Software Engineering
Professor Sridhar Iyer
Department of Computer Science and Engineering
Indian Institute of technology, Bombay
Professor Prajish Prasad
FLAME University
Week 2: Software Requirements
Requirement Gathering and Analysis

Professor Sridhar Iyer: In the previous week, we looked at the software development process. And we saw various models like the Waterfall Model and Agile Model. In this week, we will be looking at the first step in software engineering, basically, identifying requirements for a software solution. This is a very important step, because we need to spend sufficient time and effort in identifying such requirements and documenting, why? Because if we do not do this thoroughly, there will be a mismatch between what the user expects and what the software developer does. So, this week, we will focus on requirements.

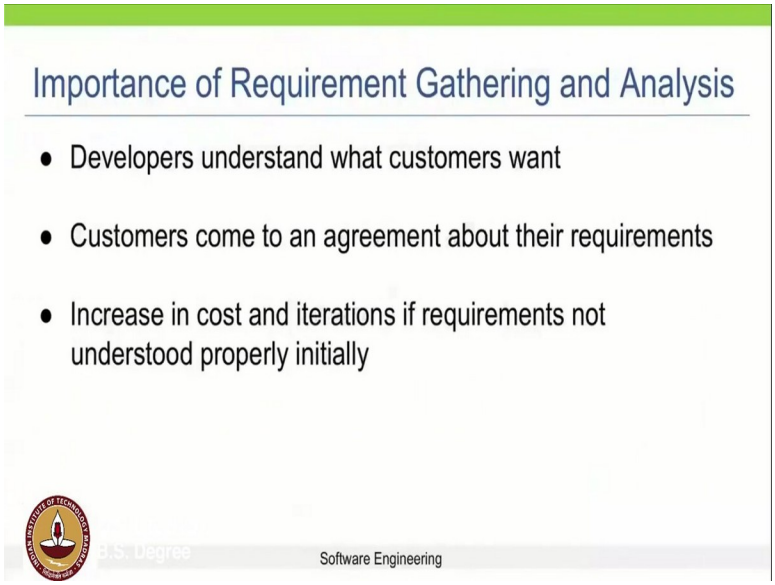
Professor Prajish Prasad: So, you might not be so convinced that identifying requirements are important. And you might also think why we need to spend so much time and effort in identifying as well as documenting these requirements? Cannot we just identify the requirements as we go about designing and developing the software system? So, let us reflect on this question.

(Refer Slide Time: 01:36)

So, learners, what are the disadvantages, if we do not spend time and effort in identifying and documenting requirements? You can pause this video and write down your responses before proceeding.

Professor Sridhar Iyer: Our vision of what the software should look like and behave is quite different from what the user has in mind. Even historically, many projects have had this problem, developers started to implement something while the users actually were expecting something else. So, there was a mismatch of expectations. What is the cost involved here, you spent a lot of time, effort and money in developing something, which was not what your customer asked for, so. So you have to make changes. So, there are more iterations in the software development process. And this pushes up the development costs very significantly.

(Refer Slide Time: 02:47)

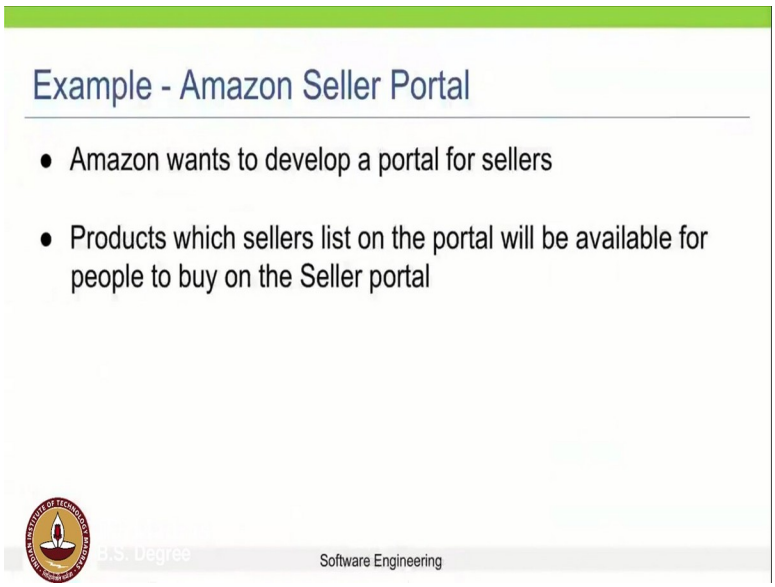


So, to summarize, the importance of requirement identification and analysis are two fold. One is that we understand developers understand what the customers want, of how the software should look and behave. And two, the customers agree that the developers are understanding is along the lines of what they are expecting. Otherwise, the cost, the time, the effort involved in developing the software can increase significantly. So, now that we have established that requirements are important, how do we go about it? Here, the first thing I need to know is what does my customer want from the system?

Again, remember, the customer is not always the end user. The customer can be an internal user, customer could be somebody from a different team. So, I need to ask the customers, what do they want? And then I look at what they want, analyze them, in order to plan how do I develop my piece of the software. All of this may sound fairly abstract at this point. And we will look at an example later. But the point to keep in mind here is that first, teams gather requirements. Then they analyze them. And only after that, they think about how to develop the software.

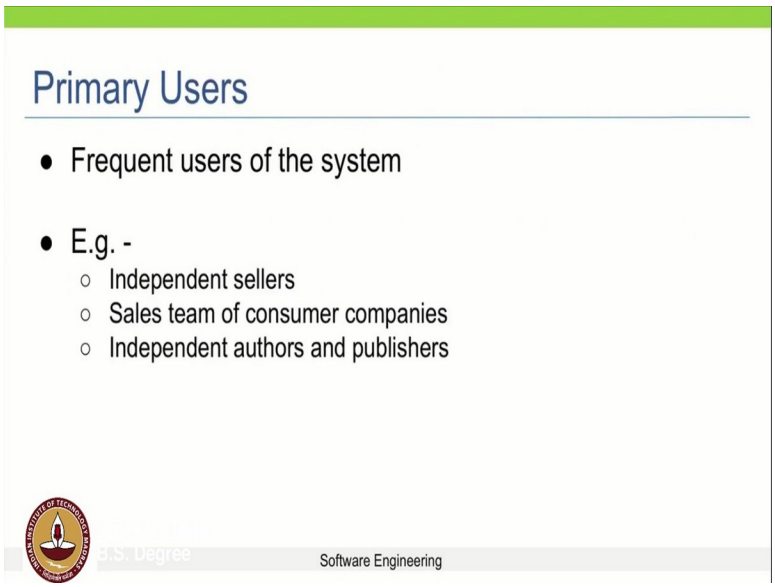
Let us start with requirement gathering. That seems very straightforward. Just go ask clients what they want, and we get a bunch of requirements. In reality, it is not that simple. First of all, there are different kinds of people who are going to be your customers in the system. All these users are called stakeholders of the system. So, when you introduce a software system, it influences different types of users. They may belong to your own organization, other organizations or end users and they may have different roles and profiles. So, what is required is to be able to understand the requirements of all these users in a broad, holistic manner before diving into designing the software. Let us look at this with an example.

(Refer Slide Time: 05:20)



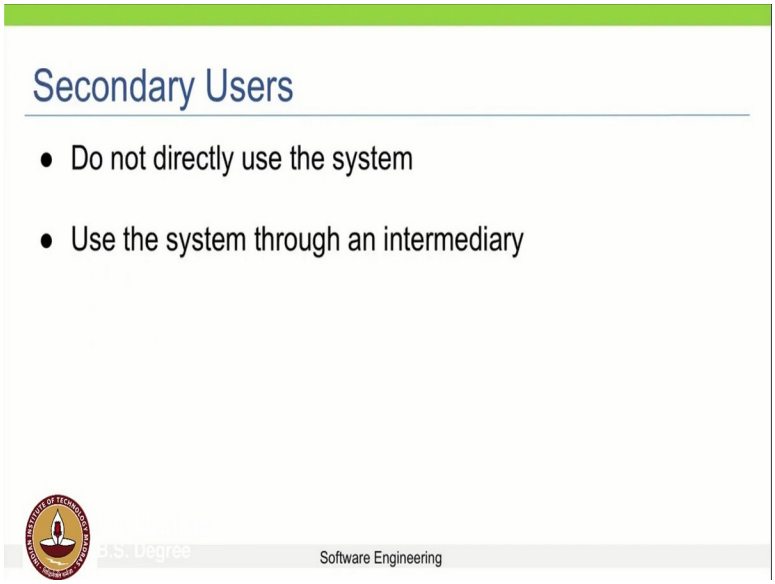
Professor Prajish Prasad: So, let us take an example from Amazon itself. And let us say Amazon wants to develop a portal for people who want to sell their products online. The products which sellers list on the portal will be available for people to buy on Amazon.

(Refer Slide Time: 05:41)



So, let us think about the frequent users of this seller portal. And these users are known as a primary users of our system. So, who do you think are primary users of this Amazon seller portal? Well, they can be the independent sellers who sell specific products. Primary users can also be the sales team of consumer companies, like those who sell electronics like mobiles, computers, etc. Even independent authors and publishers who want to sell their books online can be the primary users of this seller portal system.

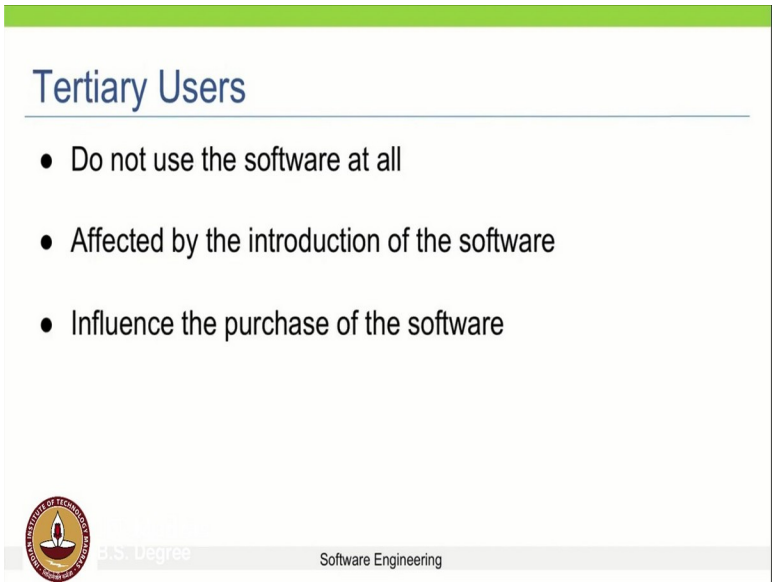
(Refer Slide Time: 06:29)



Then there are other types of users who are known as the secondary users. And these users, they do not directly use the system, but use the system through an intermediary. So, can you think of some secondary users of this seller portal? So, earlier, we mentioned that the sales

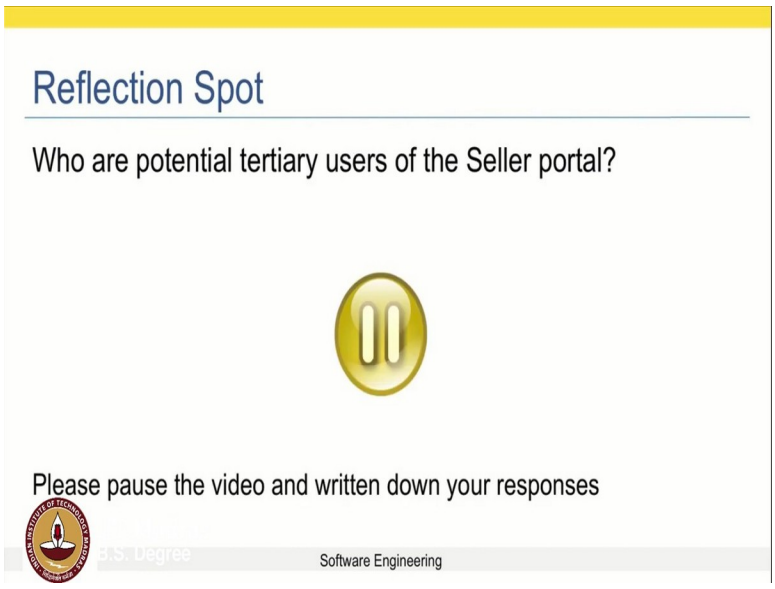
team, they can be primary users. And such sales teams usually have managers who periodically want to check the sales numbers, the profits and margins, etc. So, the sales managers can be the secondary users of this Amazon seller portal.

(Refer Slide Time: 07:16)



And then there are users who do not use the software, but are affected by the introduction of the software or they even influence the purchase of the software. So, these users are known as the tertiary users of the system.

(Refer Slide Time: 07:35)



So, learners, can you think of tertiary users of this Amazon seller portal? Who do you think our users who do not use the system at all but are affected by it in some way? You can pause

this video and write down your responses before proceeding. So, who are the tertiary users? Well, logistics shipping companies, they are responsible for delivering items to buyers. So, they are potential tertiary users of the system. Another type of tertiary users can be banks. So, the portal has to interface with banks to ensure that timely transfer of money happens to buyers based on the sale of their products.

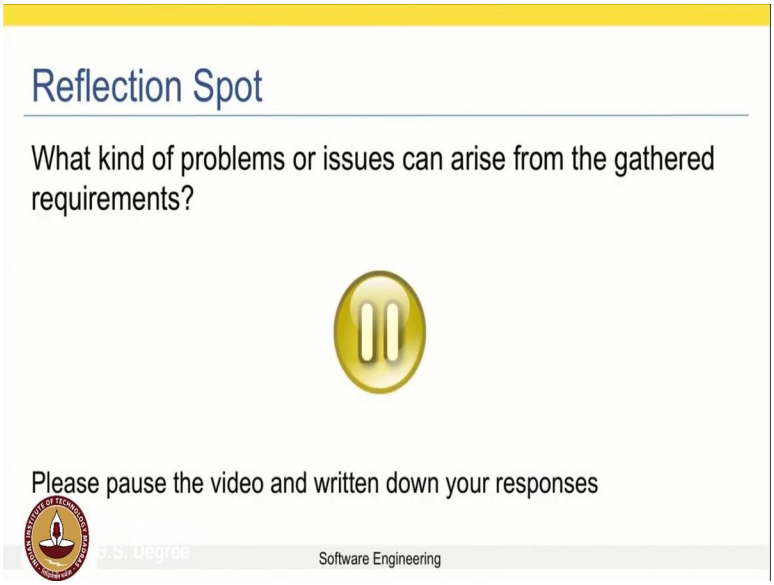
Whatever people who buy on Amazon, even they are tertiary users. They are buying depends on what products sellers sell on the portal. So, all of these users are possible tertiary users of the system.

Professor Sridhar Iyer: We saw that there are different users of the system. And we have to seek requirements from these users. So, how do we go about it? Do we randomly tap of users and ask them? Or do we conduct systematic interviews? There are many ways of doing this. In many cases, a basic interview might be sufficient. But there are also ways like studying existing documentation, having focus groups of different categories of users and also by observing what users are doing with the system or what do they need. We learn about all this in the next video.

Professor Prajish Prasad: So, now let us say we have gathered these requirements and now the next step is to analyze them. So, how do we go about doing that? So, during requirements analysis, one analyses the gathered requirements to form a clear understanding of the exact customer requirements and to way it out any problems in the gathered requirements. And these problems can arise because we will be gathering requirements from various stakeholders and from different sources. So, let us take the example of the Amazon seller portal which we discussed earlier.

Let us say we got a bunch of requirements from different clients, different stakeholders. So, now what kind of problems or issues can arise from the requirements gathered? Let us reflect on this question for a moment.

(Refer Slide Time: 10:47)



So, what kind of problems or issues can arise from the requirements gathered? Pause this video and write down your responses before proceeding.

Professor Sridhar Iyer: One issue can be that requirements are not clear. They might be ambiguous. What does this mean? For example, the requirement of independent sellers may be that they need to manage their inventory. But this is ambiguous because what does manage really mean? It depends on how you interpret it. I may interpret manage as being able to add one more product, you may interpret manage as being able to modify the price of a product or some other feature of the product. Somebody else may interpret manage as in a third way, say delete some products, add some new products and so on.

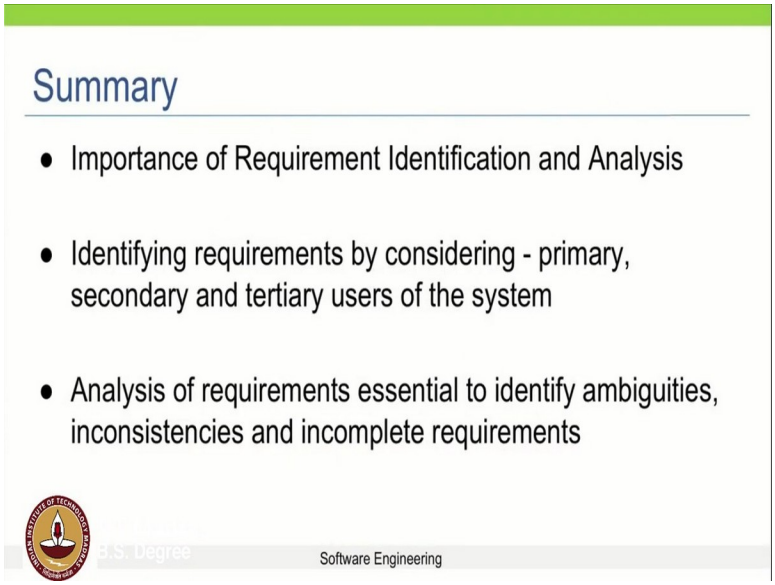
So, the point is that there are several interpretations possible. And developers will build a feature based on their own requirement. So, what we have to ensure is that the idea of what is required is the same both from the user's perspective and the developer's perspective. Another issue can be that the requirements are inconsistent or sometimes contradicting each other.

For example, if you think of the Amazon portal, it may be that independent sellers may want their payment credited every week. On the other hand, a banking system may say that the payment can be credited only bi-weekly. So, these two requirements may actually conflict with each other while building the system. So, as developers, we need to resolve these requirements in one way or the other. For example, you might say that for independent sellers, we might tell them that this is all that can be provided as a feature.

Once people know what the constraints are then you can go ahead with building the software. The third issue could be that requirements are incomplete. What does that mean? It means that some aspects of the implementation may have been overlooked, simply because neither the customer, nor the developer were able to imagine the all the details of the system before beginning to develop it. For example, independent sellers may mentioned that they want to be able to track orders. What does track orders mean? Does it mean that you want to track which orders are delivered? Or does it mean that you want to track the order while it is in transit? Or does it mean that you want to keep track of what orders have been returned, etc?

So, which type of orders does the seller want to track is an important information that the developer needs before being able to proceed.

(Refer Slide Time: 14:14)



To summarize, in this video, we have seen the importance of Requirement Identification and Analysis, identifying requirement by considering primary, secondary and tertiary users of the system. Analysis of requirement is essentially to identify anomalies, inconsistencies and incompleteness among the requirements. Once we have done this, we are in a better position to start designing the system.