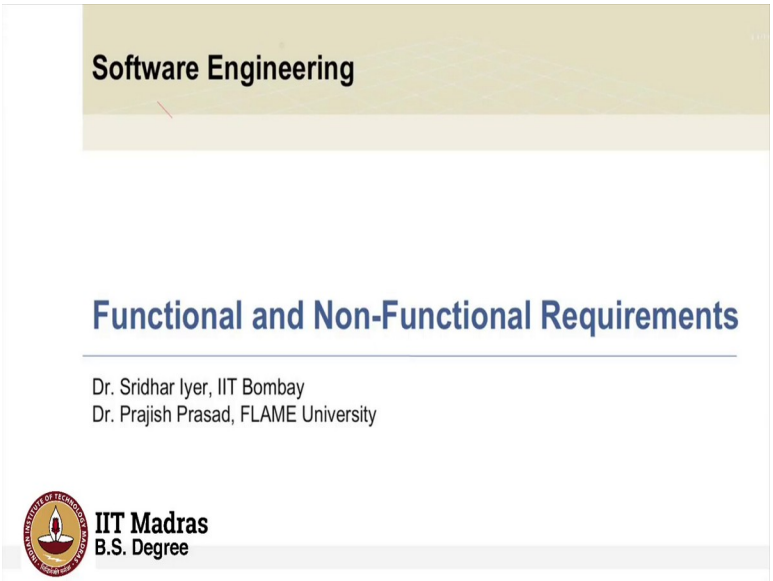
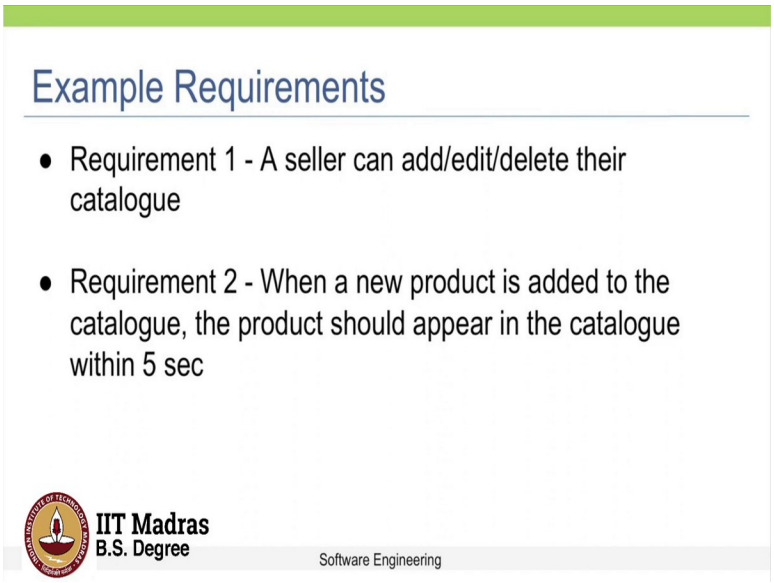
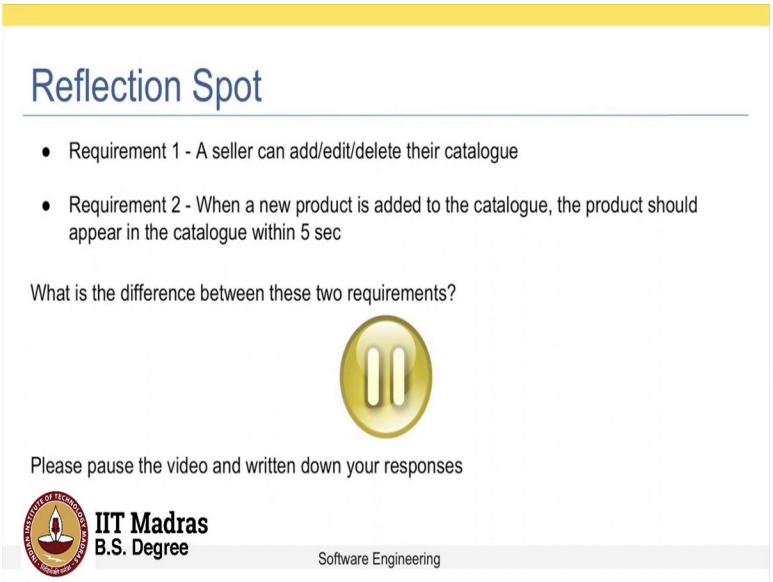
Software Engineering
Professor Sridhar Iyer
Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
Professor Prajish Prasad
Department of Computer Science
FLAME University
Functional and Non-functional Requirements

(Refer Slide Time: 0:20)

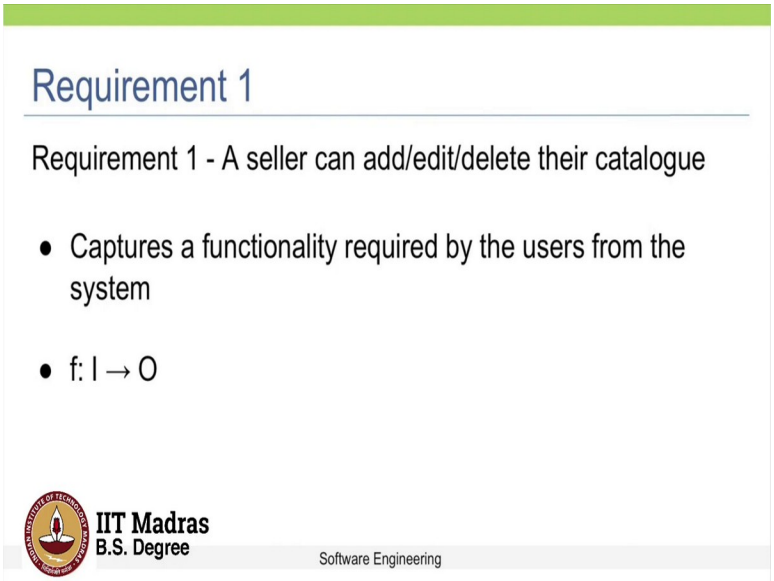


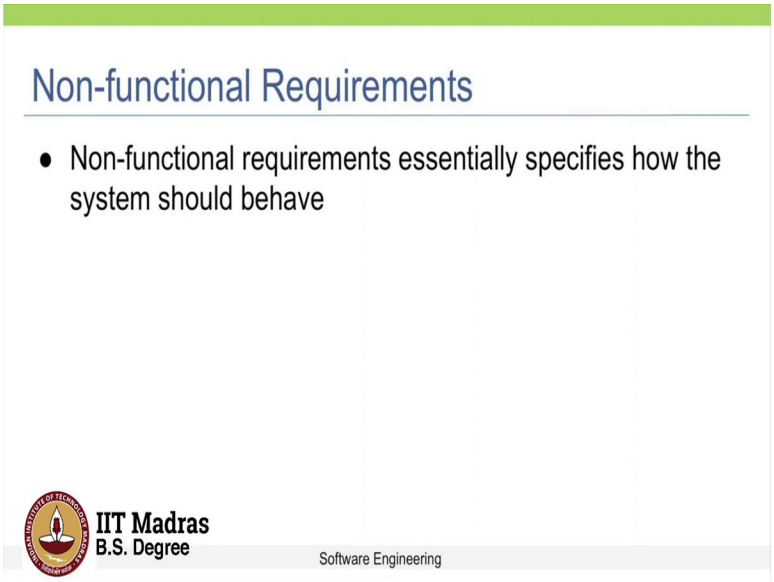


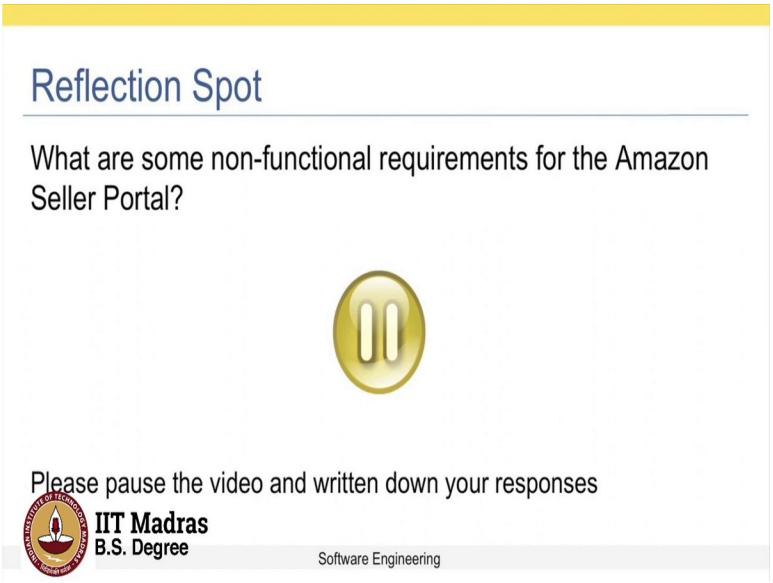


Professor Sridhar Iyer: In the previous video, we looked at how we can identify requirements using several techniques? Such as interviews, looking at the documentation, questionnaires, and so on. You might have noticed that not all requirements have the same characteristic, or similar type of characteristics. For example, consider these two requirements. A seller can add or delete items from their catalogue, this is one requirement. Second one is, when a new product is added to the catalogue, it must show up on the user's interface within 5 seconds. Can you say what is the difference between these two requirements? What is a nature of these two requirements? How was that different? Pause here for a moment, think about it, and then proceed.

(Refer Slide Time: 1:28)







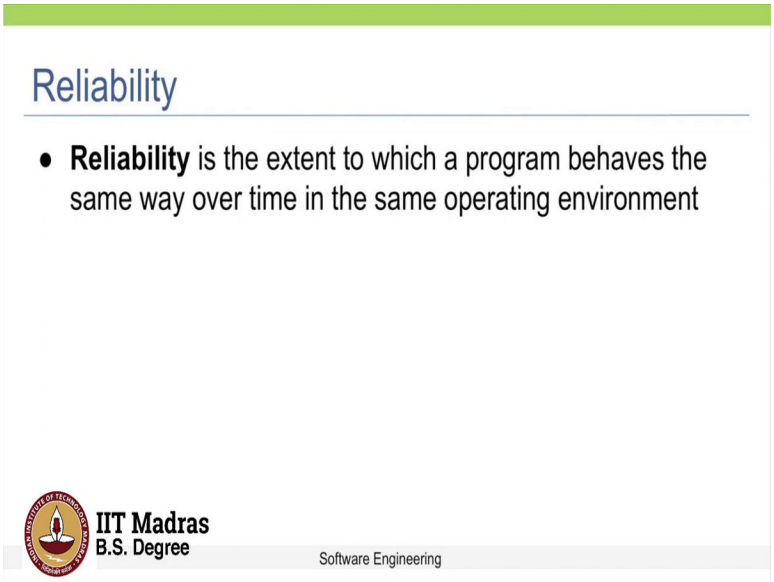
Professor Prajish Prasad: So, let us look at the first requirement. So, requirements 1 captures the functionality required by the users from the system. So, the user performs some actions, give some inputs, and based on that input, the system provides an appropriate output. So here, you can see that the user can maybe add, edit, or delete some things from their catalogue, and the output will be a result of these actions. So, these types of requirements are known as functional requirements. So, functional requirements can be considered similar to a mathematical function, where the function transforms an element in the input domain I to a value in the output O. So, the functional requirements of the system should clearly describe each functionality that the system would support along with the corresponding input and the output dataset.

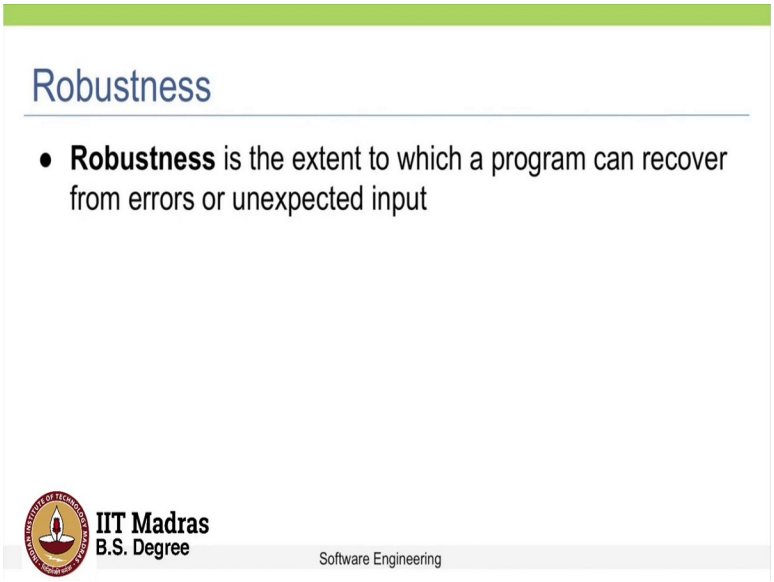
Professor Sridhar Iyer: So, coming back to the requirements, we have saw that requirement 1 of being able to add or delete a product from the catalogue is a functional requirement. The

requirement, second requirement that we talked about that is that the product should appear in the catalogue within 5 seconds. It is not a requirement that can be expressed as functions, that is as accepting some input and producing some output data. So, such requirements essentially specify how the system should behave, not what the system should do. So, such requirements are called non-functional requirements.

It is a little difficult to understand both these terms in the beginning, especially because we are used to the word non-functional interpreting the word non-functional in a different way. So again, let us take the example of the seller portal. We have talked about one non-functional requirement which was that the product update should appear within 5 seconds. What are some other non-functional requirements that you can think of for the system? Pause here, write down one or two non-functional requirements and proceed.

(Refer Slide Time: 4:11)



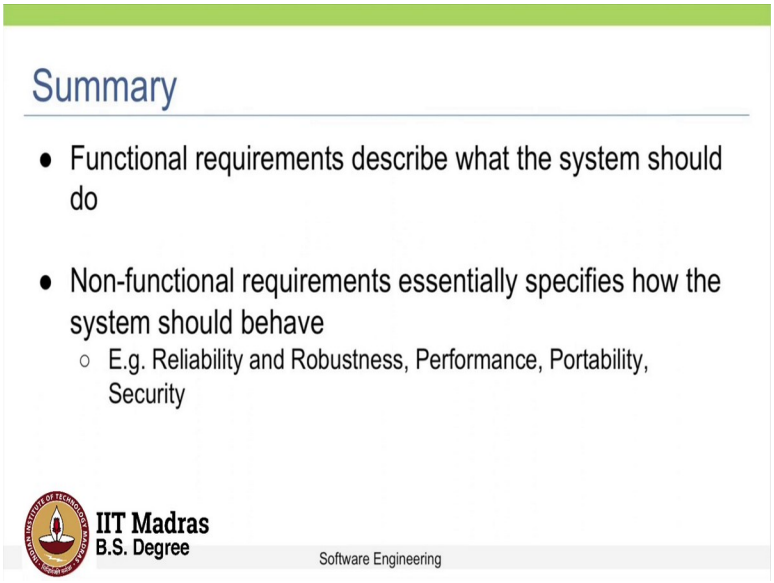


Professor Prajish Prasad: One type of non-functional requirement is reliability. So, a client would want the system to be reliable. So, reliability is the extent to which a program behaves the same way over time in the same operating environment. So, in the case of software, we need to ensure that the software does not crash often, otherwise users will get frustrated. For example, in the seller portal reliability is even more important. We must ensure that the portal handles all operations, especially when the inventory changes based on the buying and selling from customers.

Professor Sridhar Iyer: Now, in addition to reliability, we also want the system to be able to handle unexpected situations. What does it mean? For example, a user may just give some garbage as input, what is the system do now? So, such features, or non-functional requirements come under the category called robustness. So, robustness is the extent to which a program can recover from errors and unexpected input. For example, in the seller portal, we have to ensure that the service handles large numbers and high traffic.

When we specify requirements related to reliability and robustness, we are in a way ensuring that the system behaves the way we intend it to behave. And we are not exactly specifying what we want the system to do? Or we are not checking the functional requirements. So, that is the key between functional and non-functional requirements.

(Refer Slide Time: 6:10)



Professor Prajish Prasad: So, to summarize, in this video we looked at functional and non-functional requirements. We saw that functional requirements describe what the system should do? Whereas, we saw the non-functional requirements like reliability, robustness, they essentially specify how the system should behave? So, in addition to reliability and robustness, there are other non-functional requirements as well, such as performance, portability, security, interoperability, etc. So, we will not look into those in detail now, but it is important to consider the functional as well as non-functional requirements when we start building a software system.