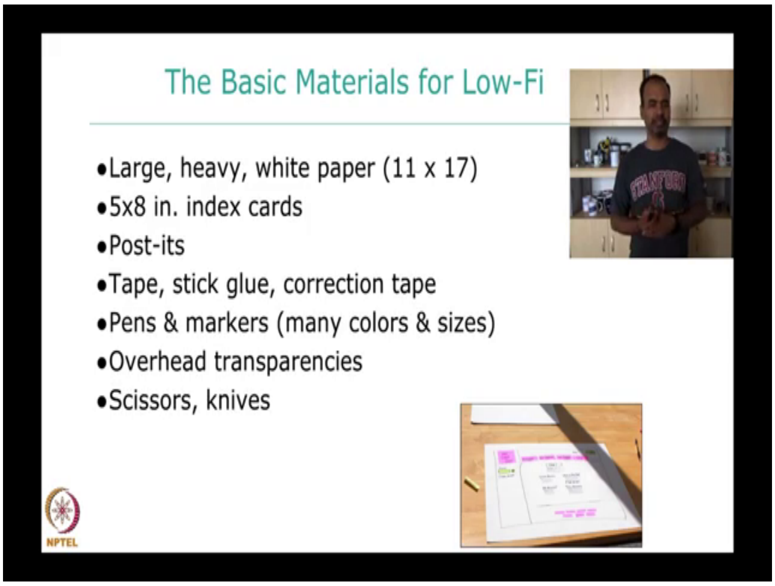
Introduction to Human Computer Interaction
Prof. Ponnurangam Kumaraguru ("PK")
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 14
User-Centered Design

(Refer Slide Time: 00:14)

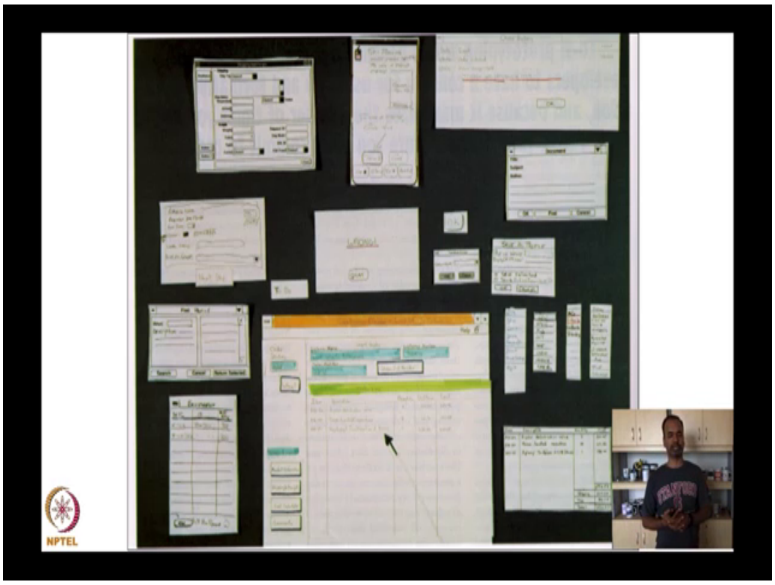


The Basic Materials for Low-Fi

- Large, heavy, white paper (11 x 17)
- 5x8 in. index cards
- Post-its
- Tape, stick glue, correction tape
- Pens & markers (many colors & sizes)
- Overhead transparencies
- Scissors, knives

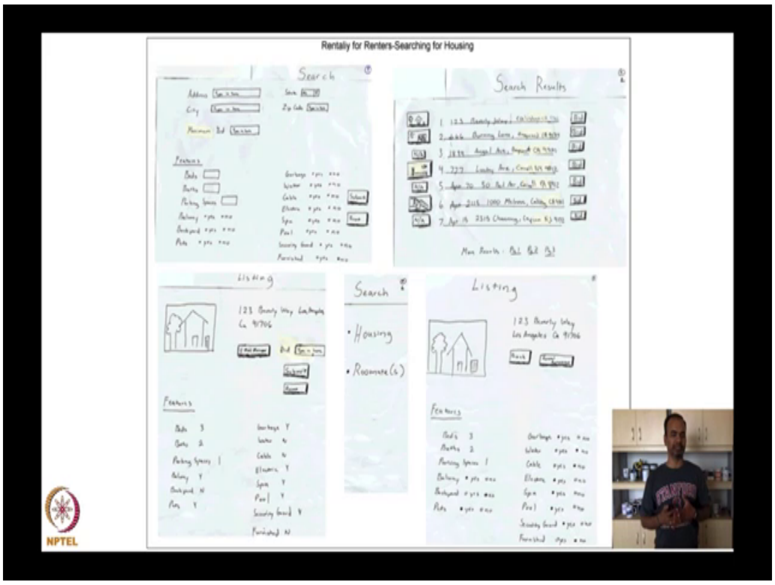
So, these are the basic materials that you need for building a, low fidelity prototypes.

(Refer Slide Time: 00:23)



Now let me show you more of the low fidelity prototypes that people to we seen people built. Here is the one which gives you screenshots or the paper prototypes of the different interfaces that an application has. This is basically showing you different interfaces that the app has or the solution has. Here is another one.

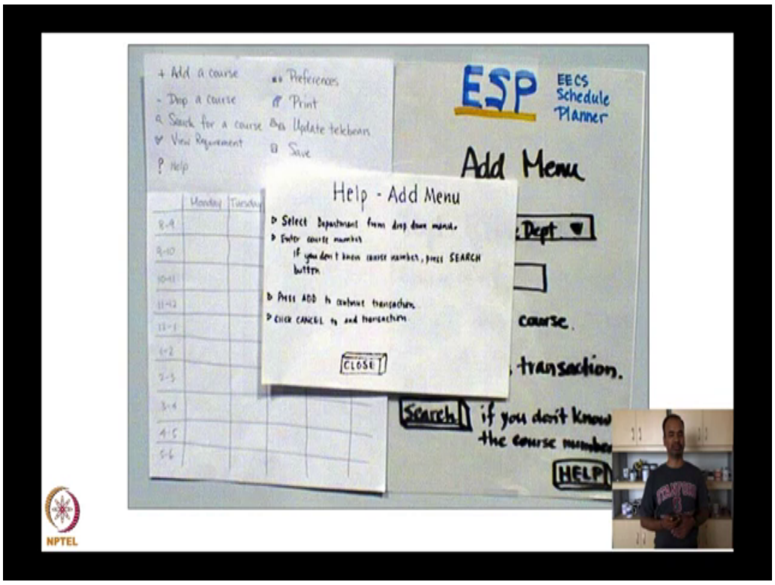
(Refer Slide Time: 00:49)



Rentally for Renters-Searching for Housing

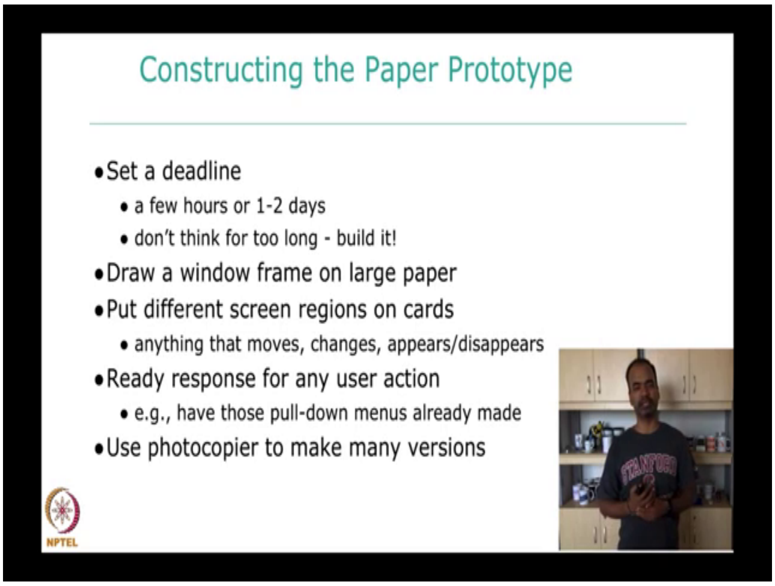
Again everything is very similar because it is all done. In paper, but there is also differences in it because they are also solving there are all solving different kinds of problems. In this you see there is an image; in this there is no image in terms of which interface it is; where the user is currently and it is listening, it is searching and everything.

(Refer Slide Time: 01:11)



Here is another solution which also has so, I think one of the other interesting things that you could do in a paper prototype or low fidelity prototype is you can also create elevations that can actually be a good way of evaluating how your interface will look for the users. I have a link to the video later which I highly recommend students in the class to take a look at it because I think some of the videos I cannot really play it as part of this recording himself because it is actually videos on YouTube. If you take a look at it I can actually talk about the video after you have seen the video there.

(Refer Slide Time: 01:51)



Constructing the Paper Prototype

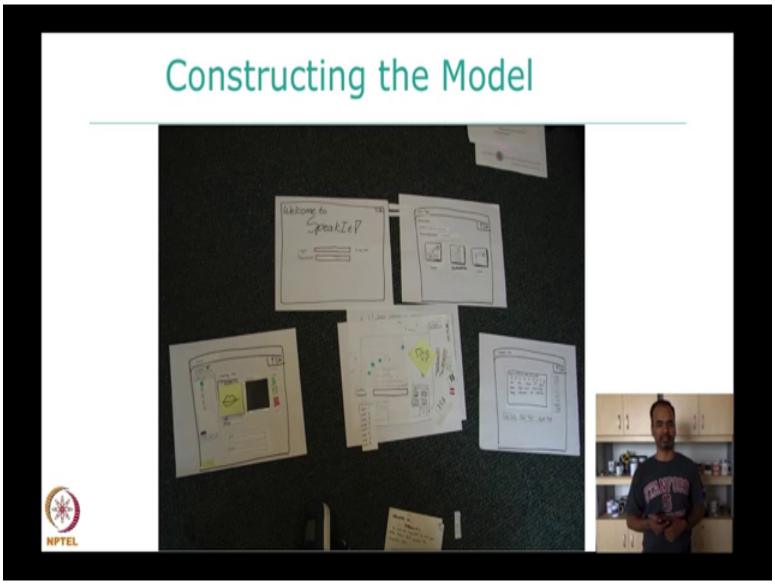
- Set a deadline
  - a few hours or 1-2 days
  - don't think for too long - build it!
- Draw a window frame on large paper
- Put different screen regions on cards
  - anything that moves, changes, appears/disappears
- Ready response for any user action
  - e.g., have those pull-down menus already made
- Use photocopier to make many versions

Let me give you some ideas about how to actually go about creating a paper prototype or a low fidelity prototype right. So, you do not want to keep multiple days or multiple weeks for setting up developing the low fidelity prototype. Keep it really short, do not think for too long; come up with something. I think that is the whole idea for iterative design itself. So, keep it very short one or two days, create some interfaces, go to the users get feedback and then fix the design.

So, some specific inputs are basically make sure that you have a kind of a interface or design or the screen that you want to build, have a larger space by a larger paper where you can actually put things down. It is also legible for the users, put different screen regions on cards, ready response for any user reaction which is that if the user is saying that now I am going to click on the left to see the screen slight behind or are going to actually try and edit my credit card details in the Paytm app.

So, you should be able to provide the next screen that the user may see when they are interacting with the app and also do not waste time in creating many things that are already there. A photo copying or printing out from your machine; you should try and reuse everything that is already there. Do not create something's; do not create things that are already there. For example, the security building an app where you want to create something like an Uber, you can take actually basic nuts and bolts of Uber interface itself and then build on top of it instead of actually creating the entire Uber interface on a paper product.

(Refer Slide Time: 03:33)

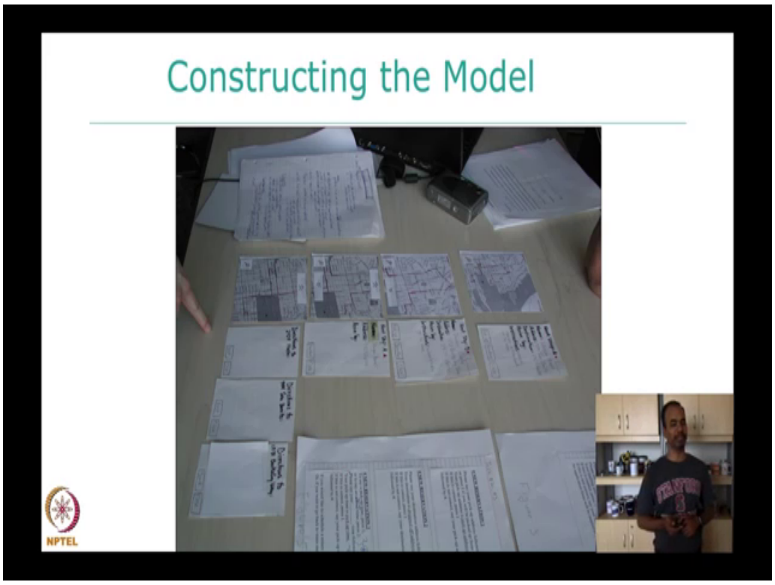


Constructing the Model

Some more examples which is showing you that interfaces; paper prototype interfaces that would be created there are multiple ways of actually getting inputs onto the interface prototype that you build; which is one is just to get the users interact with the paper prototypes, low fidelity prototypes that you develop and get users only to move or when the users are speaking you can you can let give the user this is what the interface is.

So, in this you can actually let the user give you feedback where they are looking at the interface and when they are actually asking questions. You can keep feeding in which interface the other module is; you keep all the interfaces low fidelity prototypes that you build on a table and then they kind of move things around or you can give a paper and a pen and you can start getting the users to give you ideas on how the interface should look like. I have some screenshots of all of hand lets go through one by one. So, this is one where you could actually move around the interface; users could move around this.

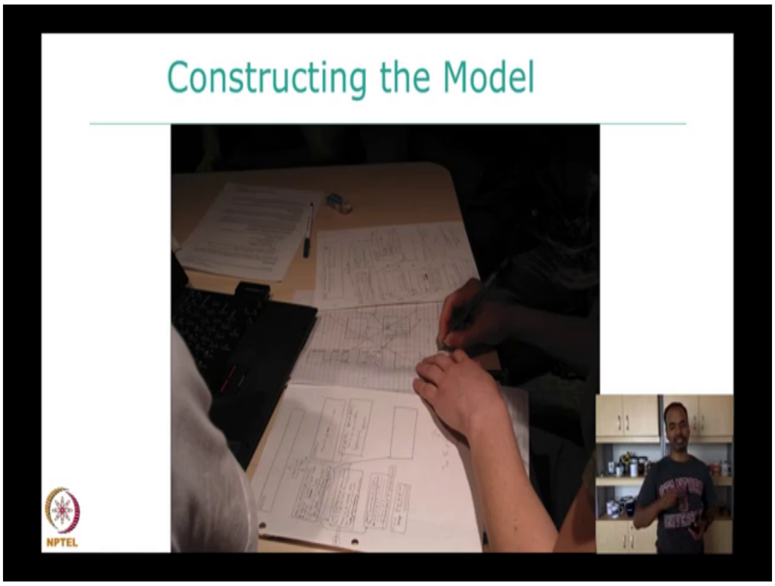
(Refer Slide Time: 04:37)



Constructing the Model

Again the same kind of thing where a users have all the interfaces in front of them and they are actually moving the interface around to make to suggest how the interaction should look like.

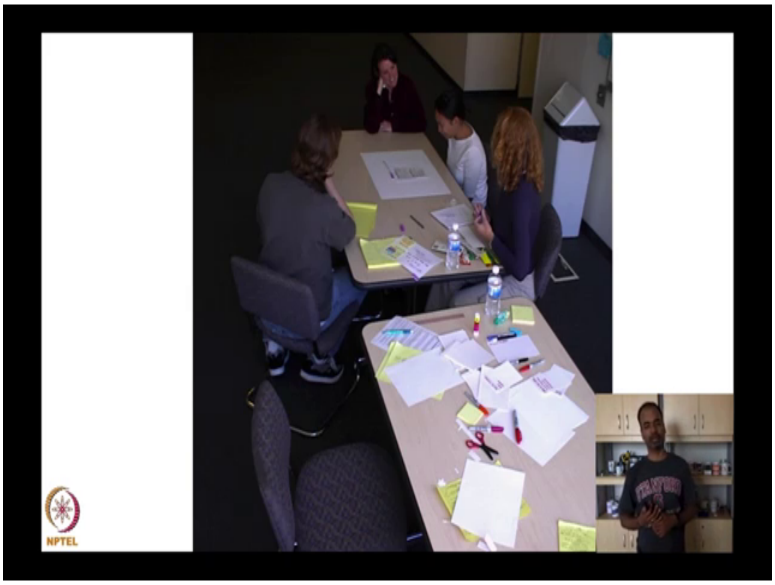
(Refer Slide Time: 04:50)



Constructing the Model

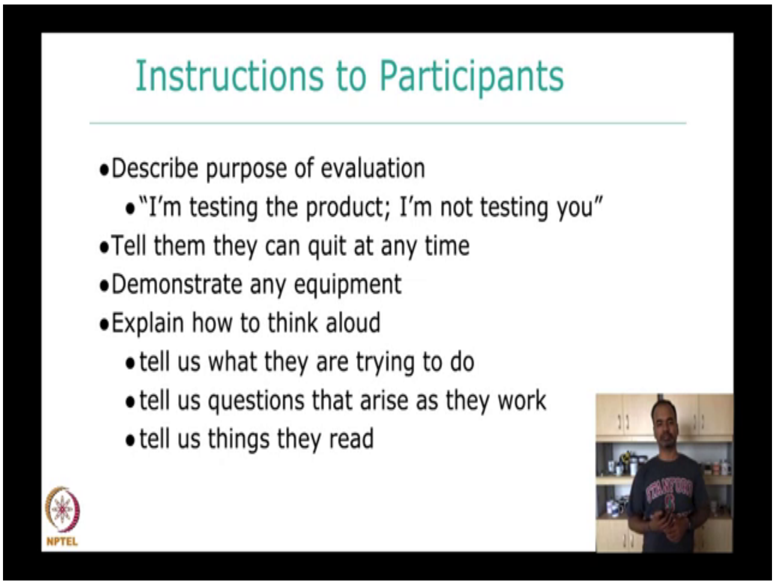
There is another one where the user is actually using a pen and a paper marking with marking the way that the interface would look like right. So, they are creating mark ups. So, they are creating the paper prototypes ideas just for the pen and paper; not really cutting it and pasting it that.

(Refer Slide Time: 05:06)



Also, this is kind of a session where administrators people who are involved in designing and getting inputs from the user. So, there the user is sitting there and giving inputs on this is what the interaction means and then the users or the administrators can actually immediately get feedback from the user and probably can, I mean if you see there are people sitting around trying to take some notes, trying to see what the user is doing and that is how I would highly recommend when you are doing it. You should not be the only one who is actually asking the question, writing notes, capturing everything. There should be at least more than a couple of people giving inputs, capturing the inputs that the users are saying in writing down notes. It is showing the interface and probably even better.

(Refer Slide Time: 05:51)



Instructions to Participants

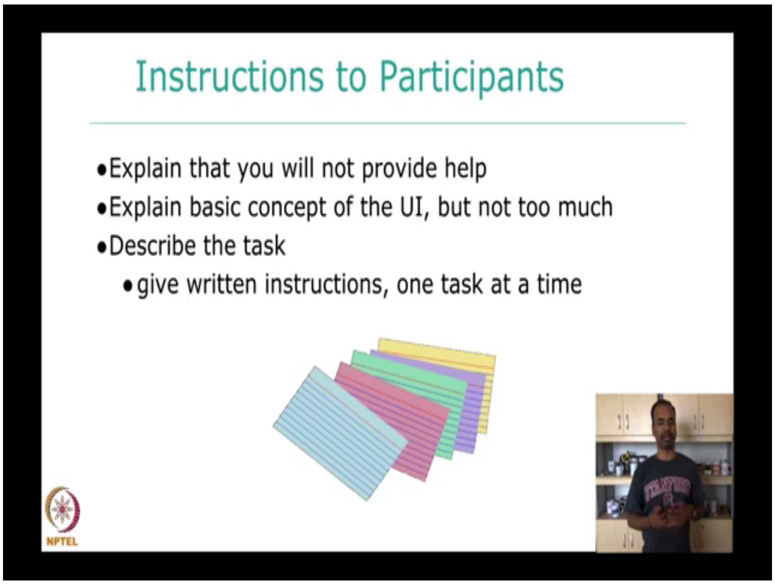
- Describe purpose of evaluation
  - "I'm testing the product; I'm not testing you"
- Tell them they can quit at any time
- Demonstrate any equipment
- Explain how to think aloud
  - tell us what they are trying to do
  - tell us questions that arise as they work
  - tell us things they read

Now, let me give you more instructions to the participants that you after get. These are not just instructions only for the low fidelity. In general I think it will be nice if you can keep these instructions in mind. Describe the purpose of the evaluation that is make sure that they understand what the evaluation is all about. It is it is testing the product; it is not testing them.

So, the statements that you can you clear uses and testing the product I am not testing you. Tell them they can quit at any time, demonstrate any equipment how to use it get them to at least get a feel of what the equipment is. Explain how think aloud works because I think one of the biggest disadvantage that you will have and think aloud or in this low fidelity evaluation prototypes is that low fidelity prototype evaluations is that you will end up actually seeing people who are more talkative give you more inputs compared to actually people who do not speak much.

So, you want to explain what I think aloud is; make sure that the participant that you have got; understands what I think aloud it is and is able to actually speak; is able to actually communicate his thoughts; basically things speak everything that he or she is thinking; explain that you will not provide help which is that when the user is actually doing the evaluation; I think it is a bad idea to stop them and then say that oh why did not you do that? Why would you not do that?

(Refer Slide Time: 07:00)

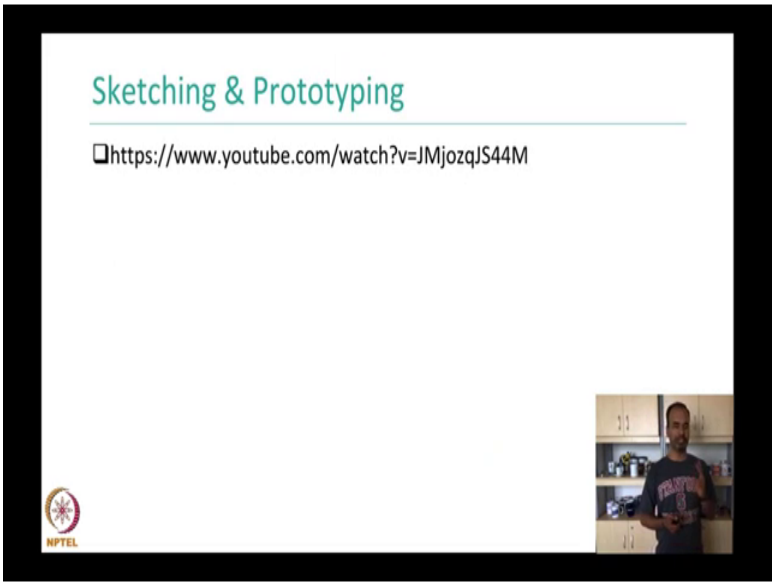


Instructions to Participants

- Explain that you will not provide help
- Explain basic concept of the UI, but not too much
- Describe the task
  - give written instructions, one task at a time

If they are really stuck you can help but otherwise do not get involved and give you help and make it very very clear that you will not give you will not help them when they are doing the evaluation. Explain the basic concepts because I think it is it is always nice for everybody to know the participants to know what is going on; what is the kind of basic interface that they are going to look at. Describe the task given that you must have done until now, task analysis you should actually build a set of tasks; give the task to the user; one task at a time; see them how they react take inputs and then move on to the next task.

(Refer Slide Time: 07:45)

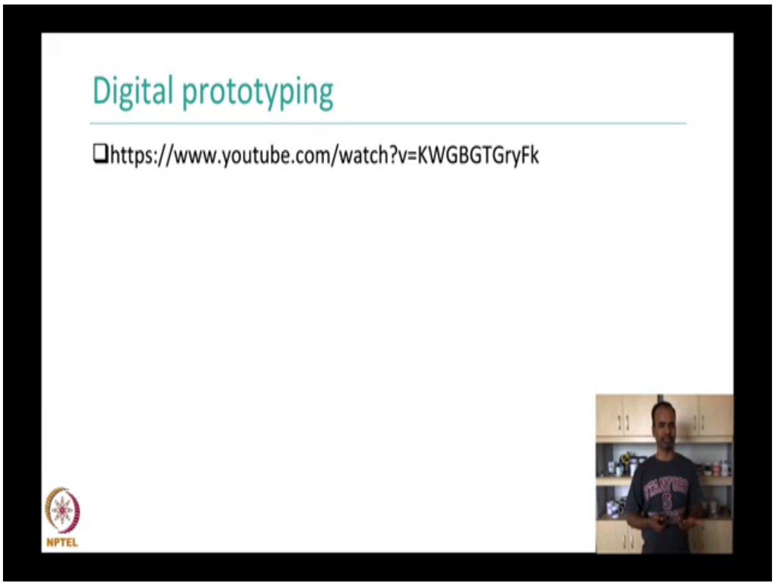


Sketching & Prototyping

❑https://www.youtube.com/watch?v=JMjozqJS44M

As we get out the two videos that I said I will actually let you to watch it. I will also put this link in the YouTube, in the document and the website or on the hand on the YouTube comments, so that you can actually take a look at these videos. Because I really want you to take a look at these videos to understand what is sketching and prototyping is and how there are some nice inputs that you can get in terms of developing these prototypes.

(Refer Slide Time: 08:13)

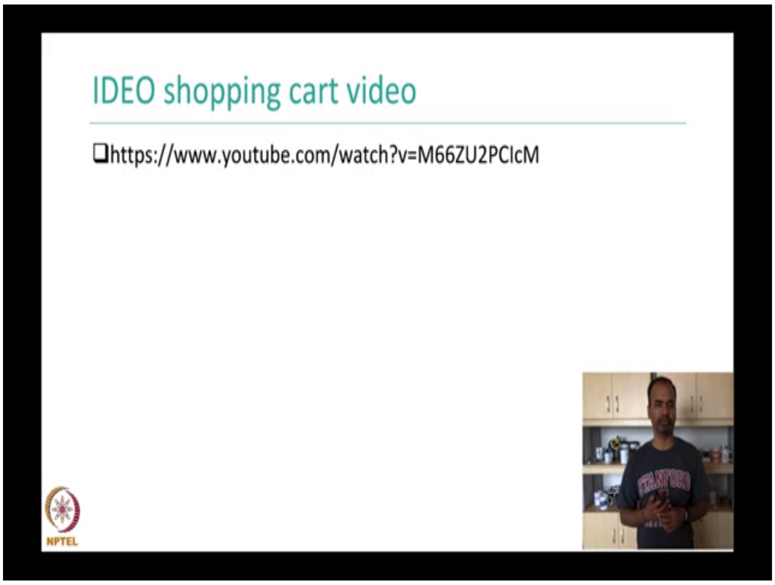


Digital prototyping

❑https://www.youtube.com/watch?v=KWGBGTGryFk

And then it is also about digital prototyping. So, these are three part videos of Google that you will actually look at which talks about sketching and prototyping and then digital prototype. I let it you watch it. I will also come back and talk about these videos after you watched the video. And there is another video that I will actually also put a pointer to if the video for IDEO; where they actually talk about how they go about they creating, getting inputs from the users and trying to develop that as user inputs.

(Refer Slide Time: 08:48)

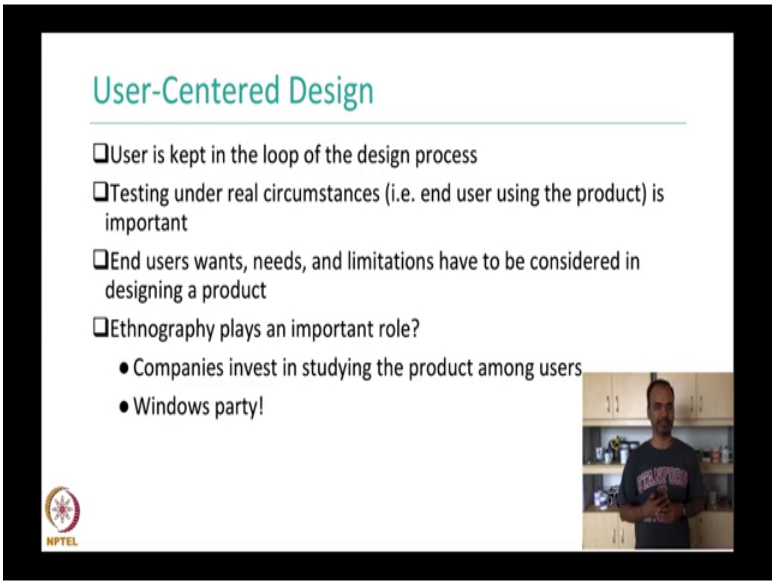


IDEO shopping cart video

□https://www.youtube.com/watch?v=M66ZU2PCIcM

Here is a link to the IDEO shopping cart video that I mentioned. Please take a look at the video. What I will do is I will actually talk about all these three videos later in the week today. Just getting over the details of what was there in the video. So, this will help you also to actually understand the important parts that you should take away from the video. So, I would highly recommend you to watch the videos before you actually watch the part that I am going to be talking about the videos.

(Refer Slide Time: 09:17)



User-Centered Design

□User is kept in the loop of the design process

□Testing under real circumstances (i.e. end user using the product) is important

□End users wants, needs, and limitations have to be considered in designing a product

□Ethnography plays an important role?

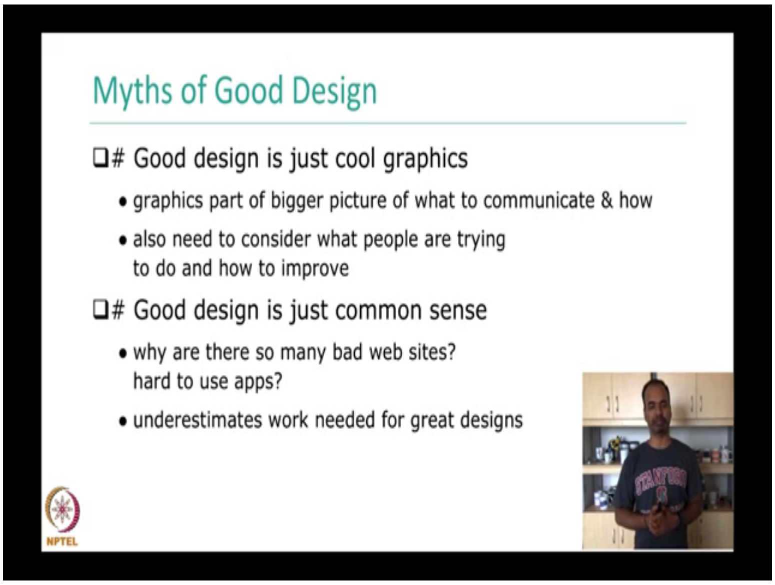
- Companies invest in studying the product among users
- Windows party!

User-Centered Design right: So, now moving on to the topic of we did some iterative design low fidelity brought the users in mind. So, let us just talk a few minutes about just the word or the concept theoretically what is user-centered design; first time I am introducing the word user-centered design. User is kept in loop which you understand that is why I was kind of building it over the users. So, at this point in time it should be very intuitive or sometimes even to some extent trivial Testing under real circumstances which is what you would do contextually your task analysis. End user wants, needs and limitations have to be considered while designing a product which is necessary because otherwise you will end up designing something which users may not use.

Ethnography plays an important role; which probably you have seen even in the past that and knowing how people are using the systems in their own location, in their own environment is actually very very important. Companies invested in the products among users.

Many I have some people who may now work at Intel or even windows, where they actually do give away their products to the end users and they even fund a little bit which is for windows party kind of environment is; they fun with the little bit where you can actually showcase the product that you are going to test among the end users and get feedback and give it back to the company that is given you co product right. It is basically window is taking giving away the operating system of the tool or the application, you use the product at your home, invite your friends to showcase the tool and windows will find out or companies may find out.

(Refer Slide Time: 10:52)



Myths of Good Design

- ❑# Good design is just cool graphics
  - graphics part of bigger picture of what to communicate & how
  - also need to consider what people are trying to do and how to improve
- ❑# Good design is just common sense
  - why are there so many bad web sites? hard to use apps?
  - underestimates work needed for great designs

Let me briefly touch upon these concepts of Myths of Good Design and again let us the slide blank because I wanted you to think about before going to the next slide or before going to the next minute in the video. I would like you to actually think about what are the myths of good design? What do you think there are some myths which people think that is good design but it is really not. Please take a minute or two and think about it and if you have some ideas please pass it on to the mailing list also. I would be keen to look at what you came up before actually looking at the slides.

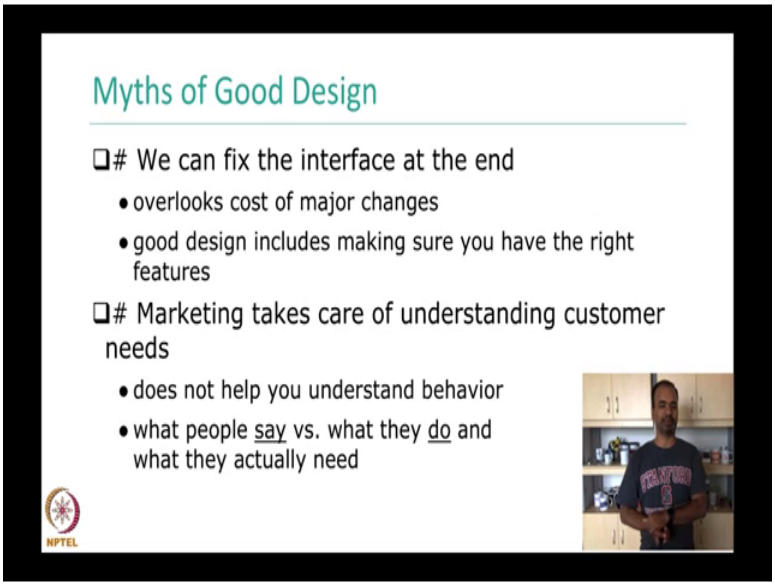
So, you know some I think is actually myths, which is sometimes again it is intuitive. Good design is just cool graphics; people just think that I will just made cool graphics and it should just communicate what I want to communicate with the users, which may not be the case. A graphics a part of big picture of what you want to communicate and how; also need to consider what people are trying to do and how because; just imagine if the Uber app that that you using where the cars are let us take about a half centimetre versus three-fourth of a centimetre; that is probably the distance in which the other size of the car that is placed in your phone.

That pretty imagine if this was a big car; right if it was let us take 2 centimetre by 2 centimetre or 2 centimetre by 3 centimetres. So, that is going to be actually pretty ugly right. So therefore, you want you want to be very clear on what kind of graphics you can use while actually building your solution and you want to be very careful with it. Good design is just common sense; many people actually have argued that design is nothing but just common sense right.

You just have to be you just have to think a through which is what think a lot sessions are? A cognitive walkthrough; you just think a lot is do this cognitive walkthrough; you should have good design but that is not the way. That is right why are dealing that case? Why are there so many websites services that we use? We see which are actually copy in their interface. Underestimate work needed for great designs because I think it is not that very easy to come up with good designs.

Now we have a lot of design patterns which is another part of the course also which I will go through; in detail about what are the design patterns that are available; which designers like you and programmers like us can actually use. But, but I think using design patterns does not just give you good design also right. It does not give you the design that you would like to have. Design patterns are built for something we will actually look at later advantages and disadvantages of design pattern but design pattern helps you generally to get some basic things in place not necessarily everything a good place.

(Refer Slide Time: 13:35)



Myths of Good Design

❑# We can fix the interface at the end
- overlooks cost of major changes
- good design includes making sure you have the right features

❑# Marketing takes care of understanding customer needs
- does not help you understand behavior
- what people say vs. what they do and what they actually need

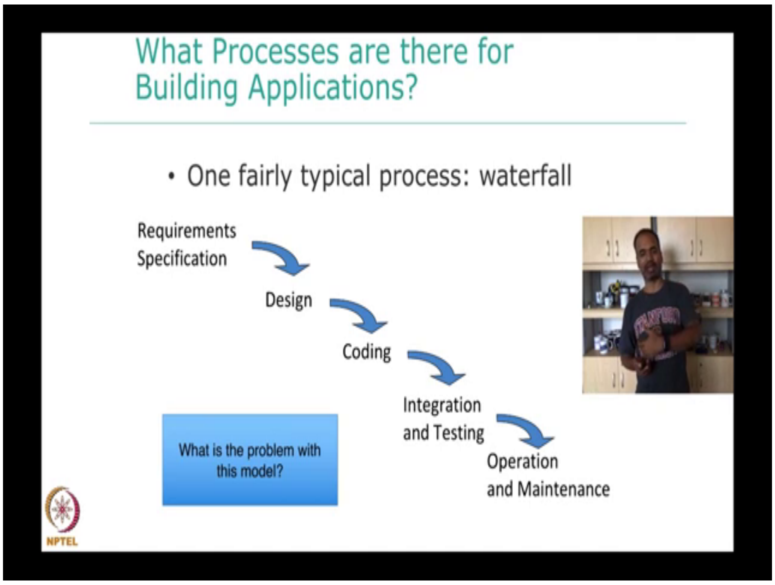
Oh we can fix the interface there. That is probably another way that people would have which is. Let us just build it and then we will think about the interface at the end. We do not really care about the interface to sort of with. All those the cost of needed changes because you come back to it later; there are there are going to be more changes major changes required and it may actually cost you more than actually doing it well in the first time. Good design includes making sure you have the right features right; features, see I think that is always the trade off that you have to keep playing, functionality slash features versus interface right; these two are the things. That you want to always keep balancing out in everything every thought process that you have. Marketing takes care of user understanding; I do not really need to think about users; marketing people will do that and they will just come back and give it to me; which we have actually seen in earlier slide about contextual inquiry and market research right.

Contextual inquiry you get a thousand things about a few people and then market research is actually you will actually get a lot of people but only about a few things. And what people say versus what people do and what they actually do. I actually have a lot of interesting experiences in understanding this particularly in the context of security or the kind of studies that I have done in the context of phishing is what people say that a why actually do not really open the emails which has which is coming from an account that I do not have.

For example, I do not have a bank account in bank A; that bank is sending me an email to change the password I will not open it right. This is the kind of thing what they say. What they actually do is actually something different right. If you send an email saying oh please open this link and you will get a 1000 rupees voucher for this or you get a 35 percent discount on your next visa order or something. People end up actually opening it right. So, what they actually say and then what they actually do and they what they actually need is actually very different right.

So, with that I will actually stop about what the myths of good design are. Now I will actually get into how to motivate. What I am what we have been talking about called interactive design; there is another software engineering view of the same thing also.

(Refer Slide Time: 15:47)



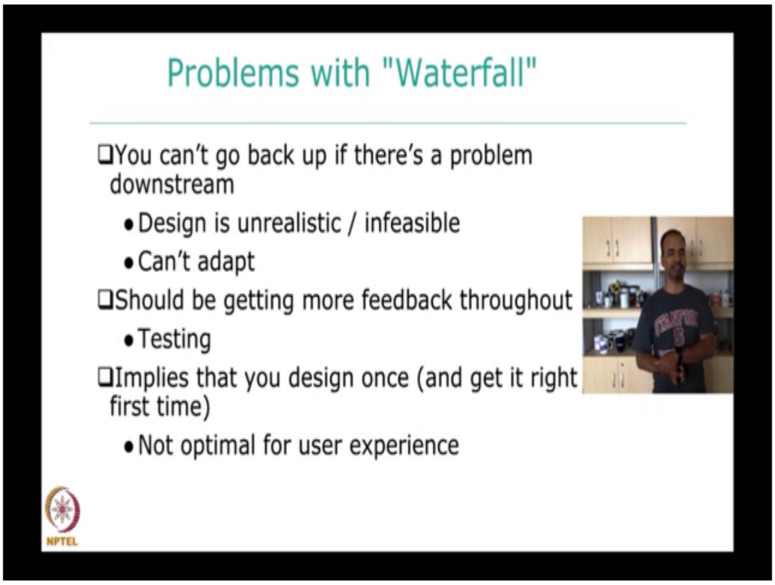
What Processes are there for Building Applications?

- One fairly typical process: waterfall

Requirements Specification
Design
Coding
Integration and Testing
Operation and Maintenance
What is the problem with this model?
NPTEL

I am sure many of you have seen this waterfall process of designing applications building. Application which is go to the end user like system requirements, design code, integrate and test and then operations and maintenance right. So, this is a very waterfall flow which is everything goes down, developing things you it is hard to come back because you do a integration and testing which you want to actually go back will fix some design or design requirements it is actually going to be hard. I have been part of some projects where this kind of processes is being followed and it is actually extremely hard to go back and fix. Every fixes that you want to carve is also cost right. It is it is expensive; it is hard to get the developers to convince to be convinced to actually make the changes also. Right the problems are it is in one direction you cannot actually go back and fix all that right.

(Refer Slide Time: 16:39)

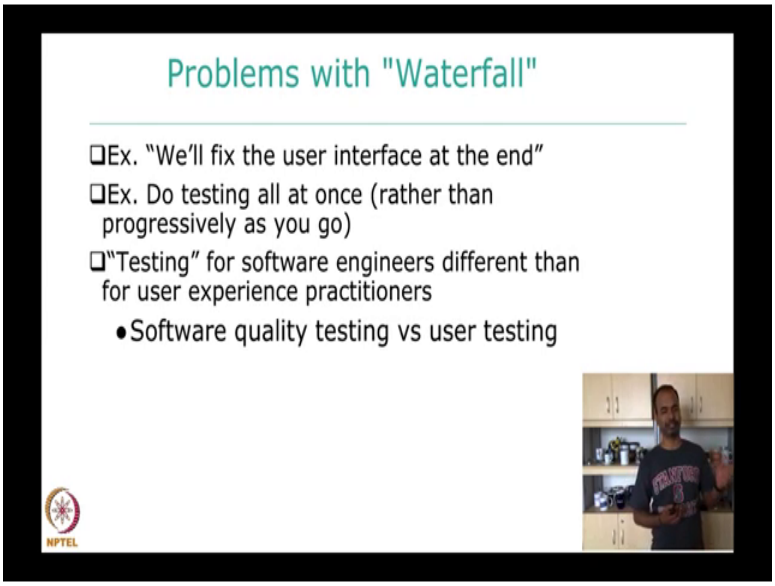


Problems with "Waterfall"

- ❑You can't go back up if there's a problem downstream
  - Design is unrealistic / infeasible
  - Can't adapt
- ❑Should be getting more feedback throughout
  - Testing
- ❑Implies that you design once (and get it right first time)
  - Not optimal for user experience

You cannot go back up with if there is a problem in downstream should be getting more feedback throughout the testing; implies that you decide ones not optimum for user experience. And so, it is it has these so to say assumptions which are being made which may or may not be true. It is actually hard to get the design work done through this waterfall design will fix the user interface at the end.

(Refer Slide Time: 17:00)

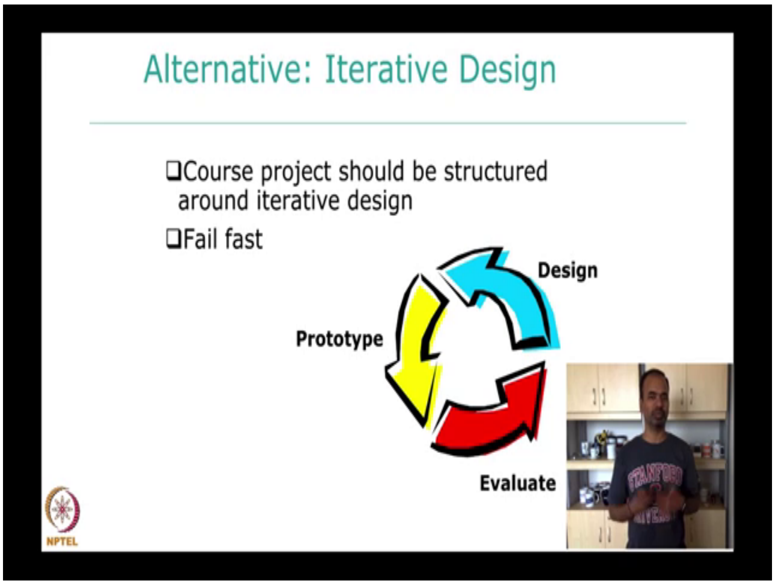


Problems with "Waterfall"

- ❑Ex. "We'll fix the user interface at the end"
- ❑Ex. Do testing all at once (rather than progressively as you go)
- ❑"Testing" for software engineers different than for user experience practitioners
  - Software quality testing vs user testing

Do testing all at once testing for software engineers different than user experienced right. So, now I have actually many of my students now who were working as fulltime, full stack; so to say design experience or user experience engineers right. We may have heard about full stack developers; now there is also fullstack design engineers, user experience engineers right. What are these people these; people are actually starting from the way of starting from the way of thinking of a solution, converting it to the final solution, deploying and getting user feedback and revising it; everything that they do. That is what the full stack designer is; now full stack user experienced engineer. Whereas, these kind of things right for software engineers is different; for the way that testing will be done by a user experience engineer is very different from my testing that will be done by a software engineer. So, those are the problems with the waterfall model.

(Refer Slide Time: 17:59)



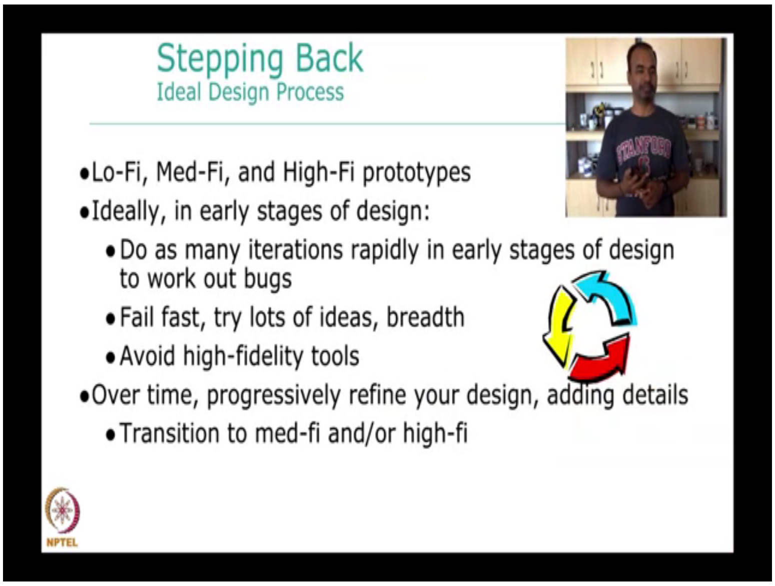
Alternative: Iterative Design

- ❑Course project should be structured around iterative design
- ❑Fail fast

Let us look at what do we want to do right. In this course we have been emphasizing this whole idea of user-centered design, iterative design. So, I would like you to actually take a look at this slide which actually talks about user-centered design, course project should be structured on the iterative design which is that; I really like would like to see some of you do some ideas, try out with your own apps that you want to build or take up a problem that is in there with respect to the interface and try and actually modify it right.

Which is that faith fast is an idea by which you say I am going to design, I am going to get some prototype down, I go get some feedback, I fix it and then come back redesign it right; fix it with terms of evaluating and then come back and redesign it. So, that is the idea for Fail fast which is actually a quite very useful technique in terms of getting your designs better. I think in terms of programming these are the terms are called agile programming and a topics around that.

(Refer Slide Time: 19:00)



Stepping Back

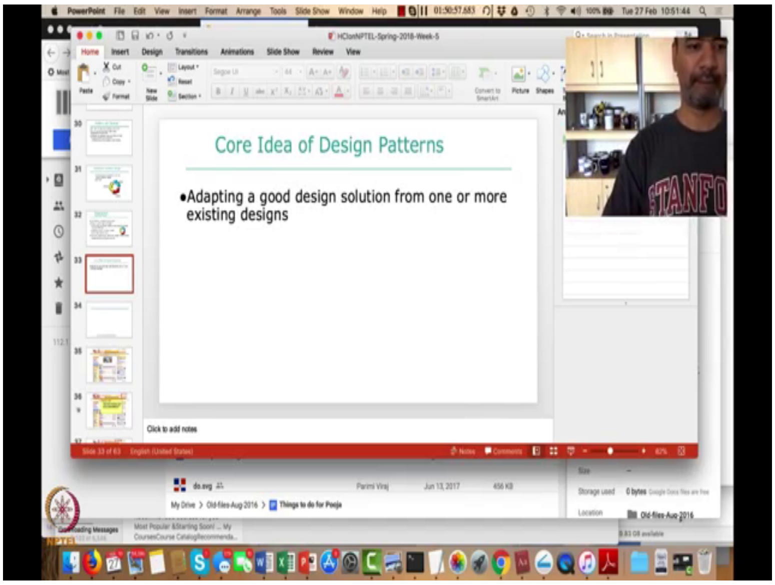
Ideal Design Process

- Lo-Fi, Med-Fi, and High-Fi prototypes
- Ideally, in early stages of design:
  - Do as many iterations rapidly in early stages of design to work out bugs
  - Fail fast, try lots of ideas, breadth
  - Avoid high-fidelity tools
- Over time, progressively refine your design, adding details
  - Transition to med-fi and/or high-fi

Ideal design process is that low fidelity, medium ability and then high fidelity prototypes. Ideally, in the early stages of design as many equations as rapidly done fail fast, try a lot of ideas, do the breadth, avoid high fidelity tools, over overtime progressively refine your design and then get to the high fidelity right. In one slide I think that captures the entire course in that sound sense right. You want to really do low fidelity, you want to really do high fidelity when you want to go from low to high with some multiple iterations in between and many inputs should be taken before you go to the final products alright.

So, with that I am going to wrap up this idea of user-centered design giving you an introduction and then even talking about this low fidelity, high fidelity and ideas of actually how to build it; how to create these prototypes right. So, now what I am going to be around to is I am going to get on to a topic called design patterns.

(Refer Slide Time: 19:54)



Core Idea of Design Patterns

- Adapting a good design solution from one or more existing designs