

Human Computer Interaction (In English)

Prof. Rajiv Ratn Shah

**Department of Computer Science and Engineering
Institute IIT Madras**

Prototyping and Smart UI - Part 1

Hello everyone. Welcome to the class of human computer interaction. So this week we'll be reading about prototype and smart UI. As we have discussed earlier, just having an idea is not enough. There should be something tangible that we can try, that we can feel so that we can provide the feedback and see whether the functionality we require is there or not. Similarly, We need to have smart UI because in the advancement of technology, different users have different needs and that is something can be adapted by the advancement we have.

So this is a quick recap of what we have discussed in earlier lecture. So we studied about data requirement followed by data gathering, data analysis, and we used a case study to discuss the data requirement gathering and analysis of a real-world problem. We also discussed a very important topic related to the ethical concerns of the data. So we had a panel discussion on IRB, and we also discussed the same in our lecture.

We also had a tutorial on data gathering and analysis by Rithvik, I hope you have thoroughly enjoyed it. So this week we'll be talking about what is prototype? What is prototyping interaction design? What is smart UI? We'll be having a tutorial on prototyping as well as a discussion on prototyping with a group of brilliant students. So let's get started. So what is prototyping? we have discussed in our earlier lectures but just to formally tell it is a one manifestation of a design that allows stakeholders to interact with it so this is something tangible that you have that you can play with it that you can try and see whether does it have the required interfaces whether does it have the required personality you're looking for So in other design field, a prototype is a kind of small scale model. It could be a miniature car.

It could be a miniature camera. It could be a miniature building or town and so on. As Tom mentioned, prototyping is the conversation that you have with your ideas. Because just having an idea is not enough. You need to somehow translate into something tangible.

And building this prototype is a kind of conversation that you are having what is really feasible, what is really possible. So in order to build a prototype and in order to probably

translate your idea into something tangible, we need to start somewhere. And that's where basically you start with conceptual design. So the conceptual design is the initial phase in the design process where the fundamental ideas and structures are outlined. It focuses on emphasizing a high level framework for what people can do with it, how a system will function and how the different stakeholder and the user that you have going to interact with it so conceptual design defines the core concepts user interaction and the system requirement before detailed design begins because before you're designing something you need to understand how the system is going to work you need to have some visualization about what all kind of different stakeholders you have and how they are going to perform their required actions, what kind of input they will be providing to the system, how they will be interacting with the interfaces that you have, and what kind of output the user will be getting.

So it understands problem space and current requirement, and most importantly, it empathizes with users of your system. So it helps in creativity and brainstorming techniques. So it mood, board may capture design feel. There are different alternatives. So you can have a kind of scenario and prototyping.

These help in conceptual design. So the key aspect of conceptual design. So the key concept of conceptual design in prototyping, it involves firstly you need to define your goals. So what is the primary objective of the prototype and what problem does it addresses? Next is identify the user needs. So that is something we also discussed in need finding phase in earlier lecture.

So what are the core tasks and interactions the user should be able to perform or probably want to perform? Outlining the feature, so which features are crucial in this phase and what functionality should be included. As we have also discussed in previous lecture, it is not possible to probably complete everything but all a user want. So you have to prioritize their needs. So that's where basically you also need to take a call, what functionality should be included, probably in version one and so on. So user interaction model, so it talks about how will user interact with the system and what is the user journey.

So in a way, I mean, for a given task, how a user perform, how the user start and probably do some steps in between to accomplish the task the user want to do. Information architecture, so So it tells about how will information be structured and how will the content and navigation be organized. And this will also help us in understanding how to make a simple navigation so that user can achieve the goal by going through the different navigation it has. So why conceptual design is important in Prototype? I believe it's quite clear. So it first clarifies the scope.

because if you don't limit the scope, probably you will not meet the expectation of the user, or probably you may create a hype, which end of the day is going to create a problem for you. So it helps in focusing on the primary user interaction without getting lost in details. It guides the development. So for example, it provide a clear direction for building low fidelity to mid-fidelity to high-fidelity prototypes that we are going to discuss in later slides. So it enhances the communication.

So it acts as a shared vision for stakeholder, designer, and the developers. As we discussed, prototyping helps in the communication between the user need and probably the designer that we have. It reduces the risk, identify potential problems early, saving time and resources in the later design stages. Because it's important to fail early, it's important to understand your users early rather than building a full-fledged system and probably create a disappointment for your users. So for example, In developing a prototype for a health application, so the conceptual design phase would outline what is the core user task.

So it could be like scheduling appointments or probably the tracking symptoms that you have. It also requires navigation flow from home screen to the detailed health record that you have. and the essential features such as search for functionality or probably the reminders or probably the shared health record with the doctors and so on. So a conceptual design is the blueprint for successful prototyping and it ensures that prototypes are aligned with the user needs and the system goals. So you can say that the conceptual design is a kind of bridging the gap between the user needs and the system goal that you have.

So if you talk about the journey, you can see that it starts with conceptual designing, which helps us in building low-fidelity prototype. Further, you can expand it to mid-fidelity and high-fidelity prototype. And after several rounds of iteration, once you are satisfied with the current version of the solution or the product that you have you can launch it and in between you can get the different feedbacks and again several iterations can happen even after that and in order to fill the gap between the user understanding and probably the system interface that you have, it is very important to use some kind of metaphors. So interface metaphors combine familiar knowledge with the new knowledge in the way that will help the user understand the product. So in the previous lectures, we have also discussed about the metaphor related to copy, paste, probably recycle and so on, probably search.

So which is something we relate from our daily life to the problem that we are solving. So it includes three steps. First, understand the functionality, identify the potential

problem areas and finally generate the metaphors for that. And how to evaluate the metaphors? So how much structure does it provide? How much is relevant to the problem? Because you can't use a metaphor which doesn't relate to the problem that you're solving. Suppose you want to send an information, then you can't use a metaphor search for it.

So search metaphor is primarily for finding, retrieving, searching the information. Is it easy to represent? Will the audience understand it? And the audience will only understand if you can use the common sense knowledge of the users and try to probably do something similar for the terms they are using in their daily life. And how extensible is it? So there are different consideration between interaction and the interface types we have. So for example, which interaction type? So how the user invokes the actions? So that could be instructing, conversing, manipulating, exploring, and responding. So these we have discussed in greater details in the real lectures.

Do different interface types provide an insight? Shareable, tangible, AR, and so forth so like you can probably think an example of chatgpt here probably you can see probably i just ask who is your founder without even naming who is the founder of chatgpt or open ai or something so it got the context and that is the beauty of large language models and we'll be discussing about them in greater details in the upcoming lectures so it understands the concept so it understands the context so chat GPT replied I was created by OpenAI an artificial intelligence research organization and OpenAI was founded by several notable figures in the tech industry including none other than Elon Musk. Probably many of you are not aware. So though now he's not part of OpenAI, but he's the one who have founded along with the other tech expert you can see here. Sam Altman. If you recall, I have mentioned many times In our earlier lectures about Y Combinator, if we have a brilliant idea, why don't you just go and seek funding for that? And Y Combinator could be one such organization which makes Airbnb from million-dollar company to probably \$100 billion company.

So, yeah, so he's the former president of Y Combinator, and now you see where he is taking OpenAI these days. Similarly, we have Greg, we have Ilya, we have John, we have other people so these founders along with many other researchers and engineers created OpenAI to ensure that general artificial general intelligence benefit all the humanity and you can see there are different several metaphors is used here so for example for this particular speaker icon that you can see so read aloud so for example what if a user that you have who is conversing with chat GPT is probably visually impaired in that case probably a feature accessibility features like this read aloud will read the answer for you and you can hear what it is similarly i mean you can see this is something similar icons probably looking for you so that is something for copy and these are probably providing

the feedback you like the response or not because chat gpt can probably sometimes hallucinate And similarly, this is something it's more I can about doing a game, some kind of regeneration and something. So in a way, what all the metaphor they are using in their interface is familiar to the common sense knowledge of the human being or the users. And it is, I believe it's quite clear which interaction type is here and how you that invokes these actions. by clicking different icons that you have and basically the corresponding metaphor that you have is also help you telling you the meaning of that when you hover your mouse over these icons.

So let's expand the initial conceptual model. So what function will product perform? So basically it help you letting you know that will the product what the product will do and what will the human do. So, what are the functions related to each other? Will that be sequential or parallel? So, similarly categorizations for example, all actions related to privacy on a smartphone. So, what information is needed? So, often you see that what data is what data is needed to perform the task and how is the data to be transformed by the system. So again, we have discussed all this in greater details in the previous lectures.

So once you have conceptual design, so you have to build something concrete, otherwise it will be just an idea which nobody can play with it, nobody can understand its feasibility and so on. the difference between conceptual and concrete is the emphasis on the solution that you have so many aspects to concrete design so probably you will have color icons button interaction device and so on so that is more towards something tangible So user characteristics and the context. So it is inclusive input and output modes. So we discussed about different accessibility guidelines. So WCAG is one of the popular guidelines that probably you should have when you are going from conceptual to concrete design.

So you should also support cross-cultural design, so related to language, colors, icons, information architecture, and indigenous knowledge and the perspective. how you can generate the prototype that is something or how you can generate the more towards concrete design so that is something you can do using building some storyboard how this different steps involved of a user in completing something So for example, in this case, consider a scenario. So break down your scenario into some steps so that you can complete step one, then two, three, four, five, and six, and finally complete your task. So here basically you can see sketching out a storyboard prompts designer to think about the design issue, generate a card with prototype from a story board or from a use case. So like in this case, you can see that in case of the story board, a user first, in this case, basically the Thomas family gather around, Claire joins and Claire basically joins remotely.

So it is about the communication between a person joining virtually and probably the

remaining are there so it will basically tells their initial idea and similarly you can see that the different steps are going to complete sending a kind of detailed email about it so regarding card prototype So this is another way of creating a kind of prototype. So card prototype enable rapid and low cost ideation, quick user feedback and collaborative design by visually organizing information and facilitating testing, prioritization and iteration. So like in this case, you can see that you can create kind of a card prototype if you are basically going to some place for tourism or some other purpose so for example where do you want to go my passport was issued in so when you are going for international trip or something you need to inform where are you from what is your citizenship that is the nationality you can ask similarly what destination you want to go so based on that again it can figure out the additional information for example visa and many others so why are you going there so based on the purpose it can suggest the appropriate visa for you and similarly you can see that and again i mean this is kind of doing the same thing as compared to the second one probably asking in different ways like in this case where do you want to go where do you want to go and your destination so one good thing you can see about the second prototype card based prototype we have we are using minimal number of word and probably quite clear so my passport was issued in is probably same as the nationality and so on so we have also discussed about creating persona so which help you understanding the user requirement need and expectation so use persona card based prototypes or stickies to model the user experience and so visual representation called design map or you can see the customer or user journey map that we'll be discussing slightly more details in the next slides so in the experiment So in the experience map, basically you try to understand the whole user journey across the system, the feeling of the user, how was the experience, how easy or tough the task was, how complex the system was and everything talking about. So there are two common representations. So one is wheel-based, another is timeline-based.

So tools for creating Experience Map include Mero, Mural for visual mapping, uExpressia, Sapli for specialized journey maps and Adobe XD Sketch for visual visualizing user journey and creating prototype. So Miro we have discussed in one of our hands-on session. So an experience map can be drawn as a timeline or a wheel as mentioned earlier. So each suited to different types of user journey. So specifically timeline, it is a more linear format emphasizing a chronological sequence of events, ideal for experience with a clear start and end.

For example, e-commerce purchase, where you start with searching some product or item that you're looking for and eventually you end with buying the product. So it helps identify the gaps, visualize dependency, and create step-by-step narrative. At the same time, Wheel, it is a circular, cyclical format focusing on repetitive and iterative process ideal for ongoing interactions like loyalty program because if you keep using the system

and the product that you are using the company will never want to lose you they always want to retain all their customers so that's where basically different the companies or the products have the loyalty program and they give some incentive for keep using the system If you talk about one of the examples we had discussed in one of the earlier lecture is about, for example, credit card payments. So credit card payment you can do through N number of applications, but probably the cred, if you use, probably you can get something to probably play with. If you get 777, you can get the bigger rewards.

Otherwise you can get some points, virtual coins, or probably some small money and so on so it highlights continuous cycles interconnectivity and bottlenecks in recurring stages so this is an example of experience map drawn as a wheel so in this case you can see that so basically here the person want to travel so in this case you can see that before traveling so there are some different touch point you can see that so in this case like contact travel department book tickets receive tickets check in at the train station and probably train for the airport blah blah blah and something you can do and probably during your journey you can do a number of other things and there will be corresponding touch points and the interactions both the plane safety procedure take off and so on and similar for example watching a movie choice of meal times and so on and probably once you're done with your airline journey probably you may reach to your destination so that is before during and after here basically you can reach to your hotel you can check in you basically meet the driver and similarly a number of things you can do luggage collection and so on so this is the way you can see that you can draw an experience map probably from going from one place to some other what all the experience you are going to have. So, and similarly there is another experience map drawn as a timeline. And in this case, you can see that there are different timeline. So map title and basically these legends can help you understanding. So what is the step, comment, question, idea that you are going to have.

And in this case, you can see that here design map is Megan delivers the presentation and how this presentation delivery process takes can see that it start with Megan logs on to the presentation system Megan sees the presenter page Megan sees that her slides are ready and she does a some last minute flip through and Megan realized that there are probably some issues in some slide so she fixes an error in one of her slides and finally Megan sees that audience members are starting to arrive and so now you have to do the presentation but in the meanwhile there may be some questions there may be some comments there may be some idea that is all you have to show as a timeline so so timeline which one to use so timeline is the best for capturing experiences that unfold in a clear linear order, allowing detailed analysis of each step's progression. And wheel is well suited for visualizing ongoing or repeating process, showing the interconnected nature of the user's experience as a continuous journey. so example of both you can see that in case of timeline example booking a flight online with stages from search to booking to

payment and finally to post flight feedback but in case of wheel as i mentioned it's more about continuous getting or involving your user and probably get their their user journey so it is example of for example a loyalty program for the retail store where journey cycles through the earning points reading rewards re-engaging repeating in a loop keep buying using the points earning the points spending the points and so on so that is about users experience map So, let us get your get our hand dirty. So, in that case finally, let us how you can construct the prototype. So, again there are you may construct some prototype which may involves a physical computing system.

So, for example, Arduino board is one of the most popular. uh system people use these days so you can build and code protect using some electronics so several toolkits are available as i mentioned including arduino lily pad and so on and they are designed for use of wide range of people And for the construction, often you need existing libraries, existing APIs, existing methods. So for that, basically, there are often well-defined SDKs. That's called software development kits. So you can use these programming tools and components to develop for a specific platform, iOS, Android, and so on.

So it includes IDE, documentation, driver, sample code, and API. and it makes development much easier. So for example, Amazon's Alexa skill kit for voice-based services, similarly Microsoft Kinect SDK for motion tracking, Apple's AR kit for augmented reality, similarly OpenAI's SDK for more communicating with the chat GPT and so on. So that is brief introduction about what all the prototype is and probably how they are useful and how we have a progression on building the prototype so let's talk about what is prototyping in interaction design because end of the day our goal is to build some kind of interactive system some kind of human computer interaction system and so on so in interaction design a prototype can be Among the many other things, it could be a series of screen sketches, it could be storyboard, it could be PowerPoint slide, video stimulating the use of systems, a lump of wood, cardboard mock-up, or probably even a piece of software with limited functionality written in the target language or even in other language. So the idea is, can we have something tangible? why prototyping is helpful it can help you in testing your ideas it can help you in evaluating and getting the feedback as through the your users stakeholders can see hold interact with the prototype more easily than just see a document or probably a drawing or just listen to your ideas.

So identify usability issues early so that you can fix, you can incorporate any changes at the early stages to avoid kind of losing time, losing money, and so on. So iterate design based on the user feedback because based on the feedback you are getting at different stages regularly, you can update your system accordingly. And team members can also communicate effectively. prototype encourages reflection and important aspect of the design so prototype answer questions and support designer in choosing between the

alternatives for example often you don't know what is the best way of doing something if you recall the example of card based prototype we have discussed there are two different cards probably serving the same purpose but which one to choose so probably you are going to choose something more optimal more easy to use and so on what all kind of prototypes are there so they're primarily three prototypes you can say that so low fidelity prototype here the fidelity primarily referring to how close the prototype resembles the final products in details and interactivity so that's why low fidelity is more about i mean there is very little closeness or you can say very little resemblance between the system that you are going to have and probably what prototype you have at this moment mid fi uh mid fi prototype is more about somewhere in between than probably what is a very uh clean and probably a better version of probably very close version of the real product that you have in the high fidelity prototype so prototype should be quick and dirty so that you can again get the feedback issues benefits at the early stages and the important part of is learning not the artifact and what is the low five prototype let's go into greater details so it is simple rough and often hand drawn sketches or models so it can be used early in the design process to brainstorm ideas and get initial feedback It is quite quick, cheap, and could be easily changed. So there are different techniques, such as paper sketches, whiteboard drawing, card sorting, and post-it notes.

Wizard-oz and wireframes using some basic tools like pen, paper, or sticky notes, and so on. So these are the different tools you can use for building even the low-fi prototypes. or models so they are used early in the design process to brainstorm ideas and get initial feedback It is quite quick, cheap, and could be easily changed. So there are different techniques, such as paper sketches, whiteboard drawing, card sorting, and post-it notes. Wizard-oz and wireframes using some basic tools like pen, paper, or sticky notes, and so on.

So these are the different tools you can use for building even the low-fi prototypes. For example, designing an app layout using sketches on paper to show the screen transition is helpful. So similarly, the card-based prototype that we have seen is one such example of a low-fi prototype. A paper prototype allows you to get ideas quickly and change them even quicker.

So that is the benefit of having a low-fi prototype. So, Adobe Express, I mean, we are going to have a couple of sessions with Adobe Express. to have some hands-on experience in solving different stages of the whole human-centered design process. So we are basically using it for prototyping, we are using it for evaluation, and so on. So, a little about Adobe Express. So it is a user-friendly, cloud-based design tool for creating visually stunning graphics, videos, and web pages.

So Adobe Express is one of the most popular tools that we have these days, which can help you simply simplify your creative design. So it offers pre-designed templates and a drag-and-drop interface for quick and professional results. So this is the interface of Adobe Express, and you can try it. So the key features it has are a template, ease of use, creative tools, cross-platform access, and collaboration. And in the case of HCI, prototyping UI design quickly for stakeholder presentations, crafting user journeys, and visuals for experience mapping that we just learned.

So, with Adobe Express, someone said, creativity is just a few clicks away. Isn't it amazing? I believe so. It's quite amazing. And while it stands out due to its very interesting features, no prior design knowledge is required. The people who are working with the designer that you have, it's not like everyone has a kind of.

.. A good understanding of design concepts and so on. design principles and so on So it helps you in building something creative even if you don't have any design background. So it integrates with Adobe Creative Cloud for advanced options that you can explore. and it saves time while delivering professional-quality outputs So, some real-world examples that you can see are, for example, small business marketing. So, design branded materials like advertisements, business cards, or social media posts again. Using the different templates and many other resources that are available, you can do so.

For educational use, you can create engaging presentations and learning aids for your students. For content creation, it enhances personal and professional blogs with custom visuals. Additional resources for Adobe Express. So for further readings, kindly go through these links; similarly, for providing any feedback. To Adobe around Adobe Express or any suggestions? Or even any new features that you think would be useful, feel free to provide them at this link.

So as part of this course, we have a couple of sessions with Adobe Express, which include designing mock-ups for mobile apps using Adobe Express. So the process of design is a process using drawing artwork. Generative technology uses text templates and LLMs to rewrite various texts in a professional style. And finally, project work to create a brochure using Adobe Express features.

I hope you recall our flagship project, Project Wave. So where the aim is how we can facilitate learning for visually impaired persons or students through the different sensors you have. similarly these are some lo-fi prototype for the gloves they want to develop so like in this case wearable gloves with a flex sensor to detect finger folds and buzzer with the haptic feedback so in this case you can see that this is the frontal view this is the side view how it is look like similarly you can again in the One of the benefit that we discuss

of low fi is probably you can do some kind of evaluation early stage itself. So for example, if you just finalize this and start building these gloves, what if the user don't like? So that is the idea. I mean, you can build some alternatives. So for example, in this case, you can come up with any alternative low fidelity design.

So in this case, the model uses skin friendly adhesive bolts and tensile wire to detect finger pulls based on the wire tension and so on. So again through which I mean you can design the different options and again so all those the low fidelity prototype may have some drawbacks or probably the benefits and but you can take a call based on the different constraint you have to choose the final one. Similarly you can see you can do some kind of storyboard which is a series of sketchboards showing how user might progress through a task using the product and so on similarly for storyboarding you might have seen that often it requires some kind of drawing so what if you don't know quite good drawing so i think it's fine i mean you can just think of some very common you can say the elements often you use in the design and just make up so for example just for making a person you can do something simpler like this similarly the giving receiving transferring and devices something like simple drawing you can make and probably go ahead with your prototyping Another low fiddle prototype as we discussed earlier as well. It's more about the index card.

So for example, let's take the same example. Where do you want to go? My passport was issued in and why are you going there? What is the purpose? So in case of wizard OZ flow traffic, in this case, basically a user think that they are interacting with a computer, but actually there is no computer. There is one real person sitting behind it and responding to the output rather than a system is giving. So like in this case, you can see that here the person sitting behind the screen and it is responding to all the answer which user has asked. So it is usually done in design to understand the user expectation. And what is wrong with this approach? I mean, In a way, I mean the user is expecting or assuming that the response he is getting is kind of gold standard getting from the system, but eventually it is coming from a real human who is sitting behind the system.

So the next is mid-fi prototype. So where basically it provide little more detail as low-fi. It is focusing on structure, layout, flow, but without final visuals. So in this case, you can see the example would be a wireframe of website's main page showing navigation link and basic interactivity without visuals. And there are different techniques you can see that digital wireframe, clickable wireframe, basic navigation, and these are the different tools you can use for this. And this is the similar example you can see for a project where you can have a mid-fi of the website.

So from scratches now you can see some elements that may not, it may not be probably

all the buttons or the things you have is working, but at least it's a good balance between the simplicity and usability and helping you to irritate effectively. And so this is for the glove designing that we're talking about. It's probably a move from sketches to more something tangible but not probably working properly. As you can see the connections are missing. It's not there but it's more giving you a very rough design how it will look like.

So in case of high fidelity, it is more detailed than all these prototypes that closely resembles the final product in terms of visuals and interactivity. So it is used for usability testing and final design validation. So there are several techniques, as you can see here, and the tools, again, many of the tools we have discussed earlier, Ruby XD, InVision, Figma, and so on. So example is a fully functional prototype of a mobile banking app.

showcasing final UI color scheme and the user interaction. And this is again the HiFi prototyping as you can see here for the website that we are building for the app. And similarly, you can see here so this is now the hi-fi prototype that you have with all the proper connections connected you can see here which probably mid-fi was missing and in low-fi it is more you can see that so that is more the sketches that you have in the beginning so the I believe the difference is quite clear. So in low-fi, initial sketches for the Google search homepage to determine layout simplicity. So for mid-fi, if you talk similar other example, wireframes of Netflix streaming interface to test navigation before visual design. And in case of high-fi, it's a fully interactive, for example, Uber app, testing real-time ride booking interaction and so on.

So often we need to do a lot of compromises in prototyping because it may not be possible to complete all the features. So these compromises are necessary to manage some scope, focus, resources, and so on. And there are primarily two kind of compromises we made in the system. So that is the first one is called horizontal compromises, and the second one is the vertical compromises.

So compromises in the prototype must not be ignored. and product needs engineering so prototyping is the conversation that you have with your idea as we discussed earlier as well so in case of vertical prototyping so it's more about in-depth exploration so it focuses on building a complete features or a function in details rather than focusing on everything just focus on one and complete it so it prioritizes depth over the width allowing you to explore a single functionality or interaction fully so it helps in understanding technical feasibility and the usability of the specific features so for example creating a prototype of a search function on an e-commerce app so this vertical prototype might include averaging from the search input filtering search result to the detail but it may not have adding to the card functionality which is probably a slight different activity and the

benefit is so it provides a detailed look at least for one aspect in the design useful for testing specific interaction and getting user feedback and helping identify technical constraint at the early stages and the the cons it has is like lacks a holistic view of overall user experience but because you are now limited to just one functionality it is time consuming to develop a detailed prototype for every features so that is the issue with it second as we discuss is horizontal prototyping so it focuses on a broad range of features with minimal details for each so it prioritizes breadth over the depth as contrast to vertical where it prioritizes depth over the width so giving an overview of the system's entire interface and features and it is quite useful in evaluating the overall user experience and interface flow For example, creating a prototype for an entire navigation flow of a mobile app without delving into the details of each function. It may show the main screens, navigation elements, and general layout for an e-commerce app, but it may not be able to let you do searching of an item in all possible categories. So in a way, it's just giving a glimpse of what is happening, but uh doesn't show you how it works perfectly so the benefit is provides a holistic view of the entire system useful for assessing navigation layout and overall flow facilitates in understanding of how user will interact with the various features and it lacks the depth needed to evaluate specific features in detail and it may look and it may overlook Technical constraint or usability issue with the individual functions. And when to use vertical or horizontal prototype? So use virtual prototype when you need to validate or test a critical feature in depth. At the same time, if you want to use horizontal prototyping, then you want to get an overall feel of the user experience and interface And you can use horizontal prototyping when you want to get an overall feel of the user experience and interface flow and how overall system look like or probably work alike.

So often a combination of both approaches is used in iterative design because so that you can have the best of both worlds. So prototypes are worth a thousandth meeting as Mike said. So there are iterative cycles of prototyping, start with low-fi, move to mid-fi, high-fi, importance of testing and iterating at each stage, and there is feedback loop that iterate based on the user feedback. To summarize, prototyping is essential, helps in understanding user needs, testing concepts, and validating designs.

There are different levels of fidelity as we discussed, low-fi, mid-fi, and high-fi. Similarly, two aspects of design, conceptual and concrete. Conceptual design develops an outline of what user can do and what concepts are needed to understand the product. And concrete design specifies the design details, for example, layout or navigation. So there are three approaches in initial conceptual modeling, interface metaphor, interaction style, and interface design. interaction styles and it expand an initial conceptual model by considering whether the product or user perform each function and how these functions are related and what information is required to support them.

So after that, basically, you can generate prototype using scenarios and the use cases. We also discussed about physical computing kits, SDKs to facilitate transition from design to construction. We discussed different tools and techniques to do so. An iterative process is one of the key of building prototype or understanding the user needs.

and finding the issue with the system. Prototyping is not linear. It is more iterative involving multiple rounds of user feedback and testing. Finally, in prototyping, it compromises are necessary to manage scope, focus and resources. And the key point is fail early to succeed sooner, as Tim said. With this, we stop here for the first part of the lecture. We'll start second lecture up.