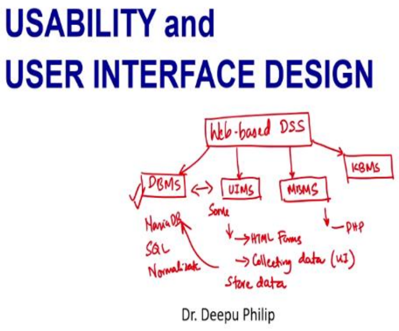
Computer Aided Design Systems
Industrial Practices using Big Analytics
Professor Deepu Philip
Department of Industrial and Management Engineering
Indian Institute of Technology Kanpur
Professor Amandeep Singh
Imagineering Laboratory
Indian Institute of Technology Kanpur
Lecture 40
Usability and User Interface Design

Good evening, everyone. Welcome to yet another lecture of the Web-Based Decision Support Systems for business decisions. This is Doctor Deepu Philip, I am from IIT Kanpur, and this is a new lecture in the series in the subcomponent of what we call as the User Interface.

(Refer Slide Time: 00:39)



USABILITY and
USER INTERFACE DESIGN

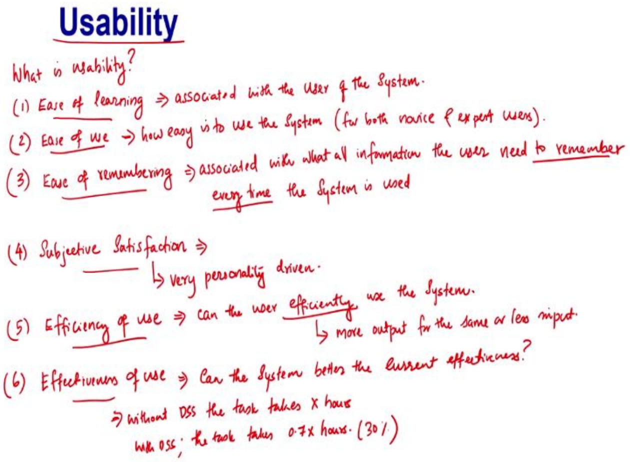
Dr. Deepu Philip

So, if you quickly look into what we have been discussing in the class, we have what we call the Web-based DSS, and we have already covered the major topic as part of it as DBMS (Database Management System), then the second part we were supposed to cover a little bit of it. We are a little bit covered, it is UIMS (User Interface Management System), then the third one is MBMS (Model-based Management System) and then the last one we had was KBMS (Knowledge-based Management System). And, we are already in the DBMS, we have covered most of the aspects of it, and we have covered some aspects here. And, now we will be covering more aspects here.

We will do what we call HTML forms and then collect data through UI and then integrate these two. How do you store data? So, those aspects we will be discussing in the coming lectures as part of it. And then we will go into what we call the MBMS (the Model-based Management System). where we will be learning a little bit of about, we will combine some of this including PHP and etcetera like that, how to code those aspects.

So, you might have already studied the different aspects of Maria DB or MySQL, etcetera. You studied SQL, you read normalization, all those kinds of things. So, that was part of what we call our DBMS. Now we try to integrate the User Interface Management System for that. So, we already did a basic lecture on Usability and User Interface, this is the advanced version of it before we move into HTML forms.

(Refer Slide Time: 02:39)



Usability

What is usability?

(1) Ease of learning ⇒ associated with the user of the System.

(2) Ease of use → how easy is to use the System (for both novice & expert users).

(3) Ease of remembering ⇒ associated with what all information the user need to remember every time the System is used

(4) Subjective Satisfaction ⇒
↳ very personality driven.

(5) Efficiency of use ⇒ can the user efficiently use the System.
↳ more output for the same or less input.

(6) Effectiveness of use ⇒ Can the System better the current effectiveness?
⇒ without DSS the task takes x hours
with DSS; the task takes 0.7x hours. (30%)

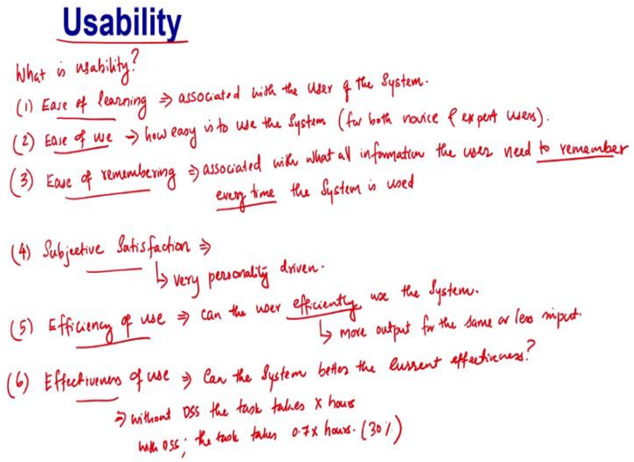
So, without much delay, let us talk about Usability. And, the fundamental question that a lot of people ask is, what is Usability? What is it? There are so many ways, so many definitions of this available. The one major definition of it is available, it is

1) Ease of learning- Ease of learning with whom? It is associated with the user of the system. How quickly can the user learn the system and gain proficiency in its usage?
2) Ease of use- Again, how easy is it to use the system? This is more likely (for both novice and expert users). That discussion is part of the ease of use.

3) Ease of remembering- So, the question is associated with, what all information the user needs to remember every time the system is used. So, the idea here is that what all things the user has to remember every time, how many things that you need to remember all the time, is that part of it.
4) Subjective satisfaction- We know that satisfaction is a very personality driven trait. So, in this case, satisfaction is personality driven. So, the subjective satisfaction is, can most of the people who are associated with the subject be satisfied or given the wide variety of people who are using it, can the majority of them be provided obtain satisfaction by using the system?
5) Efficiency of use- Can the user efficiently use the system. So, efficiency can be, think about it as more output for the same or less input. So, this is the thought process we have. Can you efficiently use a system? So, that is an efficiency angle to this.
6) Effectiveness of use- Can the system better the current effectiveness. So, for example, without DSS, the task takes x hours. With DSS, the task takes 0.7 x hours. So, a 30 percent time reduction is possible in this regard. So, this means your effectiveness as a decision maker has improved as part of this.

So, just going through this, we talked about Ease of learning once, then Ease of use, then Ease of remembering, Subjective satisfaction, Efficiency of use and Effectiveness of use. These are the major aspects from which the discussion of Usability comes into picture in Decision Support Systems.

(Refer Slide Time: 07:56)



Usability

What is usability?

(1) Ease of learning ⇒ associated with the user of the System.

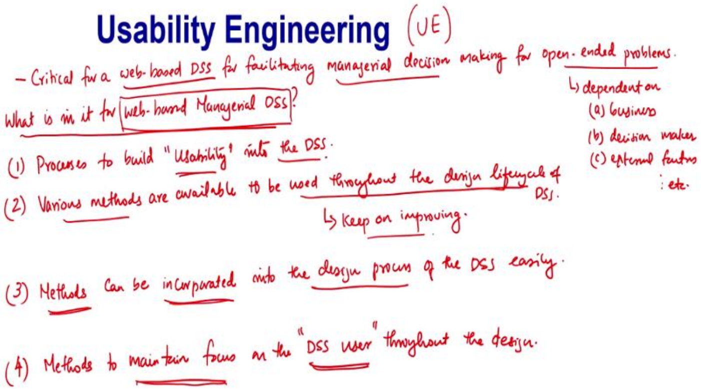
(2) Ease of use → how easy is to use the System (for both novice & expert users).

(3) Ease of remembering ⇒ associated with what all information the user need to remember every time the System is used

(4) Subjective Satisfaction ⇒
↳ very personality driven.

(5) Efficiency of use ⇒ can the user efficiently use the System.
↳ more output for the same or less input.

(6) Effectiveness of use ⇒ Can the System better the current effectiveness?
⇒ without DSS the task takes x hours
with DSS; the task takes 0.7x hours. (30%)



Usability Engineering (UE)

– Critical for a web-based DSS for facilitating managerial decision making for open-ended problems.
↳ dependent on
(a) business
(b) decision maker
(c) external factors
: etc.

What is in it for web-based Managerial DSS?

(1) Processes to build "Usability" into the DSS.

(2) Various methods are available to be used throughout the design lifecycle of DSS.
↳ Keep on improving.

(3) Methods can be incorporated into the design process of the DSS easily.

(4) Methods to maintain focus on the "DSS user" throughout the design.

So, then there is a branch, which is called 'Usability Engineering', sometimes people call it UE. And that is very critical when you develop a User Interface for a Decision Support System.

- It is critical for a Web-based DSS for facilitating managerial decision making. You are doing managerial decision making for open-ended problems. So, the idea here is that Usability Engineering is very critical, especially when you have Web-based DSS that facilitate managerial decisions. And, these problems are open-ended, it is dependent on the business, decision maker, external factors, etcetera, these are all part of this. Now, how do you build?

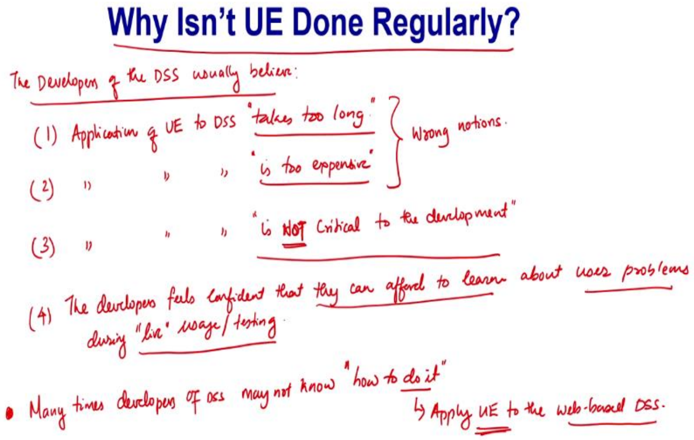
So, then what the question is, what is in it for Web-based managerial DSS? That is the aspect we need to think about is,

1) It includes processes to build Usability into the DSS. It can be into product services, etcetera, but for us it is, we can build Usability into the DSS. Usability in a sense, when we talk about this, it can talk about the ease of learning, ease of use, all these kinds of things. So, it allows processes to build this Usability into the DSS.
2) Various methods are available to be used throughout the design lifecycle of DSS. So, there are so many methods available, multiple methods are available that can be used throughout the design lifecycle. It is not just a one-time process, it is repetitive, so you keep on improving or keep on bettering. So, many of these aspects, effectiveness, efficiency, etcetera, can be improved as part of this
3) Third aspect of this is methods can be incorporated into the design process of the DSS easily. So, you can incorporate methods, you can incorporate processes, practices, etcetera, that you can incorporate them into the design process. So, when you are designing it, in various stages you can do that.
4) The last and most important part is methods to maintain or methods are available to maintain focus on the DSS user, throughout the design. So, the centric aspect, the main focus, is to keep on focusing on the DSS user. The person who is going to use the Decision Support System to make the decision or assist with the decision making that needs to be thought through in the design.

So, the question is, if you look into it, in the Web-based managerial DSS, why do we use Usability engineering? It is very clear with this that you can build Usability into the system. There are so many methods available.

You can use it throughout the design lifecycle so that you can keep on improving the system. And, there are various methods that are available that can be incorporated in the entire design process of the DSS. It is a design and update, that is kind of thing. And, it also ensures to maintain focus on the DSS user throughout the design.

(Refer Slide Time: 13:10)



Why Isn't UE Done Regularly?

The Developers of the DSS usually believe:

(1) Application of UE to DSS "takes too long"

(2) " " " "is too expensive"

Wrong notions.

(3) " " " "is NOT Critical to the development"

(4) The developers feels confident that they can afford to learn about users problems during "live" usage/testing.

- Many times developers of DSS may not know "how to do it"

↳ Apply UE to the web-based DSS.

So, then if it is so beneficial to people and it will help the user, help other people, etcetera, etcetera, etcetera. So, then the fundamental question is, why is not UE done regularly or user experience? Why do not we do that diligently in this? So, the main aspect is the developers of the DSS usually believe, the developers, the people who develop the DSS usually believe that, what do they believe?

1) Their main belief is that application of UE to DSS takes too long. This is one of their false beliefs. If you want to apply user experience to the DSS, it will take a very long process.
2) Then, the second one is that the application of UE to DSS is too expensive. This is their other thing.

So, these are two, what do you call as, wrong notions or wrong beliefs as part of this. So, these two things were what developers believed.

3) Then, the third one is they believe that application of the UE to DSS is not critical. The key word is not. It is not critical. It is not critical to development. This is another thought process is not critical to the development. This is what another belief is. It is not a wrong notion, but it is more like it is a belief. The belief that you can get away with it.
4) Then, the fourth one is that these two are tied together. The developers feel confident that they can afford to learn about user problems during live usage or testing. So, they start believing that you do not need to do it from the beginning, you can ignore the user,

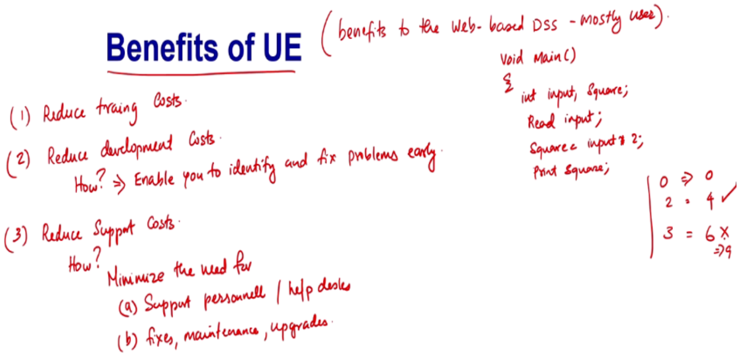
develop something. And then, once you do the live usage or testing, then document what the user problems are and then let us learn from it and let us fix the problem at that point.

So, these are the four major notions, or four major aspects why developers do not or developers shy away from doing user experience regularly. And also, last part is there is another one, another major important part is this is not a very well documented aspect.

- ❖ Many time developers of DSS may not know how to do it. So, what is the way to do it? The way to do it is, apply UE to the Web-based DSS.

They might be good at developing decision models, they might be good at building a forecast model, a classification model, this model, that model, all those kinds of things. They might be very good at it, but they may not know very well how to apply a user experience model to the Web-based Decision Support System. So, how to do this, actually? You may want to say, I want to apply user experience, but they may not know how to do it. So, that is also one other aspect that you should not ignore.

(Refer Slide Time: 17:28)



Benefits of UE

(benefits to the Web-based DSS - mostly user).

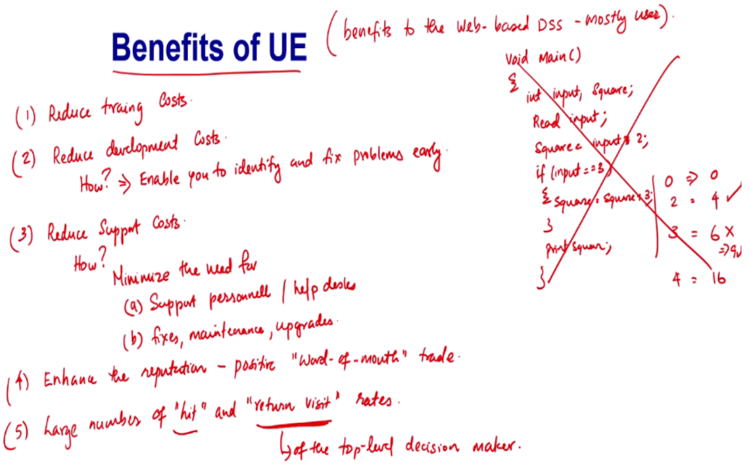
(1) Reduce traing Costs.

(2) Reduce development Costs.
How? ⇒ Enable you to identify and fix problems early.

(3) Reduce Support Costs.
How?
Minimize the need for
(a) Support personnell / help desks
(b) fixes, maintenance, upgrades.

```
Void Main()
{
  int input, Square;
  Read input;
  Square = input * 2;
  Print square;
```

```
0 ⇒ 0
2 = 4 ✓
3 = 6 ✗
     ⇒9
```



Benefits of UE

(benefits to the Web-based DSS - mostly user).

(1) Reduce traing Costs.

(2) Reduce development Costs.
How? ⇒ Enable you to identify and fix problems early.

(3) Reduce Support Costs.
How? Minimize the need for
(a) Support personnell / help desks
(b) fixes, maintenance, upgrades.

(4) Enhance the reputation - positive "Word-of-mouth" trade.

(5) Large numbers of "hit" and "return visit" rates.
↳ of the top-level decision maker.

```
Void Main()
{
  int input, Square;
  Read input;
  Square = input * 2;
  if (input == 3)
  { Square = Square * 3;
  }
  Print Square;
}
```

```
0 ⇒ 0
2 = 4 ✓
3 = 6 ✗
      ⇒9
4 = 16
```

So, then what are the benefits of user experience? How does it benefit? So, it is from the benefits to the Web-based DSS, mostly user, but also to the developer as well. But then, some of the thing developers may not want to worry about.

1) So, the first one is, the major one is, reduce training costs. Most of the time the Web-based Decision Support System, mostly meant for middle and upper-level management. So, these people do not really have too much time with them to actually go and sit and train for months together. They would rather like to come sit for a few hours, go through the system and quickly grasp it. So, if you have to conduct 10 months training for the decision maker, that is going to be very, very expensive. So, if you have a good user experience, you are added into the Web-based Decision Support System design, then the training cost for the system can be significantly reduced.
2) Reduce development cost. And, this is usually a question, how does this happen? The question is, how do you do, by UE, how do you reduce the development cost? The answer to that is to enable you to identify and fix problems early. You can identify and fix these problems early you do not have to wait for a long time to do this.
3) Then the third part is, the third major thing is to reduce support costs. So, how do you reduce the support cost? How? This is the other question. So, the main thing is minimizing the need for'

a) So, you can minimize the need for support personnel, or help desks. So, this is the support of the system or software systems are usually very, very expensive, people sometimes say can be even up to 60 percent of the development cost. So, that aspect can be reduced as part of it.

b) It usually fixes, maintenance, upgrades. This fix is something that is, these bug fixes, upgrades, maintenance, extra, they also eat up a lot of time and a lot of money. And these contracts of maintenance and up-keep are usually very expensive. I usually make a funny example, as part of this, I sometimes tell people to understand why this bug fixing or this software maintenance, if you do not design it properly, how it would impact. So, assume that you are in the process of designing a Decision Support System to find the square of a number. So, you gave a developer to do this, and the developer came up with a function or something like that. It is like,

```
Void Main ()
{
int input, Square
Read input;
Square input 2;
Print Square;
```

So, now what happens is you will do the testing and you will say, 0. The answer is 0. 2, the answer is 4. Program is good, or the Decision Support System is good, let us release it to the user. And then, the user goes there and he tries to find the score of 3 and in this method, he will get the answer of 6, where the square of 3 should be 9. So, this is a wrong answer, so they will report a bug.

So, then they will go there and say, oh, I see we will fix the problem. Then we will say this here, something like this, we will modify this and we will say,

```
If (input = 3)
{
```

```
Square= Square+3;
}
Print Square;
}
```

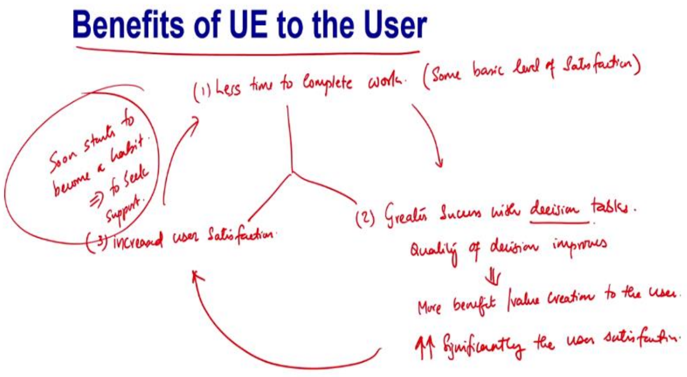
You are done. Now, when you run 3, you will get the output as 9, and the bug is fixed. So, now what will happen is, they will say, okay, what will happen when you put 4? You should get 16.

Now if you do this, it is only 8. So, now what do you do? Oh, we will add another if clause, and for all numbers until infinity, you will keep on adding this clause, which is what I will call it as fixing, bug fixes. It is a stupid way of designing a code, but you probably get the point across that if you have not thought through the logic, you have not kept the user in mind and you really have not decided what to do properly, then this kind of whole nonsense will come into picture. So, my suggestion in that regard is these kinds of fixes, maintenance and upgrades, which is not thought through and is usually a very expensive thing to happen.

4) Then, the fourth point that we want to talk about is enhancing the reputation. That is, I would call it a positive "word-of-mouth" trade. So, the idea here is that if your User Interface, if your system is very usable and the top decision maker, the CEO or CX or CFO whoever it is, finds a system very usable and they can actually do a good job about it. Then what happens is they will also go to other people and say, well look, we bought a Decision Support System from these people. They designed it, they made it, and it is so good, and this company is really good. So, the developer would end up getting a very good word-of-mouth publicity, which would ideally get them more business.
5) Then, the fifth aspect to this one is the large number of "hit" and "return visit" rates. So, it improves the hit rate and return visit rate of your Web-based Decision Support System.

So, if the top-level decision maker identifies, oh, yes, by going through this, I am able to make better decisions, so let me go back and reuse the system as much as I can. So, the person will come back, he will return to the Decision Support System and take benefits of it in making better decisions. So, that means the return visit of the top-level decision maker, we call it the CEO, CX or whatever it is, who will continue to come back to the system and then do it continuously. So, these are some of the major benefits.

(Refer Slide Time: 25:43)



Benefits of UE to the User

(1) Less time to complete work. (Some basic level of Satisfaction)
(2) Greater Success with decision tasks.
Quality of decision improves
⇓
More benefit /value creation to the user.
↑↑ Significantly the user satisfaction.
(3) increased user Satisfaction.
Soon starts to become a habit. ⇒ to Seek support.

There are certain benefits which are specifically to the user only, and these are there I would really say three of them. I will divide them into three aspects.

1) We can think about it as 'less time to complete work'. It takes the user less time to complete the work.
2) Is 'greater success with decision tasks'. That means you have taken better decisions, mostly.
3) Third one is, 'increased user satisfaction'. So, it is in a cyclic manner. So, when you have less time to complete the work, it gives you some basic level of satisfaction. You are happy that, yes, I am able to make this decision faster, so the time is money, aspect place into this. Then from there, you come to the greater success with the decision rate. So, if the process, interacting with the decision only, takes less amount of time, then you can actually spend more time studying this. So, greater success with the decision tasks means your quality of decision improves.

What does this translate to? This translates to literally what we call as more benefit or value creation to the user. So, that is the second aspect. Once these two aspects are done, this will increase. Significantly the user satisfaction. And from there, with that increase in user satisfaction, you are again coming back to the system to make better decisions and etcetera like that.

So, it starts becoming a cycle. And every time you want to make a decision, you want to come back to the Decision Support System and see what the system tells you. How is it to be used, and all those aspects. And in that process, you end up getting into this and it soon starts, it will say that, soon starts to become a habit. So, to seek support. You are looking for support in the decision making. So, once this happens, then your system is accepted by the decision maker and more things actually happen as part of this.

Now, we will get into this after a short break because the Usability problems of the Web-Based Decision Support Systems is a critical aspect for us and that we need to think through, before we start learning HTML. And that also gives us an explanation why HTML and HTML forms are important tools for decision making. Thank you.