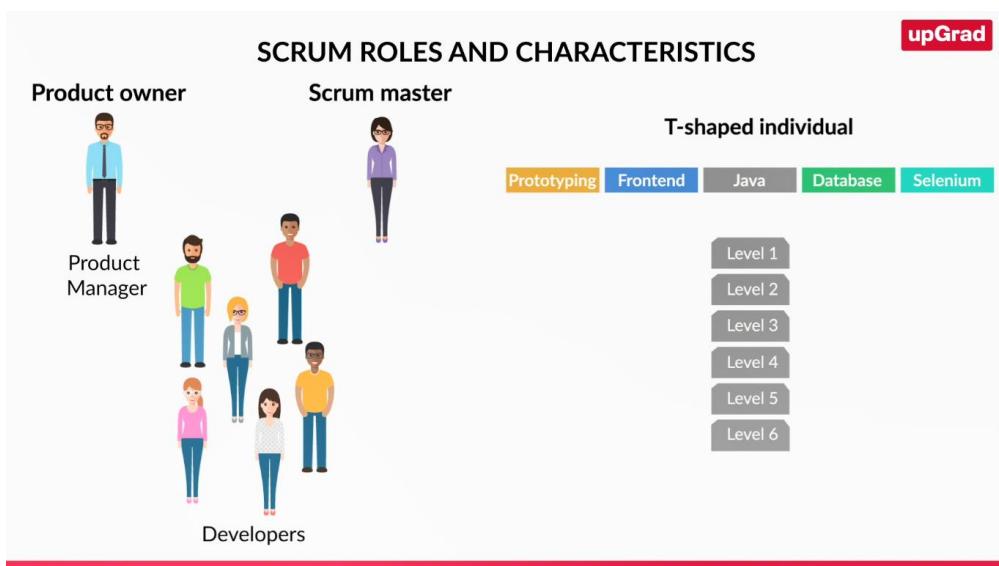


## Transcription

# Scrum Roles



In the previous session, you were introduced to the scrum framework. You learned how scrum brings out some of the agile principles and values for the teams. Now let's find out more about the scrum team and especially about the two roles of product owner and scrum master. But first of all, what are the characteristics of a scrum team?

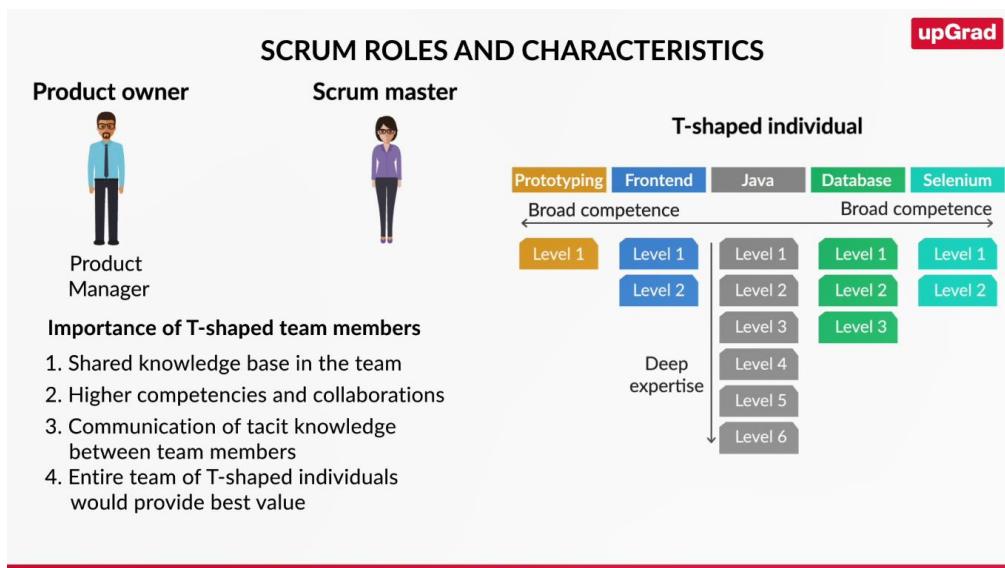


Scrum is a very simple framework. It only has three types of roles. The first role that we talk about is known as the team. Now, everybody who is a part of the team is a scrum team member. The second one is a Scrum Master who is a special role, who is one of the team members inside a scrum team.

And finally, the product owner role, some of you who are aspiring product managers, this might be closest to what you already know or what you aspire to be. So, one of the first concepts that we talk about when we talk about an agile team is the notion of a T-shaped individual.

Now what happens is most of the time, if I am a Java engineer, or if I know PHP, I'm very unlikely to be very narrowly functioning only in that domain, I probably may not step outside the boundaries of that and say, can I understand what is selenium or how do I use a balsamic, for example, for prototyping or something else.

Chances are very likely that I'm going to be stuck in only that particular domain there. The T-shaped individual takes that notion in a slightly different manner. It says you can still be the master of one subject or a discipline that you have chosen, but over a period of time, you can spread your horizon.

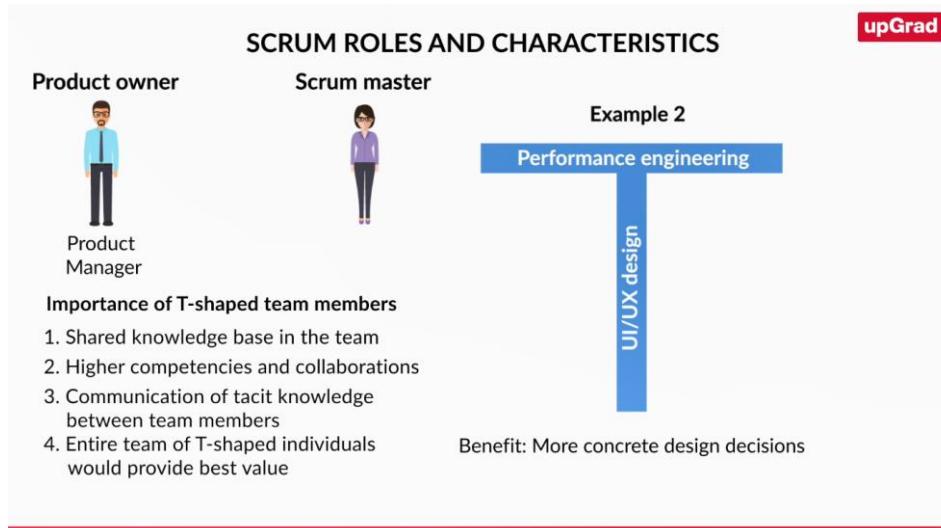


So, as to, if someone were to pictorially depict your skillset, it is more likely to look like the letter T of the alphabet, which means you have a deep expertise in one area, but you have a broad set of competencies in multiple other adjoining areas.

Now you may not be a world class expert in each of their joining areas, but you have a much than higher kind of a competency in those areas.

Why is it important? It is important because it allows you to create more shared knowledge base with the rest of your team members. It will create a much higher sense of shared competencies and collaboration inside the team. Secondly, this is also a mechanism by which the scrum teams communicate asset knowledge among themselves.

So, we talked about T-shaped individual, what if the entire team was made of T-shaped individual? If there is only one individual, it probably is not the best value for the team. The same notion as a fractal pattern can actually be scaled up at a team level.



Similarly, I might be a test engineer who is probably going to spend some time in code reviews. Now I may not really be expected to learn coding, but by getting into code reviews, I might be able to design better test cases because now I understand the design a little better, and maybe I am a designer whose job is to really design beautiful slick UI and great user experiences. And probably I want to spend my time learning about performance engineering on the team.

Now, the argument could be why is it important? But again, if you as a user designer do not have a sense of what kind of performance penalties might be imposed by a certain design choice, you might only be making half a decision. You might be making a decision in the absence of any concrete data.

So, this is the way we are basically, we can stretch this argument at a team level, across all the roles on a scrum team and make the teams more effective.



Okay, wait a bit. I need some explanation for a couple of terms. What is code review and what is performance engineering?



The slide features a video frame of a man in a white shirt and glasses speaking. To his right is a white box with a red 'upGrad' logo at the top. The title 'CODE REVIEW' is centered above three numbered points:

- 01 Reviewing the code to ensure high quality
- 02 High quality code requirements
- 03 Advantages of high quality code
  - a. Faster application
  - b. Less memory usage
  - c. Better user experience

What is code review? Code review as the name suggests is reviewing the code to ensure that the quality is good.

What do I mean by quality of code? Just like you have some cars which are of good quality, sturdy, safe, comfortable, and some which are not. Similarly, you can have a code which is written using proper standards and code, which isn't.

Code quality ensures that the code written by one programmer is easily understandable by the rest of the team. A poorly written code might not be understandable even by the same guy who wrote it six months ago.

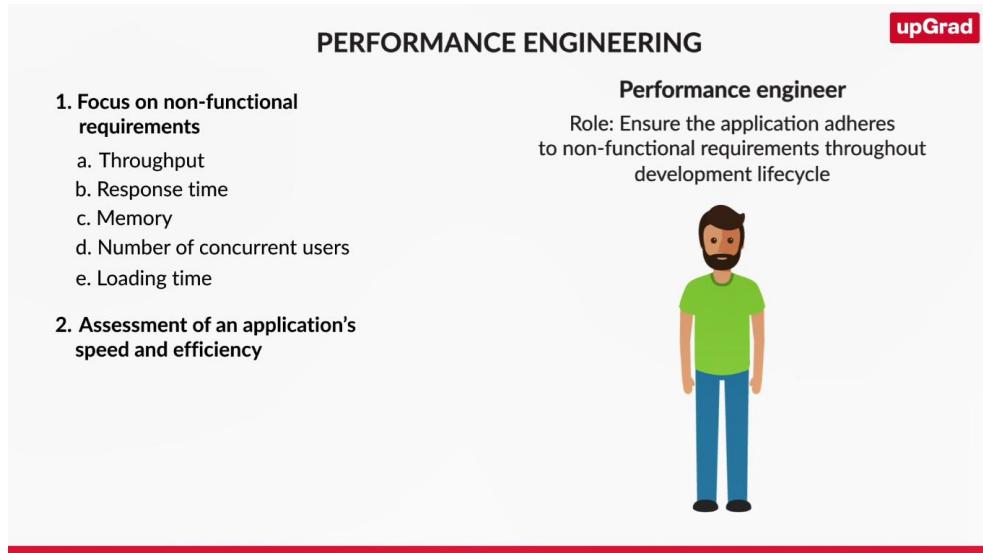
Good quality code can potentially make an application fast, use less memory, and can also improve the user experience. Hence, code reviews ensure that everyone in the team understands each other's code and hence improves agility in the team.



The slide features a video frame of the same man speaking. To his right is a white box with a red 'upGrad' logo at the top. The title 'PERFORMANCE ENGINEERING' is centered above two numbered points:

- 01 Focus on non-functional requirements
- 02 Assessment of an application's speed and efficiency

Now let's understand what performance engineering is. With the request for creating an application also come the non-functional requirements. These include things like throughput, response time memory, etc. I know these are fancy terms, right? But just to explain them briefly, they tell us how quick and efficient an application should be.

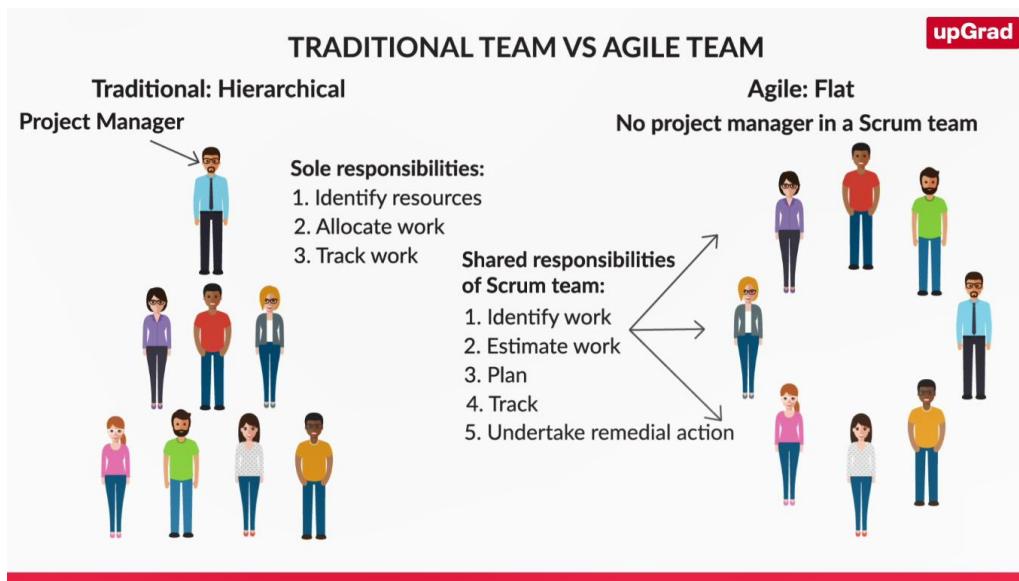


You all must have seen how some ecommerce websites might be down during their flash sales. This is due to heavy traffic on these websites. Now these non-functional requirements, like how many numbers of concurrent users the application need to support, loading time, etc., are taken into consideration by someone known as performance engineer.

The task of a performance engineer is to ensure that all the aspects of the development life cycle of the application adhere to these non-functional requirements.



Coming back to scrum and agile, how is an agile team different from a traditional team?



Now, let's first look at the traditional team. The traditional teams, typically in our industry have been very hierarchical teams. There is typically a project manager or a team leader who is responsible for a group of individuals in the team.

Each of these individuals come with their own specific competency, programming, testing, database, designing and so on. And the project manager has traditionally been chartered with the responsibility to identify the resources for the team, allocate the work to them, track the same thing and so on and so forth.

When we move away from a traditional team into an agile team, this kind of a pyramid changes the shape. We don't believe that hierarchy is a great way in a knowledge context. A small team of six to eight individuals when it is working together, the highest performance and the most effectiveness comes from working together as a flat team, working together as an agile team.

It is a scrum team that is responsible for doing the work estimation, planning their sprint, tracking their work, and finally being accountable for their deliverables.

The role of a project manager doesn't exist in scrum, as we have seen because there are only three roles in that and manager is not one of them. So, what happens to the project management? The simple answer is it does not go away.

Just because the role goes away, the responsibilities don't go away. In fact, the responsibilities get democratized and it is shared by all the team members. To that extent, some authors also say all the three roles in a scrum team, product owner, scrum master, and the team are all management positions.

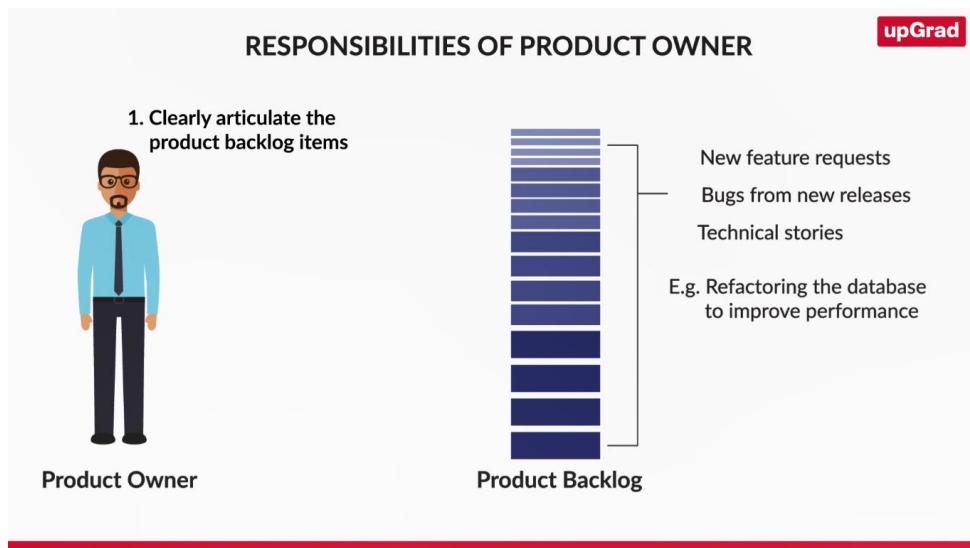
Because all of them are doing a management responsibility, which is identifying the scope of work, estimating the scope of work, planning the work, tracking the work and taking any remedy election that might be needed by the team to bring the performance back to the plan.



This video gave some more information on the scrum framework. In a scrum team, every member is a T-shaped individual who has a deep expertise in one area, but also has competencies in multiple other areas. Further, a traditional team is very hierarchical while an agile team is flat, and everyone is accountable for their own tasks. After this, let's get into the details of the role of a product owner in the next video.

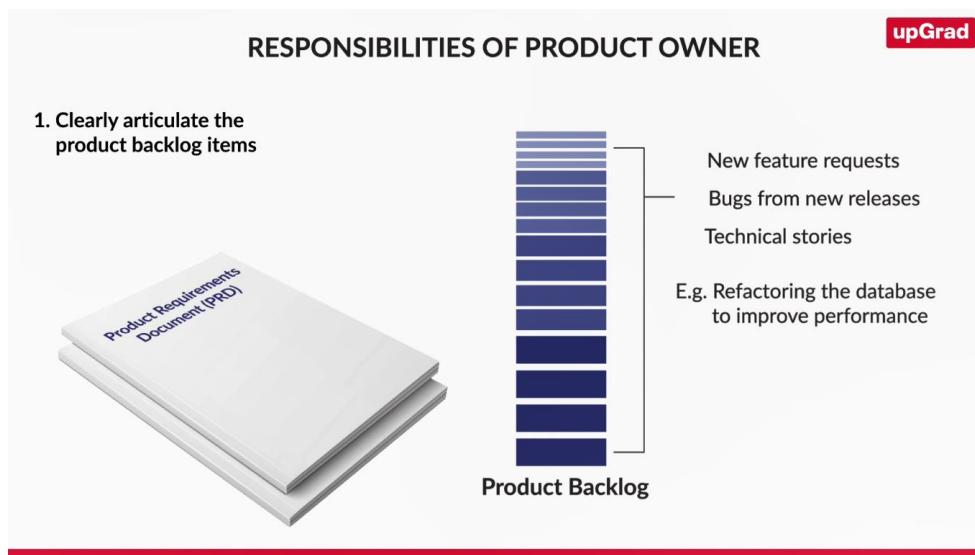


In the previous video, you saw how every member in a scrum team should be a T-shaped individual, a sort of Jack of many trades and a master of one. Now let's find out more about the role of the product owner, because that is the closest what a PM would be in the team. So, what does being a product owner entail? What are the responsibilities this individual has?



Well, three or four major points here. The first one, they are responsible for clearly articulating and expressing the product backlog items. What is the product backlog items? Anything that exists on the product backlog. These could be the new feature requests, these could be the bugs that are coming from the previous releases, or these could be technical stories.

For example, somebody might say, I need to refactor this database because the performance is very slow. It may not have a direct functionality being released to the customers, but it is needed from an architect point of view, to improve the performance of the database.



So, any of these items is collectively known as a product backlog items. Now, traditionally, we have seen that these items were either created in huge amount of details. Well, we have seen the traditional product requirements documents, also known as PRDs.

We typically have expressed them in big fat documents, 30, 40, 50 pages document. They often took a lot of time and they often took a lot of effort before these could be consumed by the development team.

In an agile team, and specifically in a scrum team, we believe that light amount of details in a product backlog item are just enough. You don't need to go and specify every single detail there.



So, the question comes, who takes care of the details? Well, the product backlog items are deliberately written at a high level so that when the teams really have a discussion at the time of implementation, they go to the product owner and say, Hey, what do you mean by this? Is that what you mean by that? And at that time, the product owner can actually give them the bring down.

The typical pattern is they go to the white board and they really chalk it out. And in half an hour, they have all the details available. That makes sure that the discussion is not caught up into too much of excessive details, even in the initial stages. But we maintain the high-level vision of it.

And when it comes to the implementation details, we understand because it's possible from the time I identified the feature, till the time I actually implemented, maybe there is a two months of time lag.

Now, a lot of things could have changed in those two months. There might be new mechanisms available. Why do we want to tie down a specific implementation to what is known today, knowing very well that the implementation will happen after two months, and a lot of changes could have happened there?

So, scrum is of the school of thought that we want to be as much lean as possible, which means defer the decision until the last responsible moment. And that makes sure that we have the most innovative ideas. We have the latest technology that we can consume when it comes to implementing a given feature. So, that's the first part of a product owner role.



Let me stop you there. You must be wondering what is a PRD?

A video frame showing a man in a dark blue polo shirt with small white dots, gesturing with his hands as he speaks. To his right is a white callout box with a red border. The title of the box is "PRODUCT REQUIREMENTS DOCUMENT (PRD)". Inside the box, there are three numbered items: 01. Contains all product requirements, 02. Enables team to understand, and 03. PRD for a login page. Below item 03 is a list: a. User goals, b. Personas, c. Requirements, d. Tasks, e. Key challenges, f. Test cases.

01	Contains all product requirements
02	Enables team to understand
03	PRD for a login page <ul style="list-style-type: none"><li>a. User goals</li><li>b. Personas</li><li>c. Requirements</li><li>d. Tasks</li><li>e. Key challenges</li><li>f. Test cases</li></ul>

What is the PRD? A Product requirement document also known as a PRD is a document containing all the requirements for a certain product. It's written to allow people to understand what a product should do, how a particular feature should function.

Let's say the function is the login page of your app. So, in the PRD for this feature, you would include the user goals, personas, requirements, tasks, key challenges along with the test cases.

We will get into the details of a PRD soon. For now, understand that even though a product owner communicates the requirement at a higher level to the team, a tentative requirement document with details is built so that later when the team approaches the product owner for more details, the PRD acts as a support document to have a detailed discussion.



Let's get back to the role of the product owner. What are the responsibilities does this team member have?



The second thing is ordering of these items. When we are looking at a sequence of product backlog items, there has to be a way in which these are ordered. We typically make buckets and say, this is high priority, medium priority, low.

But in scrum, the expectation is that there is a very clear priority order of one, two, three, four, five, six, seven, eight, nine, ten. These could be ordered based on priority, based on value, based on risk. It is really up to the product owner, as long as there is a clear, consistent way for a given product to linearly order that.

Another responsibility of the product owner is to make sure the product backlog at all times is kept visible, updated, and transparent to the team.

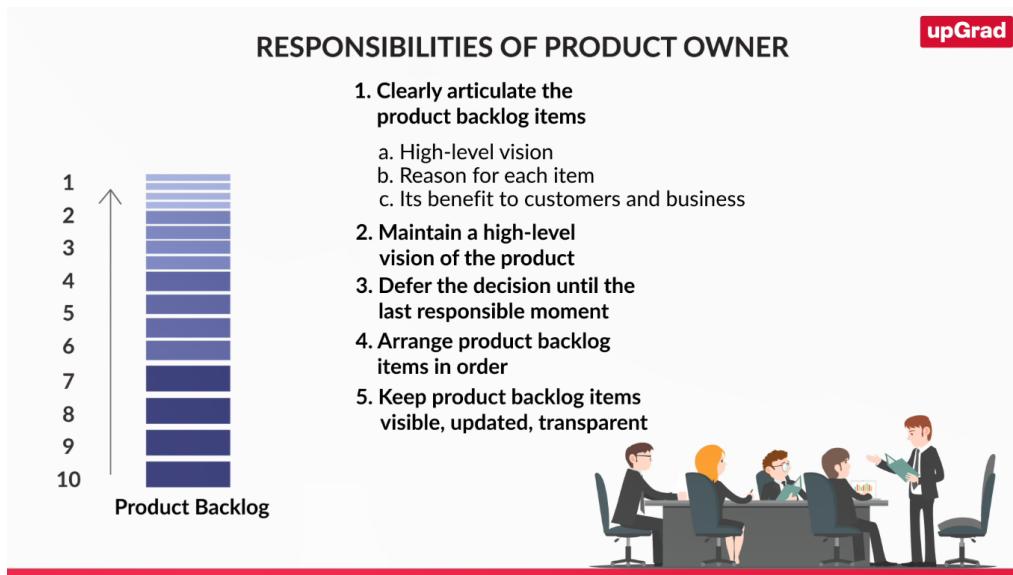
The traditional PRDs were more in the style of doing it one time, baselining it, and often the documents were stored somewhere in an electronic repository and not always accessible to all the people. Even if they were accessible, it would not normally be the expected behaviour that people would go out and look at that.

However, in the case of a scrum team, the product backlog is not only always changing, it is also always used by the team members.



We will talk about one of the scrum events known as backlog grooming. And in that event, typically the team members and the product owners are discussing and collaborating about the product backlog, what changes have happened, any new requirements that have come in, any new prioritization that happened on almost on a weekly basis.

So, this is an important part of the product owner responsibility to keep the team posted on what is coming up, what will be the activities they are likely to work on in the coming sprint and so on and so forth.



And finally, the most important part the product owner must do in terms of keeping the team on the same pages, clearly articulate and make them understand what each of these product backlog items are all about, help them understand the reason for their existence, the benefit from implementing these features that the customers are expecting and so on and so forth. So, these are some of the key activities that the product owners do.



After having understood what a product owner really does, let's check out an example of this role from BookMyShow.



At BookMyShow while development teams are working on the current sprint goal, the product owner prepares a backlog items for the next sprint planning meeting. In order to bring every member of the team on the same page, the team dedicates about one hour, it's just like a time box meeting once a week to understand technical feasibility like implementation, dependencies, approach, impact, etc. And also, to identify impediments and risks, if any.



BACKLOG GROOMING MEETING

01	Product owner explains the user stories
02	Team <ul style="list-style-type: none"><li>a. Adds user stories/epics to product backlog</li><li>b. Extracts user stories from existing epics</li><li>c. Prioritises and estimates work</li></ul>

For the upcoming backlog items, we call it as a backlog grooming meeting. Here, usually a product owner explains the user stories for the upcoming sprint. And the team helps in adding new stories and epics extracting stories from existing epics, prioritization and estimating efforts for the existing product backlog items.



You saw that the product owner is responsible for clearly communicating the product backlog items with the scrum team. He orders these items according to their priority on a scale of 1 to 10, one being the highest. The product owner also ensures that the product backlog items are updated and always visible to the team.

Lastly, he also helps the team make a sense of the product backlog and the items therein. Next, we'll learn a bit more about the product owner role. See you on the other side.



You learned about the different responsibilities of the product owner. Now, some of you might be wondering, there seems to be a lot of activities, can the product owner do all of them independently, or is collaboration with the rest of the team important?

A video frame showing a man in a striped polo shirt speaking. To the right, a white callout box titled "ROLE OF PRODUCT OWNER" contains the text "Can product owner work independently?". Below it is a table with two rows:

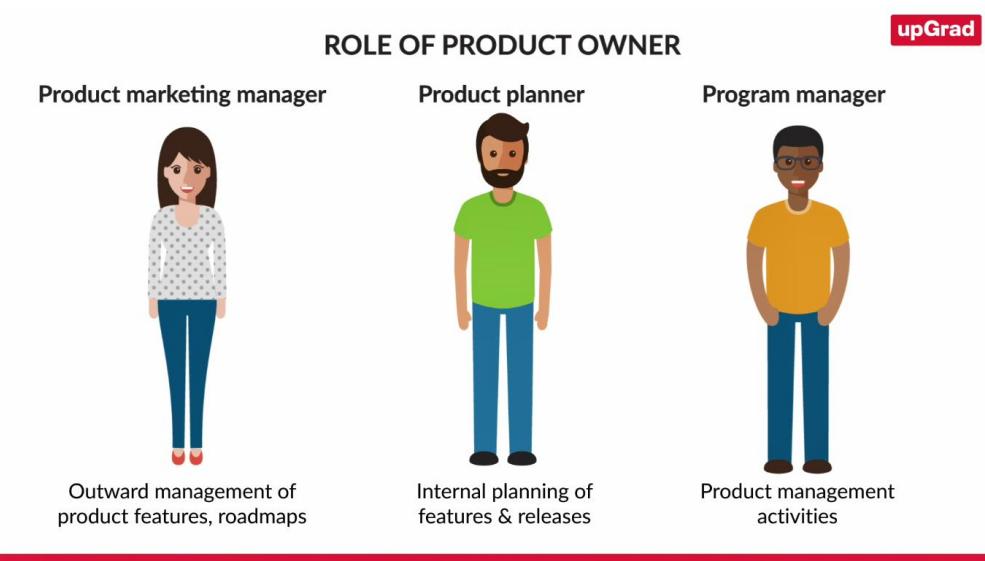
Maybe	If product and team are small
NO	Multiple locations and operations

The final product decisions must be made by the product owner

The short answer is no, the product owners cannot do all the things independently. If it's a small product, let's say only four or five people are working on the entire product, maybe it is possible to do that because the product owners have all the moving parts that they can put on a table and they can talk about it.

But suppose there is a team that is spread in three or four locations, it requires a lot of other events, like a technical marketing or customer support or manufacturing and so on and so forth, obviously the product owners will not be able to do it all by themselves.

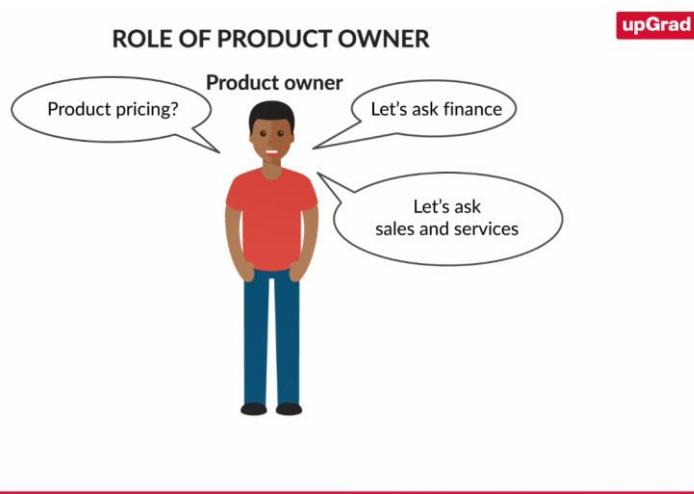
Even then the recommendation from scrum is they can take as much help from each other. So, there's nothing wrong in really taking help from all the other stakeholders. But the final decision must be made by one individual, and that is product owner.



What happens is in a lot of organizations I worked at and I've seen, there are two parts of these activities. For example, there might be a product marketing manager who's really more responsible for an outward management of how the product features and the roadmaps might be happening.

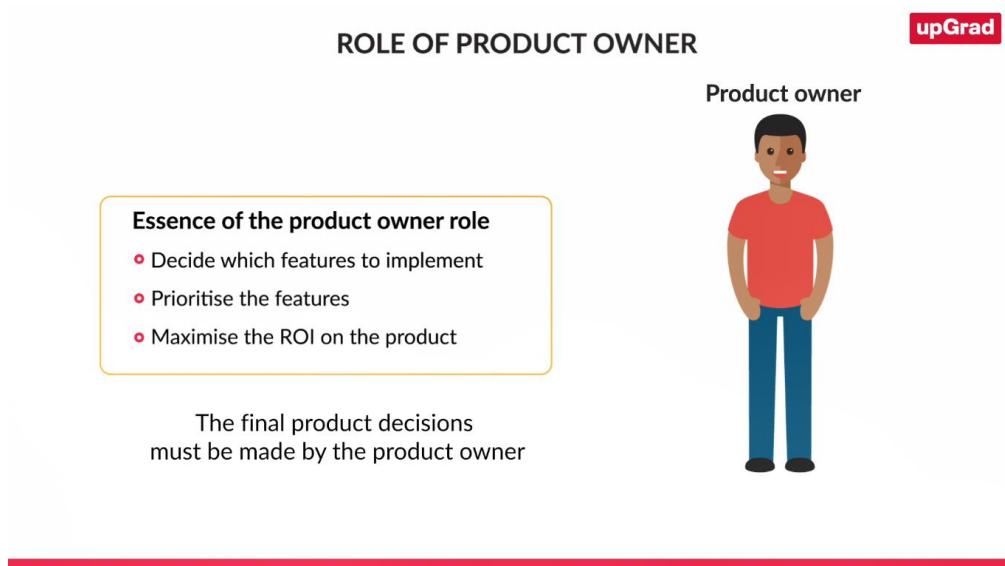
There is typically a product planner in some companies who is really more chartered with, Hey, how are these things going to be planned and how the releases will be planned and what features will go on in each release internally and so on.

So, there could be a multitude of different roles that might be responsible for. In some companies, I have also seen the role of program manager who is chartered with additionally responsible for the product management activities on the product. So, these are some of the things that make it very complex.



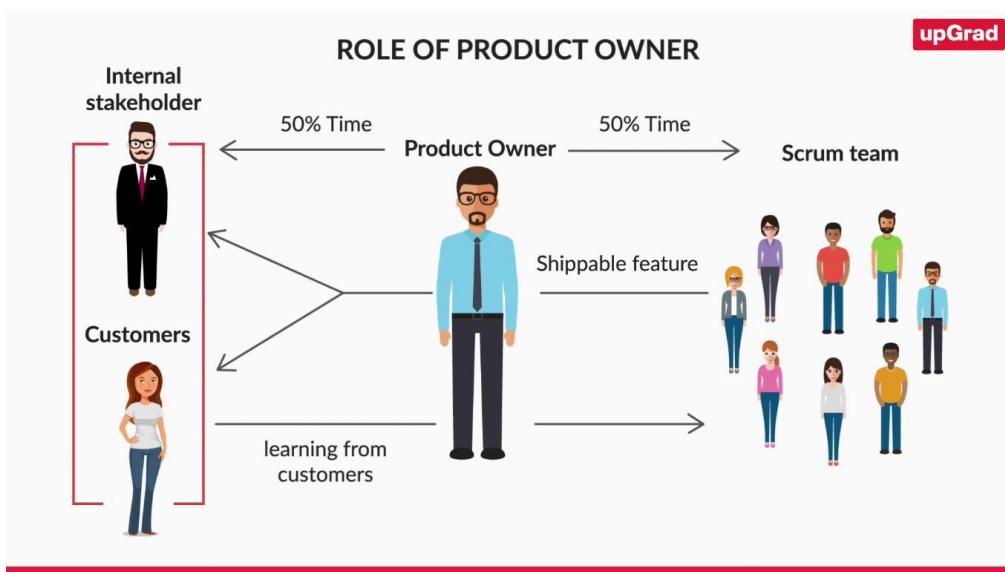
In scrum, the position that the creators of scrum took is, there can only be one role. And that one role is scrum product owner role. And there is no more finger pointing about it. If there are decisions that are, for example, pricing decisions.

Now, obviously a product owner may not have all the moving parts, all the information, all the factors known in order to come at the pricing decisions, they might need to talk to the finance team. They might need to talk to the sales and services team and so on. They will do so in order to do that.



But when it comes to really deciding which features will be implemented, what is the right priority order for the, what is the relative value for them? So, as to maximize the ROI on the product, this is the essence of the product owner role, which is undertaken only by one individual.

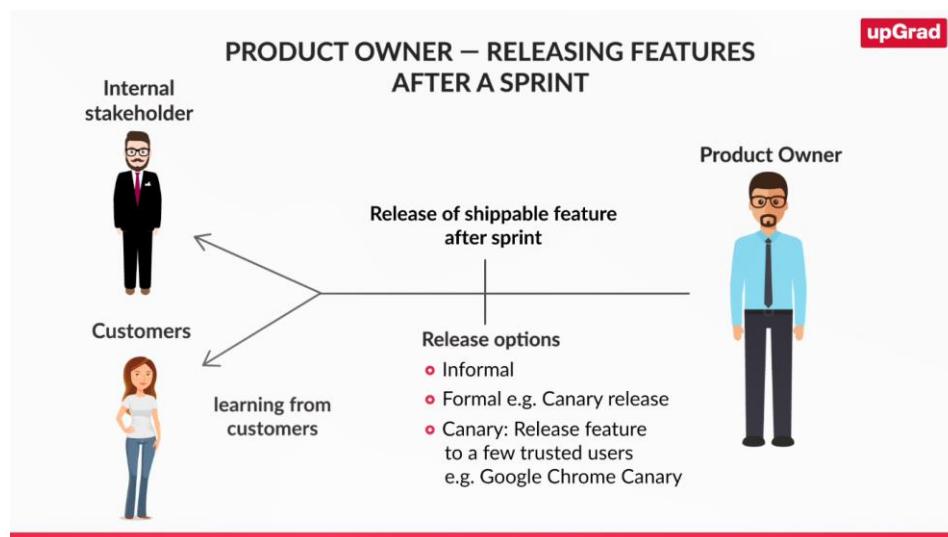
It cannot be a committee because the scrum belief is that decision made by committee will invariably short chain some of the decisions. It wants one person to be singularly and wholly responsible for the entire product features that are going on that.



Product owner says, unlike the traditional product managers, they are not the people who typically have a kind of a long absence away from the team. In fact, the best product owners spend about 50% of the time with the team and 50% of the time in the field with the customers.

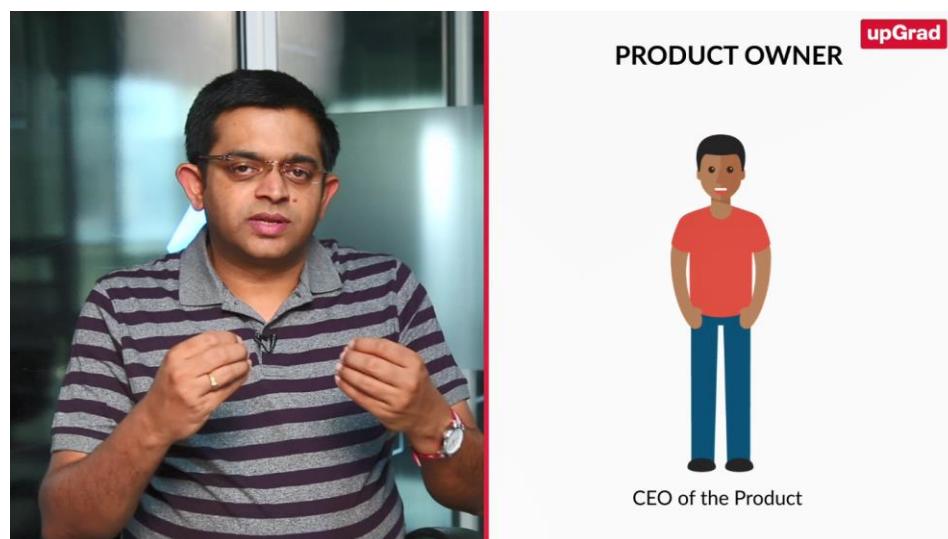
And they do that because they are able to crosspollinate the learnings from whatever they have learned from the customers, bring it back to the team, and to help the team understand so that they can keep moving. And when the team has delivered something of value, every two to four weeks, they take them back into the field.

It could be internal stakeholders like sales and marketing, customer support and so on, or it could be external customers straight away out in the field and socialize some of the early features.



They might do them in informal gathering, or they might do that in a very formal manner where they might do something like, that Canary releases that Google is famous for.

So, they basically would do some early versions like a Canary release, and they would only release it to a few trusted users. And based on the response from them, they will actually decide to make it mainstream.



So, the product management is a very heavy function. In fact, the scrum team is, in scrum, we actually call product owner is the CEO of the product. Even though they are not really the boss of the product, the importance and the stature of the product owner is so high that everything really revolves around the product owner.

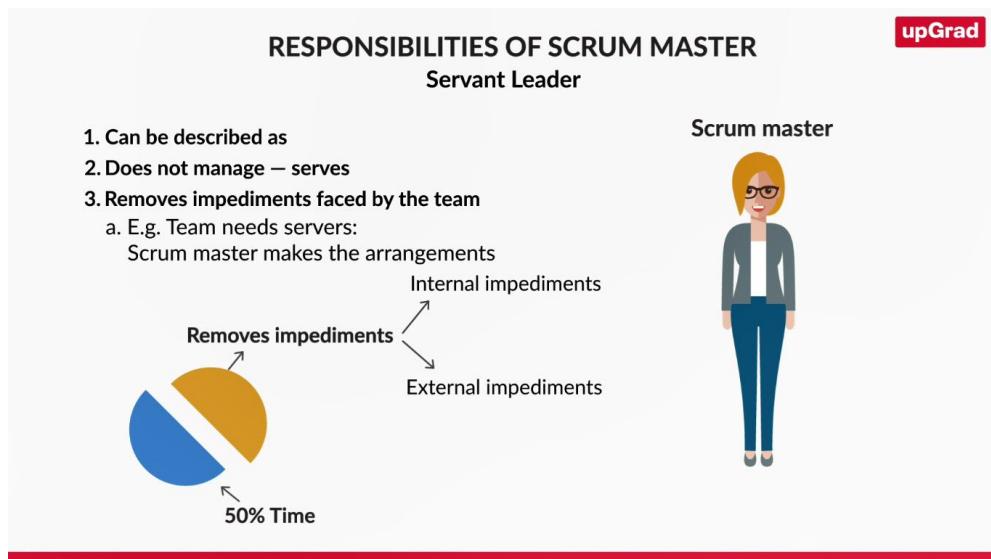


So, if it's a small product, the product owner can manage to work independently. But for a large team spread across multiple locations, the product owner takes inputs from the team members before arriving at the final decision.

Typically, a product owner divides his time 50-50 between team and the customers. In the next video, we'll dive deeper into the role of the scrum master.



After learning all about the product owner, it's time to move on to the scrum master. So, what does the role of a scrum master entail? What are the responsibilities?



The scrum master is one of the team members who has a special set of responsibilities. As per the definition of scrum, the scrum master is really the process coach or the enabler or the facilitator for the team. They are also known as the servant leaders.

These are the people who don't really have people reporting into them. They don't really manage the people on the team. They really serve the people on the team. So, their job is to find out how can they enable a smooth sailing for them? How can they remove the impediments?

For example, the team is struggling because they don't have access to two servers. You don't want two engineers to stop their productive time and go and find where they can get two servers in the organization. They will ask the scrum master for help.

And the scrum master will have a role and the responsibilities commensurate to that role where they can go back to the organization. Maybe if they have to go to the manager, if they go to the CFO, if they go to the procurement team or the materials or inventory team, but get those two servers.

How does it help? It helps because the scrum master is a designated role whose job at least 50% of the time is to remove impediments for the team. These could be internal impediments, which could be in terms of the process, method, tool, communication, and so on.

Or this could be external impediments. Servers are not available. We haven't been able to complete the hiring. We have challenge in really getting the customer on the phone call and so on and so forth. They can do it most effectively only when they are really a part of the team.



## MORE ABOUT SCRUM MASTER

What does Agile prescribe?

	YES	NO
Can there be one scrum master for multiple teams?		

So, if a scrum master serves to remove impediments, is it possible to share one scrum master across teams?



## MORE ABOUT SCRUM MASTER

What does Agile prescribe?

	YES	NO
Can there be one scrum master for multiple teams?		a. Must understand the team's work b. Must have mutual respect

I've seen some teams, they experiment with having an external scrum master, or there is a person who is kind of external to the team and is serving two or three teams at the same time.

Needless to say, such an experiment is not always successful because the scrum master is not having the right context about it. They don't have the right respect from the team members because they are not really sitting with the team members in the trenches.

So, I think it's very important for a scrum master to have a position of mutual respect. They must understand the work being done by the team members and the team members must really look up to them that they can help them with any of these things.



## MORE ABOUT SCRUM MASTER

What does Agile prescribe?

	YES	NO
Can there be one scrum master for multiple teams?		a. Must understand the team's work b. Must have mutual respect
Can any team member be the scrum master?		

As you just heard, the scrum master is there to facilitate a smooth working environment for the rest of the team. Taking this into account, can anybody be a scrum master?



## MORE ABOUT SCRUM MASTER

What does Agile prescribe?

	YES	NO
Can there be one scrum master for multiple teams?		a. Must understand the team's work b. Must have mutual respect
Can any team member be the scrum master?	a. Architect b. Database designer c. QA engineer d. Business analyst	a. Must be a leader of influence b. Must have gravitas

I like to answer it this way. Anybody can be a scrum master, is both right and a wrong answer. Let's look at both of them. The right answer is yes, anybody on the team can be a scrum master. It does not have to be limited to a development manager alone.

It could be a architect, it could be a database designer, it could be a QA engineer, it could be a business analyst on the team, if business analyst is embedded inside the team. Anybody can be a Scrum Master.

Let's look at the wrong side of the answer now. Can anybody become a Scrum Master? No, I don't believe anybody can become a Scrum Master. And let me back that up, what do I mean by that. Unlike the role of a manager in the past, which was a role of title, position and hierarchy, a scrum master is not really a leader of hierarchy, title or position.

A scrum master is really a leader of influence. A scrum master like I said, nobody reports to them. They are not positions of power. So, to that extent, not anybody can become a Scrum Master. They must be people who really have a lot of influence on the team.

I like to even call it that they must have a lot of gravitas inside the team, a lot of gravitas inside the organization. Because if they do not have that much of gravitas, they could be sending a mail to anybody in the organization and people will simply ignore that.

So, they need to have the right ability to influence the stakeholders and the team members without necessarily having the power to control or direct them. And that requires a very different thought process. And that requires a very different skillset.



After a low down on the theory, let's check out an example of a scrum master from BookMyShow.

SCRUM MASTER AT BOOKMYSHOW

- 01 Resolve all impediments of the team
- 02 Ensure that the team follows Agile and Scrum methodologies

The image shows a man with a beard and dark hair, wearing a dark polo shirt with white polka dots, speaking. To his right is a sidebar with the title "SCRUM MASTER AT BOOKMYSHOW" and two numbered items describing his responsibilities.

So, at BookMyShow, every scrum team has a scrum master. They typically do things a scrum master has to, has to make sure they do. And it's a very, very important role for a scrum master to play. So, the number one thing a scrum

master does is that they need to be able to solve all the impediments of the team, you know, the engineers or maybe the product owner as well.

Impediments such as, Hey, you know, I don't have a laptop, charger or I don't have a particular requirement that needs more, you know, clarity, right? So, any impediments that a scrum master needs to solve, it could be hardware, it could be software, it could be something that the stakeholders need to answer. So, the scrum master is responsible to solve impediments.

The second important thing that a scrum master is responsible for is to always make sure that the methodologies of, you know, scrum and agile are implemented by the team and are followed by the team, and to sort of, kind of reinforce it in a very coachable manner and keep coaching the team on the fundamentals of scrum and making sure they're always on track.



You saw in this video that a scrum master is not really a manager, but a facilitator for the team. The scrum master can be best described as a servant leader. Most of the role entails removing impediments from the team. A scrum master can be any kind of a developer, but the person needs to have influence without authority to get the job done. And that's the end of the session. You can go through a detailed summary in the next video.



## SCRUM ROLES: SUMMARY

- 01 Roles in the Scrum team
- 02 T-Shaped individuals
- 03 Traditional team vs Agile team
- 04 Responsibilities of a product owner
- 05 Product requirements document (PRD)

That was a really interesting session on the different roles in a scrum team. Let's quickly recap everything we've learned. We went deeper into learning about the scrum team which only has three roles, the product owner, the scrum master and the team, but the members are called developers.

In a scrum team, every member is a T-shaped individual, which basically means that each person has a deep expertise in one area, but also has competencies in multiple other areas. This allows you to create a more shared knowledge base with the rest of your team members.

You also learned that while a traditional team is very hierarchical, an agile team is flat and everyone's accountable for their own tasks. There is no team leader and responsibilities are democratized.

Next, you learned that the product owner is responsible for four major tasks:

1. He has to clearly communicate the product backlog items to the scrum team.
2. He has to order these items according to their priority, on a scale of 1 to 10. One being the highest.
3. He has to make sure that the product backlog is updated and remains visible and transparent to the team.
4. He has to ensure that the entire scrum team is on the same page and understands the product backlog and the items therein.



## PRODUCT REQUIREMENTS DOCUMENT (PRD)

01 Contains all product requirements

02 Enables team to understand

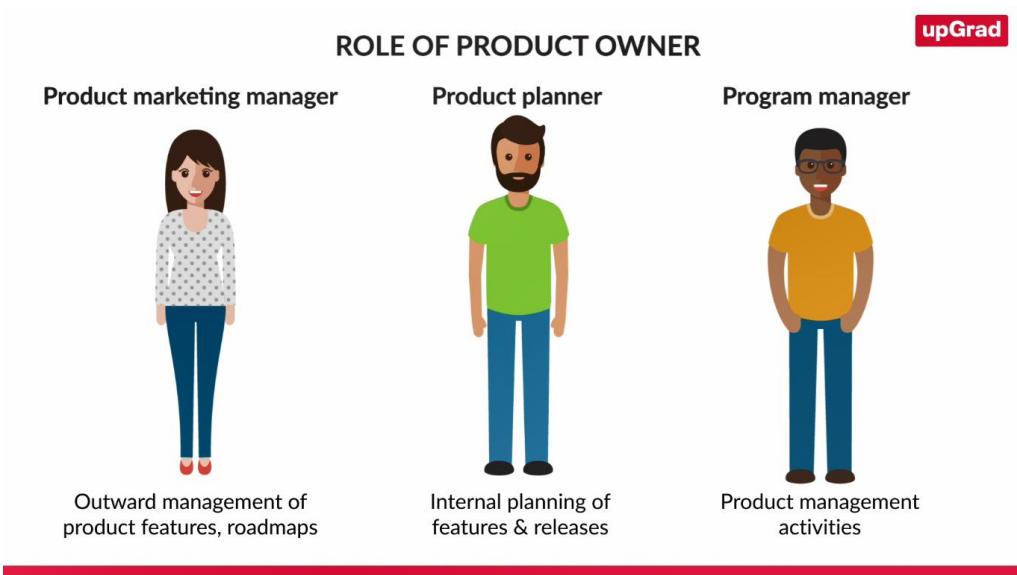
You also learned what a product requirement document or PRD is. This is a document containing all the requirements for a certain product, which explains how a particular feature should function. In scrum, the product requirements are usually communicated at a higher level, but a more detailed PRD is built as a support document in case the team has any queries.



## SCRUM ROLES: SUMMARY

06 Role of a product owner

After this, you learned if a product owner can work independently of the team and the answer goes both ways. If it's a small product with only four to five people working on it, then it might be possible, but if the team is spread across multiple location, the product owner needs to take help from the team members before arriving at the final decision.



In such a scenario, the role of the product owner maybe divided into a product marketing manager, product planner and program manager.

Typically, a product owner would divide his time 50-50 between the team and the product customers. Essentially, the role of the product owner is to be like the CEO of product, as every final decision rests with him.



As an example, we looked at the role of a product owner at BookMyShow. While the team is working on a sprint, the product owner prepares the backlog item for the next sprint planning meeting. He conducts backlog grooming sessions, where he explains the user stories for the upcoming sprints. The entire team then discusses the prioritization and the effort estimation of existing product backlog items.



SCRUM ROLES: SUMMARY

- 06 Role of a product owner
- 07 Product owner and stakeholders
- 08 Product owner and backlog grooming at BookMyShow
- 09 Responsibilities of the Scrum master

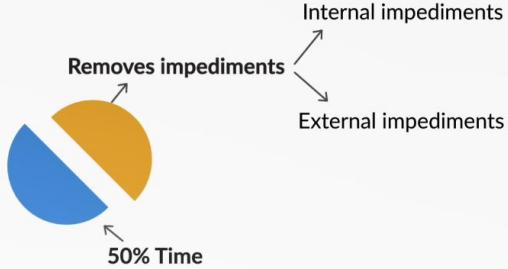
Next, we went into detail about the scrum master. A scrum master is not a manager, but the facilitator or enabler for the team.

## RESPONSIBILITIES OF SCRUM MASTER

Servant Leader

Scrum master

- 1. Can be described as
- 2. Does not manage – serves
- 3. Removes impediments faced by the team
  - a. E.g. Team needs servers:  
Scrum master makes the arrangements



This role can be best described as that of the servant leader. It entails dealing with any impediments the team might be facing, be it in terms of process, method, tool or communication. A scrum master can be any kind of developer, but the person needs to have influence without authority to get the job done.



## SCRUM MASTER AT BOOKMYSHOW

01 Resolve all impediments of the team

02 Ensure that the team follows Agile and Scrum methodologies

Next, you saw an example of the scrum master at BookMyShow. He has two major responsibilities, one to solve all the impediments of the team. And second to ensure that scrum and agile methodologies are followed by the team.



## CODE REVIEW

01 Reviewing the code to ensure high quality

02 High quality code requirements

03 Advantages of high quality code

- a. Faster application
- b. Less memory usage
- c. Better user experience

Related to the theory, we came across two terms, code review and performance engineering. Code review simply means to review the code written so that the quality is maintained. This will ensure a fast application, efficient use of memory and a great user experience.



## SCRUM ROLES: SUMMARY

- 09 Responsibilities of the Scrum master
- 10 Scrum master at BookMyShow
- 11 Code review
- 12 Performance engineering

Performance engineering means monitoring how quick and efficient your application should be. This would include loading time, response time, and memory.

And that's it for this session. In the next one, we'll take a closer look at scrum artifacts. See you on the other side.

No part of this publication may be reproduced, transmitted, or stored in a retrieval system, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publisher.