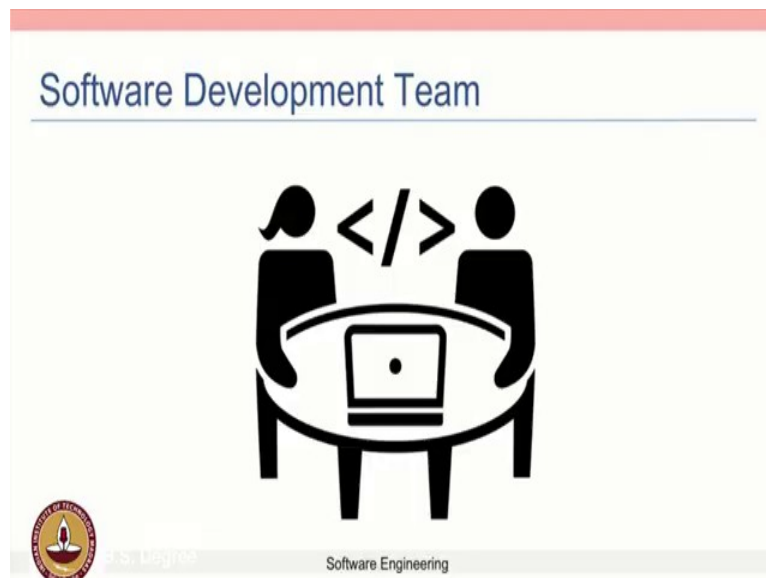


Software Engineering
Professor Sridhar Iyer
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Professor Prajish Prasad
Computer Science, FLAME University
Software Development Process - Software Design and Development

Sridhar Iyer: In the previous video we talked about the first steps of software development cycle; namely gathering of requirements. So what is the next step? Most people would think, okay, I am ready to start coding and they would jump to writing code. Is that really the next step? Let us see this with an example.

(Refer Slide Time: 0:48)




Imagine that you are a small team going to implement the Amazon Pay feature from the requirements. All of you are really excited to start implementing and directly jump into coding.


(Refer Slide Time: 1:02)

Reflection Spot

What difficulties are you likely to encounter if you directly start coding?



Please pause the video and write down your responses




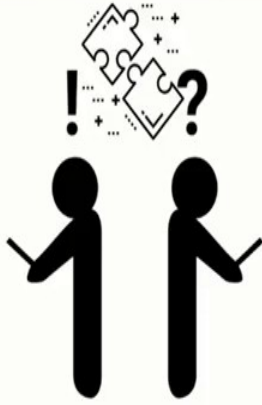
Software Engineering

What difficulties are you likely to encounter? Pause here for a moment and think about it and write down one or two difficulties that may arise if we directly jump into coding after gathering the requirements.

(Refer Slide Time: 1:27)

Issues during integration

- Different developers may have different ideas about how the functionality should be implemented



Software Engineering

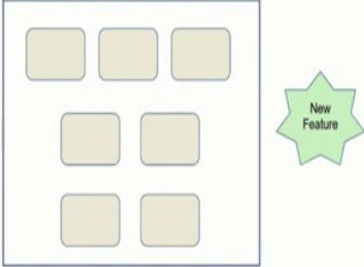
Prajish Prasad: So one issue can be that different developers they have different ideas about how the functionality should be implemented. So even if you have discussed about what should be done with your teammates you will most likely make changes in your code.

For example, you might add an argument to an existing function in your code and this can result in an error in the code of your colleague, which uses your function to develop her component. And this can be more problematic when different components created by different people have to interact with each other.


(Refer Slide Time: 2:10)

Difficulties while adding new features

- Adding new features - big picture view of the system is necessary



The diagram illustrates a system architecture. It consists of a large rectangle containing seven smaller yellow squares arranged in three rows: three in the top row, two in the middle row, and two in the bottom row. To the right of this rectangle is a green star with the text 'New Feature' inside it, indicating the point of integration for a new feature.

 B.S. Degree Software Engineering

And such a scenario can be a disaster. It will needlessly make the software complicated and the product will not be delivered on time. Also if I want to add new features 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: 2:38)

Software Design

Requirements

- Goals the implemented system should have
- Should cater to the need of clients

Design

- Big picture view of the software system
- Provides a structure to the software system

 B.S. Degree Software Engineering

Sridhar Iyer: So this big picture view of the system is created in the design phase of the development cycle. We create this so that we know how to organize the code before we jump into implementing it and this provides the structure to the whole software system.

(Refer Slide Time: 2:58)



Prajish Prasad: So after coming up with the design then we can go into implementing the system and write code. So learners here is a reflection spot for you. So what all comes to your mind when you see software developers writing code? How do you think people work in this development phase? You can pause this video and write down your responses before proceeding.

(Refer Slide Time: 3:37)

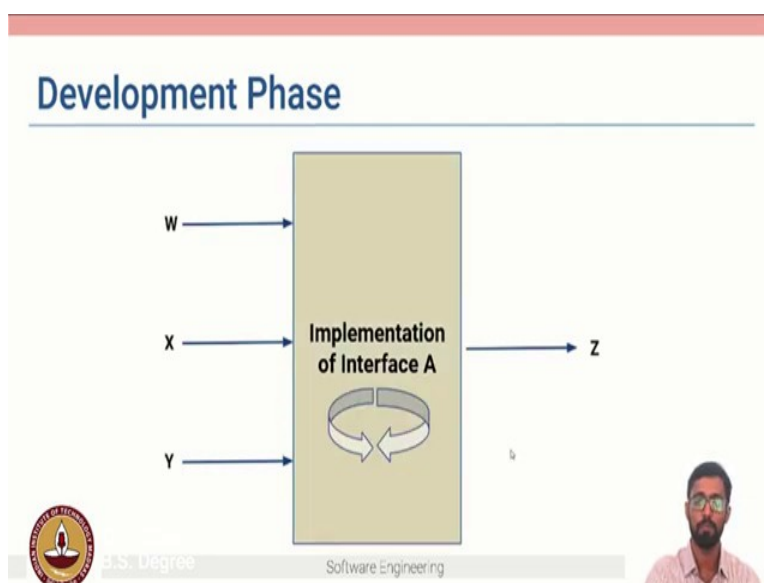




So when a feature is being implemented multiple developers work together and write code for the feature. They use tools like GitHub to collaborate and write code. And very often coding is done in a distributed manner with developers working in different locations and even in different time zones.

Hence, it is very important that everyone working on the code base has a consistent understanding about what the code does. And that is the reason why developers write documentation for their code and write precise interface definitions. So now what is an interface?

(Refer Slide Time: 4:27)



An interface is a description of the actions that your functions can do without describing the implementation in detail. The interface shows what requests are accepted and in what format the corresponding response is given that is the output. The code being implemented by the interface can change as long as the interface remains the same.

Sridhar Iyer: So in the development phase software developers write code based on the requirements and the design of the software system.

(Refer Slide Time: 5:11)



Development is usually distributed. These systems are too big for one person or one small team to develop. So multiple people write the code which implement the required functionalities, often multiple people are sitting in multiple time zones to do this. So to ensure now that we can put everything together and it functions everyone in the development team needs to write what are called interface definitions.

What does that mean? It basically means that what is the language that two components can talk. How do I expect a query? What is the form that I expect a query to take and what is the form in which I will provide the answer? So that is the simple way of thinking about interface definitions.