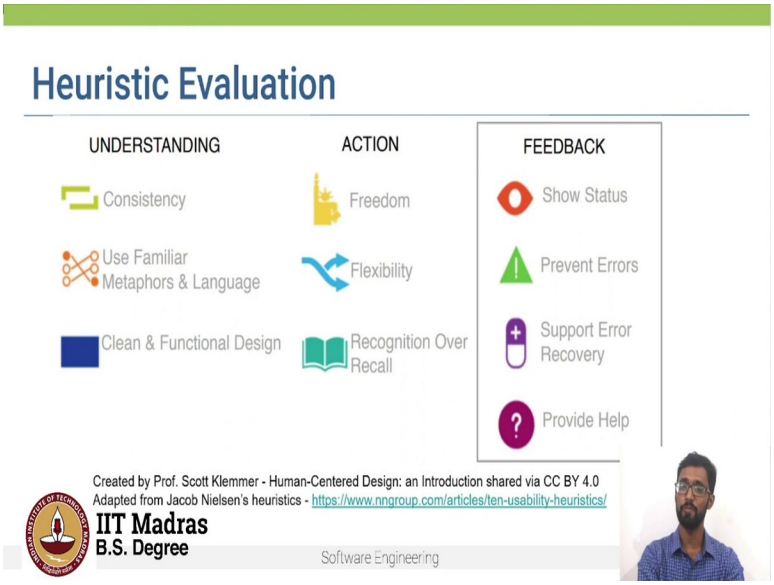
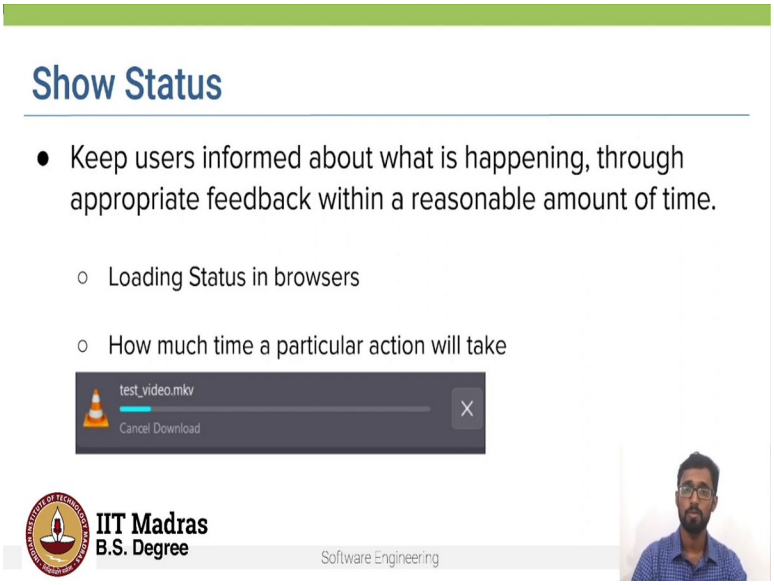
Software Engineering
Professor Sridhar Iyer
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Professor Prajish Prasad
Computer Science
FLAME University
Evaluation using Design Heuristics - Heuristics for Feedback

(Refer Slide Time: 00:24)



In this video, we will be looking at certain heuristics for feedback, for providing appropriate feedback to users.

(Refer Slide Time: 00:35)

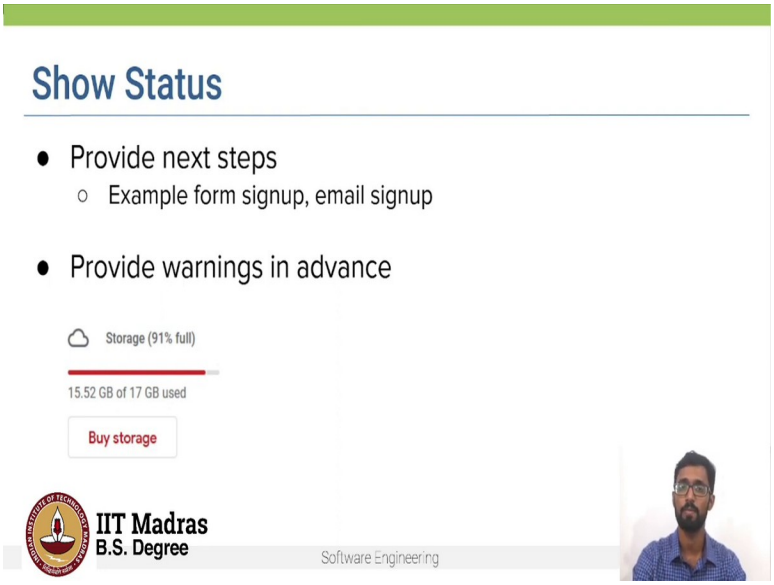


So, the first heuristic is showing the status. So, this means that we need to keep users informed about what is happening through appropriate feedback. And this feedback should be provided within a reasonable amount of time. You have seen multiple instances of this in several applications.

So, for example, the loading status in browsers, so, it can be very frustrating if I have to stare at a blank screen while the page is loading. So, is the page actually loading, is there an error. So, if I do not have an animation, which shows the loading status, then it is very difficult for me to understand what the problem is.

Another example is showing status of how much time a particular action will take. So, let us say if I am downloading something, how much time will it take to download, so that is a useful information which can be given to users.

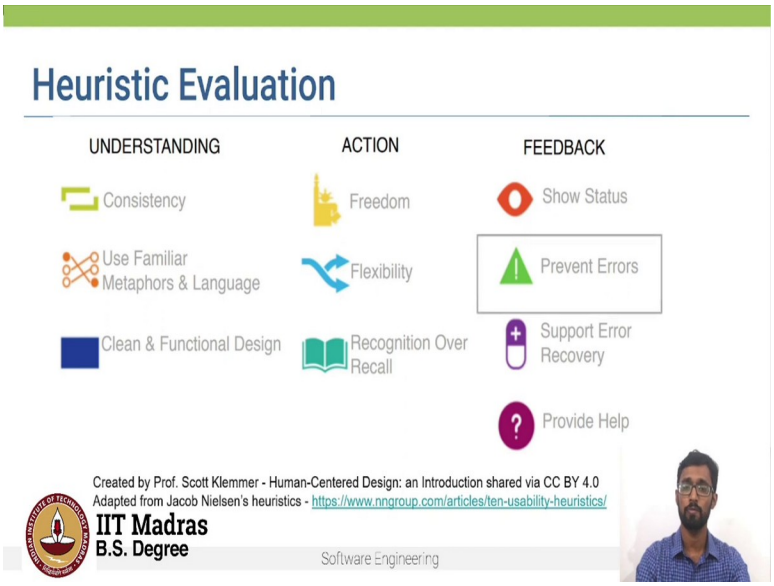
(Refer Slide Time: 01:44)



Another example of showing status is to provide next steps to an action. So, for example, if you have filled a form or if you have created a new email address, so what are the next steps? Otherwise, as a user, I will not know what I should do next.

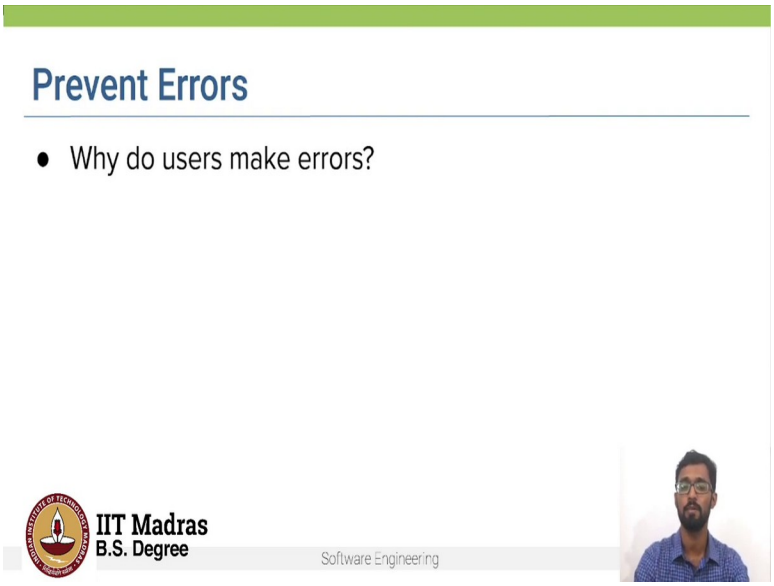
Another example of showing status is to provide warnings in advance. So, it is important to let users know of some action which is needed from their side. So, for example, if the Gmail space is running out, appropriate information on Gmail can help me take appropriate steps such as buying storage.

(Refer Slide Time: 02:43)



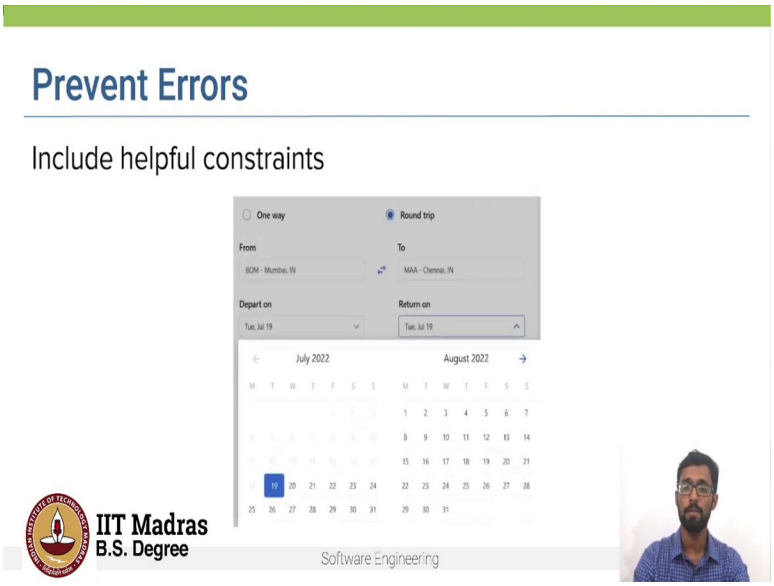
The next heuristics are for preventing errors.

(Refer Slide Time: 02:48)



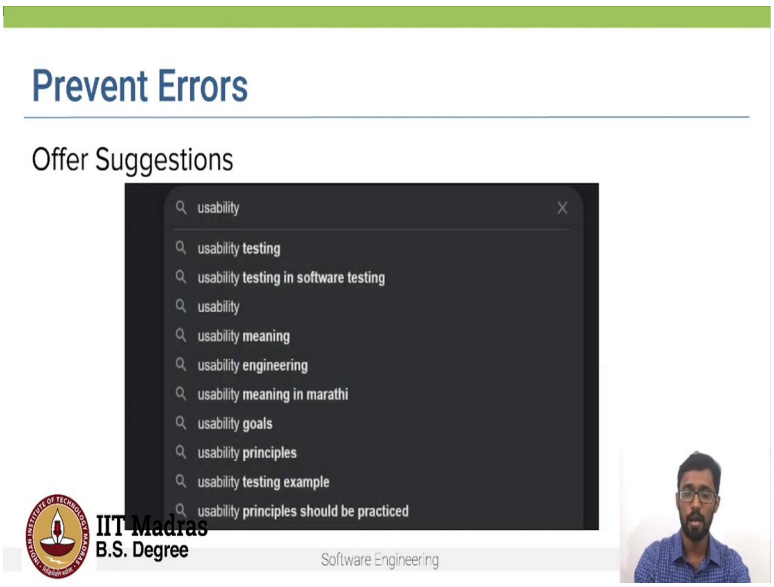
So, why do users make errors? So, many times we often attribute errors which users make to something that the user did. But it can also be the case that certain aspects of the user interface lead users to make these errors. So, what are ways in which we can prevent users from making errors in the first place?

(Refer Slide Time: 03:18)



One example can be to include helpful constraints. So, for example, this is an interface for booking a flight, booking a flight from and the return journey as well. And here you can see that the return date cannot be before the departure date. So, I have mentioned the departure date as July 19. And the dates before July 19, have been grayed out. So, in this way, the interface is constraining the user by not allowing them to fill a date before 19 July. And hence we can prevent errors of a user adding a date which is before the departure date.

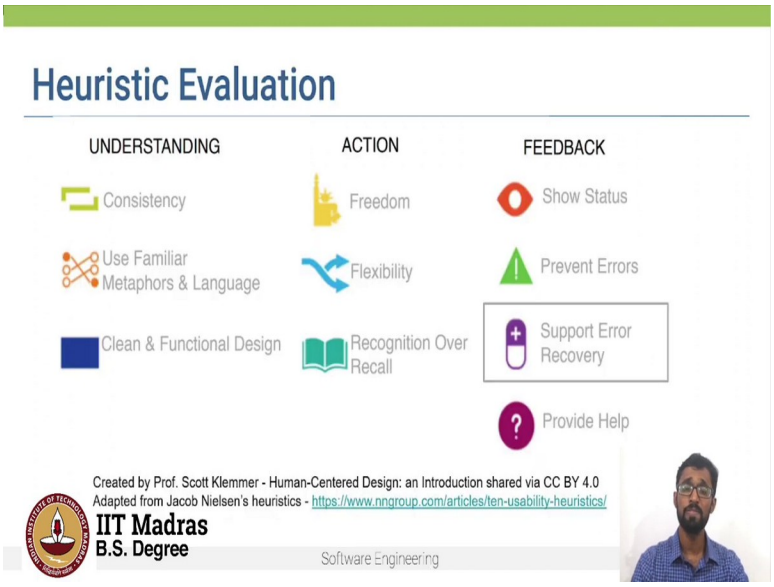
(Refer Slide Time: 04:13)



Another way in which we can prevent errors is to offer suggestions. For example, I may really not know what to do. And the user interface can provide suggestions so that errors will

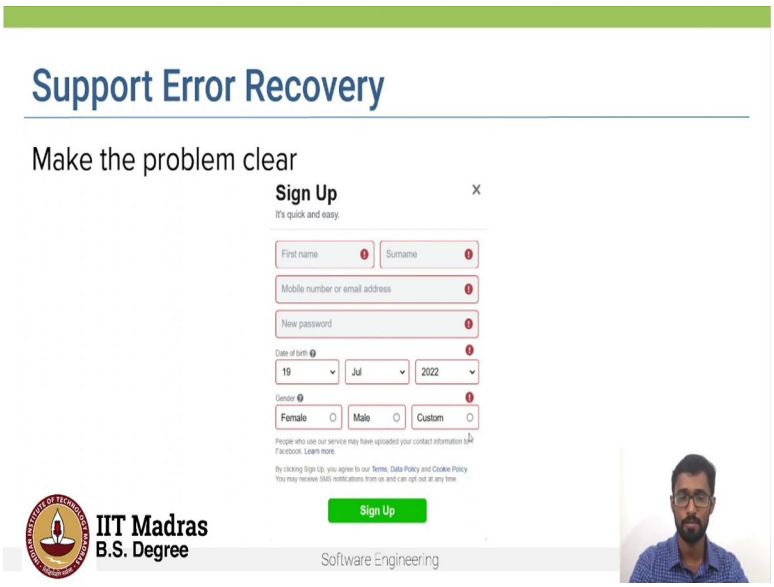
be prevented later on. So, for example, on the browser, I type usability, and I am really not sure what I am looking for. But Google auto suggests provides me a list of options as we see here. And this is useful for me to choose the particular aspect of usability I am looking for.

(Refer Slide Time: 05:00)



Now that we have seen how we can prevent errors the next heuristic is to support error recovery. So, how can we help users once they have made an error? How can we support them to recover from these errors?

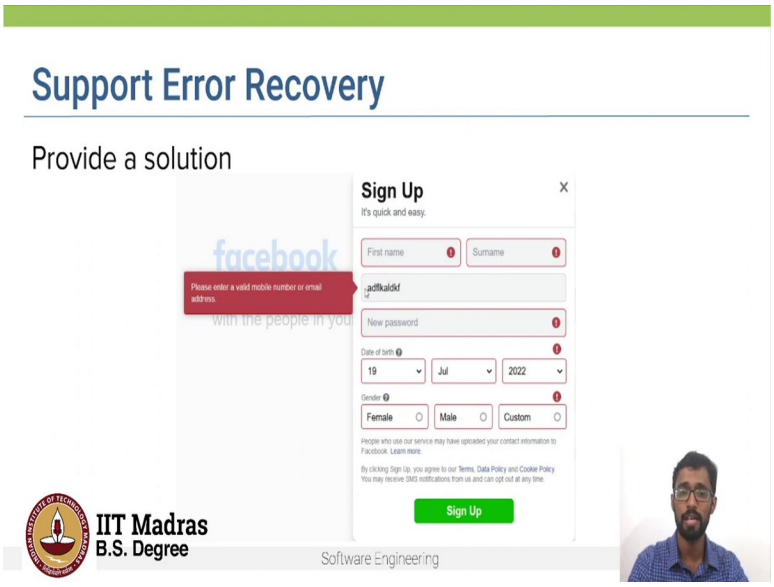
(Refer Slide Time: 05:17)



So, one way to do it is to make the problem itself very clear. So, users might be interacting with your UI for the first time. And it is essential that you specifically mentioned what the

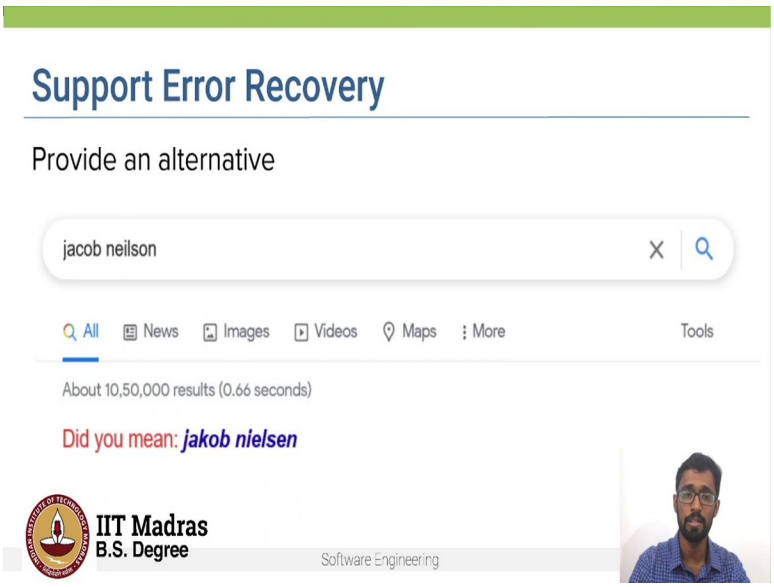
problem is. So, for example, this is Facebook signup form. Here, a user has clicked on sign up without entering any of the details. And here you see that each of these fields are highlighted in red. And thus, the interface makes the problem really clear for users.

(Refer Slide Time: 06:02)



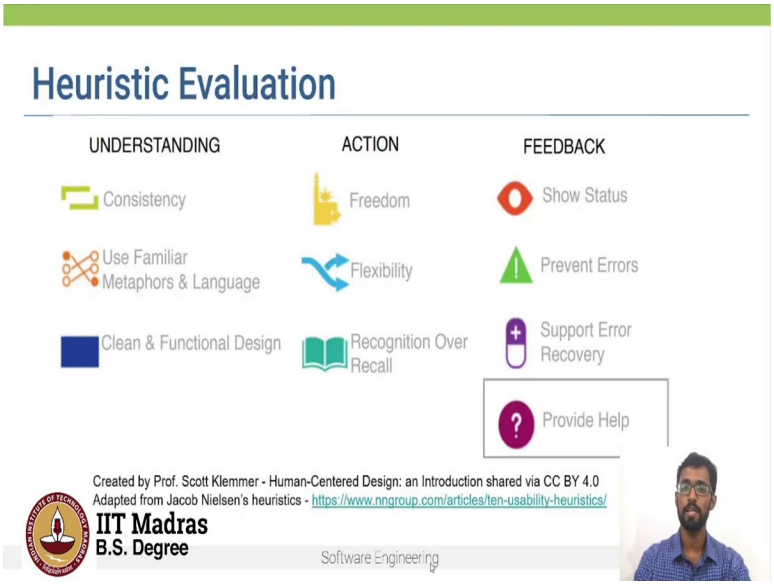
And many times, we should not only make the problem clear, but we need to provide a solution as well. So, in the same interface, you can see here that in this field, the user had to enter an email, a valid email or a mobile number. And the tooltip here specifies the solution to this problem. So, it is important that our user interfaces provide appropriate solutions when users make errors.

(Refer Slide Time: 06:36)



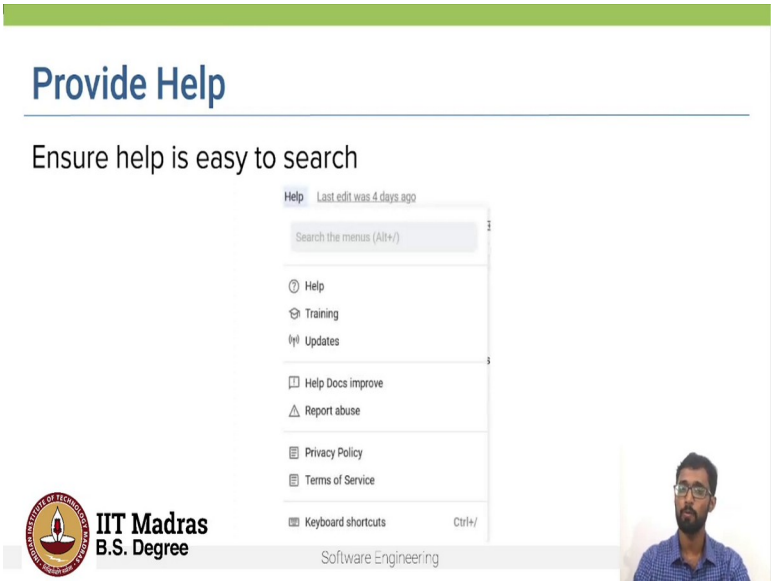
Another way to support error recovery is to provide an alternative to users. There might not be an easy solution or the right solution. But you can always offer different alternatives or different pathways for learners to reach their goal. So, for example, here in the Google search, I have typed in Jacob Neilson and the spelling of Neilson is wrong. And many times you have seen Google auto suggesting the correct search query. So, this is an example of how the user interface is providing an alternative for the learner and helps them support or helps them recover from their error.

(Refer Slide Time: 07:26)



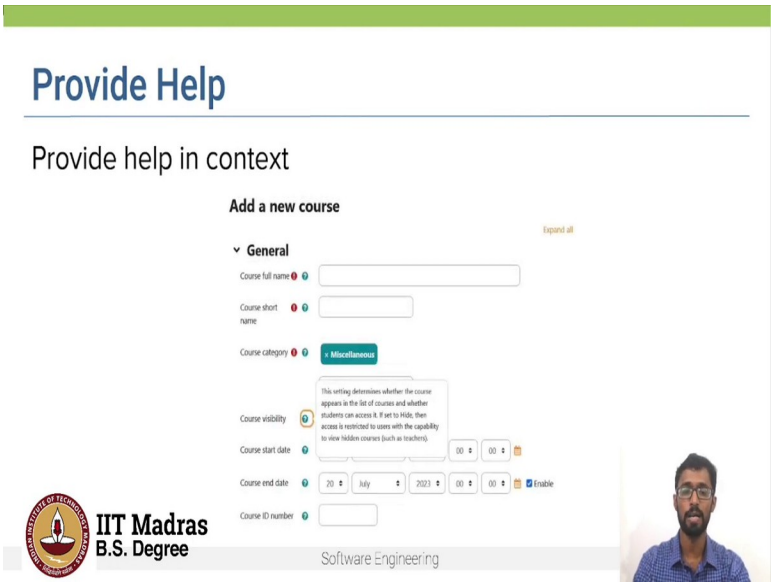
The final heuristic is providing help to users.

(Refer Slide Time: 07:32)



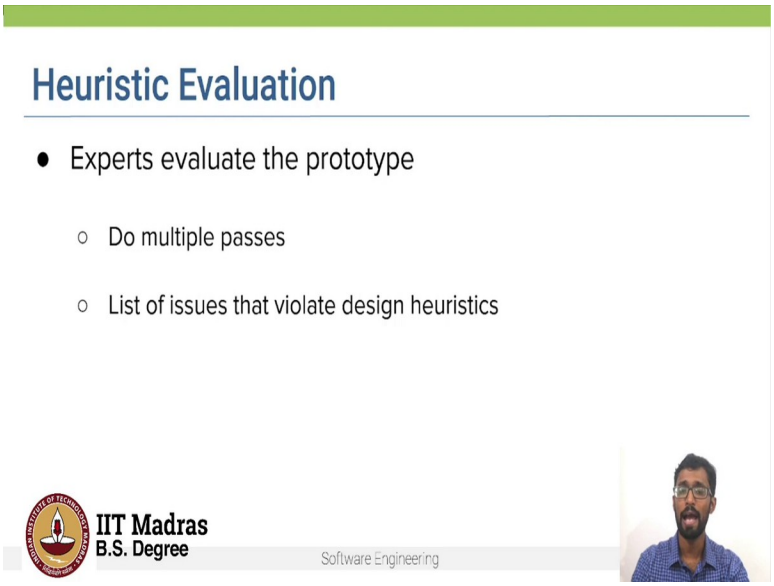
So, we need to ensure that help is easy to search. And you must have seen that most tools have a menu item known as help, which provides important help topics, trainings, updates, etc.

(Refer Slide Time: 07:49)



Another important aspect is to provide help in context, right in time, when and where the user needs it. This is the interface of a learning management system known as Moodle. And in this case, you can see that if I want to add a new course, I can fill the details of the course and the course gets created. However, as a new user, I may not know exactly what course visibility means. So, just next to each attribute, like the course category, the course visibility, you have a Help icon, which will provide you details about that particular field. This way helps us provided in context, exactly where and when the user needs it.

(Refer Slide Time: 08:48)



So, in this and in the previous videos, we have seen several heuristics by which we can evaluate user interfaces. So, after the initial prototypes are ready, you can allow experts to evaluate your interface. So, experts, they do multiple passes through your prototype, and produce a list of issues that violate some of these design heuristics. So, collecting feedback from several experts will help you fix multiple usability issues in your prototype. And these heuristics are also kind of a self-check for you and can enable you to design effective and usable software interfaces.