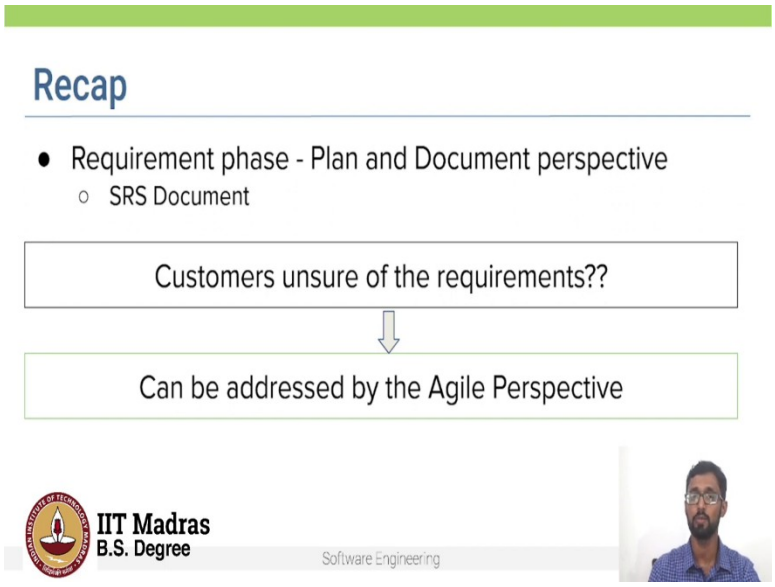
Software Engineering
Professor Sridhar Iyer
Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
Professor Prajish Prasad
Department of Computer Science
FLAME University
Behaviour Driven Design – User Stories

(Refer Slide Time: 0:19)



In the previous videos, we looked at the requirement phase in the Plan and Document perspective, how a software team can gather and analyze requirements, and how they can organize this information in a document known as the SRS, or the software requirements

specification document. We saw that SRS documents serves as an agreement between the customers and developers.

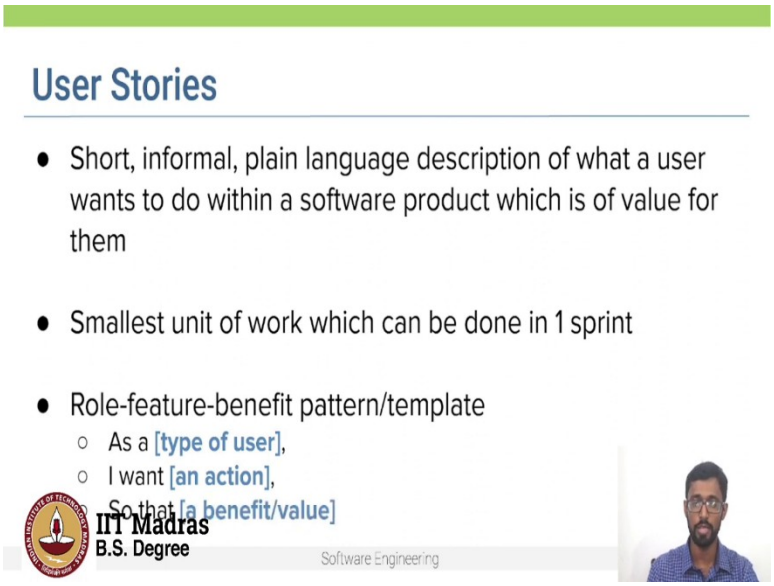
However, in many projects a customer themselves are unsure of the requirements and their understanding of their needs evolve and can change over time. So, how can we address this issue? The agile perspective can address some of these problems. So, in week 1, we saw that the Agile process involves working closely and interacting frequently with stakeholders to develop and refine requirements. It also involves building a working prototype and refining it by adding new features in sprints or iterations. And these iterations run typically every 2 weeks.

So, in the Agile perspective, we are basically in the maintenance mode as soon as we come up with some requirements, and then we start implementing features which match these requirements. So, this agile lifecycle begins with what is known as Behavior Driven Design.

(Refer Slide Time: 2:04)

So, let us look at what Behavior Driven Design is. Behavior Driven Design or BDD, it asks questions about the behavior of an application before as well as during development, so that stakeholders are less likely to miscommunicate. So, even in this case, requirements are written down. But these requirements are continuously refined to meet the stakeholder's requirements and expectations. The BDD version of requirements are known as user stories. And user stories take the place of what SRS was in the Plan and Document perspective.

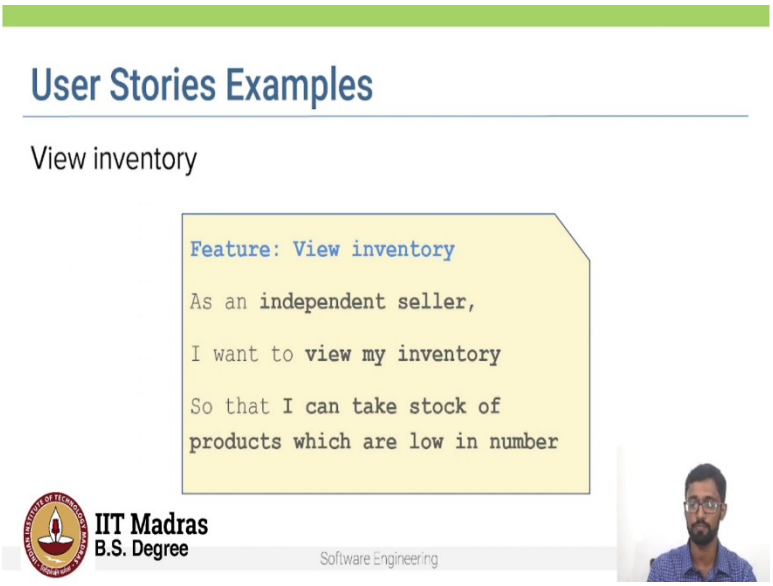
(Refer Slide Time: 02:56)



Now let us look at what user stories are. So, user stories are short, informal, plain language description of what a user wants to do within a software product, which is beneficial and of value for them. So, user stories come from the Human Computer Interaction community, or the HCI community. And these user stories are developed in 3 inch by 5 inch or 3 by 5 index cards.

So, we write these user stories in small carts. And user stories are the smallest unit of work, which can be done in 1 sprint. And 1 sprint is roughly 1 to 2 weeks. And user stories follow a particular pattern or template, which is known as the role-feature-benefit pattern, which states that as a particular type of user, I want a particular action, so that this that provides a particular value or benefit or the use. So, let us look at some examples.

(Refer Slide Time: 04:24)

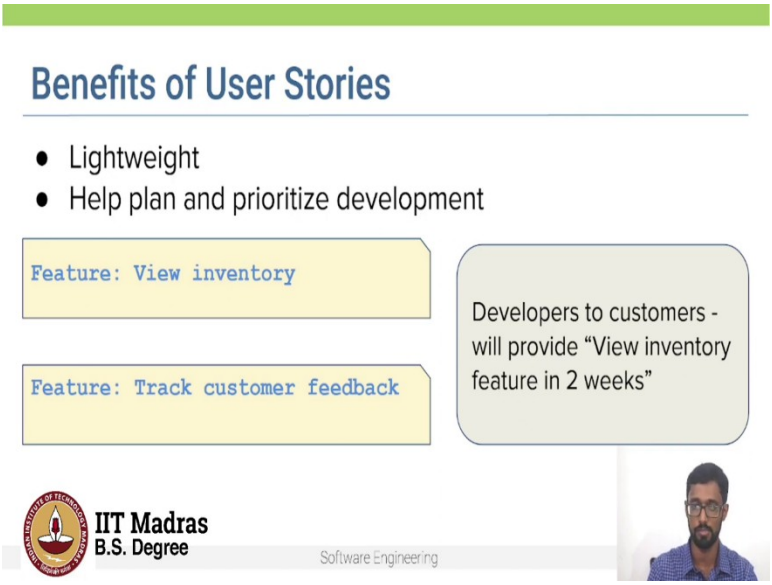


So let us take the example from the seller portal problem that we discussed in the previous lectures. Here is a requirement that we gathered from several stakeholders about viewing an inventory. So, user story for this is shown here. The feature is view inventory. So, as an independent seller, which is a type of user, I want to view my inventory, which is an action So, that I receive a particular benefit that is I can take stock of the products which are low in number.

(Refer Slide Time: 05:07)

Let us take another example of tracking customer feedback. So, what will the user story for this look like? So, here again, we follow the same pattern as an independent seller, which is the type of user, I want to view my customer's feedback for each product. And so that I can get a sense of the pertinent issues in my product. So, this is the value which we are providing to the user.

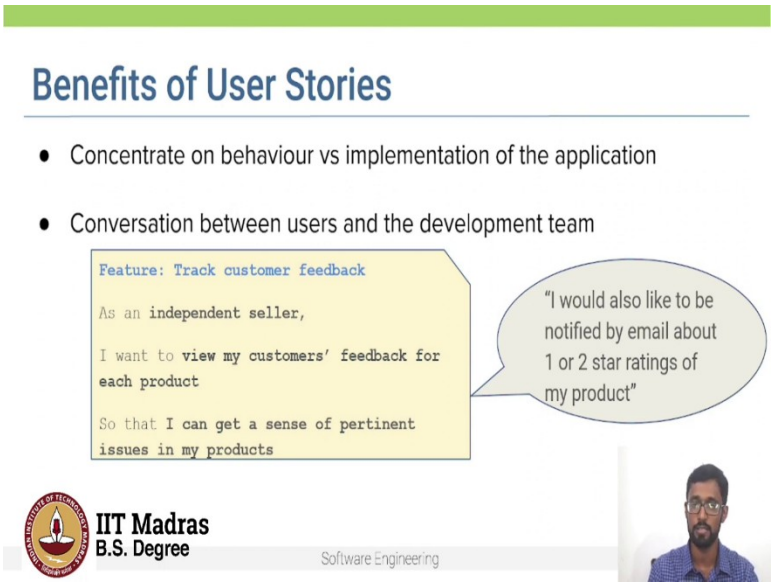
(Refer Slide Time: 05:46)



So, what are some benefits of these user stories? One, we saw that they are lightweight versions of the requirements, as opposed to the SRS documentation, which we saw earlier. Second, User Stories help stakeholders plan and prioritize development. So, for example, if I want a feature, which has to be implemented first, maybe for to get immediate feedback or to get a feel of the system, I can prioritize the development of the user story corresponding to that feature.

So, for example, we have two features of viewing inventory and tracking customer feedback. In this case, the client might want to implement the catalog and inventory first. So, the development team can tell the customers that we will provide the view inventory feature in 2 weeks.

(Refer Slide Time: 06:52)



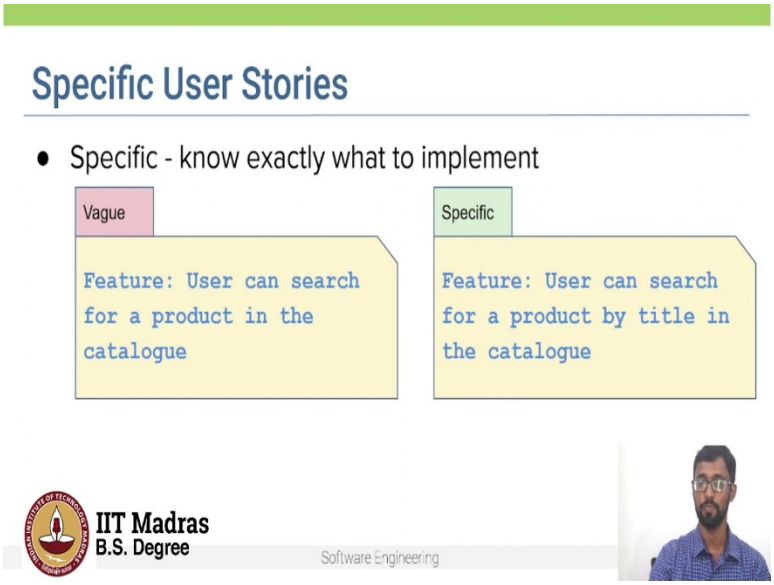
So, if you have noticed, we have not told the customers how we will implement these features. So, by concentrating on the behavior versus the implementation of the application, it is easier to reduce misunderstanding between stakeholders. Another benefit is that it facilitates conversations between the users and the development team. So, more than creating user stories, it is this conversation this back and forth between the different perspectives of each participant that will bring better simpler and more valuable solutions to user's problems.

For example, let us take this feature of tracking customer feedback. So, when we show this to a client, the customer or the client will realize that maybe I would also like to be notified by email about the products which have low ratings of 1 or 2 stars. And such modifications are common in the Agile perspective. And user stories can facilitate such modifications.

(Refer Slide Time: 08:13)

So, what are some guidelines to create good user stories? The acronym SMART can help us come up with good user stories. SMART user stories are Specific, they are Measurable, Achievable, Relevant, and Timeboxed. So, let us look at each of these in detail.

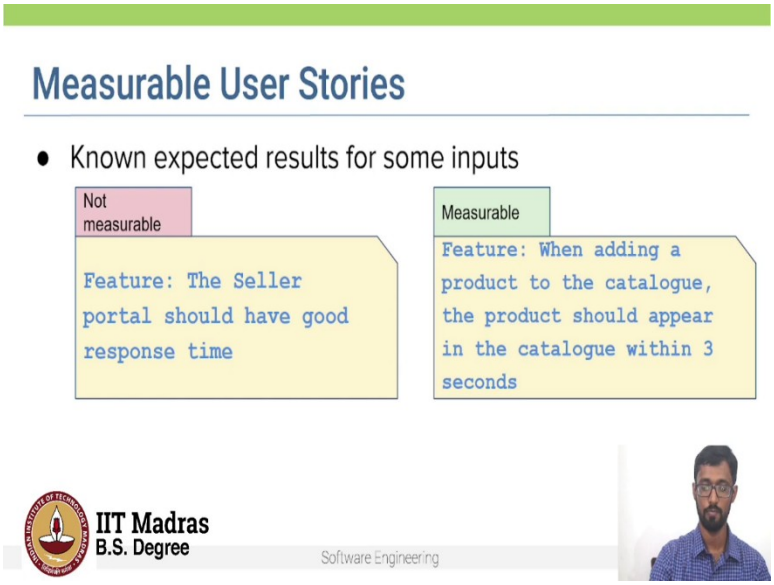
(Refer Slide Time: 08:43)



First, user stories should be Specific. So, specific user stories are essential so that we know exactly what to implement. So, let us consider this user story, which states that users can search for a product in the catalog. So, this is a bit vague, because we are not exactly specified. For example, what type of product it is, or based on what attribute can be searched

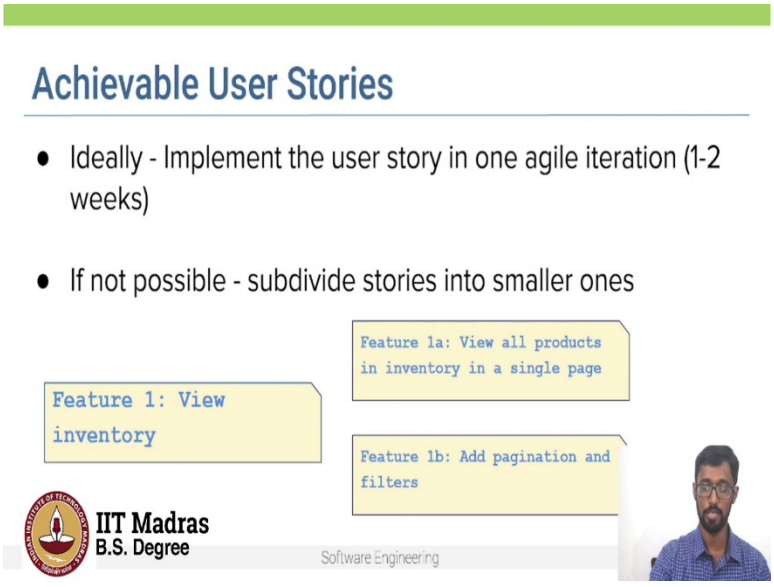
for a product. A more specific user story can be that the user can search for a product by title in the catalog.

(Refer Slide Time: 09:33)



Next user story should be Measurable. So, each story should be testable, which implies that there are known expected results for some good inputs. So, let us take this feature, the seller portal should have good response time. So, what do we mean by good? This is vague and it is not easily measurable, a more measurable user story can be, so when adding a product to the catalog, the product should appear in the catalog within 3 seconds. And this is measurable, and this is testable. So, make sure that your user stories are measurable.

(Refer Slide Time: 10:25)



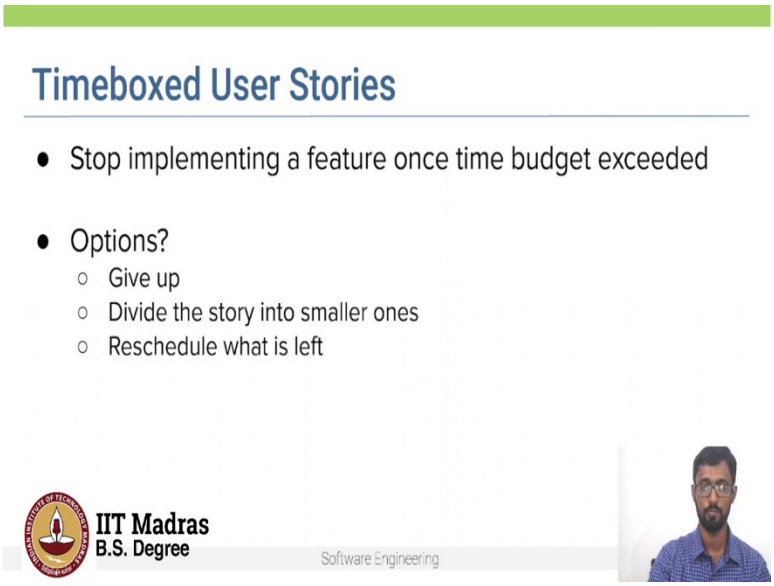
Next user stories should be Achievable. Ideally, we should be able to implement a user story in one agile iteration that is in 1 to 2 weeks. And if you are not able to complete one story per iteration, then they are too big. And you need to subdivide these stories into smaller ones.

So for example, view inventory might not be able, we might not be able to implement this in 1 week or in one iteration. Hence, we can divide it into two or three features that feature 1a is we implement the feature of viewing all products and the inventory in a single page. And then we add pagination and maybe filters can be the next feature. So, in this way, we make sure that one user story is completed in one agile iteration.

(Refer Slide Time: 11:38)

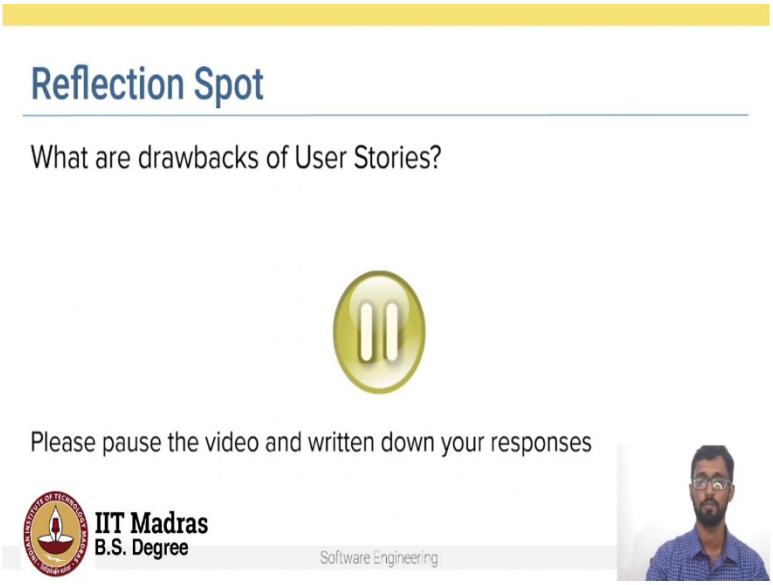
Next user stories should be Relevant. So, user story must have some business value to one or more stakeholders. So, how can we be sure that a user story is relevant? We should ask questions; we should ask questions like why is this feature needed? Is it really important is it adding any value? So, what if this feature is added? So, these questions can help us and ensure that the user stories which we add are relevant?

(Refer Slide Time: 12:18)



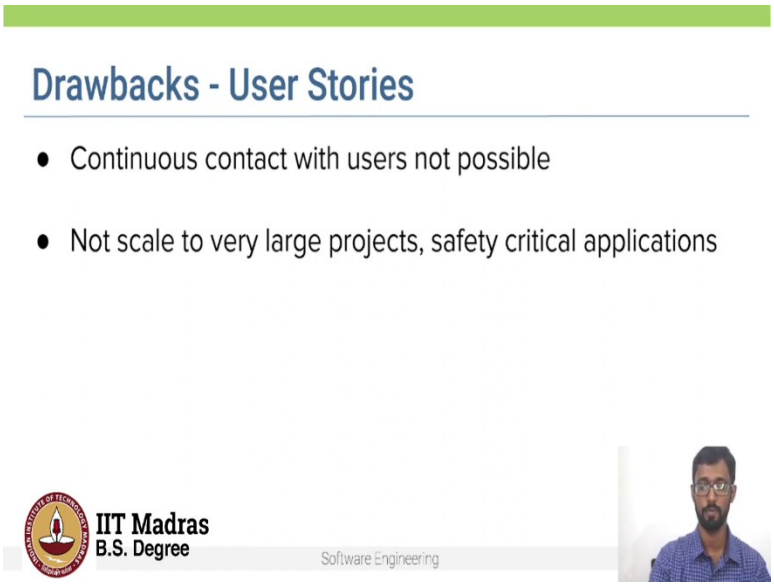
Next, user stories should be Timeboxed. So, Timeboxed means that we should stop implementing a feature once we have exceeded the time budget. So, what should we do if we have exceeded the time budget? Once we either give up? Or as we saw earlier, we divide the story into smaller ones. And or we reschedule what is left according to a new estimate, we revise our estimates. So, it is important to make sure that you are on schedule.

(Refer Slide Time: 13:10)



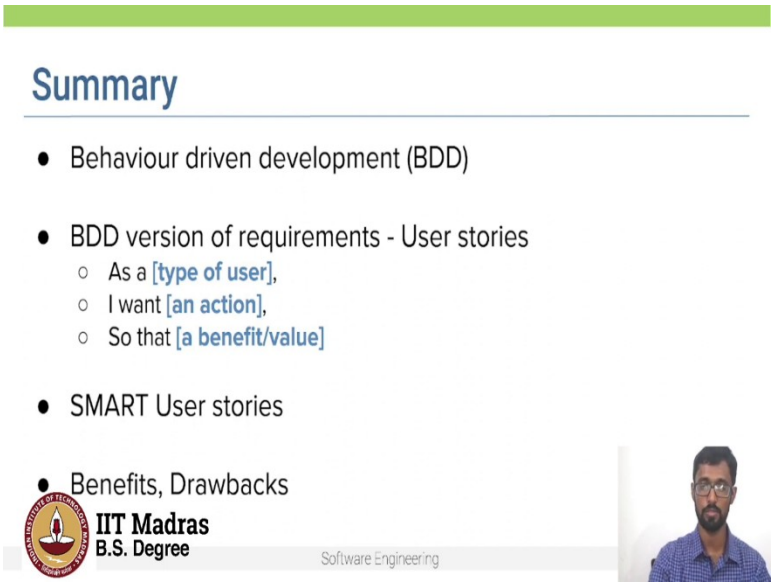
So we have seen the benefits of user stories, but let us reflect on this question for a moment on what are certain drawbacks of user stories? Are they always good? Can they not be used in certain situations? Please pause this video and write down your responses before proceeding.

(Refer Slide Time: 13:39)



So, what are certain drawbacks of user stories? One, it may be difficult or too expensive to have continuous contact with customers throughout the development process. Because customers may not want to participate or they may or may not be available at all times. Second, this approach may also not scale to very large software development projects or even safety critical applications because in such projects, extensive planning and documentation are required before the actual implementation, and especially for safety critical applications, there is no room for errors or refinement. So, the key point to remember is that software development methodologies depend on the type of application you are building.

(Refer Slide Time: 14:43)



So in this lecture, we looked at behavior driven development or BDD, BDD ask questions about the behavior of an application before and during development, so that the stakeholders are less likely to miss communicate. We saw that the BDD version of requirements is user stories. We saw the roll feature benefit pattern or template.

We looked at the characteristics of good user stories, that is smart user stories that it should be Specific, Measurable, Achievable, Relevant and Timeboxed. And we also looked at the benefits and drawbacks of user stories.