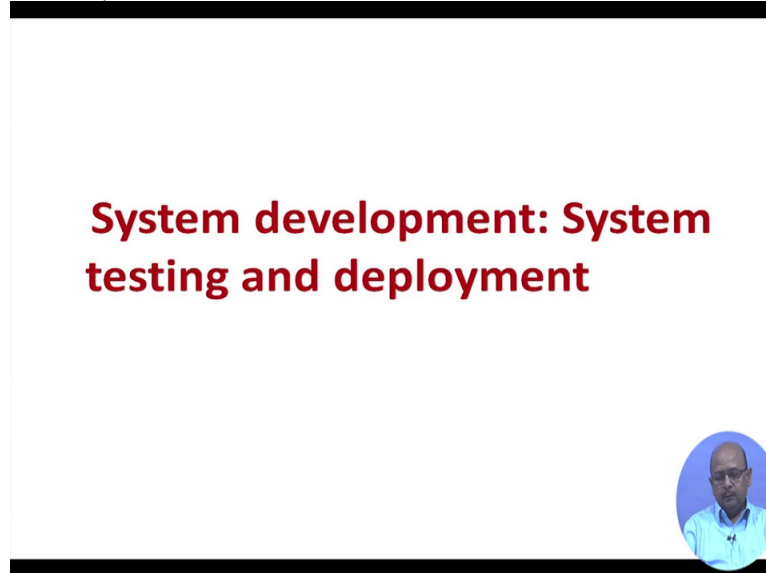


**Knowledge Management**  
**Prof. K B L Srivastava**  
**Department of Humanities and Social Sciences**  
**Indian Institute of Technology-Kharagpur**

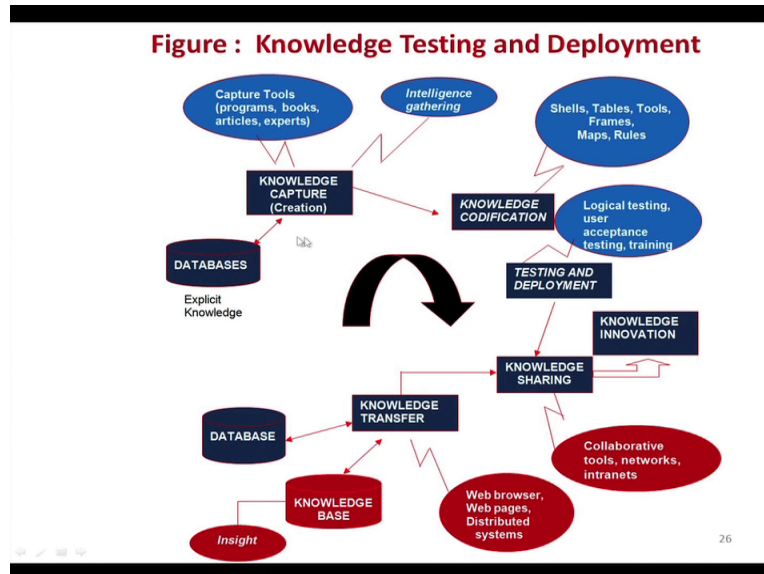
**Lecture 13**  
**System Development:**  
**System Testing and Development**

(Refer Slide Time: 00:34)



Okay. So, we just talked about the codification system now after codification system. We are going to do move to the next part that is system development and that how you go about testing and deployment of the knowledge management system.

(Refer Slide Time: 00:40)




And this stage happens once you have captured the knowledge from the databases and then you have gone for codification. From codification you are moving to the next stage that is testing and development. So, in testing and development basically the main focus is on testing user acceptance and training. We have to discuss about these issues related to system deployment that is testing and deployment, so we are going to talk about these issues.

So, to start with you see that one system is developed then it is related with knowledge sharing and it contains of these kinds of things.

(Refer Slide Time: 01:22)

**Key Definitions**

- building the system right?"
- User acceptance testing checks the system's behavior in a realistic environment. Answers the question, "Have we built the right system?"
- Deployment refers to the physical transfer of the technology to the organization's operating unit



So, now we are going to move to some of the definitions, okay. Now, you are to see that knowledge management system that you have developed is the right kind of system. It means

that it is useful for the organisation is able to build the gap in the knowledge state and it is also going to be linked with the business schools of the organisation.

So, you remember in the beginning we talked about aligning KM strategy with business strategy. And that is where we tried to link the KM strategy with knowledge strategy and also knowledge strategy with the KM strategy. So, you have to see that how once this gap is identified, you have captured the knowledge base that is required by the organisation, you have been able to codify it using various systems either framing of rules or and what you call a decision tree.

And with the help of this and then you have technological system to support it. You have to see that whether you have been able to build up, right kind of system or not, okay. So it is very, very important in order to validate the system. If the system is not built up rightly then you have lot of problem. It could be related to storage, it could be related to retrieval, it is could be related to use whatever it is ok.

What are the various parameters of examining whether the system is built in a right way or not, okay. Starting with the first whether the database are created knowledge base that we have created whether it is organised, classified in a structured format or not. Whether it is codified properly ok, whether it is put into the system ok, whether you have a good user interface, whether it is IT enabled system or not.

Whether it is run by; the most important thing is that whether the content is relevant for the knowledge users that are or not. The knowledge base that is there employees of the organisation and various other stakeholders, so, these are the various question that could be asked which is going to tell you whether the system is being built in a right way or not. Now we will talk about two things that testing and deployment.

What do you mean by testing? User acceptance testing that is you have to check system's behaviour in realistic environment. The thing is that have we built a right system or not. And then deployments then once you have built a system then you are going to deploy means that you are going to transfer that to the organisation of the system so that it is going to operate.

Now I am going to give you one example related to testing and deployment in our context say in our organisation in IIT Kharagpur we developed a ERP system. It is enterprise resource planning

system. Enterprise resource system was developed taking into the consideration that we were having other system but they were fragmented and not integrated system.

So, ERP system is the most integrated system which had databases related to all the stakeholders whether it is the faculty, whether it is staff, okay, whether it is the vendors and suppliers or whether it is the students, ok. Integrating all these databases into one system will help organisation to reduce its cost.

And improve its efficiency to have all the information whenever it is required. So, these were the benefits that we are trying to derive having an ERP system. Now, once this ERP system was built then, we wanted to check whether the system is working well or not. Now, in the beginning, there was lot of problems or troubles, okay.

Because sometimes a system was not getting the right kind of information or there was information lag was there or correct kind of information was not actually fed into the data system. So, all kind of problems were coming. So, this kind of problem will happen in the beginning.

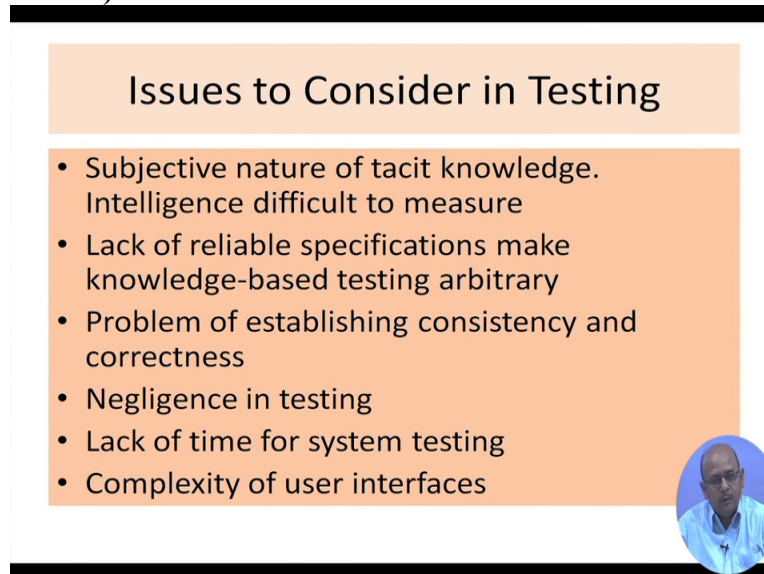
When you are going for user acceptance testing, you have to see and get feedback from users who are the various stakeholders of the system. And get their feedback, whether the system is working for them or not. Whether it is working for the vendors and suppliers, whether it is working for the students, whether it is working for the faculty or whether it is working for the staff.

So, if it is working well, then, you can say, yes that you have built the right kind of system. If it is not working then you cannot say that you have built a right kind of system. So, user acceptance is very, very important, okay. So, user acceptance basically once you develop a system ok, you need to ensure that how does it work in the right realistic environment that when it is going to be there in the organisation whether it is going to be useful or not.

So, if you test it and found that yes it is going to be useful, so what you do, you try to start using from one particular setup or unit. And then you check it for that particular unit it is working and if you find that it is working. Then you can move to other systems and deploy the system for the all the units of the organisation.

So, you start with one unit of the organisation and then you can include or incorporate the entire organisation, see whether it is working or not.

**(Refer Slide Time: 06:40)**



### Issues to Consider in Testing

- Subjective nature of tacit knowledge. Intelligence difficult to measure
- Lack of reliable specifications make knowledge-based testing arbitrary
- Problem of establishing consistency and correctness
- Negligence in testing
- Lack of time for system testing
- Complexity of user interfaces

Now what are the issues that you are going to consider to test a system, okay? Now, there is lot of issues because when it comes to codify tacit knowledge, okay which is highly subjective, okay. How it is measured? How it is codified, ok? So, you have to see whether that tacit knowledge is properly codified, because of its nature, okay?

If is not properly codified then it is not going to be very, very useful. Suppose you are going to observe certain processes but if you miss certain things out of these processes probably the other person who wants to make use of those processes will not be able to use it. So, you have to see that whenever you are going to codify tacit knowledge make sure that it is done completely.

And nothing is left out because if anything is left out, then, it remains subjective. You are not going to make it explicit to the person, okay because it is very difficult to measure the ability or the process which the related to individuals that is basically the tacit knowledge. Then the second part is lack of reliable specifications makes knowledge-based testing arbitrary, okay.

How you are going to check consistency of this system? What are the criteria that you are going to make use to test the system, okay. Whether you have standard criteria or use some random criteria to test it, okay. So, you need to identify those criteria which are very, very consistent to see whether this knowledge base system is going to work or not second point.

The third is consistency and correction. How consistently the system is working, okay. For example today the system is working, tomorrow it is not working. It is creating trouble; it is creating errors, okay. Then, you can say that yes the system is not reliable and it is not consistently working well and there are lots of problems and issues.

And these problems or issues could be for various reasons. It could be related to technical errors or it could be related to input errors also. You have to see that there is no technical error or there is no input error. Input error means the data side, all the data that is fed is correct and then you can say that input error is not there and the technical error means that the kind of system that you deployed for storage and retrieval is proper.

So, that it is arranged and organised in a proper way and it does not create any problem and this negligence in testing, okay. Sometimes what happens? We develop a system and we start using it without going for testing, okay. So, it is very important to go for some kind of pilot testing before you put it to use for entire organisation. It is very, very important, okay.

If you neglect this issue probably it may fail, if it fails probably the entire operation related to knowledge management system going to be a wasteful expenditure, okay. Then, sometimes there is a excuse that there is no time for checking the system. And let us start using it, okay. There could be a short time gain, but in the long term it is not good because if you do not check the system for its reliability, for its consistency, for its accuracy then what will happen?

If the system fails, then who is to be blamed, okay? You are going to blame the knowledge developers or you are going to blame the experts or you are going to blame the technology, okay. It is important to ensure that system is tested before it is put to use ok and then you need to solve the problem related to user interfaces, right.

This is user interfaces could be, it could be of different level, it could be of end user level or it could be at the intermediary level, right, it could be at the top level. Like there are certain data which could be accessed only by the top management people. Make sure that that kind of interface to identify those people at the intermediary level or those who are going to operate the system, okay.

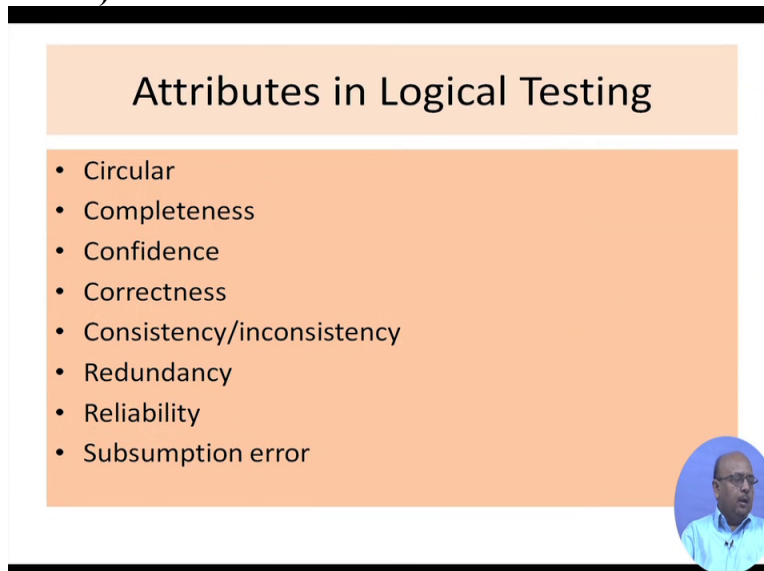
At the user level you are going to provide it to all the customers, relevant customers and then you have a system for user interface like for a students who have systems or faculty who have a

system, okay which is integrated with the ERP system. So, this you have to see that how complex you are going to make use of this user interface.

Say, for example, if internet is not working, like I want to access the relevant information using internet, okay. So, with whom I am going to interact and relate with, it could be the developers or system management people because as an end user I cannot correct the problem. So, in the intermediary level, system analyst or the developers or the programmers must be in contact with the end users so that they can solve this problem, right.

So, you need to develop this user interface, in such way so that it is easy to use especially for the end user and then you develop different layers of users at different level. That is very, very important and that is what needs to be considered.

**(Refer Slide Time: 11:58)**



### Attributes in Logical Testing

- Circular
- Completeness
- Confidence
- Correctness
- Consistency/inconsistency
- Redundancy
- Reliability
- Subsumption error

Next, what are the attributes of the logical testing? Circular, circular means what? It has to move in a circle. Starting with one point then, you move to that point. What does it mean is that ok, it has, there is a cycle, okay. So, if you look at, we talked about knowledge capture then, codification then, move to the next part that is testing and development.

Then, you move to the next part and finally again you moved to the same cycle. So, since the knowledge management system is cyclical in nature. You make sure that each and every stage you are going to test the validity of the system ok. And then make sure that it is complete, okay. Completeness it must be complete in itself ok.

And then, confidence, those who are using the system must be confident. In the sense, that this system is good is relevant they can make use of it and there are several other issues related to confidence. It means the issues related to privacy, okay. For example, I am going to use my login and password to access certain data, okay. It could be related to me, but the same data should not be available to others, except at the intermediate level or at the senior level, right.

So, the user must have confidence in the system before they start using it. And for that you also need to train them that how to develop the, use the system and then you must develop them in the kind of confidence that is required to use the system. And then correctness; Correctness how accurate the system is.

Accurateness in the, accurateness of the system depends upon kind of, I mean the quality quantity of data that you have put in. I am not talking about the quantity of the data, but I am talking about the quality of the data. So, how you ensure that the quality of information that you have put in, okay? That is very, very important and that is going to tell you whether the data the input, that is put into the system is correct or not, okay.

Then, consistency and inconsistency, that is related to reliability. How reliable that information is? Whether you are going and asking, whether you are getting similar kind of information or every time you are getting different kind of information from the system. So, if you are getting different kind of information related to particular query it means the system is not reliable.

You put your query to the system related to particular aspects and you get consistent information which is similar in nature then, you can say yes, it is consistent; you are getting reliable information. Then redundancy, so you also need to read out the redundant part on a regular basis that is very, very important.

Because the knowledge base or the database that you are created the information that you are putting, okay should not be obsolete or outdated. Make sure that you are not going to have redundant data. So, the principle is garbage in and garbage out, okay. It means whatever garbage that is collected in the bin it is to be taken out. So, that you can put in new data into the bin and that is very, very important.

Otherwise what will happen? This bin become redundant so that the knowledge management system that you have developed, make sure that you are going to update it with relevant

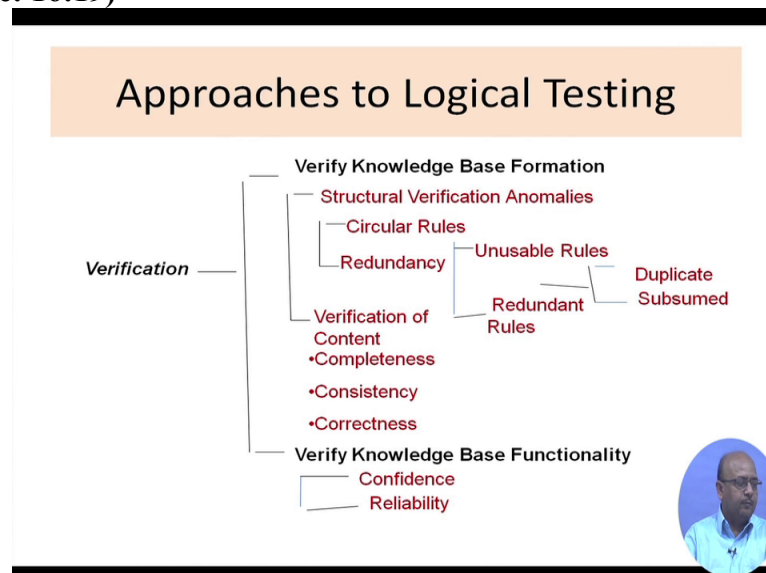


knowledge on a regular basis. Otherwise, after a period of time it is going to be redundant. So, this testing has to be done on a regular basis to ensure there is no redundancy in the data that is there.

And then, you have already talked about reliability factor. You make sure that data is reliable, okay. And then, there could be some errors or some functional errors and that could be identified. So, you should be in a position to identify the error in the data, okay. So, the system developers or the knowledge developers who are going to codify the knowledge into the system have to ensure that how they are going to identify the error if any, there is into the system, okay.

So, these are the attributes which could help you to identify and test whether knowledge system the knowledge management systems that you have built up is the right kind of system.

**(Refer Slide Time: 16:19)**



So, this is that, process of verification for testing, that we have just discussed about it. Now if you look at it, it talks about the verification process, okay. There are two ways in doing it: Verify knowledge-base formulation, how the knowledge base is created, knowledge base is formed and whether the knowledge base is functioning effectively or not.

So, this talks about capturing and codification and this talk about functionality that is used, okay. Now, if you look at this, look at the structural verification anomalies, circular rules and redundancy. This is to be identified ok, now circular rules we have just discussed about it, then redundancy that we have already talked about it. Make sure that there is no data redundant.

When you are going to create knowledge make sure that it is updated and redundancy is avoided, okay.

So, you need to identify certain rules of logic to ensure that all redundant data is kept out, okay. And in the process you have to see that certain data which is which may be duplicated or subsumed is waded out in the process. So, and you are also going to not only verify the anomalies in the data, so it is consistent, but also verify the content in terms of its completeness, consistency and correctness. So, these are the criteria for content verification.

So, if you look at this we talked about completeness, confidence and correctness it is very, very important to ensure the content that is put into the knowledge management system, it is complete in itself, okay. So, you do not have to look at that kind of information elsewhere. It is consistent; it is very reliable and correct. The content is the quality of the content is to be looked into.

Now these issues are related to knowledge base formation. Now, moving from knowledge base formation to next part that is, verify knowledge base functionality. When I am talking about knowledge base functionality, it is basically related to two issues: that is confidence and reliability. So, if people have confidence, ok they know that secrecy is not going to out, okay. They are trained to use it, then, probably they would be in a better position to use it.

And then, you have to check that reliability, when consistently you are getting all the information that is required, right. There are certain things that need to be looked into that you need to avoid: redundancy in the data, make sure that is complete, consistent and correct and at the same time you make sure that people are able to make use of it.

So, if you have developed a good system, then, it is enabled through intranet and other things probably and people are trained then probably you can verify functionality of the knowledge base. It means that it is functional people are going to use it for productive purpose, right. So, these are the approaches to logical testing that is what we discussed about here in a graphical format.

**(Refer Slide Time: 19:34)**

## Key Testing Errors

- Circular errors tend to be contradictory in meaning or logic
- Redundancy errors offer different approaches to the same problem; duplication of knowledge
- Unusable knowledge is knowledge that comes up if the conditions succeed or fail
- Subsumption errors in rules, if one rule is true, one knows the second rule is always true
- Inconsistent knowledge, where the same inputs yield different results



Now, moving from this to next part is that how you identify the errors in the system, system of the process, okay, like circular errors. What is a Circular error? Circular errors - it tends to be contradictory in meaning or logic. It means that see, circular errors means that it is a cycle. One leads to other. So it is happening in the similar direction otherwise you can identify that yes, it is not logically connected it means that it is not correct, okay.

So, you cannot go for creation of knowledge or codification of knowledge before it is captured and created, ok. So, this process to be followed in its, in the proper cycle and that is what I am talking about the circular errors. Next part is redundancy errors ok, so, you have to identify problems related to the data. So, there is no duplication of data knowledge I mean the similar knowledge in the different form is not available.

Similarly, the redundancy also talks about obsolete and outdated knowledge which is put into the system. So, you need to ensure that obsolete and outdated information is kept out of the purview of the people; because it is of no use and going to keep lot of space in the system and it has no value. So, you need to ensure that only those knowledge content is available which is of relevant and value to the people, ok.

And then, you also need to identify unusable knowledge, okay. Unusable knowledge is knowledge that comes up if the conditions succeed or fail, okay. So, you need to identify those knowledge which is not being used by the people, okay because if the knowledge is not being used by the people then there is no question of having success or failure, okay. See we get lot of books and journals in the library.

Now, the thing is whether all of us are using them properly or not. If it is not being used then how you are going to say that okay this is good or bad; whether you are going to succeed or failed using this. So, it is very important is to ensure that you get only that kind of thing which is going to be help, which is going to help you to succeed; okay. Otherwise you are going to fail.

And there are some subsumption errors basically if one rule is true one knows the second is always true. What does it mean? How are we going to explain this subsumption error in the rules? For example if you say okay, A is X, okay then we can say B is X. Can we say that A is B? Logically you can say, because if A is X, B is X, it means  $A = B$  because A and B denotes to the similar things, right. But if say A is X and B is Y then we cannot say that  $A = B$  because they denotes two different things.

So, if one rule is true that other, other one that knows that the second rule is also going to be true. So, if one is A is X, B is X the second rule is A equal to B, so that is correct ok, that is subsumption errors, okay. We assumed that okay this is based on the first rule, okay; so, logically you are trying to connect first with the second rule, okay.

Now, inconsistent knowledge, okay; Inconsistent knowledge is whether same inputs yields different result ok. So, suppose you have data okay. If you are going to analyse the data, whosoever analysis data, you come out with the similar results, okay. But if two people are analysing the data, the same data using two different approaches but coming out with two different results it means either there is a fault in the method or the data that is to be identified.

So, when I am talking about inconsistent knowledge either there is a fault in coding the data. So, either data is not proper, data is not correct or the methodology that you have adopted to analyse the data is not correct, okay, right. So, you have to see that if there are similar inputs it provides similar results; otherwise it is not consistent, okay.

**(Refer Slide Time: 23:45)**

## Steps in User Acceptance Testing

- Select a person or a team for testing
- Decide on user acceptance test criteria
- Develop a set of test cases unique to the system
- Maintain a log on various versions of the tests and test results
- Field-test the system

32

Then moving to the next part that is steps in user acceptance testing how you go about it? So, you need to create a team for testing purpose that is very, very important. Selective personality for testing and this team should be responsible for validating and verifying the system, okay and then you decide the criteria, okay. You can use the rules of logic that we just talked about, okay.

So, you have to decide on user acceptance testing criteria. So, these are the criteria that will be using okay while verification of the knowledge or functionality of the knowledge. In both the cases, you need to identify certain rules like you want to see whether it is complete, correct, and consistent and at the same time there is no redundancy error.

And then you also see that whether the system is working or not, okay, in the sense that people are have confidence and reliability. So, the thing that just we talked about can be used as a criteria for user acceptance testing. So, you need to decide which criteria you are going to adopt for testing, acceptance testing, right and then, you develop certain test cases.

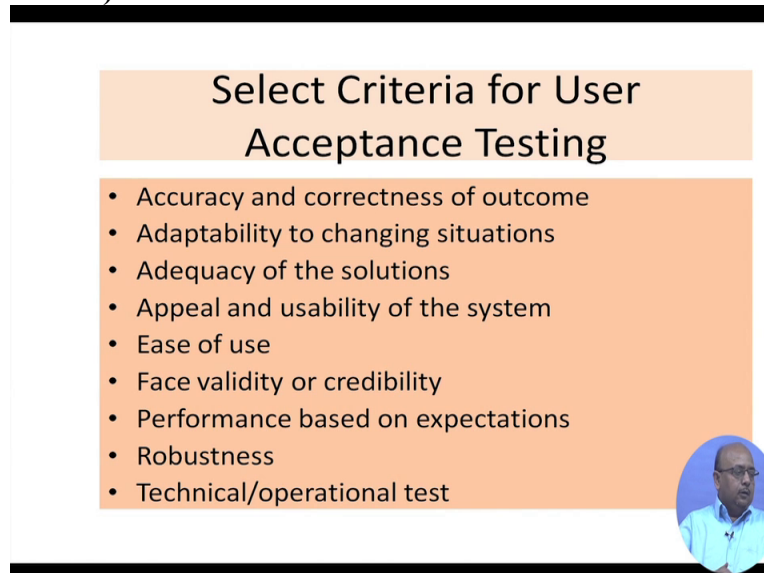
So, you take some departments from different parts of the organisation and try to use it; and see whether it is being validated to start with. That is what we know as pilot testing. You should go for pilot testing to start with, to check whether the system is working in certain cases or not. If it is working then you go for deploying it into other places, okay.

Then, you maintain a log of various versions of test and the results. So, you also document the test that test results so that you will be in a position to say or communicate to the people and give feedback that how well the system is working, okay. You tell them, okay that these are the

criteria's and on these criteria we have tested the system and it is working or not. And these are the results of the system.

And then and you go for field testing, tested in the organisation and see whether it is working or not. If it is working, well and good. And if it is not working then, again you look at the entire cycle again starting with knowledge capturing, codification and then you move to that testing and deployment to see where the fault lies.

**(Refer Slide Time: 26:04)**



**Select Criteria for User Acceptance Testing**

- Accuracy and correctness of outcome
- Adaptability to changing situations
- Adequacy of the solutions
- Appeal and usability of the system
- Ease of use
- Face validity or credibility
- Performance based on expectations
- Robustness
- Technical/operational test

So, we have already talked about these criteria, the selection criteria, so, this could be used as a selection criteria like how accurate and correct or outcome that is coming out of the system. Whether that testing is going to adapt to the changing environment or not, so, you have to see whether you are going to use it and able to adapt yourself system or not, okay.

How adequate solutions are getting out of the system, whether the system is able to provide good solution of the problem or not. If yes or the extent to which it may not be 100%, but maybe it is 60% , 70% so you decide what the success rate, you are looking forward to, and accordingly, you decide whether it is working or not.

Then, the next is appeal and usability of the system. How good it is in terms of use okay, whether people are able to make use of it or not and what are the problems and how you are going to address these problems. That is related to what you call appeal and usability of the system. So, there are certain problems, so you need to address them or people appeal about that to address it. So, that you are able to make the system usable.

And then ease of use, how smooth access is there, so, there could be a technological factor or other factors to make sure that you are going to develop a system where there is storage and retrieval especially retrieval and application is very, very smooth, okay. And then, you also look at the face validity.

At the face of it whether it is working or not, you look into the credibility factor. If the system is credible it should definitely work, to start with. And that is where we do not look into other form of validity but at the face of it system appears to be okay. So, first of all you see that whether at the face of it, system is credible and then you go for testing for further validity, right.

And whether and whatever expectation you have about the system whether it is meeting those expectations or not. What I mean to say is that, when you are developing a knowledge management system, it has certain objectives, okay. The objective is to create knowledge system, relevant knowledge system which people can use for their work. And whether people are really using it and ready to, in a position to solve their problem;

Then, if that is so, then you can say yes the knowledge you have deployed be a good knowledge management system and it is helping people to arrive at the solution. And it must be robust, robust in the sense that it is not easily mean bifurcated, hacked, okay ethically correct. And these kinds of issues are there in robustness. It has a very sound system, it has technically very, very sound system. So, that it is going to work and it is a very strong system, okay.

And then you go for technical and operational test make sure that you meet all the technical and operational requirements, okay before you accept the system for deployment, okay. So, these are the different kind of criteria for the user acceptance and testing.

**(Refer Slide Time: 29:09)**

## Managing the Testing Phase

- Decide when, what, how, and where to evaluate the knowledge base
- Decide who should do the logical and user acceptance testing
- Draft a set of evaluation criteria in advance
- Decide what should be recorded during the test
- Review training cases, whether they are provided by the expert, the knowledge developer, or the user
- Test all rules for Type I and Type II errors



Then, the next stage is managing the testing phase. And in the space basically what you are trying to see when, what, how are to evaluate the knowledge which is very, very important when you are going to especially this phase where you are going to test the system then you have to answer these questions that: who is going to do logical testing, okay.

Who is going to do user acceptance testing, what are the evaluation criteria that you have decided in advance. How you are going to make use of these evaluation criteria. Decide what should be recorded during the test and what you are going to document that, okay. Then, you are going to review training cases whether you have been provided by experts or not.

The knowledge developer or the user whether you are provided enough training to the people, whether it is the end user or say knowledge developers or the experts and then you type all rules of for type 1, type 2 errors. Type 1 and type 2 errors. Type 1 error basically refers to when you assume that certain things are correct but it is not.

And type 2 is that when you assume that it is correct or it is not correct but it becomes correct, okay. These kinds of error should be avoided. Type 1 and type 2 errors, okay. So, thank you that is all for this session.