## Transcription

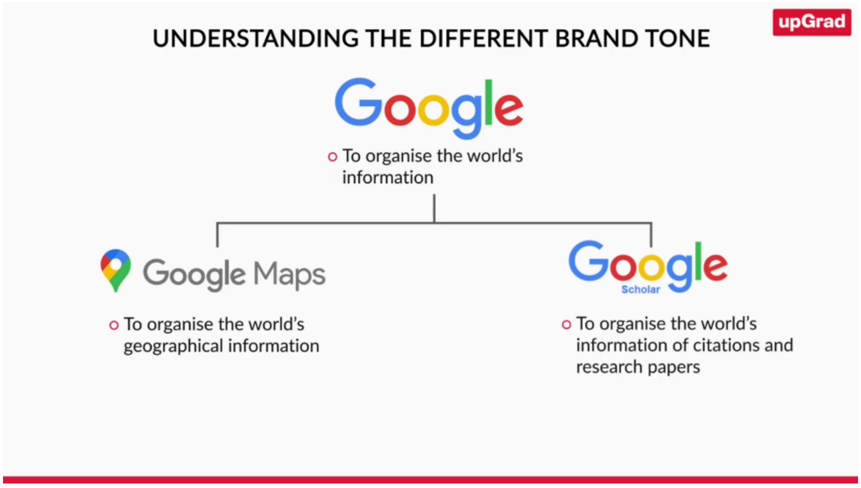
# Scrum Artifacts


Hey there. By now, you've learned about the various roles involved in a scrum team. As an aspiring PM, you must have noticed that your role would be close to that of a product owner.

So, the product owner is responsible for creating backlog item. But before getting into backlog items, let's first figure out some other scrum artifacts. The first one being the vision. Now just like me, you too must be thinking, what is the vision and why do you need it? Let's find out.


The first one that we talk about is what we call as the vision. What is a vision? I consider vision as the true north. When you are a traveler traveling in the desert and if you lose the way, you always can look up to the true North and it will always point to the right direction.

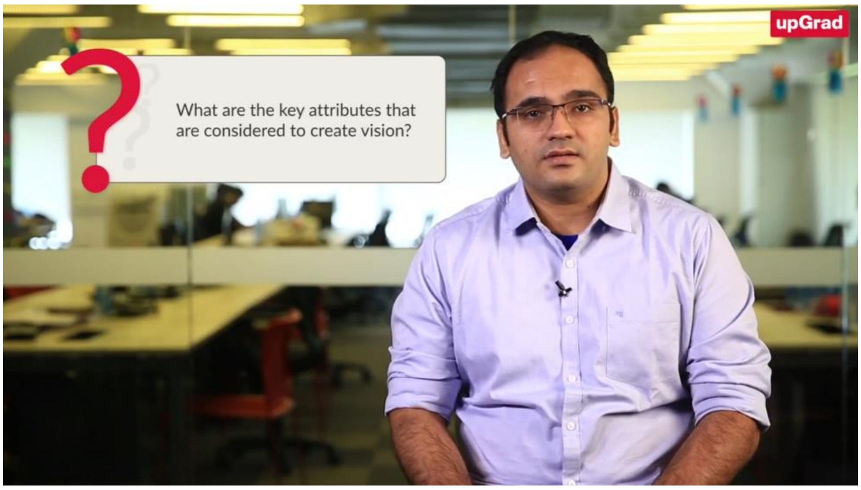
However, a true North is not something that is so low hanging that you can just go and pluck it from out of the sky. It is unachievable. You can never reach the true North, but yet the true North really continues to guide your journey and continues to help you, give you the feedback so that you can align yourself to where you want to go.



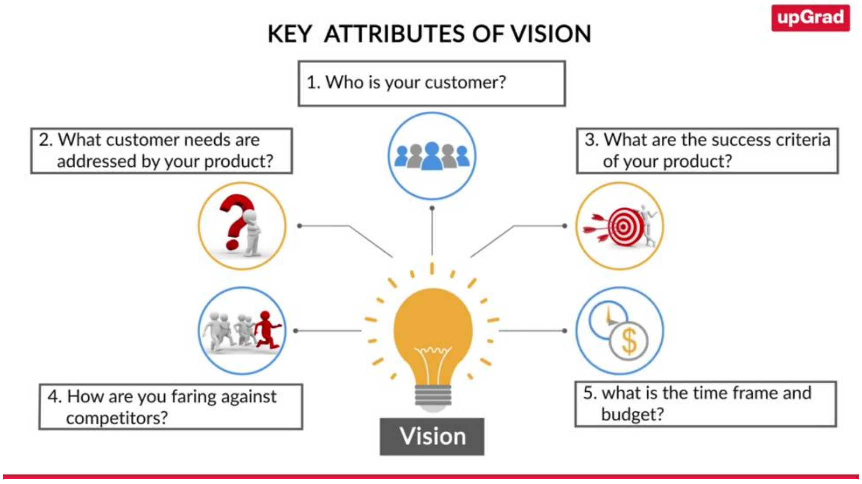

In the same way, in the context of a product, the vision continues to guide. For example, Google's vision is to organize world's information. Now it could be that when Google is really working on the Google maps, it might say that our vision is to really organize the world's geographical information.

If it is coming to the Google scholar, they might say our vision is to really organize the world's information or the citations or the research papers.


It should also be, as Jonathan Swift said, vision is the art of seeing things invisible. It should not be visible today because if it is visible today, we already know that it is something that is achievable. So, it must be something that is aspirational, provocative, expose the team into action. And yet it acts like a true North. It keeps the team honest; it keeps the team aligned to what they would like to do.




Okay. So, vision acts like true North and helps the team focus and align their goals to achieve it. Let's find out another thing. What are the factors to consider while creating a vision?




There are five key attributes that we look at. The first thing that we want to know is who is my customer? Who's going to use my product or services. Who's going to buy my product or services, who is, as we call it, who's the target segment or who's the target customer.

Now, obviously there could be a large segment of it, but we do want to understand and start by understanding who is really going to be my key customer, or sometimes we also call them as the primary personas. Who are the initial set of people for whom I'm really targeting my product and services?

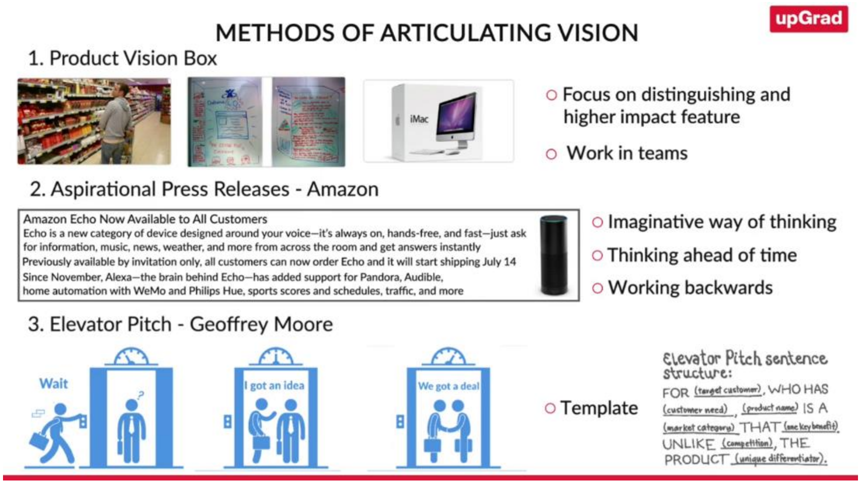
The second thing that we want to know, having known the customers, what is the customer need that we are really solving with our products and services? The third one is what are the absolute criteria for success of the product? What are the key criteria that are critical to satisfy as far as customer needs are concerned?

The fourth point is how does your intended product or services stack up as compared to the other existing products and services? In other words, how are you fairing against the competition? Because if you offer exactly the same services, people are probably not likely to accept you because they already are getting those needs addressed by the existing players.

And finally, it's important to understand what kind of a timeframe and budget we are looking at. Or as a product manager, you probably cannot justify a budget runway of three years or four years in today's world. On the other hand, the pace at which technology and the customer changes, you also cannot design a product for three years down the line. You probably need something that you can reach out to the customers much sooner rather than later.

So, it's important to understand what kind of a timeline and budget we are talking when it comes to vision. So, these are five key parameters that we look at when we look at articulating the vision for a product that we intend to design in the context of a business.


So, detailed user or market research helps you figure out the key attributes to articulate the vision. But how do you go about articulating the vision?




Now the vision itself could be articulated in multiple shapes and forms. Some of the traditional ways are just writing it on a piece of paper, and maybe you write it in one A4 sheet or two A4 sheets. And that's a traditional way.

With an agile world, we believe it should be a little more fun because nobody wants to read long documents. So here are a few ways in which you can experiment. One of the ways is what we call as the product vision box. So, a product vision box is actually a great way for a team to collaborate and understand that if I were to create a product box, which I need to then put it on the isles in my supermarket, what should I write on that?

Can I make it very crowded? And I put the whole PRD or a product backlog on that, will that make users interested in that? Or should I just make it as minimalistic and just some pictures like Apple does? Will that make it interesting enough? Or should I really focus on some most important features that we know are really going to have a big impact on people's decision to buy the product?

I would encourage you to actually work together as a team and build it. And again, if you have a team of 10 people, you have one to actually split the team into three or four teams and let there be multiple product vision box come up, because you may never get one single idea, but you might get some interesting ideas from each of the boxes, which you can then call it together and really bring them up there.

So, that's one way of articulating product vision. There is another way of doing it, which Amazon does. Apparently what Amazon does is even before the product has been created, it actually creates a press release for the product.

Which means, let us say, we are sitting here in the month of April and the product is going to go live in the month of October. What they will do is as a team, they will come up and articulate a press release. When you create such a compelling story about a product, you are actually creating a very imaginative way to make something which will be a value to the customers. You are thinking ahead in time, and you are trying to sell a story to the people.

By doing it, you are actually creating a state of a vision and then you work backwards. So, Amazon also calls it as working backwards. So, this is yet another way of doing it. It's a little textual way, as opposed to the first one, which was kind of a box in experiential manner.

Finally, there is another time-tested way of communicating the product vision, and that is something known as the elevator pitch. The elevator pitch was created by Geoffrey Moore.

And the basic idea is if you are on the 12th floor and you see the CEO or you see the next investor in the lift with you, and you have to come down from 12th floor to the ground floor, you probably have about 20 or 30 seconds. How much can you really talk about the product to make him or her interested enough in your product to kind of sell it to them? They might be your investors. They might be your sponsors. How do you really make it compelling enough in just a matter of 30 seconds? So, that is the amount of time the elevator takes from the 12th floor to the ground floor.

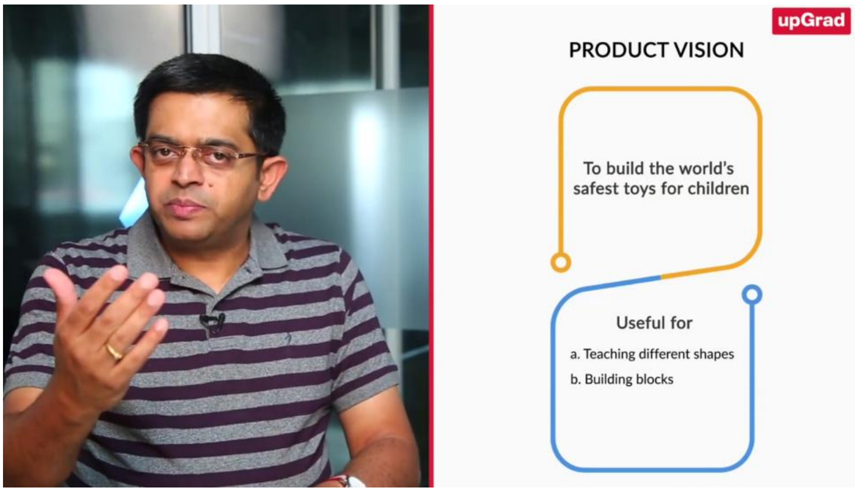
There are some templates available that Geoffrey Moore has created about the elevator pitch. And I would encourage you to have a look at them. Any of these three exercises could be a great way to get the team started into thinking, what are the most distinguishing, compelling and differentiating features that we would like to work on.

After all, you are going to invest next three to five years in realizing the vision. You do want to do something differently. And these are some of the ways in which you can call them out.


In this video, you learned that vision helps your team focus and align you to the goals that you want to achieve. You also saw the various attributes and methods to articulate product vision. In the next video, we'll elaborate on yet another scrum artifact for our roadmap.

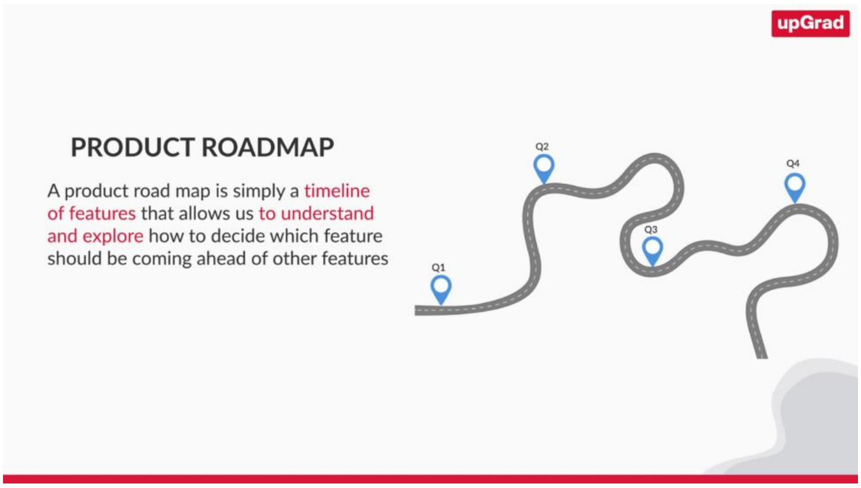

You saw how the vision acts like true north for the team to align product strategies for achieving ultimate goal, just like ancient travellers would use the North Star to navigate and reach the destination. There are several methods to define the vision of a newly emerging product, though in most companies, it's the CXO or other higher management who set the vision for a product.

Once you have a product vision, the next step is to develop the roadmap. Let's find out what is the product roadmap and how does one decide the roadmap?



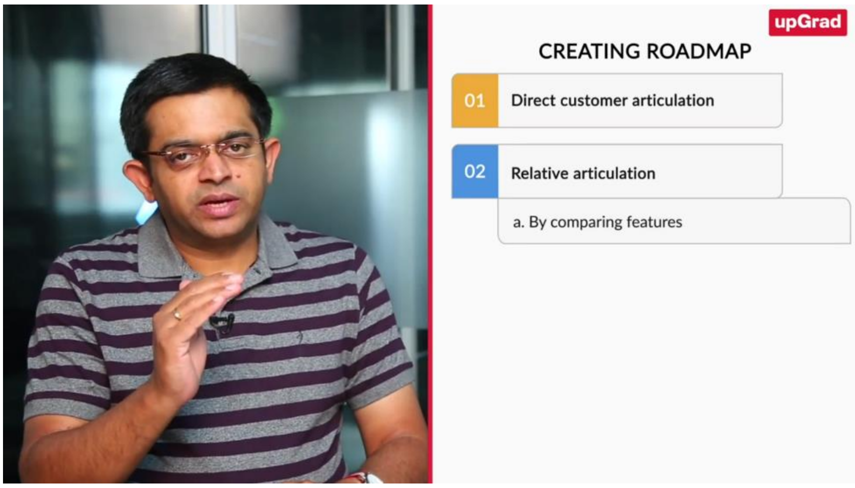

Let's say your vision is you want to build the world's safest toy for children. And you believe that toy is going to be the most interesting learning toy for the children. You want them to be able to teach different shapes, different ways of building something else, and so on and so forth.

Now, obviously your vision is going to be stretched out in future. It may not be that you can deliver that vision in a matter of one quarter or maybe one year or so. So, you will need to understand how do I really stagger my features and how do I really create versions out of it eventually meet the vision.




So, let us say, you probably have a timeline of next four quarters, the next six quarters, and you will need to find a way to put the features on the timeline and say, Hey, this goes on Q1, this goes on Q2 and so forth. In product management parlance, we call such a thing as a product roadmap.

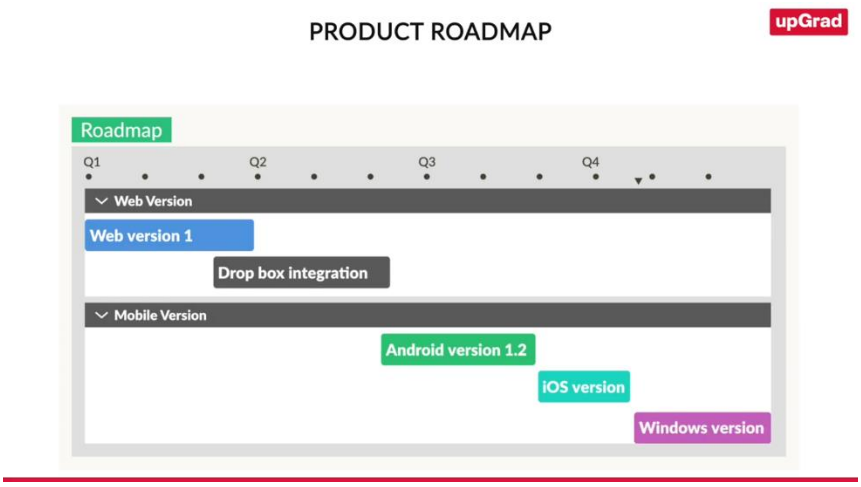
A product roadmap is simply a timeline of features, which allows us to understand and explore how we are deciding which features should be coming ahead of other features.



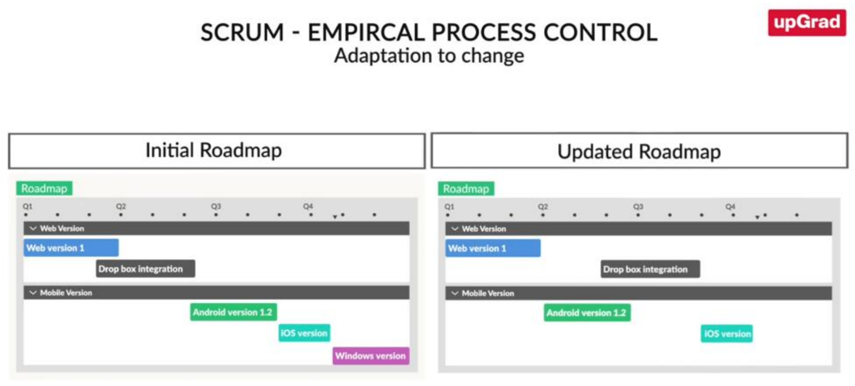

How do we decide that? Well, there could be N number of ways in which we decide it. It could be based on customer's priority. Let us say you are in a market where there's a clear value attributable to every feature.

For example, the customers might say, Hey, if I had the Dropbox integration, I would be willing to buy that $25 a year or $50 a year subscription. Now that's a very clear articulation of the value that this feature is worth this much value to the customers. Often, it is not the case that we know that, but it's easy for us to have a related articulation.

Compared to this feature, do we think this is more important or less important? Irrespective of that, I'm sure in your field, you will find some way of making a relative or an absolute articulation of the value of these features. And based on that, you would be in a position to create some kind of a roadmap.




We might have a suggested timeline or a suggested roadmap that, Hey, Q1, we want to do this. Q2, we want to do this. Then we will have an Android version. We will have an iOS version. We will have a windows version and so on. But when we actually start some of the earlier versions released in the market, we might find that there is a different customer feedback, which is against our initial hypothesis.



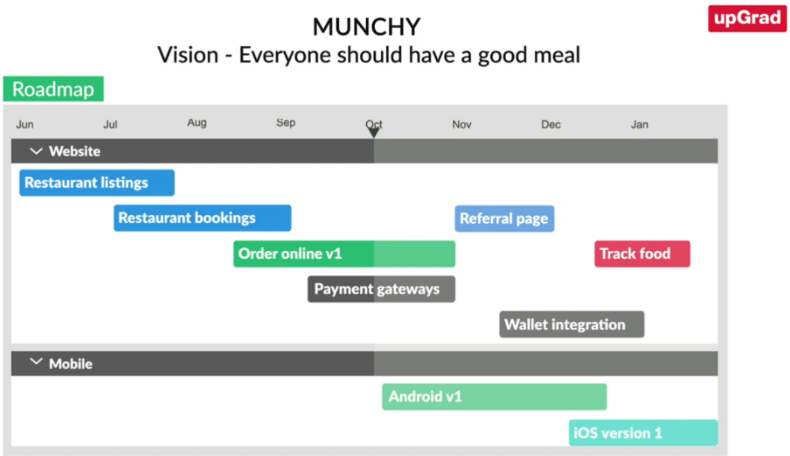

So, the empirical process control theory will tell us that, Hey, you don't want to pursue down something where your assumptions have not been positively validated. You do want to factor in those changes and you want to adapt your plans in accordance with what you're hearing from the market.

So, your roadmap will keep changing. So, net-net, if we say in the context of scrum, the idea is while a roadmap will help you create some kind of a map, you do want to make sure, as I like to put it, there is a very old saying that we have, I read it in the Swiss army manual. It says, when you find a difference between the map and the terrain, trust the terrain.

So, you do make sure that you take these inputs into account. But the key thing is the roadmap allows you to have these conversations with all the stakeholders.


So, the product roadmap primarily helps you realize your product goals. But what does the product roadmap look like for a particular product?


Let us consider a company in a food tech space, which has a vision of everyone should have a good meal. Let's name it, Munchy. The roadmap will look something like this. You have various releases or features planned against the calendar.




From the roadmap, we can see that they have completed working on restaurant listings and bookings. You are planning to launch order online feature. And currently that is in progress. They also have planned to stagger the features like wallet integration, tracking the order, etc in the coming quarter, along with the releases of version one of Android and iOS apps.

The timelines mentioned in the roadmap may not be very accurate. This is just a tentative roadmap so that the company can communicate what it is up to, to its various stakeholders.


In this video, you understood that the product road map is staggered features along with the timeline. And it is decided either by absolute or relative articulation by customers. You also saw what a product roadmap would look like for a food tech company like Munchy. In the next video, we'll take a look at product backlog.


Creating a product backlog is one of the primary responsibilities of a product owner. Before getting into how to go about creating one, let's first find out what is the product backlog? Why does it matter? And what are the characteristics of product backlog items?

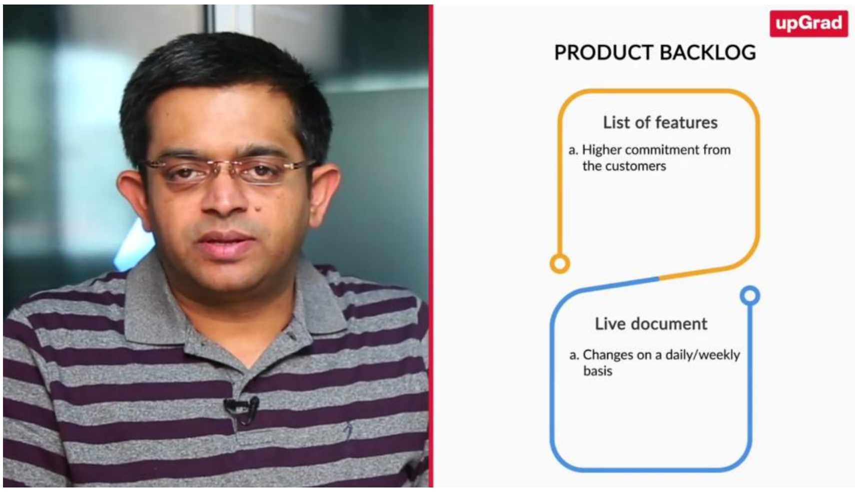
PRODUCT ROADMAP VS PRODUCT BACKLOG

| Product Roadmap | Product Backlog |
|---|---|
| No solid commitment | Higher commitment |
| Less detailed and less sanitized | More detailed and more sanitized |
| Feedback incorporated at higher level | Feedback incorporated at more intrinsic level |

A road map typically might be a little long drawn with no real firm commitments. It might have a suggested or an anticipated release timeline, but it may not really be backed by a solid commitment about it. And that's where we start getting into more of the scrum artifacts. The first one of them being the product backlog.

The product backlog is something which has a very higher level of signal to noise ratio. A roadmap could have a very low signal to noise ratio, which means we have put all the laundry list of features there. And we may not even have sanitized these requirements. We don't know whether the customers really would like to have it in Q4 or in Q3, or the customers would ever like to have that kind of a thing.

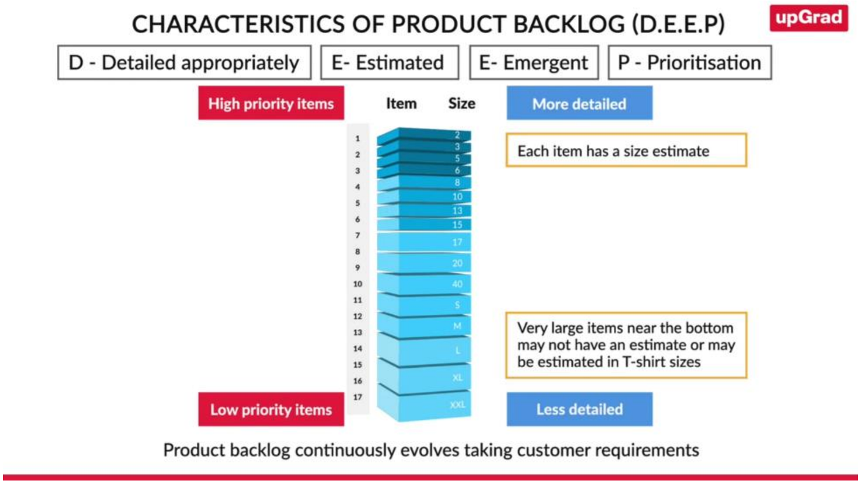
But as we keep getting more and more feedback from the customers, we keep pruning the list to a much shorter list, which has a much signal to noise ratio. What I mean by that is that there is more clear feedback from the customers that they would like to have these features.




So, now we have a product backlog. It has a bunch of features here, which have, as I said, a higher signal to noise ratio. That means they have a higher level of commitment from the customers.

One of the key things about a product backlog is it is not a static document. It's a live document. In fact, it keeps changing. If not on a daily basis, I would say at least on a weekly basis. As long as the product is alive, we want the product backlog to keep changing and it should reflect the current realities of the product.

We also have another way of describing the product backlog. We use the acronym DEEP to characterize it. And each of the words of the acronym DEEP stands for one of the characteristics.




So, let's start with them.

- D is for detailed appropriately. When we talk about detailed appropriately, what it means is that the items that have the highest priority, we have had more opportunity to work on them. And we have put them in more details as compared to the item that probably we are going to do after two quarters. We have not put enough effort. They have been analysed. They are much smoother. We really understand what are some other elements of those. So, these are detailed appropriately in terms of the priority order.

- The second one is estimated. What it means is that on the day one of my first sprint of the release, I start looking at the product backlog. I may not have had opportunity to estimate each and every product backlog item. Unlike a traditional project, where I would first go about an estimate every single thing, what I will start doing it is I will start doing the estimation of the highest priority items. And I will keep going back to the product backlog, maybe on a weekly basis till the time I have estimated entire thing.

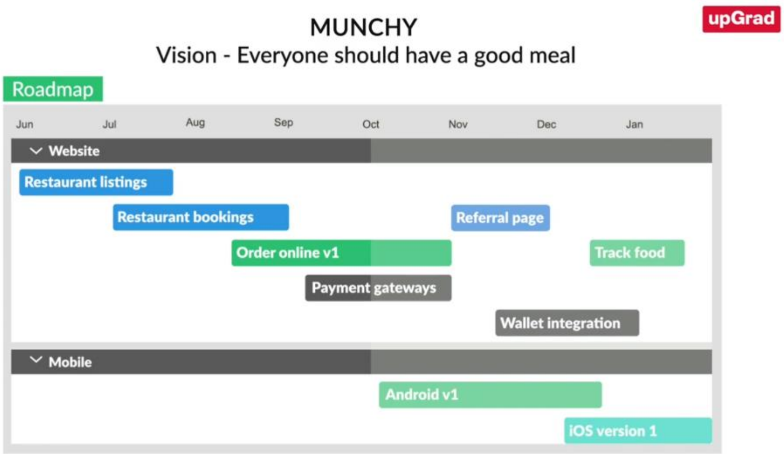
  But at some point in time, I will have the entire backlog estimated when it is known in all the details there. It may not happen on the day one, but it will happen certainly at some point in time during my sprints.

- The third important part is what we call is emergent. What it means is we don't believe that it is possible for all the requirements to be known upfront. We believe that the more we interact with the customers, the more we develop the software and give it to them, the more we are likely to engage with them and discover some of the new characteristics.

  The agile methods are really built in such a manner that we continue to evolve around the emerging requirements. We evolve our architectures; we evolve our roadmaps. We evolve our plans around the emergence of these requirements and the emergence of the architecture.

- And finally, we talk about prioritization. In scrum, what we believe is that these are my 50 high priority requirements, I must be able to articulate them as priority number one, two, three, four, going up to 48, 49, 50. What it helps me is a very clear pecking order. So, now if I can only deliver 30 of them, I know which are the most important 30 features out of that, as opposed to any 30 out of those 50 features. So, that's about the product backlog.


So, product backlog items should be detailed, estimated and properly prioritized. Now let's see what a product backlog looks like.




We've seen how product roadmaps would look like for a food tech company like Munchy. Let's say, so far, we have features like restaurant listings and booking of tables in restaurants available on the website. And the team is currently working on features like ordering food online and the payment gateway integration.

PRODUCT BACKLOG OF MUNCHY

| As a | I would like to | so that I can | Prioritization | Estimation (Story points) |
|---|---|---|---|---|
| user | like to view the menu | decide upon the items to order | 1 | 5 |
| user | select items | choose the items of my wish | 2 | 2 |
| user | change the quantity of items | easily increase or decrease the quantity | 3 | 1 |
| user | view cart | see all the items at one go | 4 | 3 |
| user | view total bill amount | check the bill before placing the order | 5 | 3 |
| user | select address and phone number | give delivery details | 6 | 3 |
| restaurant owner | see the order immediately | deliver the food quickly | 7 | 8 |
| user | view average delivery time | plan my meal accordingly | 8 | 5 |
| stakeholder | have sms partners integration | send notifications to the users | 9 | 2 |
| restaurant owner | have email services integration | send notifications to the users | 10 | 2 |
| repeat user | save my favorite dishes | so that I can order quickly | 11 | 2 |
| marketing team member | view the transaction funnel | find the reason for drop outs | 12 | 8 |
| marketing team member | view click tracking | figure out the customer behaviour | 13 | 5 |
| operation team member | to see the work flow | manage the operations | 14 | 3 |
| operation team member | able to process refunds | refund the amount for cancelled order | 15 | 5 |
| user | view delivery fee | choose the restaurant accordingly | 16 | 2 |
| credit card user | pay using credit card | do cashless checkout | 17 | 2 |
| user | get notified if restaurant cancels the order | order from another restaurant | 18 | 2 |
| user | view ratings of restaurant | easily choose the restaurant | 19 | 2 |
| restaurant owner | view payment status | know whether the payment is done or pending | 20 | 2 |

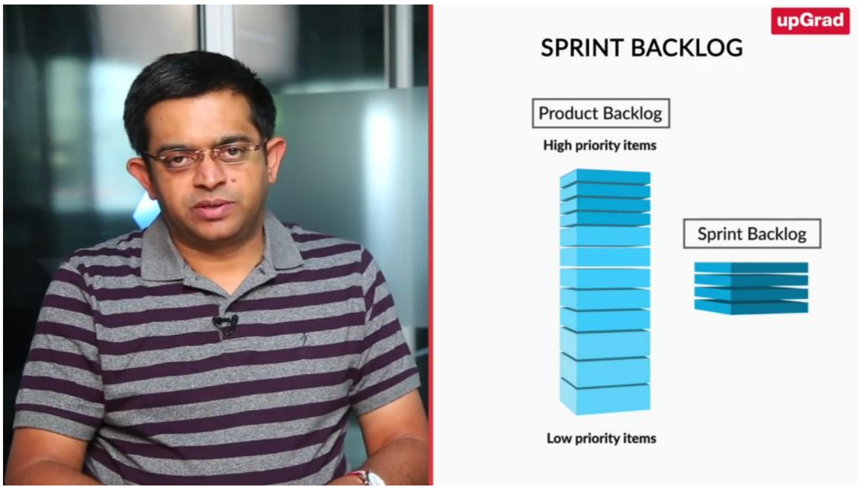
So, a product owner maintains a list of backlog items, which will go into upcoming sprints. Each feature will have a lot of user stories like as a repeat user, I would like to save my favourite dishes so that I can order food quickly or as a credit card user, I would like to pay my bill using my credit cards so that I can do cashless checkout.

As you can see, each story has a priority given and a rough estimate of effort needed. We will see how the prioritization and estimation is done in detail in the coming module. You can also observe the backlog contains some user stories for Android version planned in the future, but that is not very well curated yet.

Hence, the estimation is given on a different, like small, medium, large, and extra-large. As the priority of these items increases, more details will be added and they will be divided into smaller stories for proper estimation. This is how our product backlog looks like.


In this video, you learn what a product backlog is and why it's important to be created. You also learned about the characteristics of backlog items. In the next video, we'll cover everything about sprint backlog.

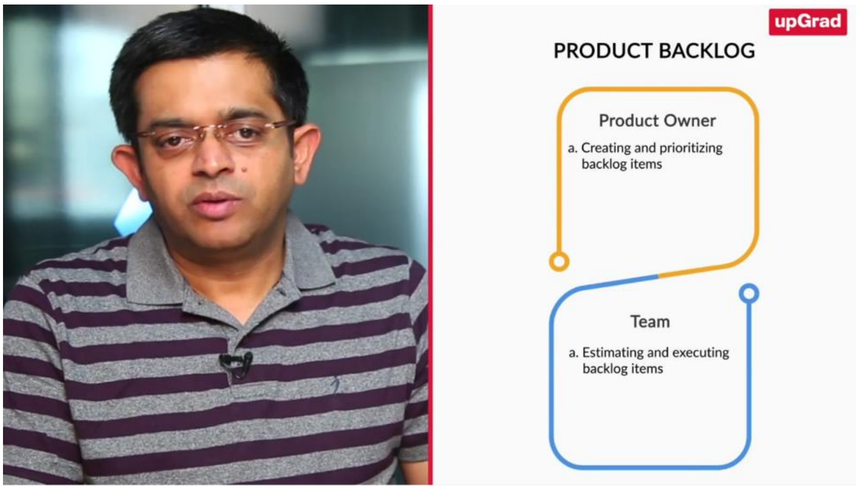

You have already seen that in a sprint planning meet, some items would be moved from product backlog to sprint backlog. Let's learn more about the latter. What is a sprint backlog? How does it create it?




The sprint backlog essentially is the highest priority requirements out of the product backlog that the product owner would like the team to deliver in the next upcoming sprint. And this is kind of a contract between the product owner and the team.

In the sprint planning meeting, the team and the product owner have a conversation. The team members can ask the questions. The product owner can clarify, explains typically the rationale behind it, the benefits behind it and so on and so forth.

Now, based on the team's assessment on how much they can deliver in the coming sprint, they might say, Hey, you asked for so much of sprint backlog. Unfortunately, we can only deliver so much of it.



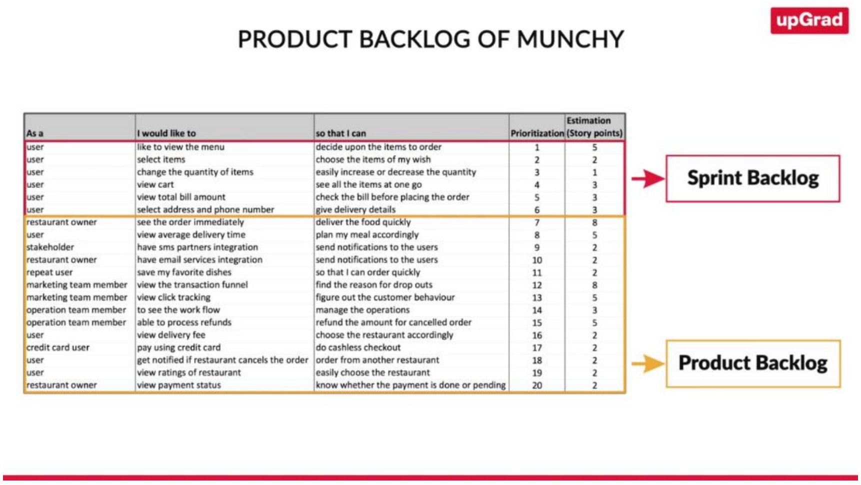

Scrum takes a position. The product owner should be trusting the team members judgement. The same way the team is trusting the judgment of the product owner, why these are the most important requirements.

So, I think it's a mutual respect that is really working at play. And we don’t want it to be a creative conflict. Sometimes the product owner wins, sometimes the team wins, but every time the best ideas win.


And at the end of the sprint planning, we have identified the most important elements that will go into the sprint backlog is the current sprint.

Scrum takes the position that we do not want to change during the time of the sprint, any of the stories on the backlog there. And the reason is simple, if we keep changing it all the time, the team has no realistic ability to estimate their work, size their work, and predict what they will be able to deliver. So, these are the small boundaries that we require everybody in the system to aggregate.


So, in a sprint planning meet, the sprint backlog is created. Let's see what it looks like for any particular sprint.



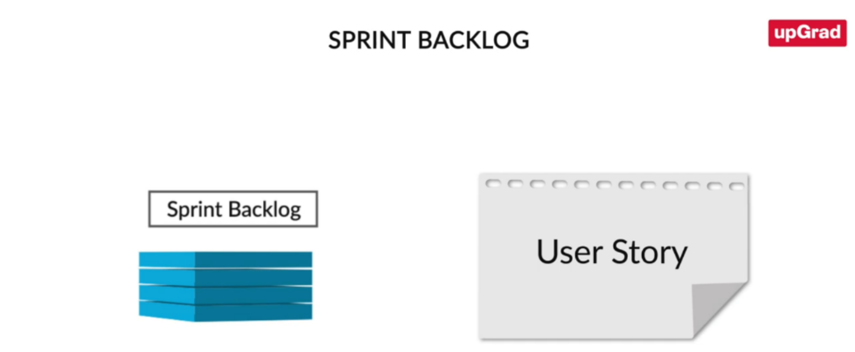

| As a | I would like to | so that I can | Prioritization | Estimation (Story points) |
|---|---|---|---|---|
| user | like to view the menu | decide upon the items to order | 1 | 5 |
| user | select items | choose the items of my wish | 2 | 2 |
| user | change the quantity of items | easily increase or decrease the quantity | 3 | 1 |
| user | view cart | see all the items at one go | 4 | 3 |
| user | view total bill amount | check the bill before placing the order | 5 | 3 |
| user | select address and phone number | give delivery details | 6 | 3 |
| restaurant owner | see the order immediately | deliver the food quickly | 7 | 8 |
| user | view average delivery time | plan my meal accordingly | 8 | 5 |
| stakeholder | have sms partners integration | send notifications to the users | 9 | 2 |
| restaurant owner | have email services integration | send notifications to the users | 10 | 2 |
| repeat user | save my favorite dishes | so that I can order quickly | 11 | 2 |
| marketing team member | view the transaction funnel | find the reason for drop outs | 12 | 8 |
| marketing team member | view click tracking | figure out the customer behaviour | 13 | 5 |
| operation team member | to see the work flow | manage the operations | 14 | 3 |
| operation team member | able to process refunds | refund the amount for cancelled order | 15 | 5 |
| user | view delivery fee | choose the restaurant accordingly | 16 | 2 |
| credit card user | pay using credit card | do cashless checkout | 17 | 2 |
| user | get notified if restaurant cancels the order | order from another restaurant | 18 | 2 |
| user | view ratings of restaurant | easily choose the restaurant | 19 | 2 |
| restaurant owner | view payment status | know whether the payment is done or pending | 20 | 2 |

We have been seeing the product roadmap and backlog for Munchie. Now that the backlog items, which will be planned for the coming sprint will go into the sprint backlog, let us say that in the sprint planning meet the team decided to take up the top six items, and hence the sprint backlog will look something like this, clearly prioritize and estimate it.

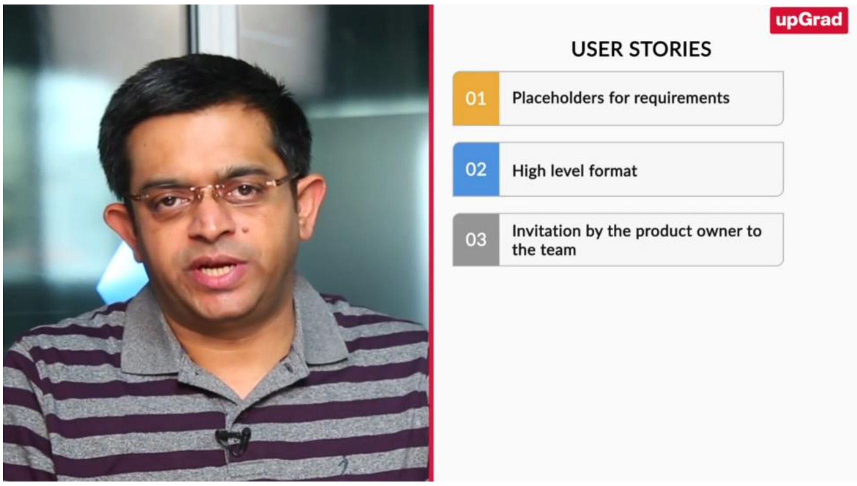
Now, as the team works on these items, the product owner will work on prioritizing and estimating items for the next sprint, taking inputs from the team and other stakeholders.

In this video, you learned what a sprint backlog is, how it is created and what it looks like for any particular sprint. In the next video, you will learn more about user stories and epics.

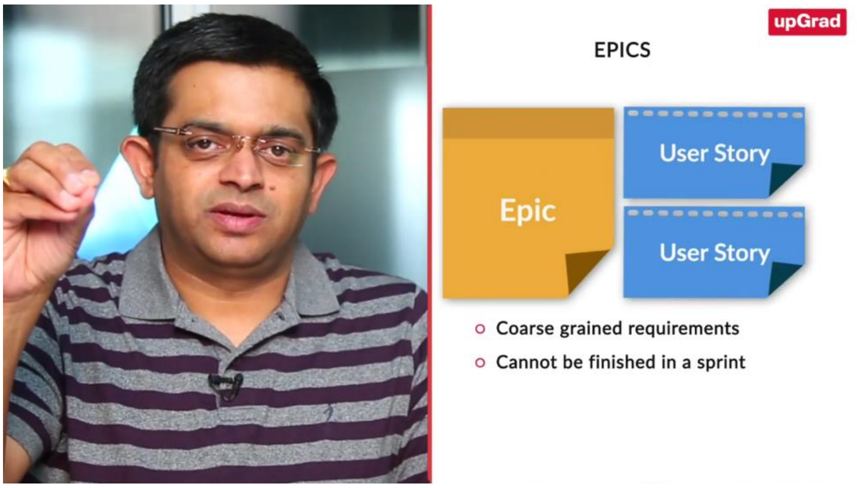

You've seen that user requirements are expressed as user stories in the backlog. Now let's delve deeper into it. What exactly is a user story? What is an Epic and what is a theme?



Now what does the sprint backlog have? The sprint backlog has requirements that we, the product owner would like to have in the software. And these requirements are expressed in a very specific manner in scrum. And we call them as a user stories or simply stories. So, let's understand what are these stories?

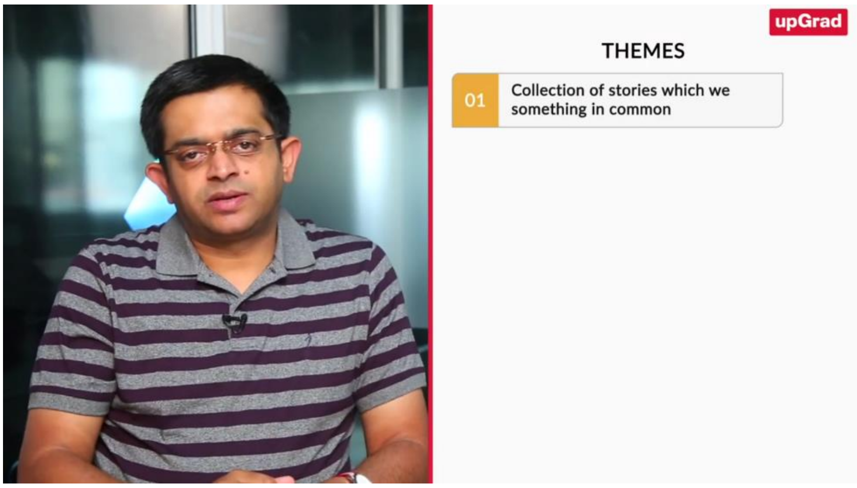



So, what are user stories? User stories are those very short place holders that specify or direct towards the user requirements. User stories are only at a high-level format, and the details will emerge out of the discussions that the team members will do as a part of that. We also call it as, it is an invitation by the product owner, to the team to come back to them and have a conversation on a future date.

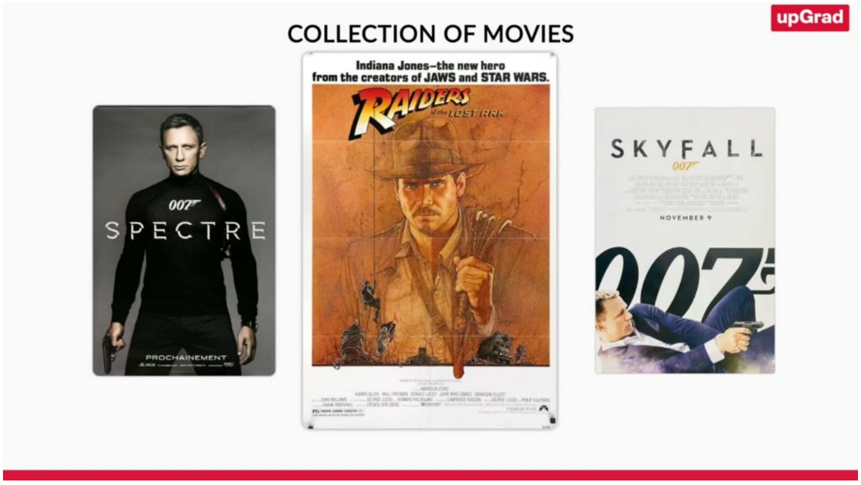



Some of the big stories we call them as Epic. And these are just simply coarse grain. Is there a mathematical definition? There is no such definition. A user story that can be done within a sprint or few of the stories that can be done inside the sprint, we will call them as a small. And a story that cannot be done inside a sprint by definition can be called as an Epic, but anything that is big story is an Epic for us.

So, when we start looking at a product backlog, we might see some of the Epic stories and these stories have to be broken down into smaller stories there, which can then be consumed in the sprint backlog.



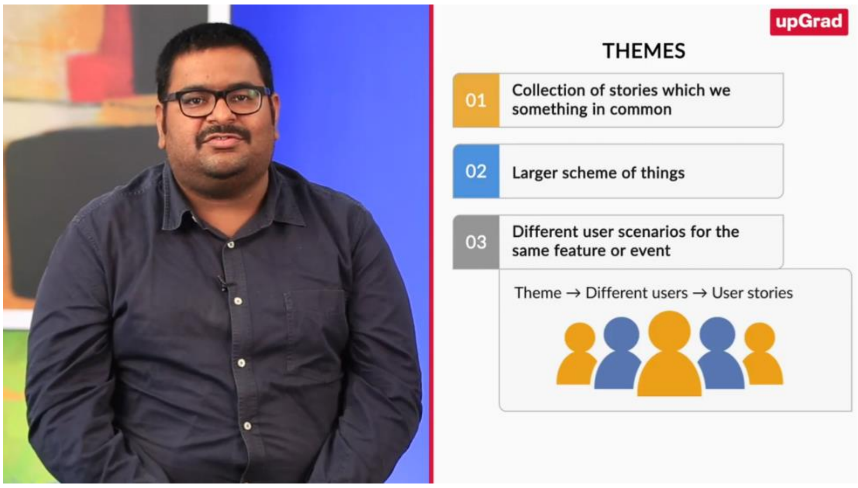

There is another context that we call in there, which is a collection of the stories. So, let us say, I have 20 stories here. Now all the 20 stories may not be related to one single, they might not have anything common.




For example, let us say, I love movies and I have lot of movies. And then when I actually started looking at my movie collection, I realize I have a lot of James Bond movies, and I have a lot of Western movies and I have a lot of action movies and I have a lot of Indiana Jones movies. I might simply call them as different themes, and that is just making me club them together into an Indiana Jones collection or a James Bond collection or a Western correction or an action movie collection. It just makes me easier for me to track all the individual stories inside a given team.


So, a theme sounds like a collection of similar things. Let's find out more by seeing what could be a theme in the product sense.

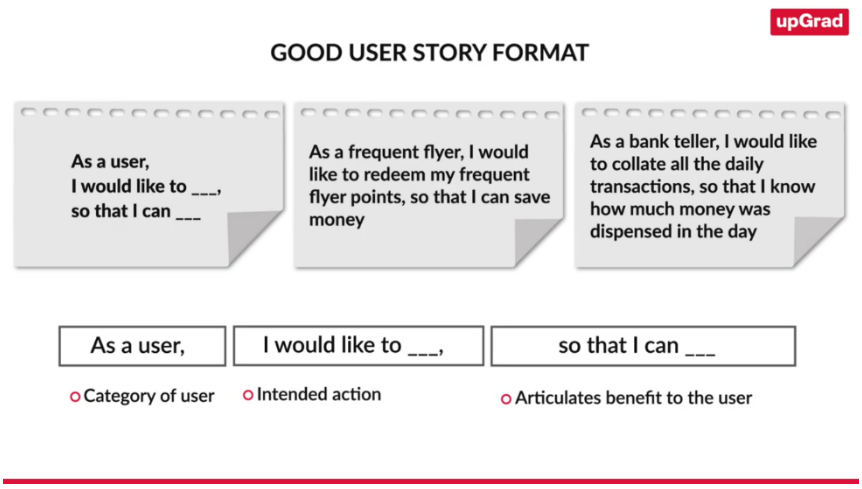

So, what's the team, right? So, let's take an example, Facebook calendar is a theme. So, when we say, okay, Facebook calendar, if we want to sync Facebook calendar with Outlook, who would be the users and what would be the use cases? So, as a user, I want to sync my Facebook birthday calendar. That can become a user story.



So, it all starts with the theme first. So, what's the larger overall scheme of things. What do you want to achieve? And then you define who are the actors or who are the end points, who are the participants? And then every participant may have different user scenarios for that particular thing.

And then you define those user stories gradually. And that's the process which we follow. And most of the time, a lot of market research, competitive research, and also internal debates help us to create the right kind of user stories.


That's great. So, whenever you write user stories, there must be a proper format. So, what is a good way of writing a user story? And what are the characteristics of a user story?



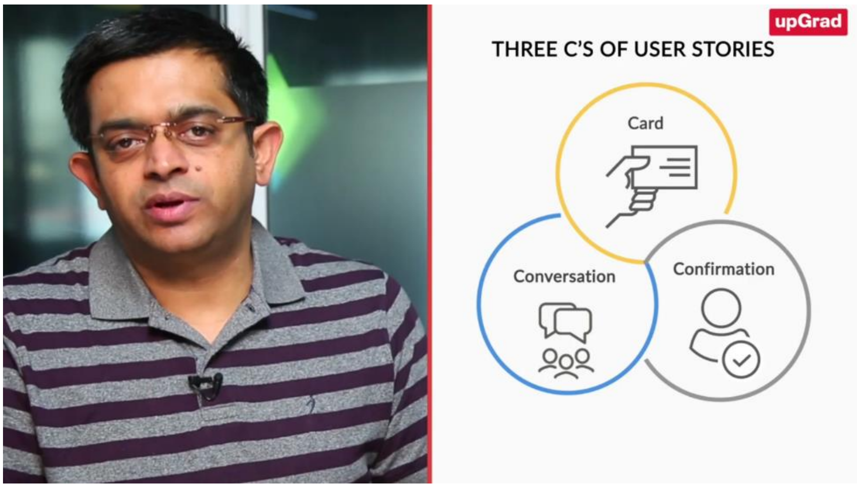

One of the suggested formats is something like this. We say, as a user, I would like to do some action so that I can get some benefits out of it. That's a very simple format. Let's take an example. As a frequent flyer of some of my favourite airlines, I would like to be able to redeem my frequent flyer points so that I can save money. It could be as simple as that. Or it could be that as a bank teller, I would like to be able to collate all the daily transaction so that I know how much money was dispensed in the day.

So, there are three components of that. One is we are trying to understand who the user and that tries to help us understand which category of the user would like to have this requirement. The second part is I want to do something and that action is the requirement.

Now we are all used to the requirement being expressed only in that second line, that I should be able to update the ledger, or I should be able to redeem my frequent flyer points, or I should be able to order online food on my food app and so on.

So, this is something which is obviously the intended action, which will translate into the requirement. The third part of the user story is so that I can articulate the benefit. Now, again, we are not used to articulating the benefit for the requirements.

Within the agile world, we believe that by articulating the value, we make the purpose behind it very explicit and very clear to the development team. Because end of the day, we don't simply deliver the functionality.




There are a few concepts associated with the user stories:

1. One of them is known as the three Cs. Now these three CS are, the first C is known as a card. And actually, it is just a four by six index card, a small card that we use to basically express the user story. We don't take a full page. We don't take a full document. We take a small index card.

   Why do we do that? We do it on purpose because we don't give enough real estate on the paper for people to write a complete story or a paragraph. We give only so much of real estate that people are forced to choose out the most important part of that requirement and write the user story and take away everything out of it.

2. The second C is known as the conversation. And as I explained earlier, the user story is a place holder for the conversation.

3. The third C is the confirmation where the acceptance criteria has been written, which allows the team to understand what will be the conditions of satisfaction. What will it be that will help a customer really accept a particular feature or a functionality?


So, we've seen the three Cs characterization of a user story wherein the first C stands for Card, second C stands for conversation, and third C stands for confirmation. In the next video, let's find out more about user story.

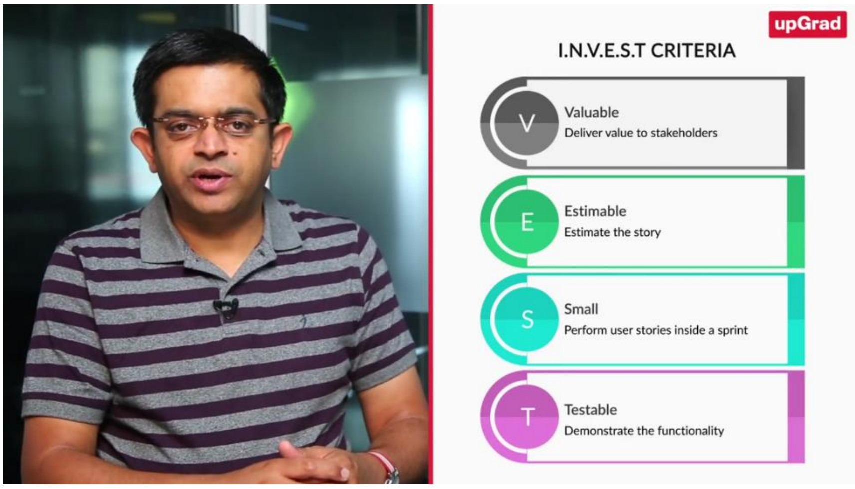

In the last video, we've seen that a user story can be best written as, as a user I would like to do dash, so that I can get dash. Now, how do you determine whether a particular story is good or not?

So, we looked at the 3 Cs of the user story, right? Even if the three Cs are followed, is every story a good story. We don't always know. So, we have a criterion that we call as invest. An invest is a criterion that we use to understand, identify, and even give a feedback, whether it's a good story or not.

- An invest is an acronym. Let's understand each part of that. I stand for independent. If the stories are not independent, they will create a dependency and it will make it very difficult for us to choose the story in any order that we feel is a value to the customers.

- The second thing is about, N is for negotiable. When I make it negotiable, I only specify the high level behaviour, and I trust the team and their judgment that they will come up with the best possible way to actually meet that requirement. And that makes sure that the team also is challenged adequately.

- V in invest is for valuable. Now valuable to us is in the context of the customers. It might be valuable for us as a development team, but if it is not valuable to the customers, we may not get a lot of feedback out of it when we deliver it to them. So, we do want it to be an end user facing functionality.



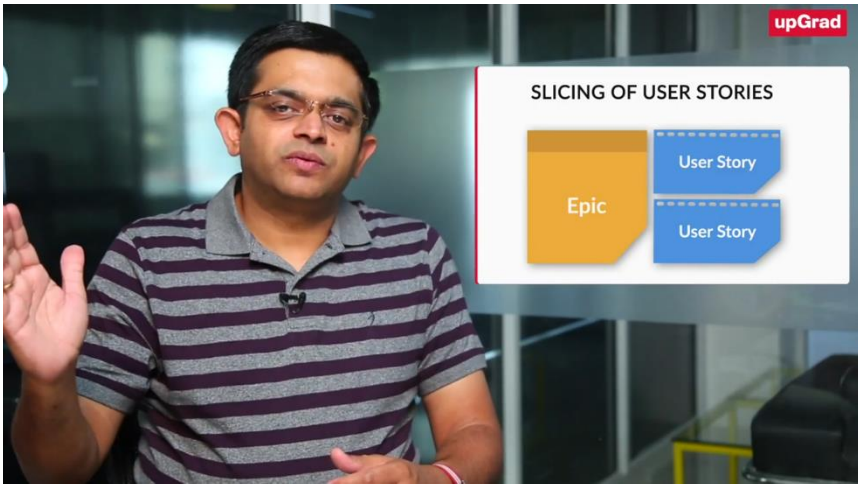

- E in invest is for estimable. We will talk about estimation in one of the future modules. But for now, it is important to understand that if I cannot estimate it, I probably have not been able to break it down into something simple that the team can run with. So, estimable is an important part of that.

- S in invest is for small. Once again, there is no mathematical definition for what is small. What we do agree on is small means I should be able to do a few stories inside a sprint. Definitely it should not really exceed the length of the sprint.

- And finally, you have the testable, which essentially means that you understand the conditions of success, you know how are you going to demonstrate the functionality, and you are in a position to know if any of those parameters is not met, then the conditions of success have not been met. Even if you are not written the test case for it at this point in time, that's perfectly fine. But in principle, it should be testable.

So, this acronym invest, independent, negotiable, valuable, estimable, small and testable. This is a great way to look at it the stories and understand if the stories are ready to be consumed by the team.

So, when the product owner explains or introduces a new story, the developer team invariably uses the filter of invest. When we apply the invest criteria, suppose we come to a point where the story is large. What are we going to do?

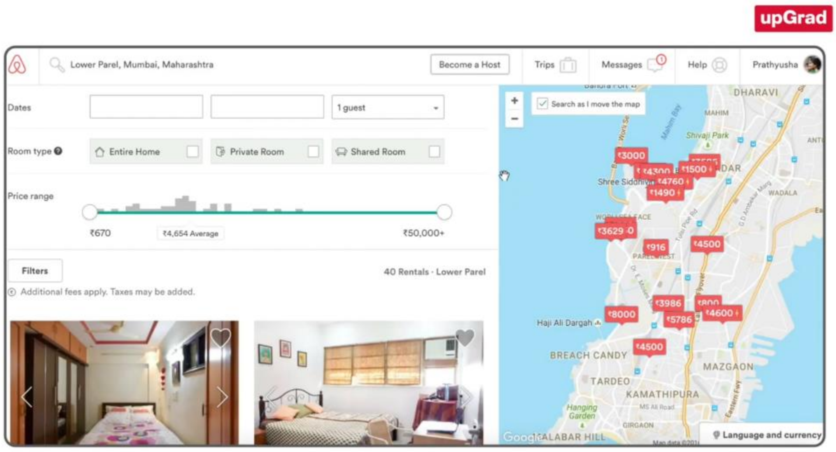



Now within the scrum body of knowledge, we have this concept of slicing down the user stories. Slicing down the user stories is all about how do we really create the vertical slices of the story so that each of the slices is an end user facing functionality. And we are able to then stagger them into our sprints or working towards the final release there.

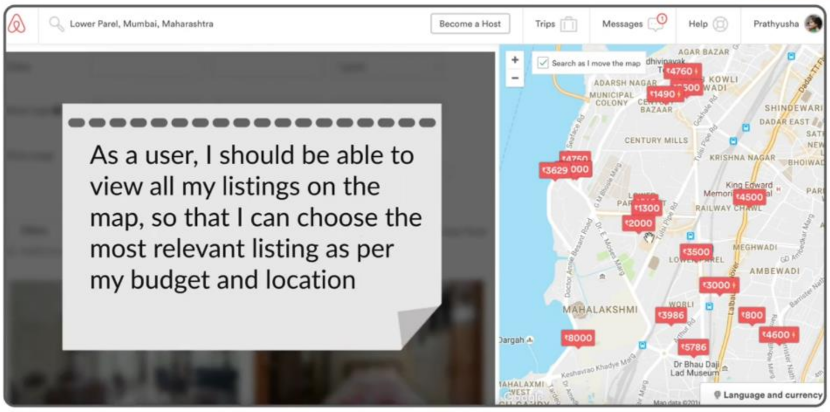

So, the invest criteria can be used to figure out whether a given user story is in good format or not. Let's see how user stories or epics would look like for any product.


e

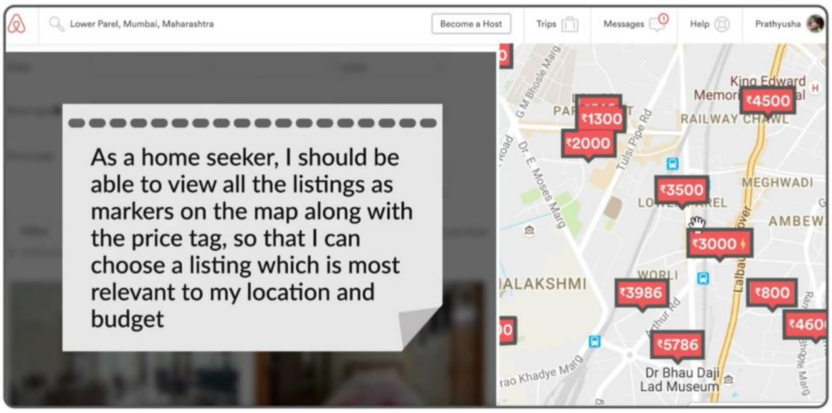
Epic is a big task, a very, very macro task, which will take good amount of time to develop. A macro task that can be divided into various micro task. Epic is more of a milestone that has to be achieved over the course of a sprint or over a course of various development iteration.



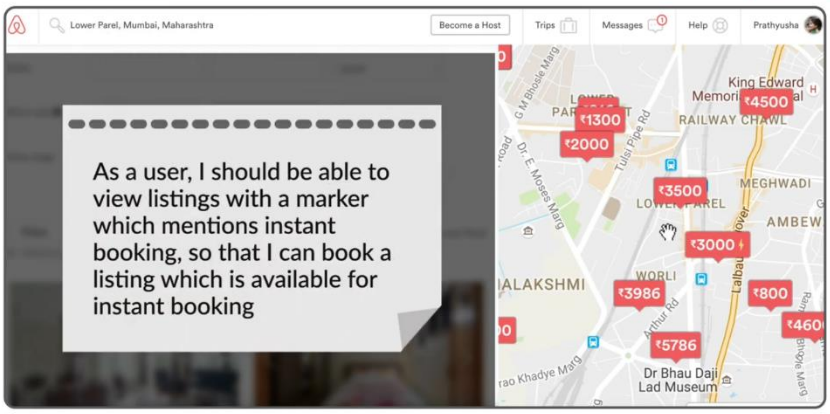

So, for example, if we have to take an example of Airbnb, so when you actually search on Airbnb, what do you see on the left-hand side of your screen, you see a map. So, an Epic for this map will be that as a user, I should be able to view all my listings on the map so that I can choose the most relevant listing as per my budget and location. Now, this is the Epic. This is like a big task, which can be further divided into multiple small user stories.



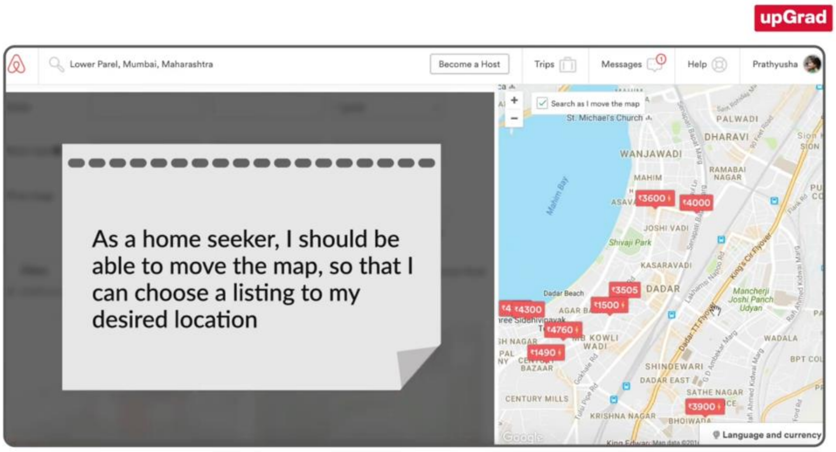

For example, one user story will be like that as a user or as a home seeker, I should be able to view all the listings as markers on the map, along with the price tag so that I can choose a listing, which is most relevant to my location and which fits under my budget.



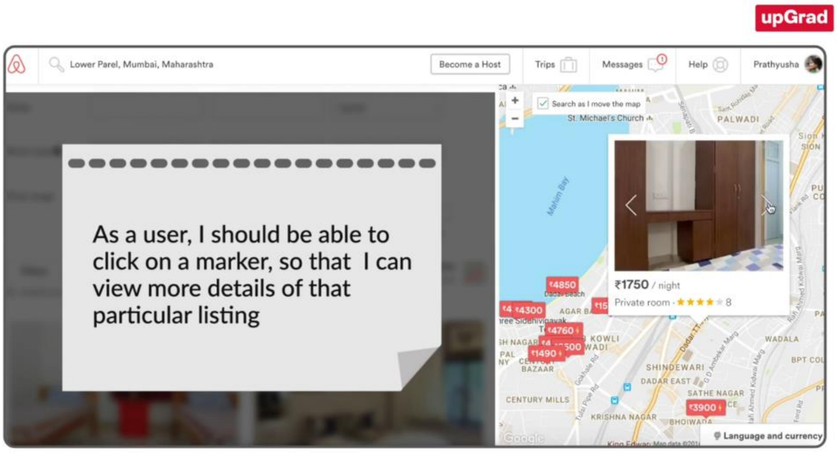

The second user study will be like that as a user, I should be able to view listings with a marker, which mentions instant booking so that I can book a listing, which is available for instant booking.



The third user story will be like that as a user or as a home seeker, I should be able to move the map so that I can choose a listing, which are available on that map frame and which are most relevant to my location and to my budget.

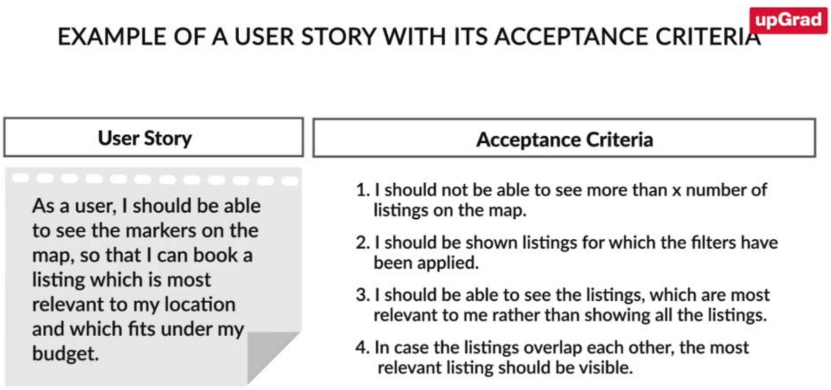



The fourth user story will be like that as a user, I should be able to click on that particular marker so that I can view more details of that particular listing.




While learning about the three 3s concept for a user story. The third C represents confirmation, which is nothing but the acceptance criteria for a given user story. Let's find out what acceptance criteria would look like for any user story.


Acceptance criteria helps a developer understand that these are the kinds of things that he/she should keep in mind while developing that particular user story. It helps a product manager keep a check on if I should accept this user story or not.




For example, if you take the first user story that we discussed that as a user, I should be able to see the markers on the map so that I can book a listing, which is most relevant to my location and which fits under my budget. The kind of acceptance criteria that I have is I should not be able to see more than X number of listing on the map. For example, if I show more than more than 50 listing on the map, the map will look very cluttered.

The second exception will be that I should be showing listings for which the filters have been applicable. So, let us say that a particular user applied a filter off by five. So, I'll not show listings which don't have Wi-fi as markers. The third acceptance criteria would be that I should be able to see the listing which are most relevant to me, rather than showing all the listings.

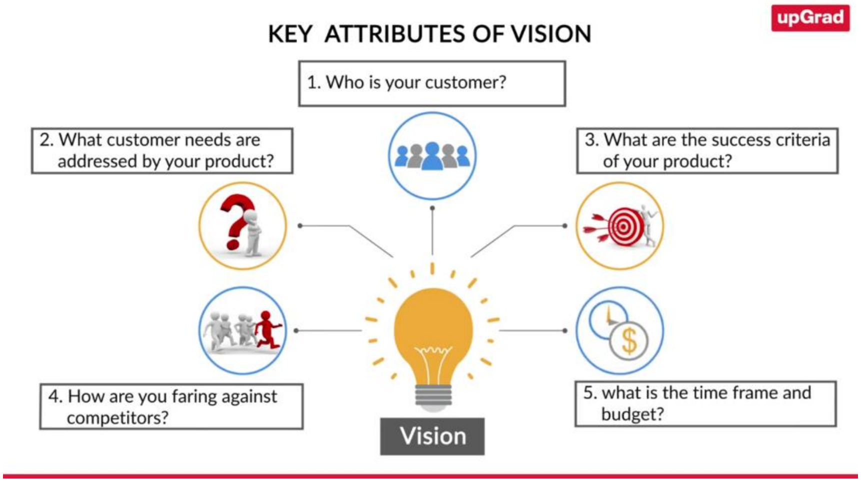
For example, if I have been looking at listings in a particular budget range, so I don't want to see listings which are out of my budget range.

So, the other acceptance criteria would be like that if two listings overlap each other, I should be able to see a listing which is most relevant to me. So, if all of these are checked and when I move this card from doing to done, only then

I'll consider the user story complete. Till then, the user story is not complete. This is why acceptance criteria is important.

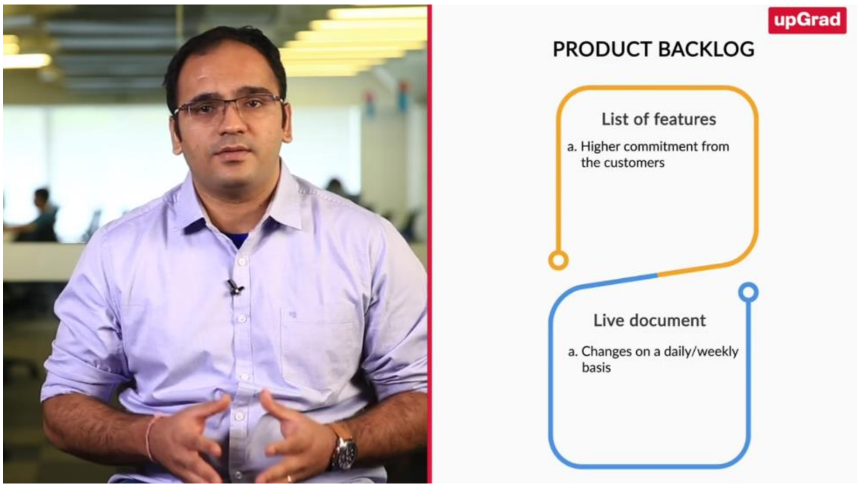

In this video, you learned in detail what user stories, epics, or themes are and how to go about writing the user story in a good format. You also saw how an Epic could be divided into multiple user stories and what acceptance criteria looks like for any given user story. Next up, a summary of the session lined up for you.


After an insightful session, let's do a quick summary. First, you learned about product vision, which acts like true North and primarily provides direction and focus to the team.

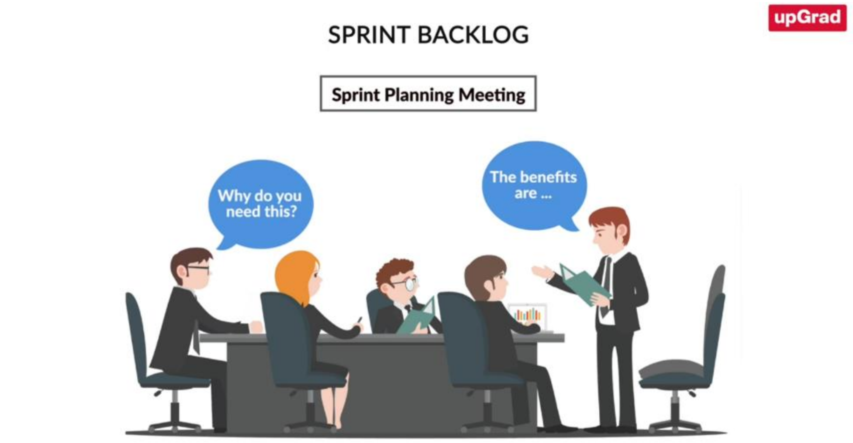



You also learned about the five key attributes that help you articulate the product vision. They are, who is your customer? What are the customer's needs that you are planning to address? What are the success criteria for your product? How are you fairing against your competition? And the last one is, what kind of time or budget are you looking at.

Then you learned various methods that are widely used to articulate the vision. Next, you discovered that to meet the vision, features are staggered along the timeline, which is called the product roadmap. The items on the roadmap are placed either on direct articulation or relative articulation of the value by customers. You saw that the roadmap has to be often changed based on current market scenarios.

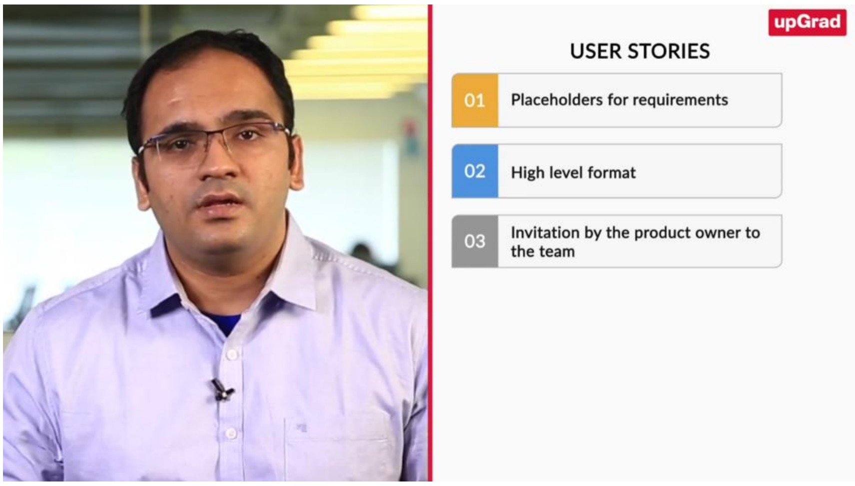



Later, you learned about product backlog, which has more signal to noise ratio compared to the product roadmap. The items in the product backlog are characterized by deep, which essentially means they should be detailed appropriately, immersion based on current scenario, estimated and prioritized.



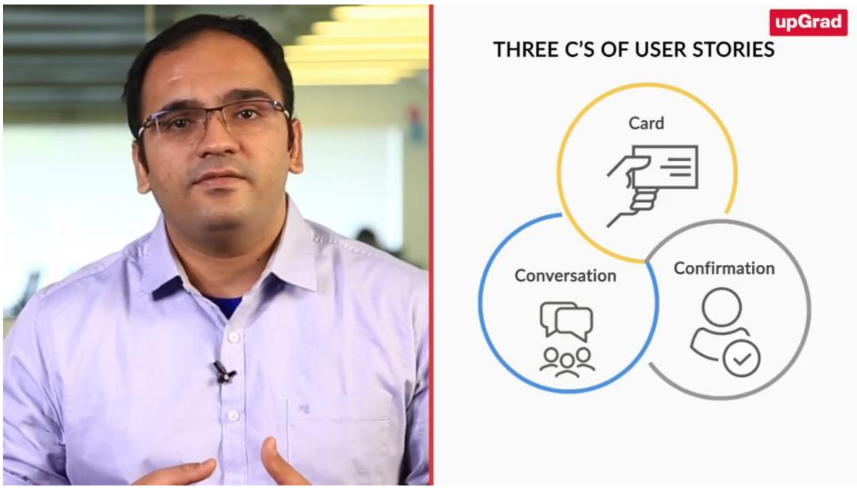

Next, the sprint backlog consists of items of highest priority in a product backlog which the team decides to work on during the sprint planning meet. One thing to remember is that once the items are moved into the sprint backlog, the changes and requirements are not resisted by scrum.

Later, you learned in detail about user stories, which act as placeholders for the team to have a conversation. Bigger user stories are taken as epics and the user stories which are similar fall under certain themes.




You learned that the suggested format of a good user story is, as a user I want to do dash, so that I can get dash, which consists of three parts.

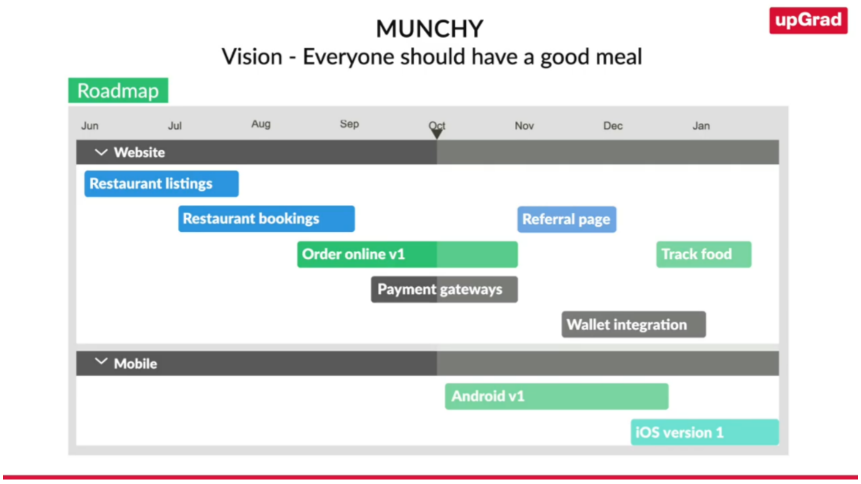
The first part makes it clear who the user is. The second part helps you know what they want. And the third part focuses on the result.



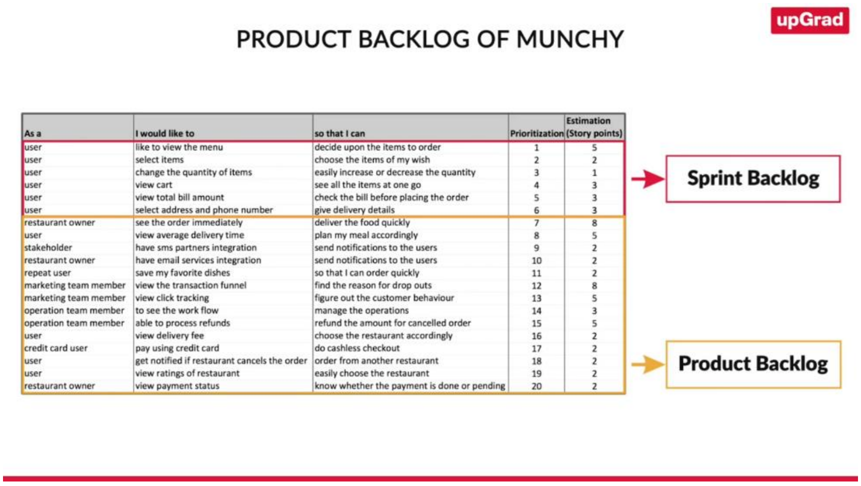

Good user stories follow the three Cs check and the invest criteria. The three Cs are card, so they remain short, conversations so that the format of the user story is retained, and confirmation which are the acceptance criteria for the user story.

To meet the invest criteria, user stories should be independent, negotiable, valuable, estimable, small and testable. Now that you've gone through the concepts covered in this session, let's recap the example you saw to understand these concepts better.

To understand the concept of vision. You saw the example of Google's vision to organize the world's data. To learn different methods of articulating a vision, you saw how the product vision box method is used, how Amazon prepares paper releases much before product planning and how Jeffery Moore used the elevator pitch.




Next to understand how a product backlog looks like, you saw the example of Munchy's roadmap, where the releases of the various features are staggered against the calendar. You also saw the product backlog items might look like for Munchie, where the top priority items are well detailed in form of user stories and properly estimated.

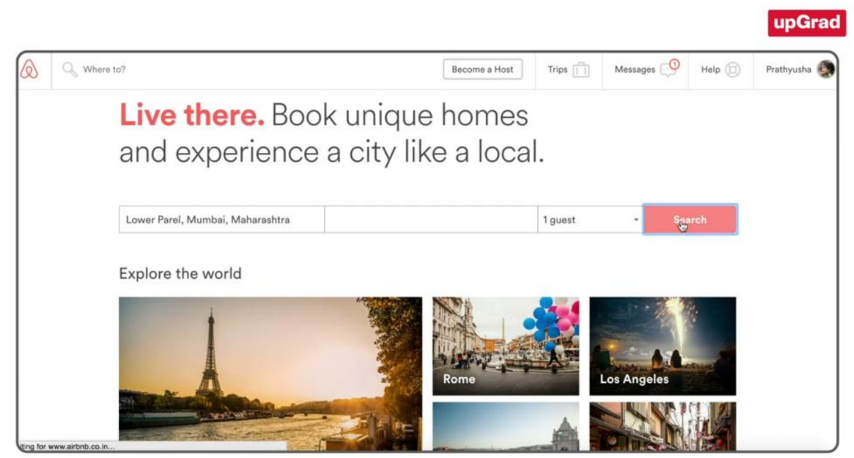


PRODUCT BACKLOG OF MUNCHY

| As a | I would like to | so that I can | Prioritization | Estimation (Story points) |
|---|---|---|---|---|
| user | like to view the menu | decide upon the items to order | 1 | 5 |
| user | select items | choose the items of my wish | 2 | 2 |
| user | change the quantity of items | easily increase or decrease the quantity | 3 | 1 |
| user | view cart | see all the items at one go | 4 | 3 |
| user | view total bill amount | check the bill before placing the order | 5 | 3 |
| user | select address and phone number | give delivery details | 6 | 3 |
| restaurant owner | see the order immediately | deliver the food quickly | 7 | 8 |
| user | view average delivery time | plan my meal accordingly | 8 | 5 |
| stakeholder | have sms partners integration | send notifications to the users | 9 | 2 |
| restaurant owner | have email services integration | send notifications to the users | 10 | 2 |
| repeat user | save my favorite dishes | so that I can order quickly | 11 | 2 |
| marketing team member | view the transaction funnel | find the reason for drop outs | 12 | 8 |
| marketing team member | view click tracking | figure out the customer behaviour | 13 | 5 |
| operation team member | to see the work flow | manage the operations | 14 | 3 |
| operation team member | able to process refunds | refund the amount for cancelled order | 15 | 5 |
| user | view delivery fee | choose the restaurant accordingly | 16 | 2 |
| credit card user | pay using credit card | do cashless checkout | 17 | 2 |
| user | get notified if restaurant cancels the order | order from another restaurant | 18 | 2 |
| user | view ratings of restaurant | easily choose the restaurant | 19 | 2 |
| restaurant owner | view payment status | know whether the payment is done or pending | 20 | 2 |



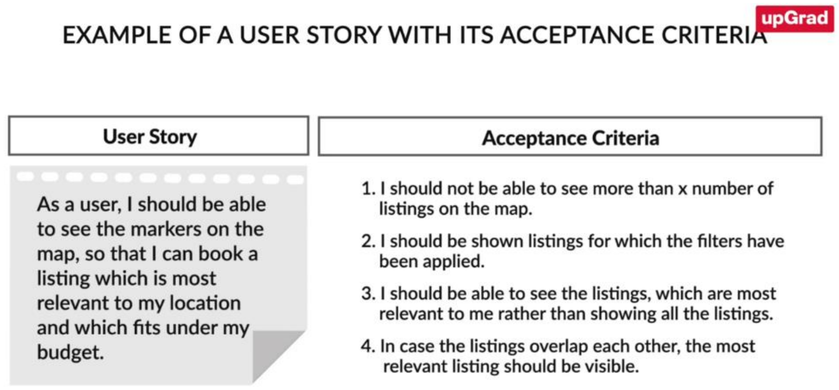
To understand how a sprint backlog looks like, we took the example of Munchy and saw the top priority items being moved to the sprint backlog. Later, you saw Facebook calendar as an example of the theme.




To understand Epic and how it sliced into user stories, you saw the example of the Airbnb map feature, which is an Epic, which is further sliced into user stories, like as a user or as a home seeker, I should be able to view all the listings as markers on the map, along with the price tags, so that I can choose a listing, which is most relevant to my location and which fits into my budget.

As a user, I should be able to view listings with a marker which mentions instant booking so that I can book a listing, which is available for instant booking. As a user or home seeker, I should be able to move the maps so that I can choose a distinct to my desired location.

As a user, I should be able to click on a particular marker so that I can view more details of that particular listing to book the most relevant listing for m.,




Lastly, to understand the concept of acceptance criteria, you took the user story of the Airbnb map features. As a user, I should be able to see the markers on the map so that I can book a listing, which is most relevant to my location and which fits my budget.

The acceptance criteria for this user story can be, I should not be able to see more than X number of listings on the map. I should be shown listings for which filters have been applied. I should be able to see the listings, which are most relevant to me rather than showing all the listings.

In case the listings overlap each other, the most relevant listings should be visible. With that, let's wrap up the session. You will learn more about how to create backlog items and PRDs in the next module. Until then, keep learning and have fun.