concepts in terms of its unique features in a scale from 1 to 5, that is a skill that you are going to use ok. You can use a much higher skill also, but I have used a skill from 1 to 5 here, right.

Now, what do you see? So, for example, if I have a concept 1, I would say ok the concept one is a highly loaded concept and it has a lot of unique features. So, I have given it the

highest number from a scale from 1 to 5 and the concept 2 does not have that many unique features, some of the things are already there. So, probably it is 2. Concept 3 similarly has the least one and therefore, it is 1 concept 4 is quite something unique features are there.

Similarly, in terms of development complexity also if you see we have rated the concepts based on our views, you know the design team members views, as 2, 4, 2, 1 in terms of maintenance and serviceability. 4, 1, 4, 4 In terms of cross platform compatibility, 2, 1, 1, 4, 5 in terms of user testing; that means, it does not mean that we have conducted user testing. It only means that we are we have seen the market, we as a design team, we know the market, we know the person and now we want to anticipate how well our users would react to.

Probably we can also we might have also have done a cognitive walkthrough, which we have discussed earlier and based on the feedback of that we can take the parameters from the based on the feedbacks of cognitive walkthrough sessions. And we can stack everything here, as part of user testing, but then you can also, you can also choose to go for real user testing, and to some extent pilot testing of your concepts show them go for cognitive walkthrough and then you can use those results here ok.

So, this now do not so, see I am showing you the table it does not mean that these are the actual parameters that you are going to use know. Again, the parameters need to be defined based on your research, your requirement based on what you prefer as the important parameters based on which a concept needs to be selected. And that should come if you work in consultation with the team members, from your stakeholders, from the design brief, from your requirements, as well as from your market study right. These are some of the examples.

So, now, putting all those numbers in stack what you see is that if you sum all those numbers, you see the concept number one got the highest rating, 17 right out of 25, that is the highest one. The next one concept 2 is the 11, concept 3 is 11; concept 4 is 15. So, now, you can go ahead with concept 17 or what you can do you can identify each of these unique characteristics.

For example, in terms of unique features this one has the highest one, in terms of development complexity this has the least one. So, higher means it has the least one and this is more when you have the less one it means it has it is more complex, it is a reverse

coded one, the scale is reverse coded. So, we have done in that way. Maintenance and serviceability; that means, all these ideas are can be maintained and serviceable, they are good and cross platform compatibility probably one this one.

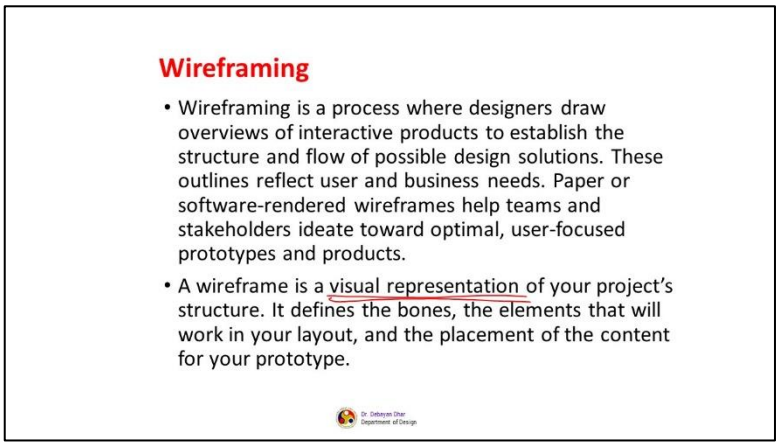
So, either you can identify these unique things and see if you can merge them together. If you cannot, then you go ahead with one concept and go for detailing. Now, when we talk about detailing we means we come up with information, architectures detail how information is presented and then the how we go for wireframes and finally, prototypes for final user testing.

(Refer Slide Time: 35:32)

Now, before wireframing, we are supposed to discuss information architecture, but in this session, we are not going to do that. In the next module where we are going to talk about usability, we would see information architecture in detail. Right now, we are jumping one session and coming down to wireframes. But for the sake of your projects of or if you want to work on this project you must ensure that you look at the module in the next session, where we talk about information architecture.

And then come back here and talk about wireframe, see wireframing and prototyping before you go for final user testing.

(Refer Slide Time: 36:09)

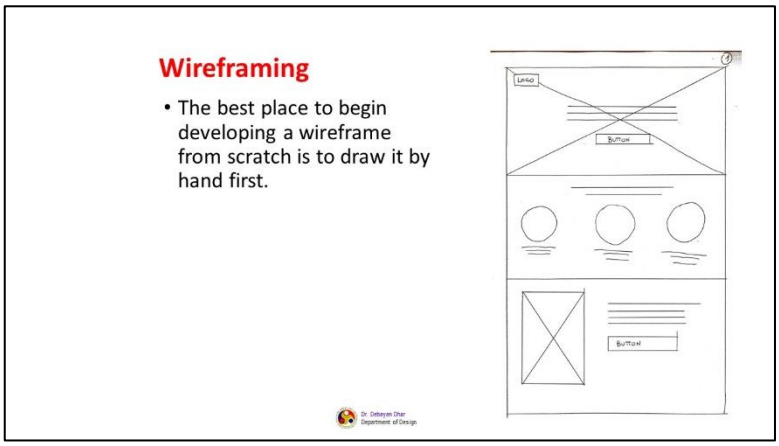


Now, what is a wireframing? So, you know in usability or in user experience design or human computer interaction, this term is widely talked about. What is wireframing? Now, wireframing is a process where designers draw overviews of interactive products to establish the structure and flow of possible design solutions. So, it actually enlists the flow.

So, for example, the task flow that we have discussed about can also be used in a way through which we arrive at the wireframes including the information architecture. So, these outlines reflect user and business requirements, paper or software rendered wireframe help teams and stakeholders ideate towards optimal, user focused prototypes and products.

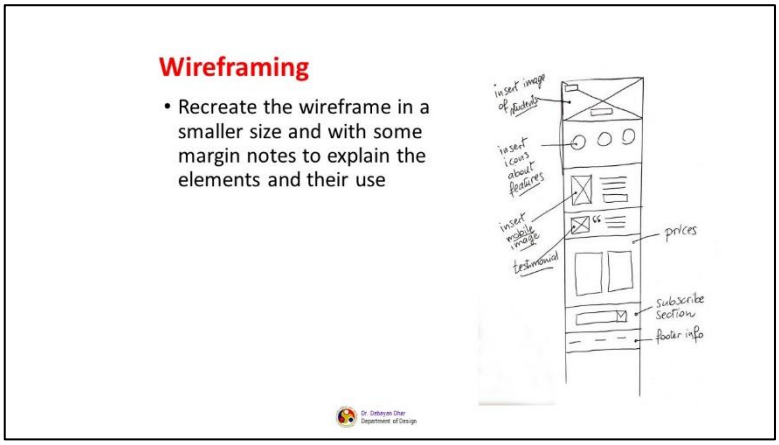
So, a wire frame is usually it is a visual representation ok, it is a visual representation of your project structure. Now, we are talking about structure, you can come up with the task flow, you can come up with an information architecture this two will help you in designing the wireframe. Now, it defines the bones, the elements that will work in your layout and placement of the content for your prototype.

(Refer Slide Time: 37:35)



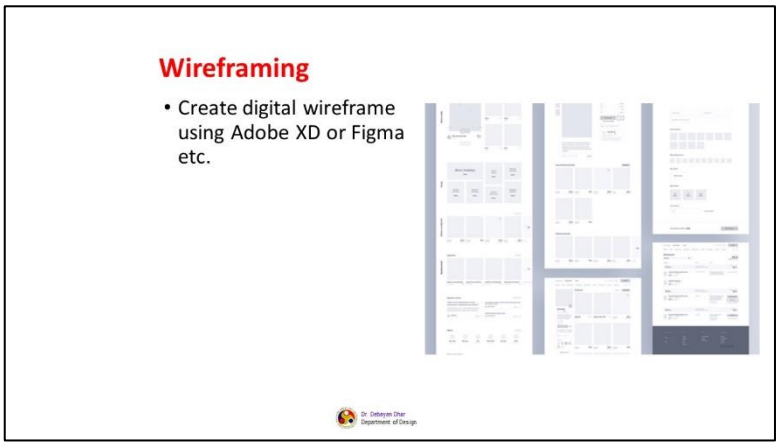
Now, the best place to begin developing a wireframe from scratch is to draw it by hand first. Take a piece of paper, draw the structure of the webpage that you want to design or the interface that you want to design. What you see in this slide is the first-hand rough structure that you can draw using a pencil or a pen to say that ok here is the way is the interface which I want the user to look at.

(Refer Slide Time: 38:04)



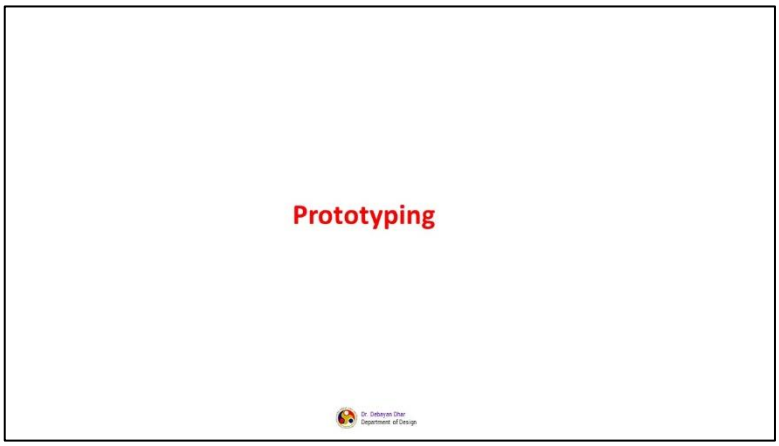
The next part is you recreate the wireframe in a smaller size and then with some margin notes. Now, these notes are important because it will allow you to understand how the task is flowing, the micro interactions aspect it aspect to it. And you also add some margin notes to explain how the elements are being used ok, you can see in this slide here.

(Refer Slide Time: 38:27)

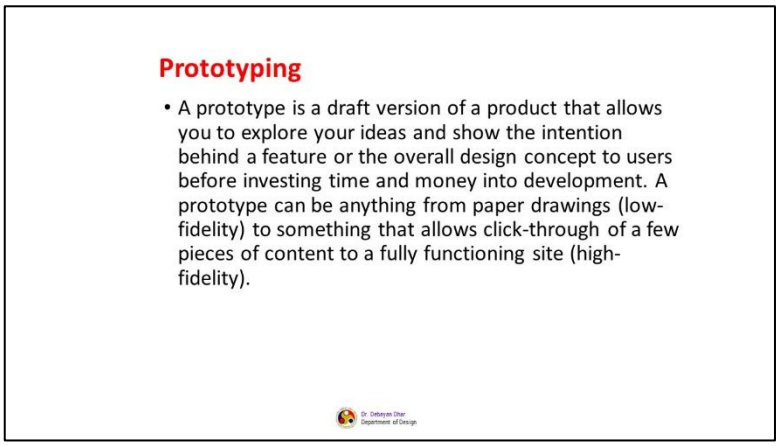


And then what you do? The next part is you create digital wireframes. Now, while creating digital wireframe many designers use nowadays Adobe XD or Figma etcetera, many people use Power Points also, when they create this kind of wireframes and what you have seen the image are the examples of wireframes. Now, these are not final user interfaces, these are not final user interfaces these are the skeletal structure of the interface that your product would have, that is what you are referring to as wireframes right.

(Refer Slide Time: 39:06)



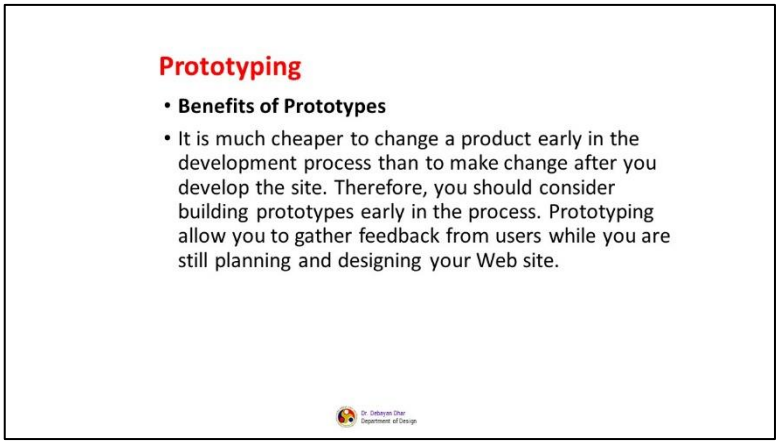
(Refer Slide Time: 39:09)



Once you are done with the wireframing the next thing that you should go ahead with is prototyping now, what is a prototype? So, a prototype is a draft version of a product that allows you to explore your ideas and show the intention behind a feature or the overall design concept to users before investing time and money into the development. A prototype can be anything, it can mean anything it can be a paper drawing.

That means, we are talking about now low fidelity prototypes and it can also be high fidelity; that means, highly functioning that allows click through of a few pieces of content to a fully functioning site.

(Refer Slide Time: 39:51)



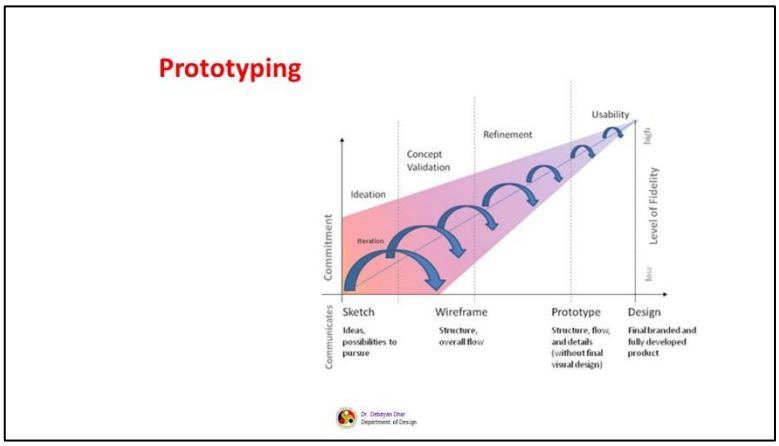
Now, you know prototypes are much cheaper to change a product early in the development process, than to make change after you develop the site. So, therefore, you should consider building prototypes early in the process. Prototyping allow you to gather feedback from

users while you are still planning and designing your website see once if you are fully developing something, if you are fully developed you have investment in invested a lot amount of resources in developing the product. What about if that thing does not work?

So, in order to ensure that you get a quick user feedback, you first develop a prototype that is at a much lower scale, but it can be a high fidelity one, it can be something the user can click through and can interact with the screen and can go for quick user testing sessions that are the benefits of prototypes. So, if you get feedback from the users and if you see that there are some changes that are required so that it matches with the mental model of the user and then you by incorporating changes you again go for feedback. If everything goes ok, you go for release.

Then it what it happens the amount that you have invested without even testing, know you can get good return on investment because once you release it, your user would be able to see that it is a product that they like they prefer. If there are minor issues they would not be able to accept that product that these are the benefits of prototypes.

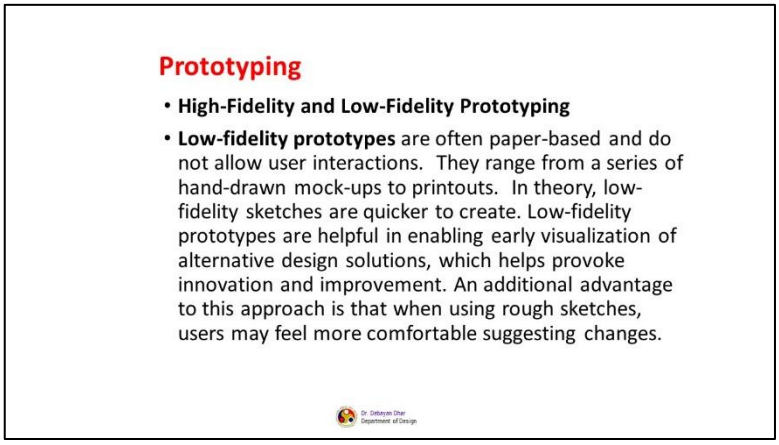
(Refer Slide Time: 41:24)



Now, what you see here is a very nice representation of the structure of what we call as the ideas which it starts their journey from being a sketch, which are you know raw ideas, then it gets converted into wireframe which is which has a structure and then to prototype, and then to final design the development part of it. So, at each stage, so the sketch that we refer to is the ideation part that we have been talked about, the wireframe is more likely it is a part of concept validation.

The prototype where we you know refine it more and we create we create a much more real life and product that looks very close to what we would like to release or would we would like to offer to the market. Where testings are done, usability issues are identified with users. And finally, the final design is a fully developed product that is you know released to the market. So, in terms of fidelity in terms of the nature of the quality of the prototypes, you can see it starts from being very raw to a high fidelity prototype which is the final product which actually is the final one that you have intended to design.

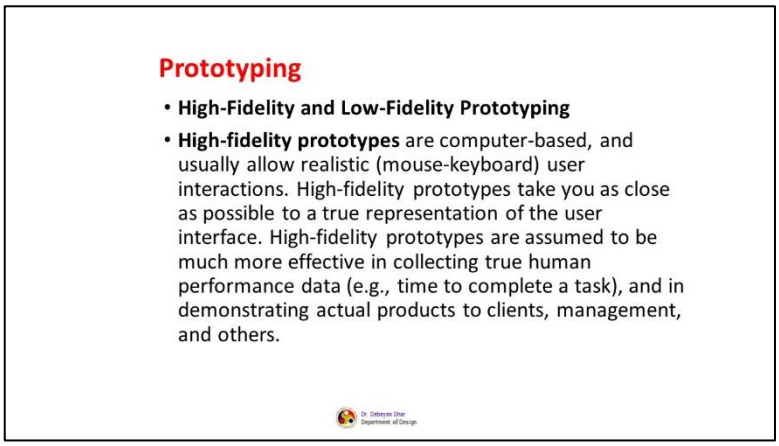
(Refer Slide Time: 42:44)



Now, prototyping as we have discussed it has two types. One is the high-fidelity prototypes and then the low fidelity prototypes. Now, in low fidelity prototypes are often paper based and they do not allow user interactions. So, they range from a series of hand drawn workups to printouts. So, in theory low fidelity sketches are quicker to create. So, low fidelity prototypes are helpful in enabling early visualization of alternative design solutions, which helps provoke innovation and improvement.

An additional advantage to this approach is that, when using rough sketches users may feel more comfortable suggesting changes. If you want to use your low fidelity prototypes and go for quick cognitive walk through sessions or user testing. They would be more open for suggesting changes, because they would see ok this is not a final product, it is just something product that is being built. So, let us give open and good feedback to it.

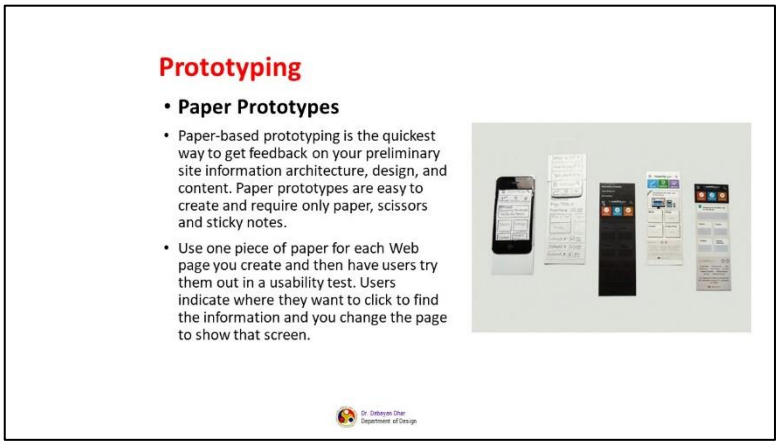
(Refer Slide Time: 43:50)



Now, in high fidelity prototypes are computer based, and usually allow realistic user interaction actual, the it would allow how the actual product would be. And high-fidelity prototypes take you as close as possible to a true representation of the user interface.

So, high fidelity prototypes are assumed to be much more effective in collecting true human performance data; that means, you can collect a lot of a analytics also, time to complete task. And it can demonstrate actual products to client's management and others. And you can conduct a full-scale usability testing with a high-fidelity prototype.

(Refer Slide Time: 44:36)



Now, many a time you know in design industry and in design groups, paper prototypes are considered to be the quickest way, through which user testing is done. Paper based prototyping is the quickest way to get feedback on your preliminary site information architecture, design and content. Paper prototypes are easy to create and require only

paper, scissors and sticky notes. You can use one piece of paper for each web page you create and then have users, try them out in usability testing phase which we are going to discuss in the next module.

Users indicate where they want to click, to find the information and you change the page to show that screen, that is how a paper prototype is being used.

Now, with this we come to the end of this module. In the next module we will start about talking about usability testing, usability valuation and we also would discuss about information architecture. Now, if you remember card sorting technique a lot about information architecture can be understood of how the information is grouped by the user in terms of their mental models by using card sorting.

So, some of the ways to understand information architecture is to use card sorting techniques, but once you have done that, now it is a situation where you are supposed to come up with your y friends. So, before that you can design the task flows and you can come up with the information architecture. So, in the next session we will discuss about that, followed by the usability valuations of the product.