Software Conceptual Design

Dr. Sridhar Iyer
Dr. Prajish Prasad
Dr. T. G. Lakshmi
Department of Computer Science and Engineering
Indian Institute of Technology, Bombay

Lecture - 04
Software Design and Development

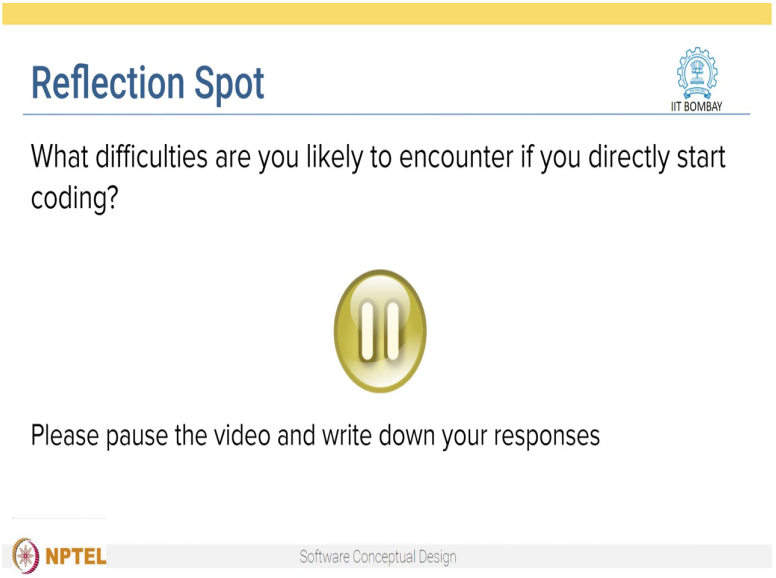
In the previous video, we talked about one of the first steps in the Software Development life cycle, that is identifying requirements from the clients Ok. So, now that we have come up with the requirements most of you would think that we can directly jump into writing code.

Yes, that is true, but there is an important step before that. Let me illustrate this with an example.

(Refer Slide Time: 00:31)

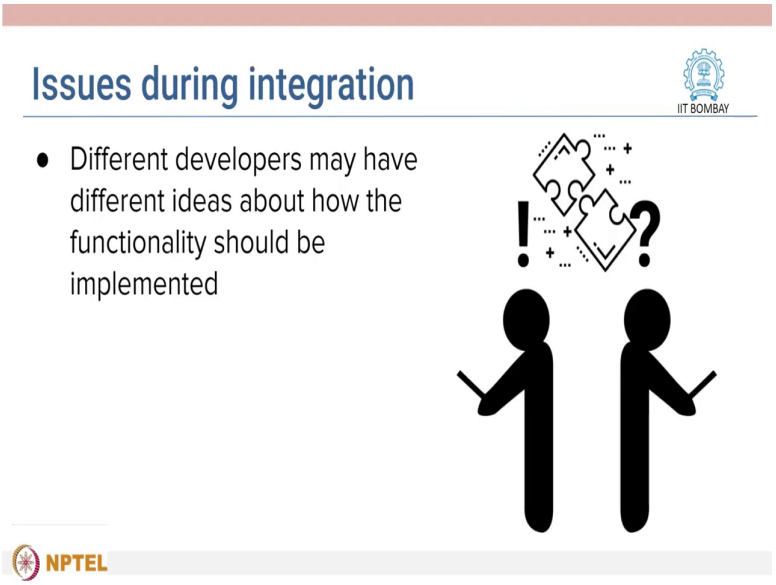
Imagine you are a small team going to implement the Amazon pay feature from the requirements. All of you are really excited to start implementing the feature and directly jump into writing the code. What difficulties or issues you are likely to encounter?

(Refer Slide Time: 00:52)



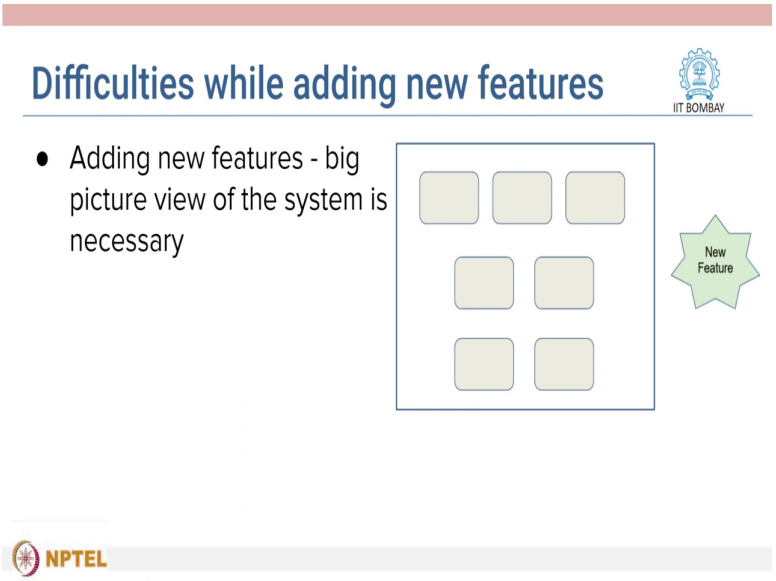
Pause the video and write your answer in the notebook.

(Refer Slide Time: 01:05)



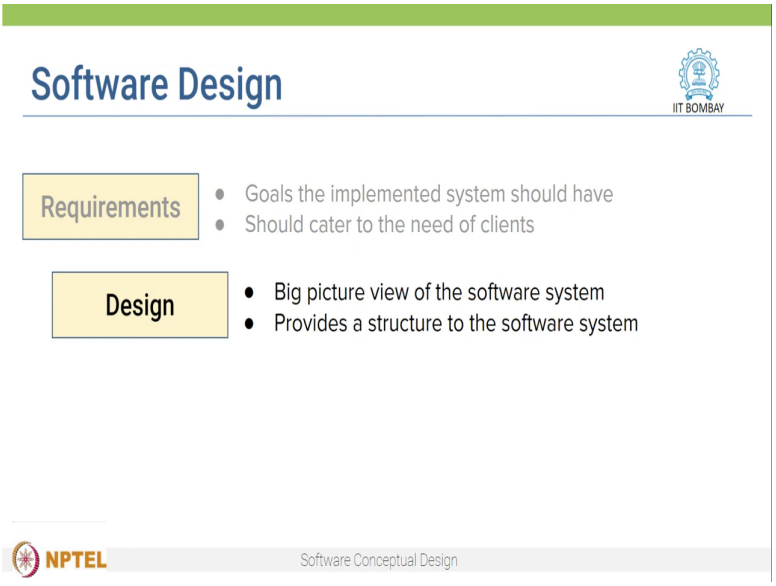
One issue can be that different developers have different ideas about how a functionality should be implemented. Even if you have discussed with your colleagues, you are most likely to make a change in your code. You might add an argument to an existing function, and your colleague who is using the same function to develop her components can face difficulties. This can be more problematic when different components created by different people have to interact with each other.

(Refer Slide Time: 01:37)



Oh, that can be a disaster. It will needlessly make it more complicated, and the product will not be delivered on time. Also, if I want to add a new feature later on it would help to have a big picture view of the system so that I can decide how this new feature can be correctly integrated into the existing system.

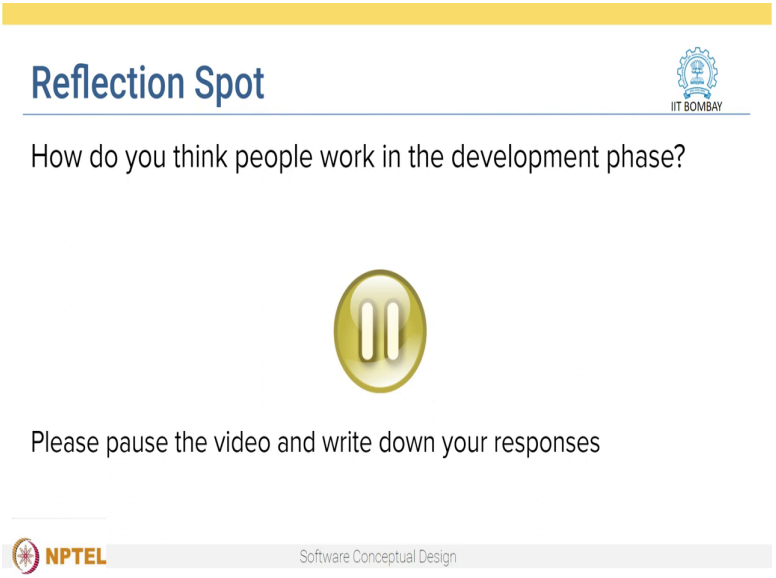
(Refer Slide Time: 02:02)



Yes, this big picture view of the system is created in the design phase of the development cycle. The design of a system is a way of organizing the code you will implement and

providing a structure to the software system. After coming up with the design, then we can go into implementing the system and write code.

(Refer Slide Time: 02:27)

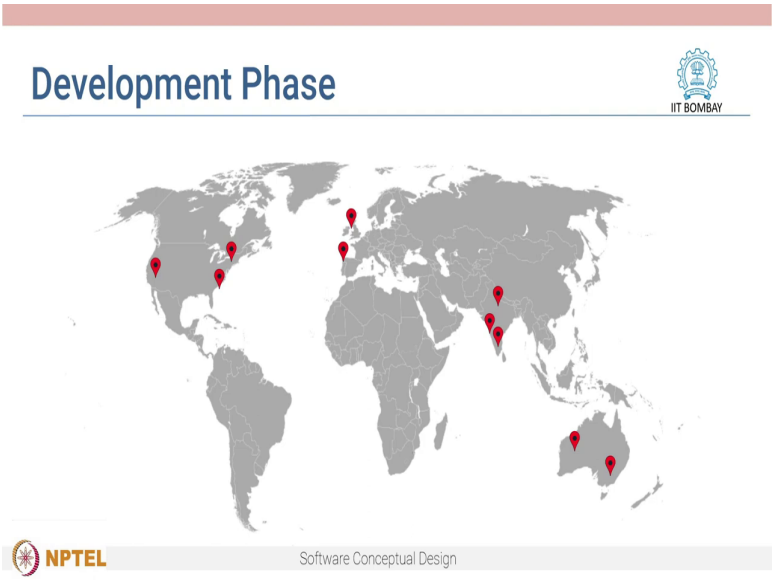


So, here is a reflection spot for you. What all comes to your mind when you see software developers writing code? How do you think people work in the development phase? You can pause this video and write your answers in your notebook.

(Refer Slide Time: 02:53)

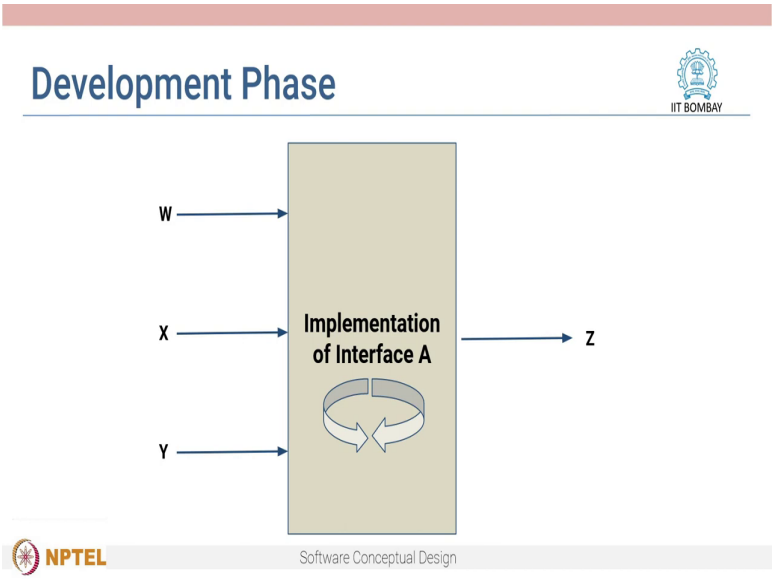
So, how do developers write code? When a feature is being implemented, multiple developers work together and write code for the feature. They use tools like GitHub to collaborate and write code.

(Refer Slide Time: 03:12)



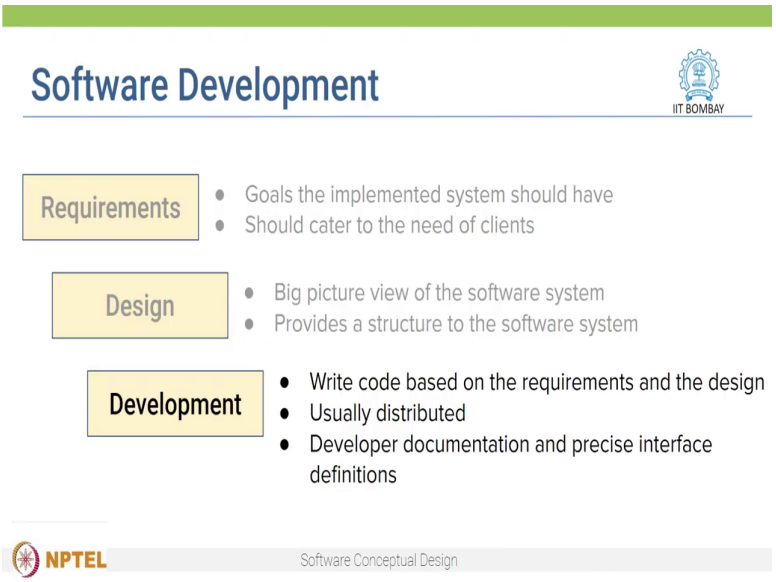
Hence, coding is often done in a distributed manner with developers working in different locations and even in different time zones. Hence, it is very important that everyone working on the code base has a consistent understanding about what the code does. That is the reason why developers write documentation for their code and write precise interface definitions. So, what is an interface?

(Refer Slide Time: 03:49)



An interface is nothing but the description of the actions that your functions can do without describing the implementation in detail. For example, in this case, interface A takes 3 inputs in the form of W, X and Y, and returns an output in the form of Z. Thus, the interface shows what requests are accepted in what format and the corresponding responses. The code being implemented by the interface can change as long as the interface remains the same.

(Refer Slide Time: 04:35)



In the development phase, software developers write code based on the requirements and the design of the software system. Development is usually distributed that is multiple people

write code which implement the required functionalities. To ensure that everyone in the development team is in agreement, developers document what their code does and write precise interface definitions.