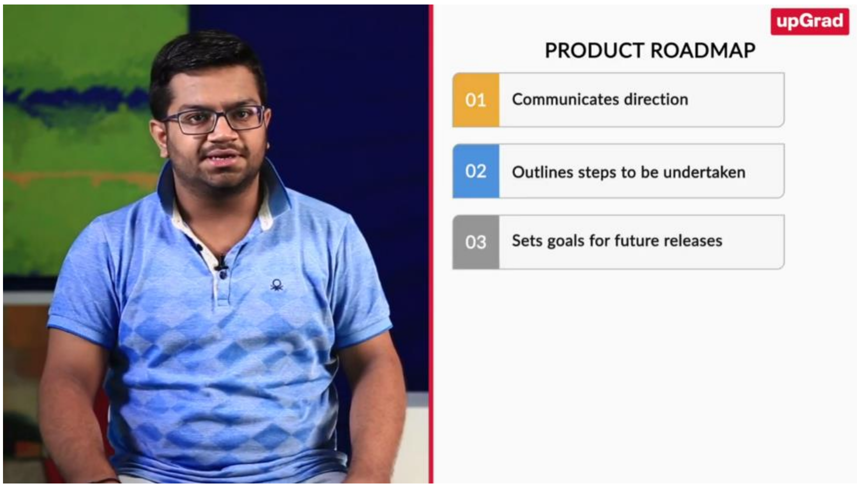
Transcription

Building A Product Backlog

Hello there. By now, you know how to articulate your product division with the help of key attributes discussed earlier. With that in mind, let's now delve deeper into the subject. Now, once you are into product management, you will hear many jargons like roadmaps, backlogs, user stories. As a PM, you have the responsibility to communicate those jargons effectively to the team members.

But how to go about creating roadmaps, backlogs, epics or user stories? Do not be overrun by complex words. We know who to ask. So, let us first understand what is the product roadmap? How does it look?



Vision acts like a true North for a product. In order to progress towards the vision, a product roadmap is created, which is nothing but a tentative timeline of major releases or launches along the calendar. It is designed to communicate direction and progress to internal teams and external stakeholders. It also shows the high-level initiatives and plan steps to get there. It helps set the goals which are achieved by strategizing the further release in a roadmap.

Every single company has a different way of road mapping. Before creating a roadmap, you need to know its audience so that you can tailor the content, focus and presentation to their needs.

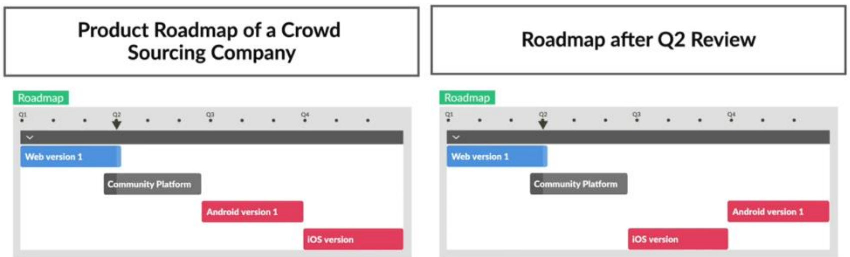
CUSTOMER FACING ROADMAP OF FACEBOOK

upGrad

For example, an external or customer facing roadmap of Facebook looks like this, where we can see how they are moving from product and platforms to developing various technologies. But the internal roadmap will be more detailed and generally not shared widely with a competitive advantage.

PRODUCT ROADMAP
Subjected to Change

upGrad

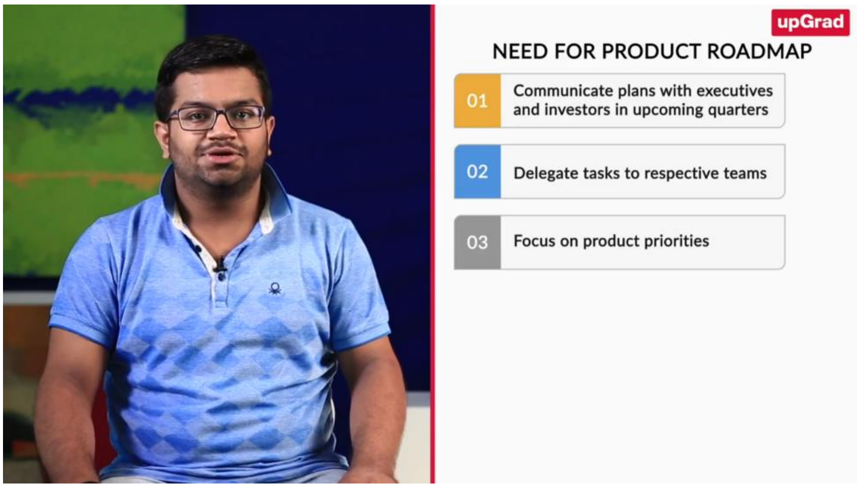


Furthermore, roadmaps are subjected to change. Let's say you are a crowdsourcing company in India and have planned an iOS app release in Q4 at the beginning of the year. In Q2, you figured out that the US market will be a lot more beneficial for your product and decided to hit US market by quarter three.

So, now we have to prepone the launch of iOS app to capture US market, as the majority of customer out there have iPhones. So, it is okay to have a roadmap as a general guide. But if you notice that you are sticking to the same timelines very accurately, then it might be possible that you are not agile at all.

Also, when companies got acquired or merged, it is not uncommon to see the roadmaps and double changes. For example, when Microsoft acquired LinkedIn, the roadmap of LinkedIn might've undergone the changes to fit the combined vision of both the companies.

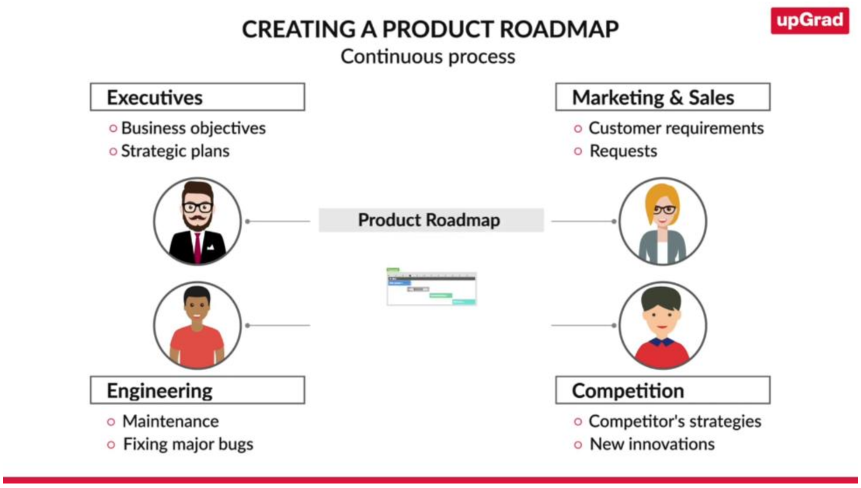
So, a product road map is like a timeline of features and targeted releases. This makes me wonder why do we need roadmaps at all?



One reason is to convey where we are heading to the executives and investors in the coming two to three quarters. It sets a sense of direction and ownership for the teams. It helps them to focus on product priorities. Also, when you develop products, especially in B2B domain, the timeline might sometimes align to an actual deadline.

We see this a lot in big companies where the investors say that we need to have so and so functionalities to compete in the market. Or maybe you work in a software services company where you sales people sell a feature that you haven't yet developed. And the customer says that I want proof of concept or version one, and so, and so by February. At that point, we place the same on the roadmap to plan further.

Alright, so roadmap is envisioned to communicate goals and strategies to various stakeholders. But how do we go about creating a product roadmap?

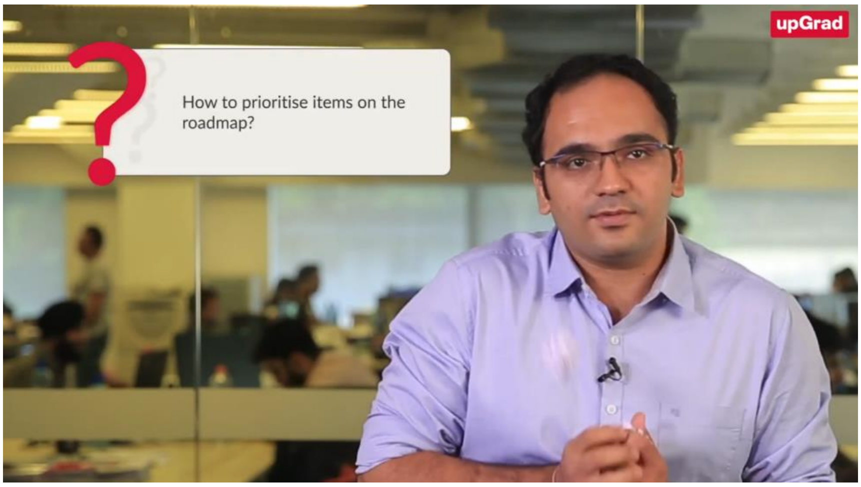


Creating a product roadmap is not one-time process. Rather, it is a continuous process through the life cycle of a product. Requirements are generated by lots of people, including customers, partners, sales, support, engineering, management, operations, and product management.

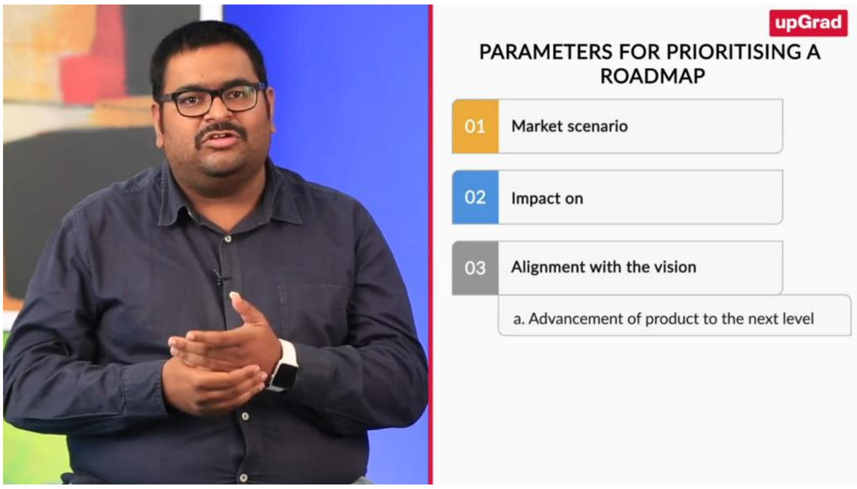
Executives convey their business objectives and strategic plans. Marketing and sales teams are good sources to know customer requirements. Engineering people might want to do some maintenance and fix major bugs.

Competitors might be launching a new feature, which saw a lot of traction, or there might be some new technology release in the market, and many of your customers are going head over heels over that. So, you might want to act upon it.

As a PM, you get a lot of inputs, suggestions, and requests from different stakeholders. It is up to the product management team to determine the priorities and make sure that roadmap is aligned with the business goals. Once the goals are decided, write down the initiatives that can be taken to meet these goals and then map them to features or releases. That in short is what we call road mapping.



So, the roadmap gets fine-tuned as we move ahead in time. But in a given situation, on what basis should I filter the inputs that are coming in from various team members and how to prioritize the inputs?



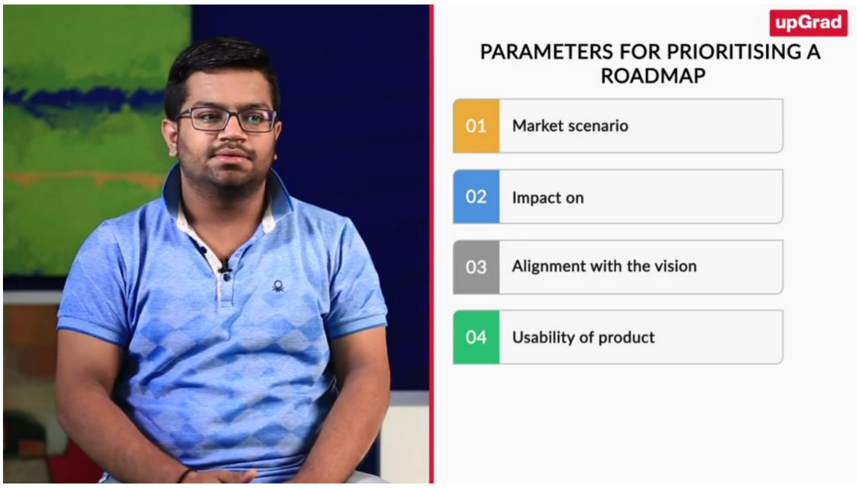
So, priorities usually are decided on three things. One is how soon you want this particular feature out there in the market because a competitor already has it, or it's the most demanded feature by the customer. So, the time criticality defines priority, that is one thing.

Second thing is what is going to be the impact of this feature on your user base or your business. So, its if it's a high impact feature, obviously prioritize at a higher level because it is going to either bring me more users or either going to bring me more money in terms of business.

Third thing is how well it aligns with my vision and whether it is progressing my product to the next level or not. So, every feature, every, enhancement which you are making should take your product one step ahead.

So, instead of doing something in parallel, all the things which you are doing in parallel, experimentation probably carry lower priority because that is not taking my product ahead. It probably is giving me a backup strategy, right? But something which always takes my product ahead in terms of functionality, in terms of satisfaction, in terms of acceptance, always take the higher priority.

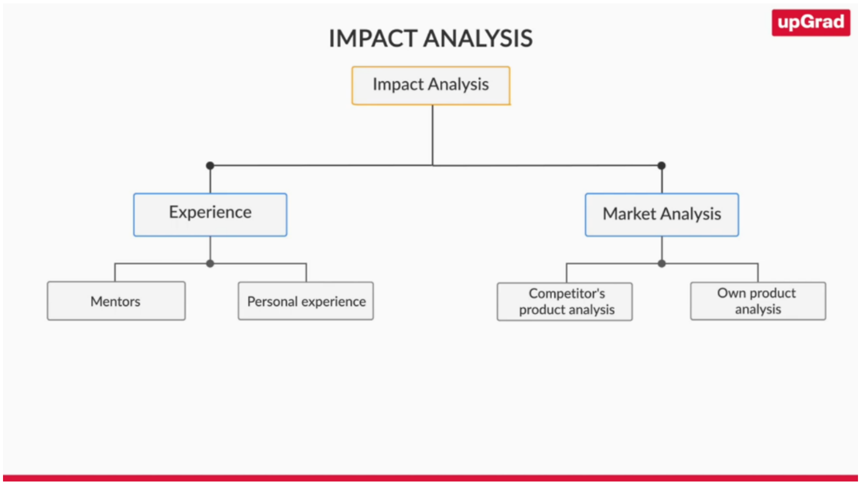
So, competition, market, share and alignment of the inputs to the vision forms the basis of a PM's priorities. Now let's find out how Zimmber prioritized the items on its roadmap.



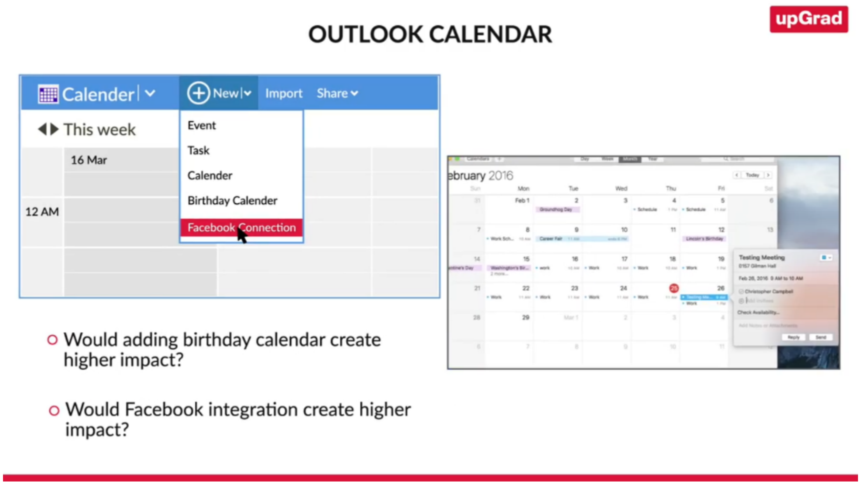
What we do at Zimmber is that we give the most utmost priority to the usability. For example, via analytics we saw that the users are dropping at some point and they are facing some issue, then we'll fix that first. Then secondly, we'll move on to the business goals, which requires new features so that we can maybe attract more customers or improve conversion.

If you noticed, in this scenario, Zimmber prioritized customer's user experience with the platform to get a hold of the market share. Once those are achieved, they move on to develop new features, to reach the business goals, which allows them to attract new customers.

We have seen that one of the parameters for prioritizing the roadmap is impact. But how do you judge, which features have a high impact on the user base?

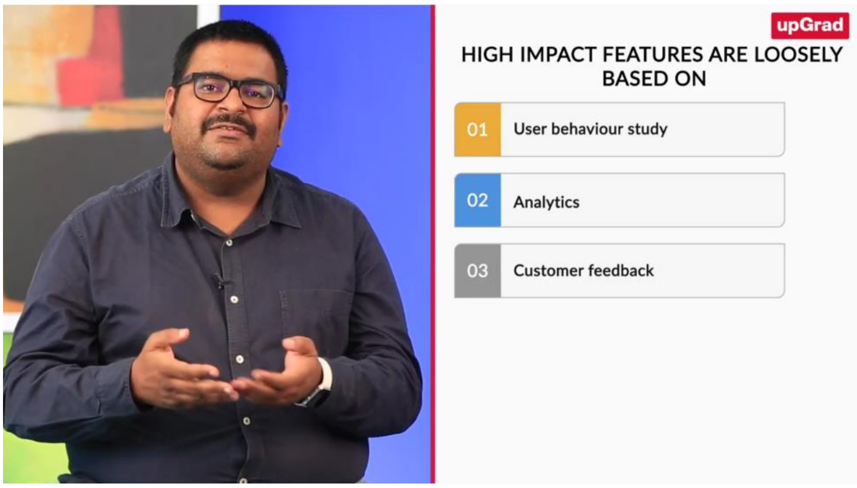


So, there are different ways of doing it. You either can do it through experience, or you can do it through proper study. So, proper study in a sense that you study what has already been built either by competition or by yourself, and what does the data say.



So, whether data proves that, okay, this particular feature if we add, so a lot of people are now using our calendar, even for casual appointments and everything. If we add birthday calendar to this, whether this would give a bigger impact, whether if we add Facebook connection to this particular calendar, whether this would be a high impact feature.

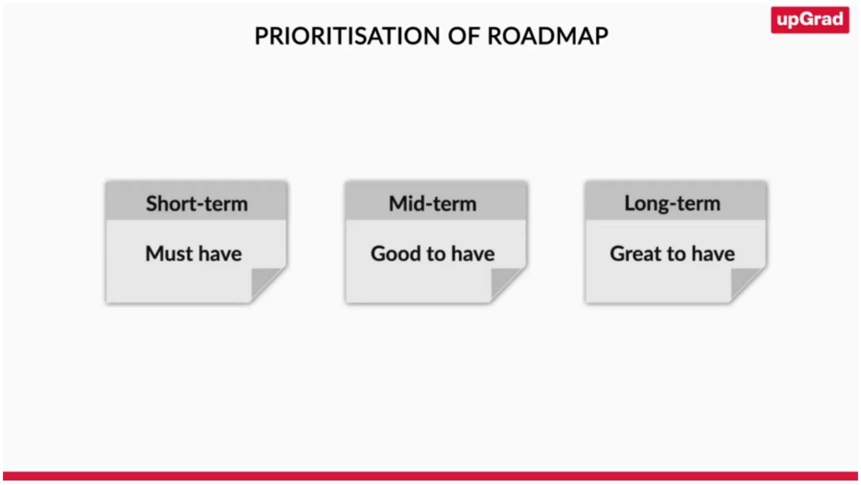
A lot of people actually, we have seen that reply to meetings saying that, okay, I'm running late. Can you reschedule? I'm running late for this meeting. So, can we add a very quick reply button on a meeting itself? It says, one single click, user can send I'm running late.



So, these are very small, but very important feature because they have very high impact because we have seen either from user behaviour study from analytics or from customer feedback, that these are some of the scenarios which they want to address. So, that actually led us to believe that, okay, this is a high impact feature.

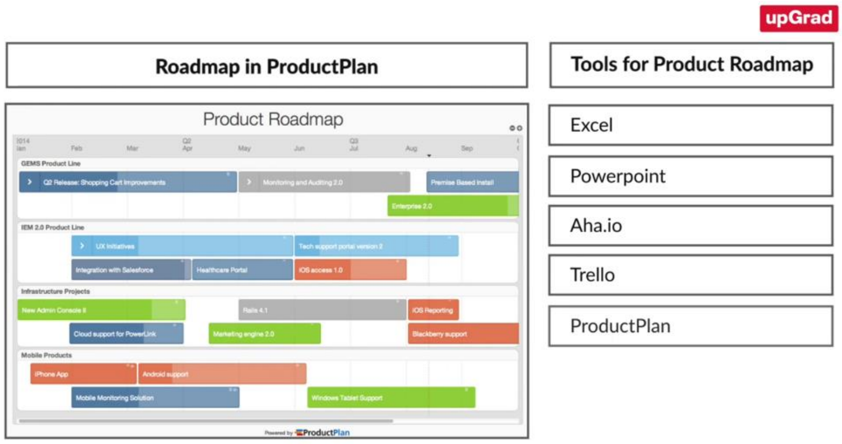
That's great. Now let's learn more about roadmaps and how prioritization is done in different companies.

Creating a compelling roadmap is not that easy, particularly in agile context where changes occur frequently and unexpectedly. What a lot of product development teams out there do is placing the request in different buckets. Here's what we are going to do now.



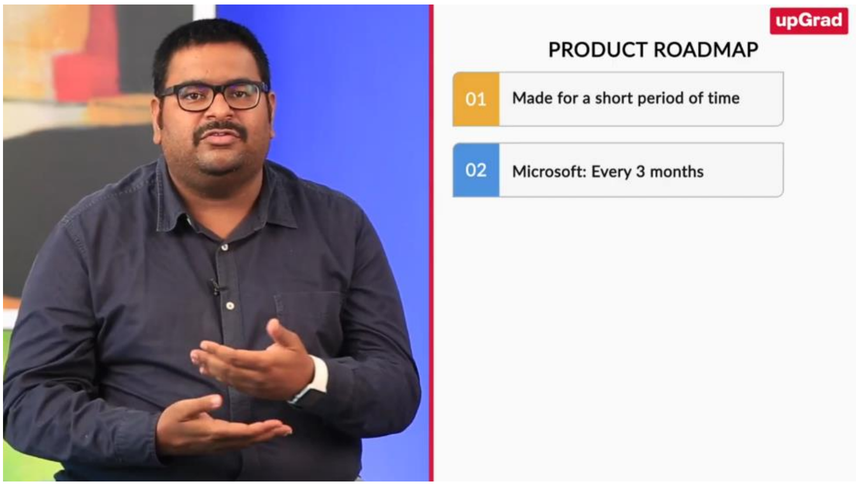
Immediately, like in next few weeks, or next month are placed in short term bucket. Next is a midterm which consists of items which you want to do immediately after short term items. And then comes the long-term bucket, which contains items, which we want to do eventually, but they're not of high priority right now.

This can be done by looking at how important it is and how urgent it is to implement it soon. So, the must have items mostly go in either short term or midterm. Good to have items might be placed in midterm or long-term buckets. And great to have, mostly will fall in the long-term buckets.



Product management team then conveys the same to all stakeholders and fix upon some tentative timelines. To accomplish this and to get a plan of record, you need a collaborative roadmap that offers ongoing visibility. Roadmaps can be created in Excel, PowerPoint, or shared platforms like aha.io, Trello, product plan, etc.

Alright, but how far into the future is the roadmap made?

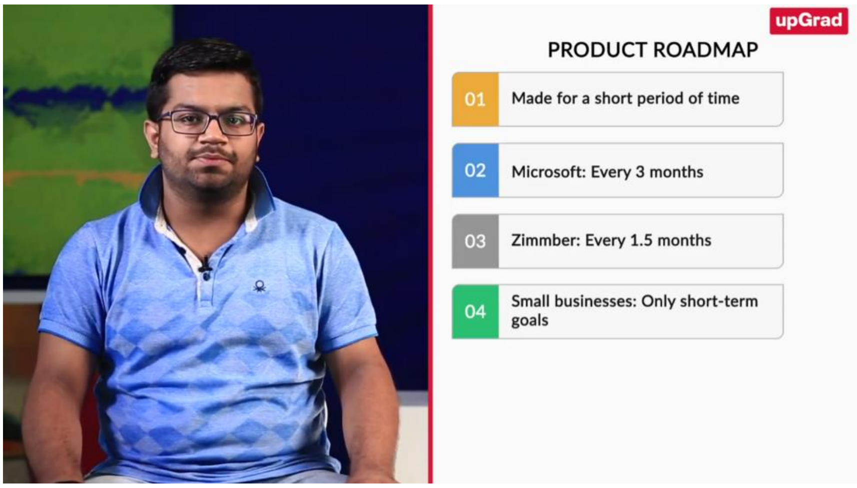


So, that far is no longer remaining far. So, we have to kind of bring it down to say two months, three months' timeframe. The reason I'm saying three months is because a lot of behavioural changes are happening. I wouldn't say innovation is happening. Innovation still has its own life cycle and timelines.

But a lot of behavioural changes are getting triggered and we have to constantly adapt to those behavioural changes. We constantly adopt to how user expectations are changing. So, that's the reason why we cannot afford planning like a year down the line or two years down the line now. We plan for say three months and say that, okay, this quarter we will do these things. These would be our targets. And these are the features which we will deliver.

And in that also, if we receive a certain behavioural change, we adapt to it by either accommodating and negotiating few features out of the current priority list if the other features which are coming in are of the higher impact, or align them in the next roadmap, which is the next three months.

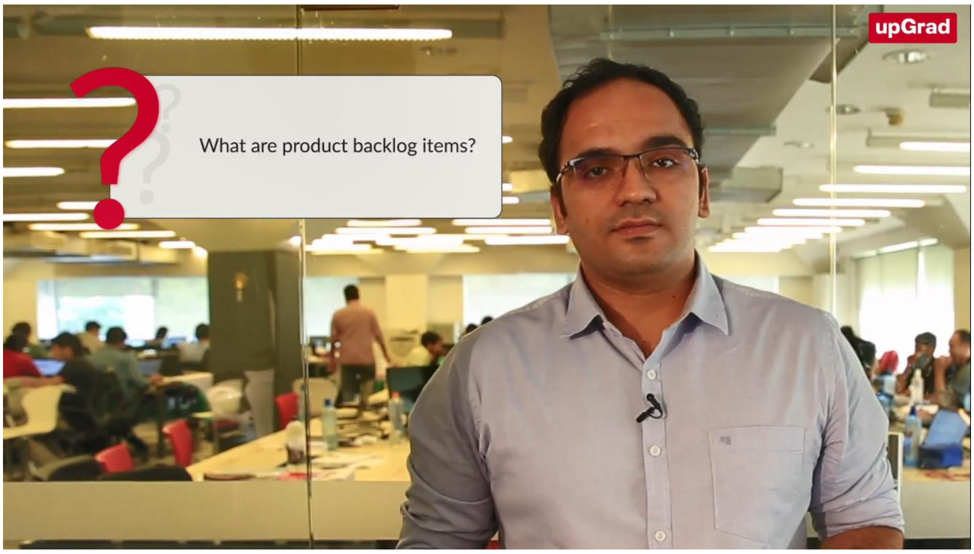
So, should we formulate roadmap for every three months?



There is no hard and fast rule that product roadmap should be made for next six months or one year. We have seen companies like Microsoft create a roadmap for every three months. We at Zimmber create a roadmap for every one and a half month. Some small businesses even choose not to have any roadmap as such, but they do have short term goals in their agenda.

It is you as a product manager who should continually keep on populating the roadmap whenever you have inputs from the stakeholders and customers.

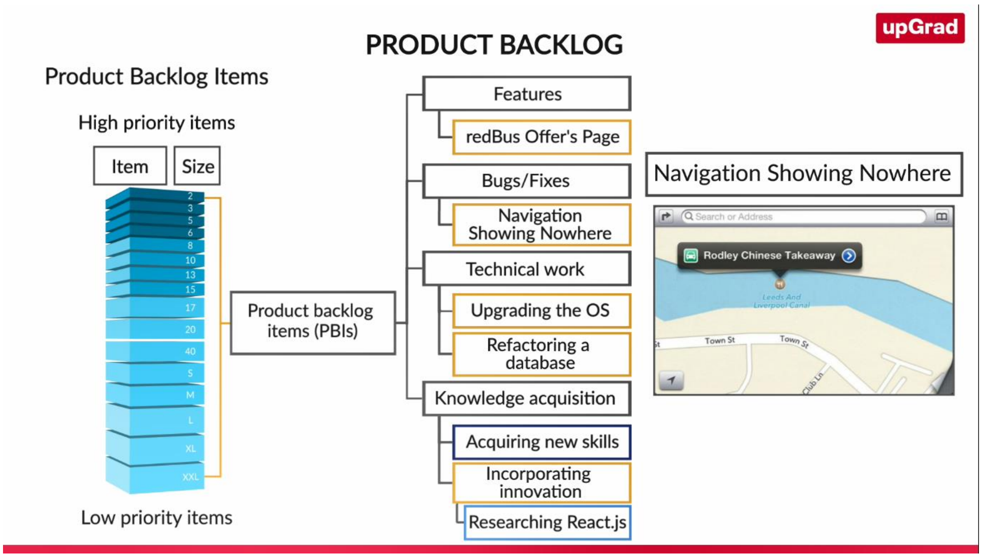
In this video, you learned that a product road map is nothing but a tentative timeline of major releases along the calendar year. You learned from Zimmber's example how to prioritize the parameters of a roadmap and the most important factor time. The roadmap needs to be accessed periodically to track progress of both short term and long-term goals.



As a PM, you might have observed that you have been juggling hundreds of demands from the management and customers alike. You might be bombarded with requests, demands, or feedb ack.

How can I handle all of this? Is there supposed to be some kind of a to do list to be prepared every day? Or what am I supposed to do with the suggestions? Keep them aside and start working on the next level feature. Let's ask our subject matter expert, what is product backlog and what are the product backlog items?

A product manager gets a lot of requests on a daily basis from various stakeholders. It is very important for a PM to maintain a to do list, categorize in terms of themes or features or perks or suggestions.



In our product management ecosystem, we call this list as a product backlog. As you have seen, product backlog is a prioritized inventory of work to be done. It is composed of product backlog items, which are often written as user stories.

Before getting into writing user stories. Let us first see what all constitute a product backlog. Product backlog items majorly constitute list of features, which represents the items of functionality that have tangible value to a customer.

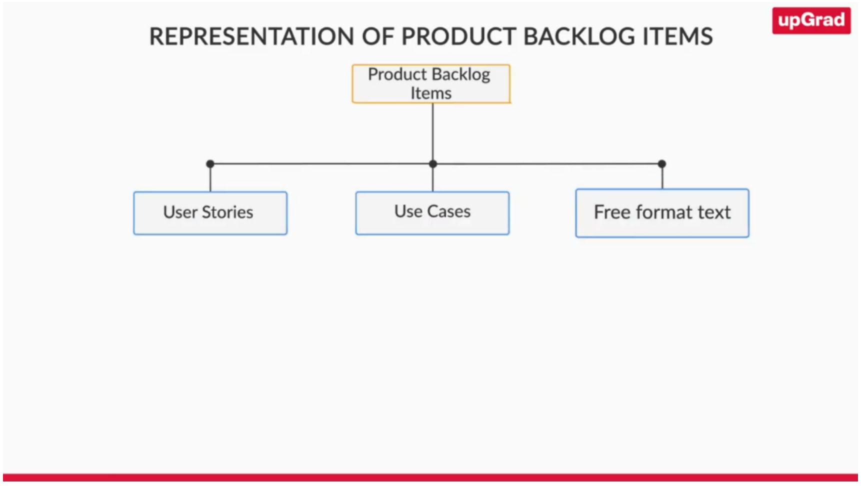
Let's take for example, the offers page of red bus, it can be considered as a feature. Items of this type can be brand new features or changes to existing features. Product backlog also may contain bugs to be solved, as we don't want to ship new features every month without fixing the bugs, which are brought to your notice.

For example, let's say you are developing a navigation tool and you found a bug where directions are given to nowhere. So, you might want to fix it. Sometimes, product backlog might also contain the items related to technical work.

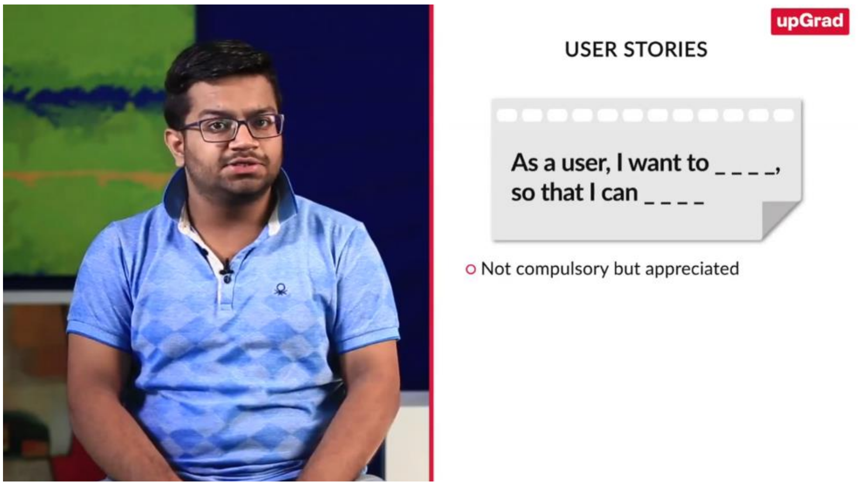
Example might include upgrading operating system on all developer’s machine or refactoring a database. These items are needed so that the team works more efficiently or allow the product to function better as a whole.

Another type you might want to include is items related to knowledge acquisition. Learning new things and incorporating innovation in the product needs some time to get everyone acquainted with new knowledge. For example, researching react JS and how you can incorporate it in your product.

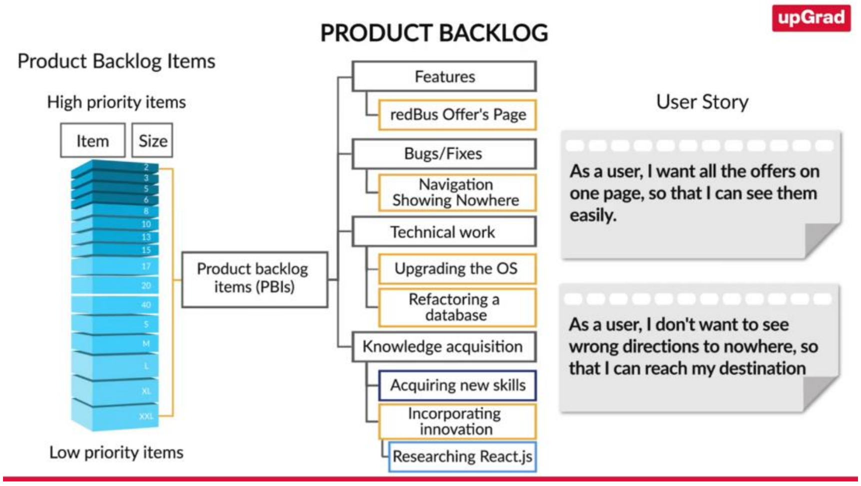
Okay. That's great. How should we represent these product backlog items?



Different companies use different formats like user stories, use cases or the simple free format text. Let's see them one by one. User stories are the well-proven approach for the agile teams. They are typically expressed like this.

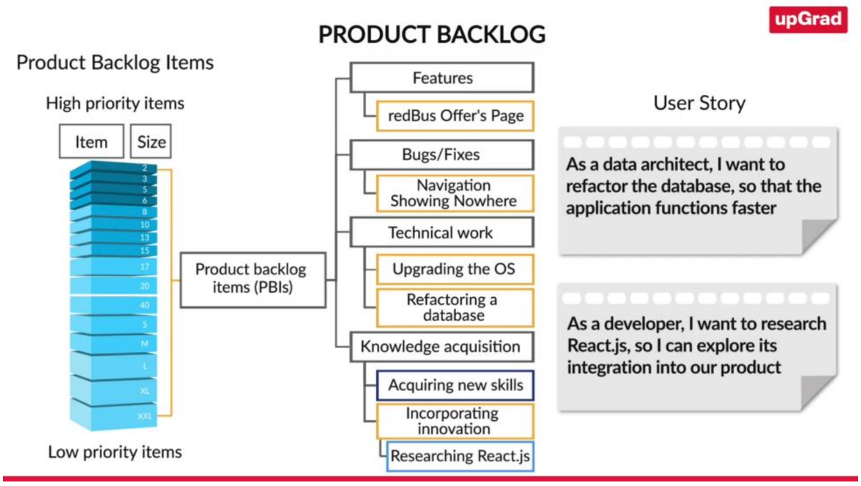


As a user, I want to achieve this particular goal so that I can do this. It is not a required practice in scrum, but it is widely appreciated because it gives a clear idea on who are we building it for, what we should build and why are we building this?

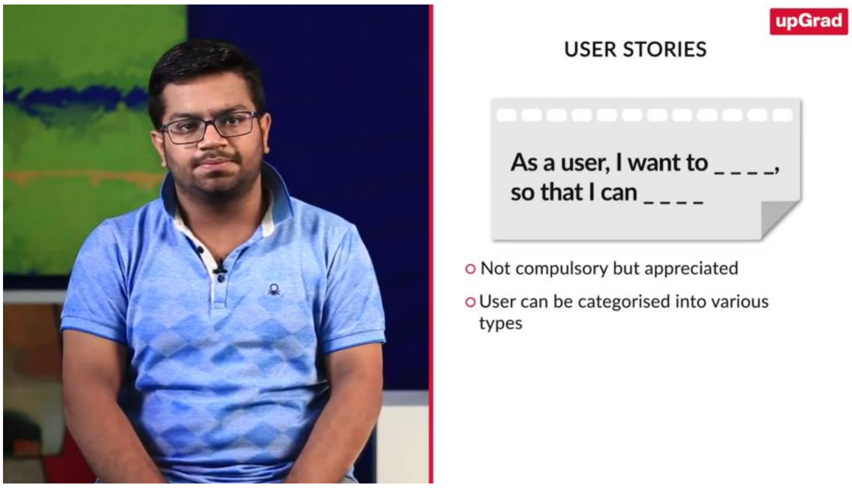


You can write user stories for any type of product backlog item. Let's take the same examples and try writing some user stories. So, for the feature offer page on Red Bus, the user story will be something like, as a user, I want all the offers at one place so that I can see or use them easily.

In case of the bug, navigation showing directions to nowhere, it can be, as a user, I don't want to see wrong directions to nowhere, so that I can reach my destination.



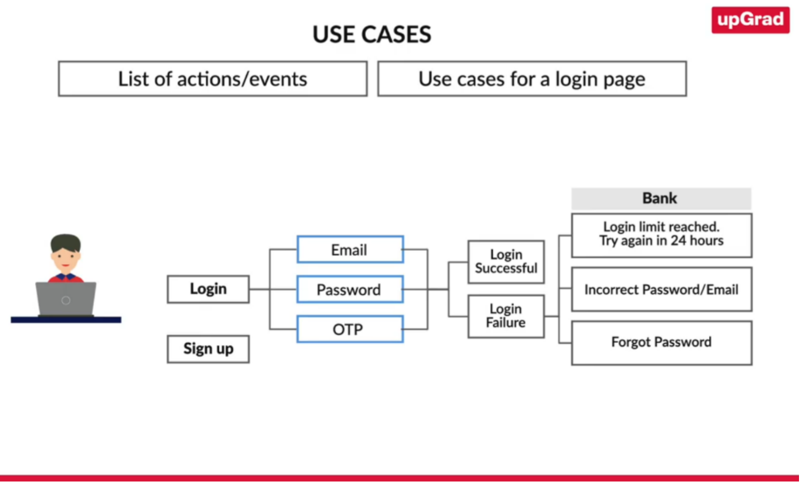
For a technical work, the user story can be like, as a data architect, I want to refactor the database so that the application functions faster. For knowledge acquisition, it can be like, as a developer, I want to research react JS so that I can explore its integration into our product.



Users mentioned in the user story can be categorized in many ways like registered user, repeated user, home seeker, broker, developer, tester, etc., depending upon the scenario.

So, you have seen how user stories could be used to describe different types of backlog items. Next, we will learn how they're represented as use cases.

In the last video, we understood that one of the ways to describe backlog item is by describing them in terms of use cases. Now let's understand use cases.



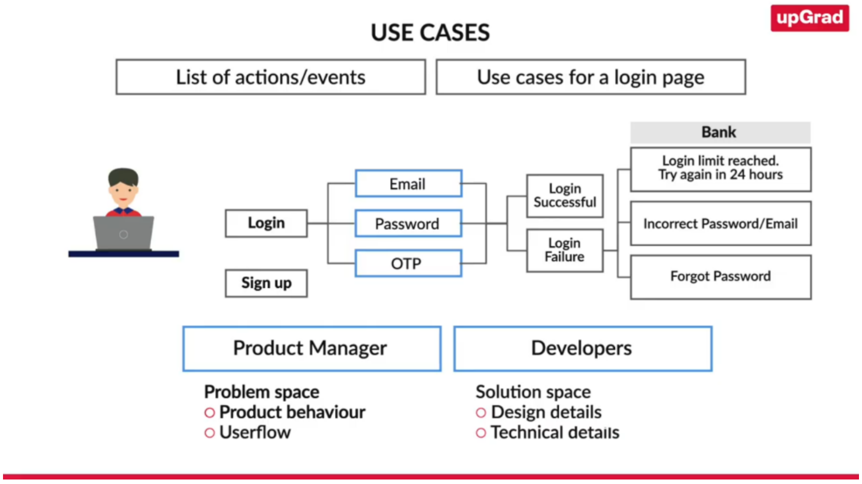
Some product managers also represent product backlog items in the form of use cases. Use cases is nothing but the list of actions or events that happen when the user is interacting with the product.

For example, if you are writing the use cases for a login page, these are the some of use cases which you will have to write. One can be a successful login, other could be what will happen if the failed login. For example, in failure, there could be one scenario where the password was wrong. And in other scenario, maybe the email that he entered was not even registered in your database.

Then one-use case could be when he forgot the password, he wants to reset his password. Another could be, he just wants to use OTP for login. One more use case is that if he's not even registered in your database, he might want to sign up.

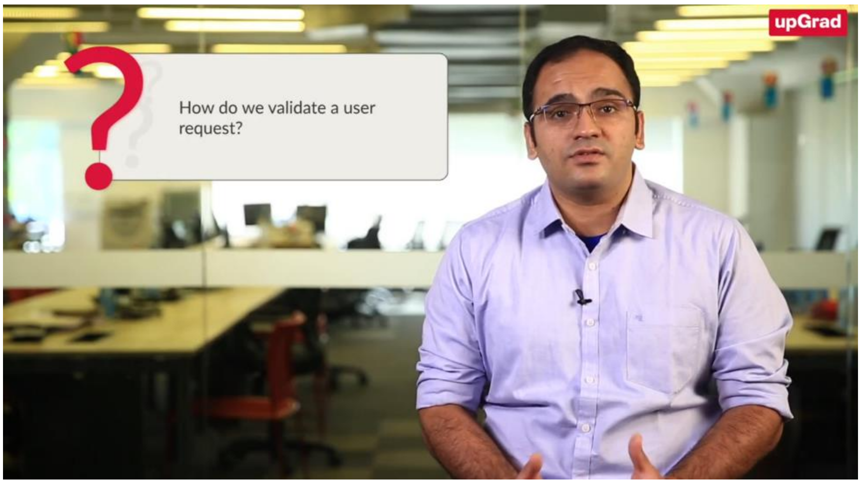
In case of a login page, a bank may have an additional page, which says you made more than five incorrect attempts, please try after one day. When designing the login page for a bank, the PM has to think of this scenario as well, and to have a popup or a separate page to tackle this use case.

So, that is as simple as that. You have to really put yourself in the user's shoes to get good at writing use cases. In product development life cycle, you will observe that code developers and product managers use the term use cases, but in different context.

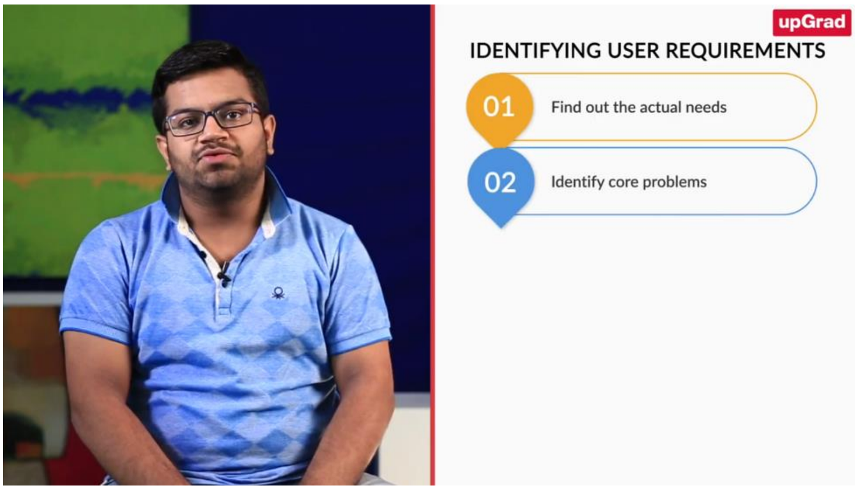


Though, principle behind both are same. The use case that is created by product managers is for problem space, and that is presented to developers to work on the solution space. Then developers use it as a base and add design and technical details to create another suit of use case so that they can have sufficient information to start designing or developing or testing the product or its features.

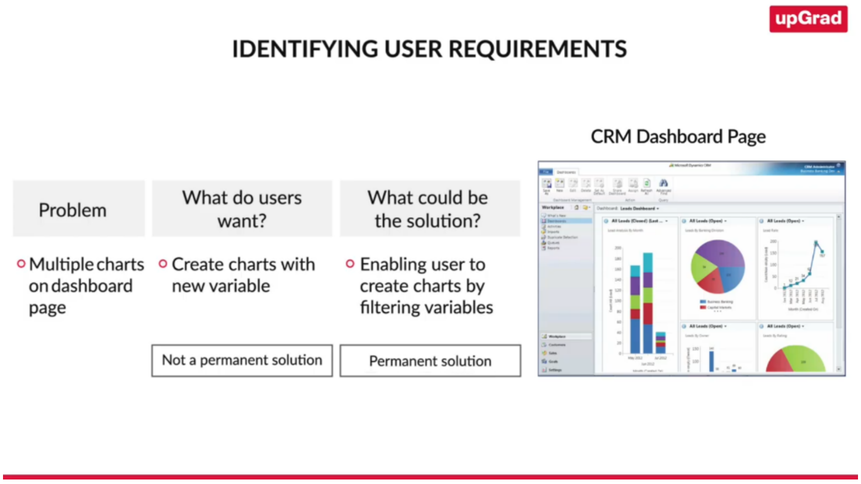
As a product manager, you should be creating use cases, keeping in mind the user flow, and shouldn't get into details of design or technical details.



On what criteria shall I base my judgment to incorporate user requests? Or how should I decide whether a user request is valid or not?

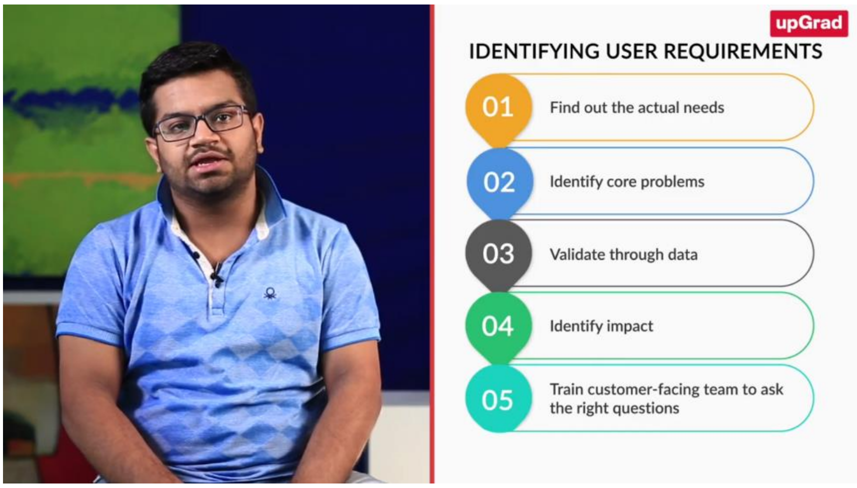


Sometimes while detailing out, the product manager should also analyse the user requests and identify the actual need instead of directly incorporating the request. Getting into the core of the problem will help you figure out the actual reason.



Let's say you have a customer relation management page for your sales and customer support team, and you've got repeated requests to set up multiple charts in page in various combinations. You might as well go and develop a product showing various charts. What if there is one more variable added in the future? You would have to repeat the same process then again.

But when you dig deeper into the problem, you can understand setting up a filter for creating charts will be a permanent solution rather than showing charts in various combinations.

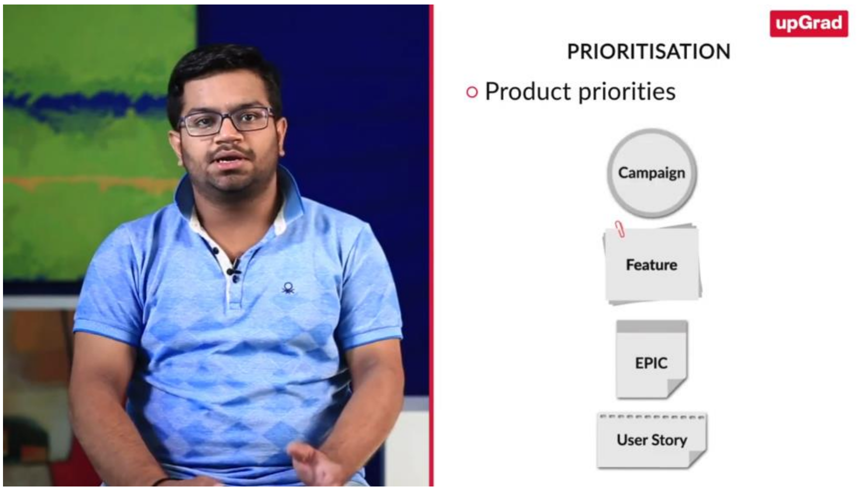


As a product manager, when you get requests or feedback from different stakeholders, write them down. Make a list of them. And when you actually get time to analyse them, try figuring out the core of the problem. Use real data, actual numbers, legitimate proof to estimate why you believe it is important rather than passing on a request from a customer.

Present a supported case for why the idea will generate more sales or improve user experience or improve traction. Also, train your customer facing team to ask right questions about the features or requests to get better insights from the customer feedback.

In this video, you learned about product backlogs and what constitutes the product backlog items. You have also learned that product backlogs can be represented by either user stories or use cases, and saw how they are represented. Now, let's figure out how to prioritize backlogs.

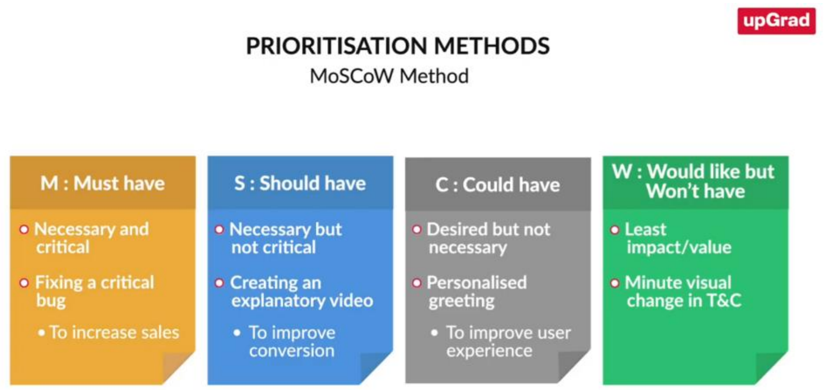
Now, considering the complexities of business, there ought to be many backlog items. I'm wondering on what basis should these items be prioritized?



As you keep on getting the inputs of backlog items, you will realize that you don't have resources to develop all features or products at one time. So, you may have to pick one by one or in batches, depending on the resources available. And that becomes simple only if you have a prioritized list.

The priority over there should be at a granular level. As a product manage, you won't be just prioritizing user stories. You might have to prioritize epics, features, and you might sometimes have to even prioritize marketing branding campaigns based on your product launches.

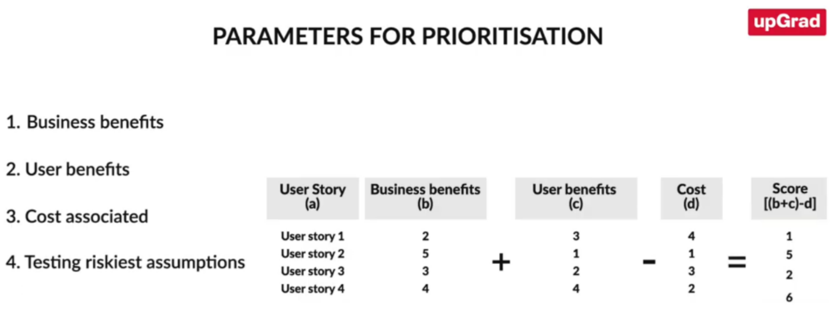
So, how to do prioritization. What are the parameters to consider? There are some common methods that are used by a lot of people out there to prioritize user stories or epics or initiatives.



Initially, you can divide the stories or epics into large buckets using Moscow method. We have M four must have, S for should have, C for could have, and W for would like but won't have.

Must have bucket containing all the necessary stories, which are critical and must be implemented immediately, whereas should have consist of necessary items, but they can be implemented in the next two to three sprints. Requirements in could have bucket are desired, but not necessary. Whereas, would like bucket contains story of least importance or less payback.

For example, fixing a critical bug, which is not allowing users to submit a booking or a feature which can increase sales by X percentage is a must have. Having explanatory videos to improve conversion can be a, should have. Personalizing a welcome greeting message with user's name in it can be a, could have. And changing the colour of that underlying on a terms of use page could be, would like, but won't have.



You may now ask the questions like, are there any business benefits? What are the user benefits and what is the cost associated in developing those stories? In order to remove the subjectivity in prioritization, you can give each story of value on a scale of 1 to 5 or 1 to 10 for each of these aspects. Once you have the value for all three aspects of a feature, add the values of a business and user benefits and subtract the score of cost from it.

Now, based on the final score, prioritization can be easily done. Testing the assumptions is another criterion. You might have already validated a lot of assumptions through user research, data or by other methods. Yet some user stories might be left where assumptions are not tested. So, while prioritizing with an Epic, the priority is given to the stories which helps us to test the riskiest redemption first.

PARAMETERS FOR PRIORITISATION

1. Business benefits
2. User benefits
3. Cost associated
4. Testing riskiest assumptions
5. Impact
6. Effort
7. Opportunity cost

User Story (a)	Business benefits (b)		User benefits (c)		Cost (d)		Score [(b+c)-d]
User story 1	2	+	3	-	4	=	1
User story 2	5		1		1		5
User story 3	3		2		3		2
User story 4	4		4		2		6

Impact-Effort Analysis	
Low effort and high impact ✓	High effort and low impact ✗
Fixing a small bug	Creating an additional page

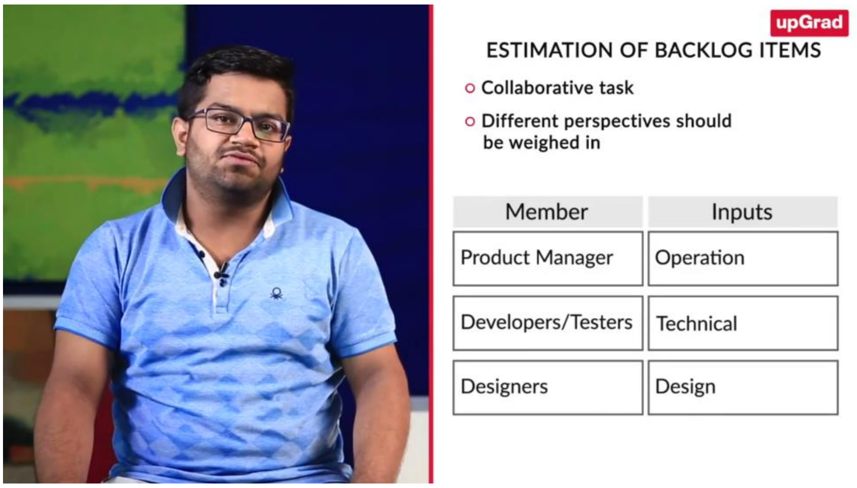
Having said that, there is no right or wrong way to do prioritization. It is completely situation based. The parameters that are generally considered are, what is the impact of this story? What is the effort to complete the story? Generally, low effort and high impact stories are given more priority than high effort and low impact stories.

For example, a small work, which is hurting your sales might be a low effort and high impact, and creating an additional page of content to drive more sales is a high effort case. Is this feature affecting masses or is it for elite few who pays a major chunk of your revenue?

What is the risk associated with not implementing this story? What is the opportunity cost? Is there a defector bug or a hotfix? Hotfixes are given the highest priority. Once the tentative prioritize list is ready, you present the same to the team to do further grooming.

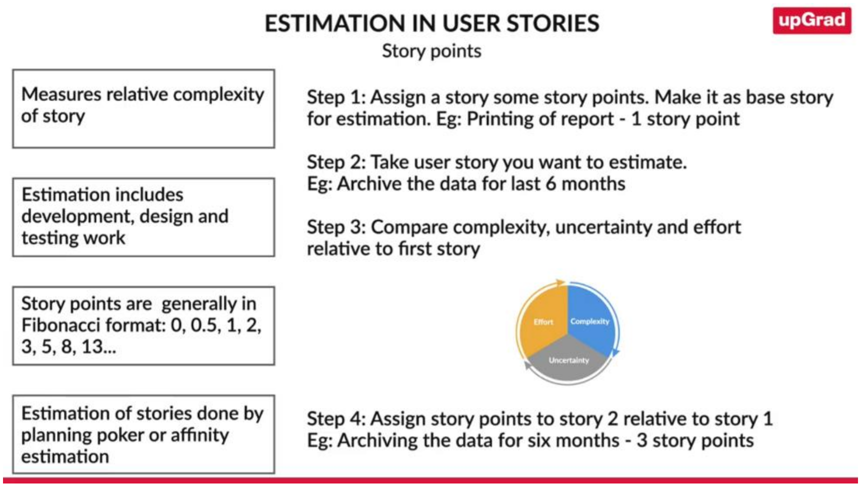
So, in this video, you learned about the steps leading to prioritizing backlogs by considering various parameters like impact, effort, business value, and feasibility. You learned the Moscow method to aid you in generating a priority bucket list. In the next video, let's learn about how to estimate the backlog items.

Once you figured out how to prioritize the backlog items, the next step is to estimate the effort and time required to complete these items. Let's now understand how estimation of backlog items is done.



Estimating user stories is a difficult task involving everyone, developers, designers. Each team member brings a different perspective on the product and the work required to deliver as a user story.

For example, if you as a PM, want to do something that seems simple, like support a new web browser, developers and testers need to way in because their experience has taught them, but might be under the surface. Likewise design changes require not only the designer’s input, but that of developers and testers as well.

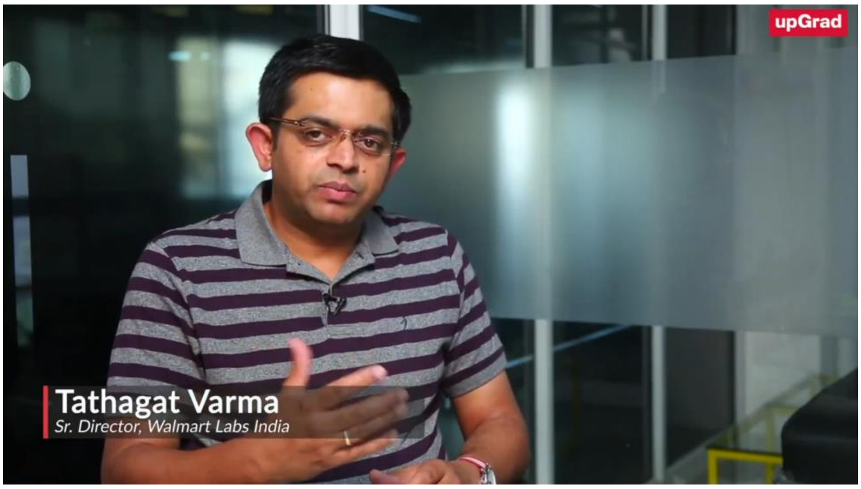


Nowadays, a lot of the agile teams are transitioning to story points. Story point measures the complexity of the story with respect to a story, which is relatively simple. So, the team together decides, let's say printing of the report as one-story point. This decision is taking considering all the development, design and testing work involved.

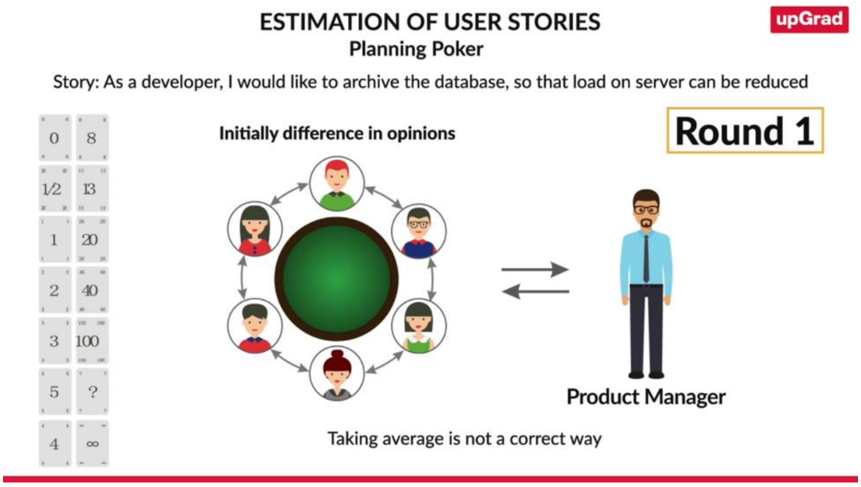
Now let's say the next story is archive the data for the last six months. So, now the story points are given to this story depending upon how complex it is with respect to the printing of the report. Considering the complexity, uncertainty, etc., let's say it is three story points.

Story points rate the relative effort of work in a Fibonacci like form, 0, 0.5, 1, 2, 3, 5, 8, 13, etc. In order to bring everyone on the team on the same page, some innovative games like planning poker or affinity estimation etc, come into channel.

But how can we include team members in estimation? How are these innovative games used to estimate the backlog?



Now planning poker is actually a mechanism in which every team member participates in estimation. So, in a scrum team, let us say we are six of us. And the product owner has explained the story about archiving the database. And we have understood the technical details of what is expected here.



Once it is done, then we will all hold the planning poker cards in our hands. These planning poker cards are essentially like playing cards with a number printed on them as you can see on the screen. The numbers could be zero and half, and then there is a Fibonacci sequence here.

Now all these cards are available to everybody on the team. What do they do? They all look at their cards and after the story has been explained, the draw out their cards, and they flash it together at the same time. Somebody might take out a two, somebody might take out a three, and somebody might take out a five.

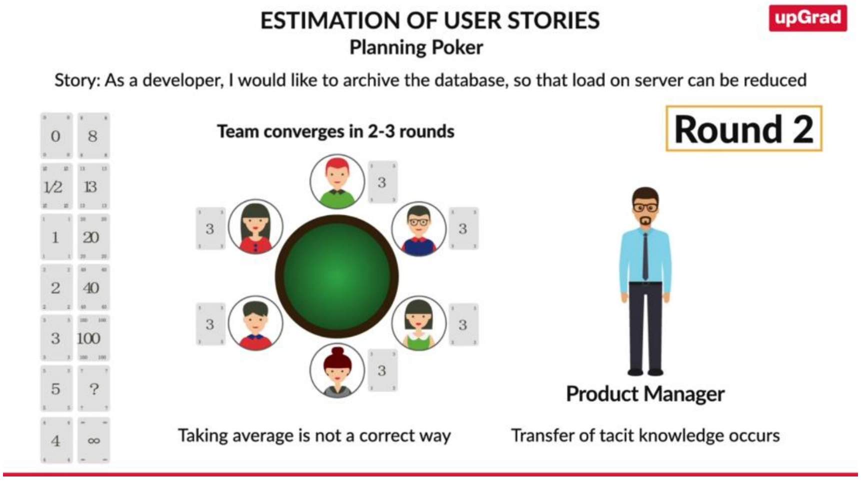
What does it really mean? All it means is the three team members are looking at the story from probably two or three different points of views. And their assessment is that compared to something that we have agreed of unit size one, this story is two times or three times or five times bigger than that.

Obviously, the team has not converged at this point in time. What do you should be the right way to proceed on that? If you thought taking average was a good choice, that's the wrong answer because taking an average is not the right approach here.

What we do instead is we actually invite the people who gave the highest and the lowest estimate and explain their point of view. Maybe the person who gave the five estimation says, I think I will have to write a function to take an archival and we will have to create an offline storage of that and so on and so forth.

And then we hear the person who has given the lowest estimation, who says, yeah, we have the APIs available for us to do the archival. All I need to do is call it and make sure that all that is done there. Now, obviously the reality will be one of these or somewhere in between. Either that API is available or it is not available.

Either ways, the entire team is going through that conversation. And what is happening is, instead of that information being available to one person, we are actually creating a knowledge sharing platform. So, even estimation is actually a platform by which the knowledge is really getting disseminated across the team.

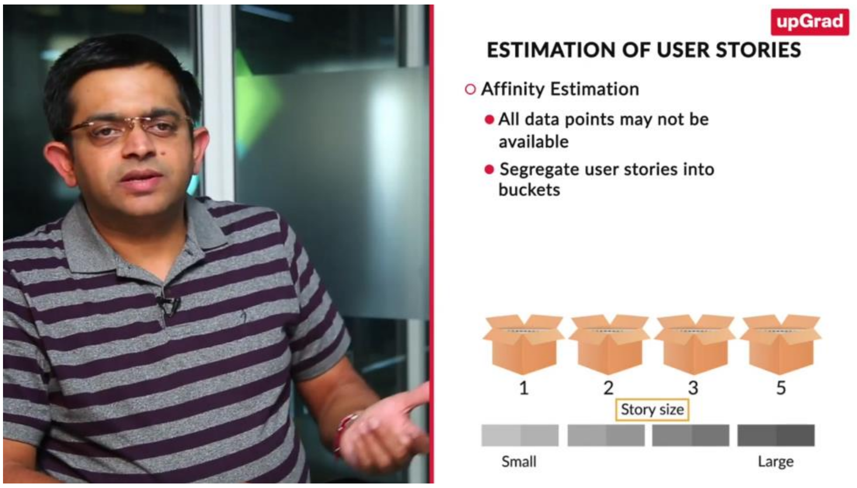


Now the team has exchanged notes and they have clarified the technical details. They once again draw out the cards. It's very likely that there will be a higher conversion this time. Most of the team conversion two rounds. It's very rare to see that in three rounds, but if the teams are still continuing in three rounds, maybe there is some other source of uncertainty that is really causing it. Maybe the teams should go back, do some more analysis and come back there.

So, we have these working estimates, which are good enough for us to get started. Planning Poker is one of the mechanisms for that. And as we keep going out, we will keep adapting our work there. So, that is one way of doing it.

In this video, you learned how estimation could be used as a communication platform to enable the team members to be on the same page. Let's now dive into stories.

You have seen how estimation can be done using poker game, but what if there are many user stories to estimate, let's learn how to estimate user stories using the affinity estimation.



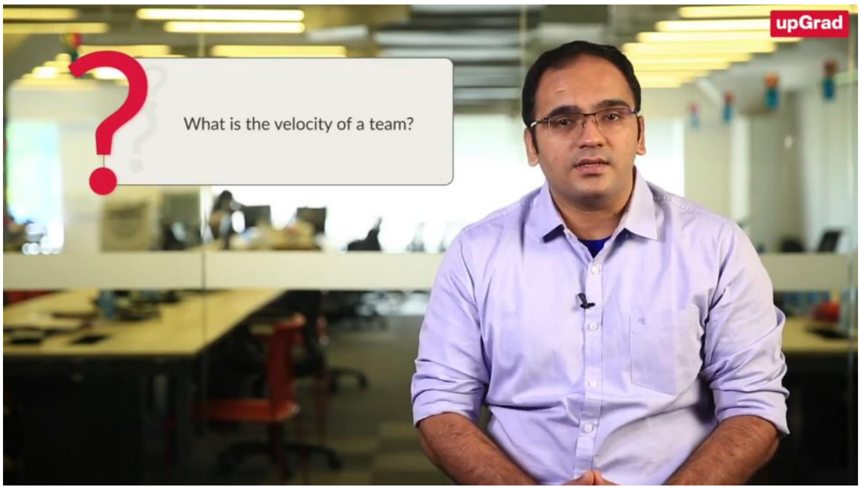
Let us say you are starting your discussion for the first time and you have like 85 stories here. You cannot keep doing the planning poker for all 85 stories. It will take you more than a day just to probably do that thing. So, is there a way that we can do it in a much faster manner?

There is one more mechanism which is known as affinity estimation. And once again, affinity estimation is again, a very lightweight mechanism to come at some kind of a consensus as a group on what are the smallest stories and what are the largest stories in a very quick period of time.

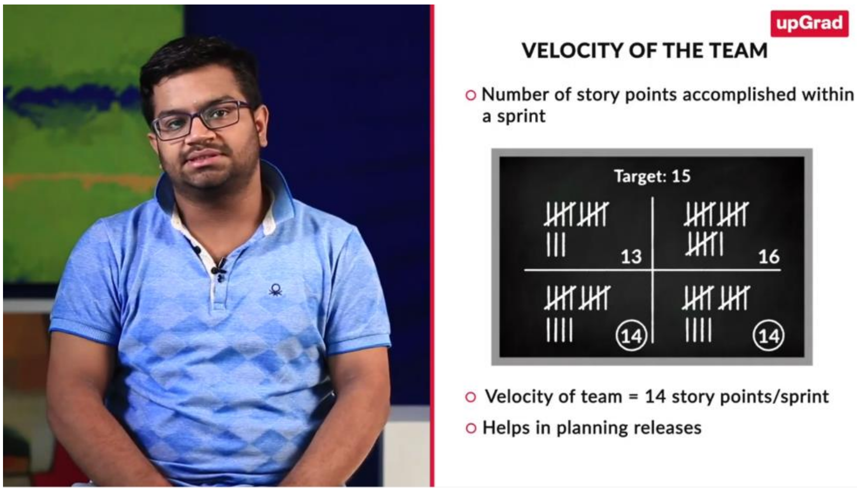
So, what the team does is, they will actually start putting these 85 stories in some kind of, Hey, where do you think it is? Is it a low, is it a high? And they kind of start putting some kind of rough order from the smallest to the largest stories.

Obviously, there could be a lot of, we don't know all the data points. So, somebody might come back and change it and say, no, I don't think this is a large story, I think this is a small story. This keeps happening till the time the team agrees that, Hey, whatever data we have today, I think this particular pattern is very clear. These are the smallest stories and these are the largest stories.

And then the team might actually start putting them in the buckets of their story size and say, Hey, which one do you think is a story size one or two or three and five, and so on. And that is one more very fast way in with just a matter of couple of hours, the team can come up with a quick and dirty estimation for a very large number of stories.

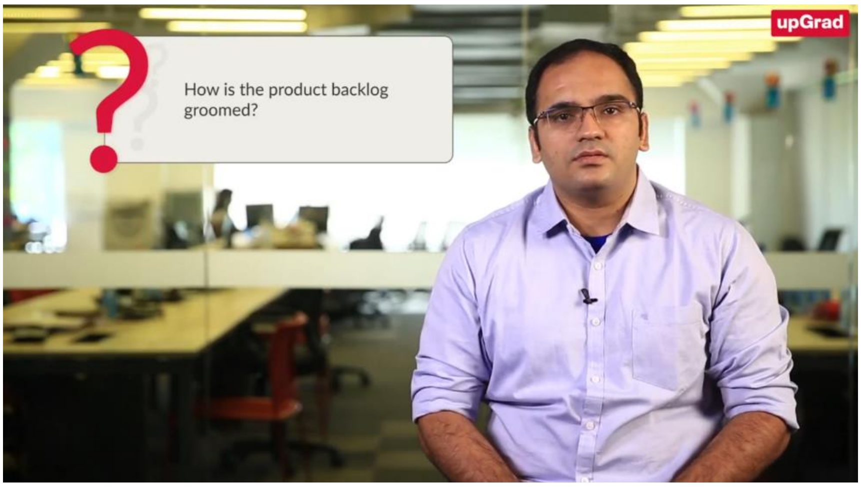


Now that you have the story points for all your stories, it's time to figure out the velocity of your team. But first let's understand what is velocity.

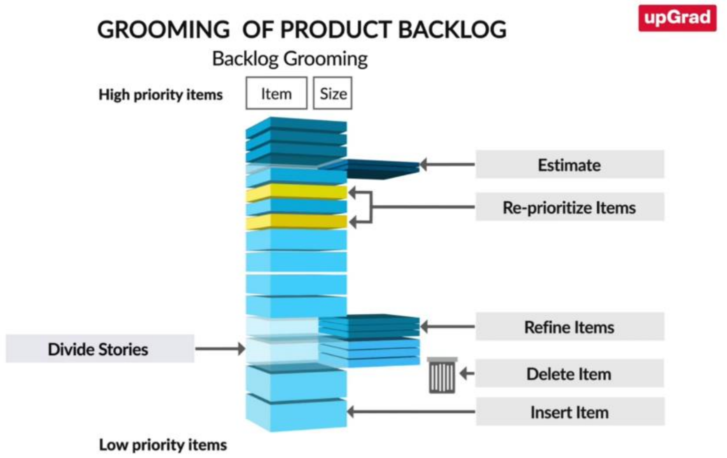


It is the number of story points a team can complete within a sprint. How do we know the velocity of the team? This is done by experimentation. First, the team might take up some 15 story points in a sprint. Let's say they finished up 13 story points. In the next sprint, they finished something like 16 story points.

Over a period of time, the team reaches a point where they can complete, let's say 14 story points per sprint. So, that is the velocity of their team, 14 story points per sprint. Knowing the velocity of team helps a product manager to assess and plan the releases.

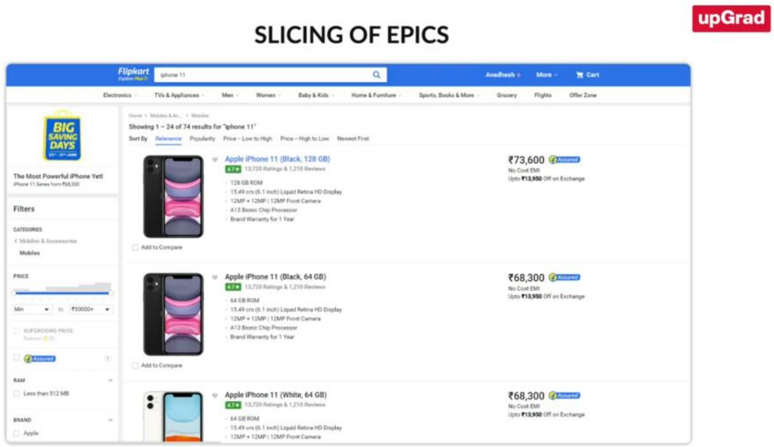


We have heard about backlog grooming in the previous module. Let's see how a product backlog is groomed.



So, generally a product backlog may contain stories or epics, or sometimes even feature level items. These are released in the order of priority and stories with highest priority are refined more into smaller user stories and more details are added, so that they can be easily estimated.

Backlog grooming happens with the entire team at one place. The highest priority items are estimated reprioritizing, deleting or inserting new items can be suggested, and large epics are divided into smaller stories.

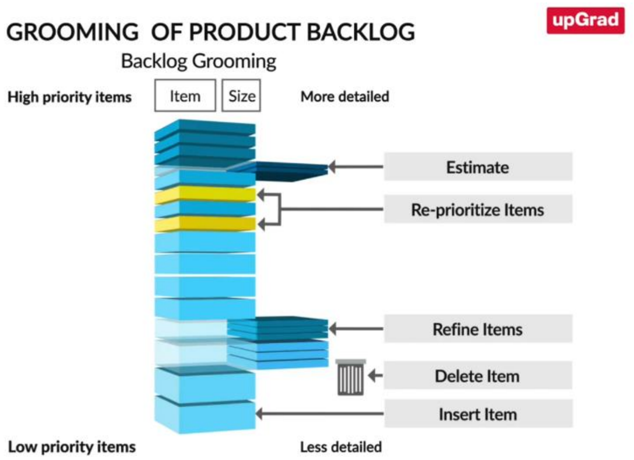


For example, if we consider a search page for an eCommerce company as an Epic, then we can have multiple stories.

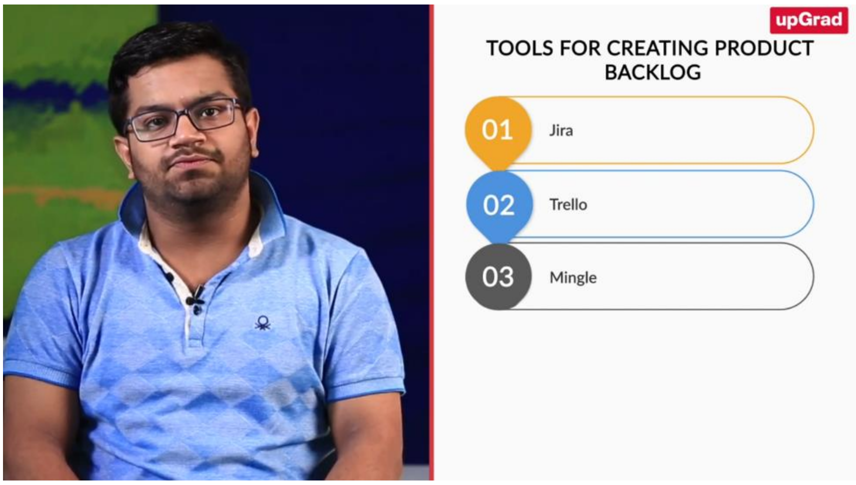
![upGrad logo]

One story could be that as a user, I should be able to do a free text search so that I can find my items easily. Another story could be that as a user, I should be able to sort my results by price or relevancy so that I can easily find my particular product.

One more story could be that as a user, I should be able to filter out the results based on category so I can see the relevant products. And one more story point could be that, as a user, I should be able to filter my search results based on the current and ongoing offers so that I can get the best deal on the product which I am going to purchase.



It is a continuous process for which a PM is responsible, but inputs are taken from the team in order to achieve this. Once the highest priority items are sufficiently detailed out and presented to the team, these items can be taken one by one to work in progress or a bunch of items are moved to sprint backlog in case of scrum.

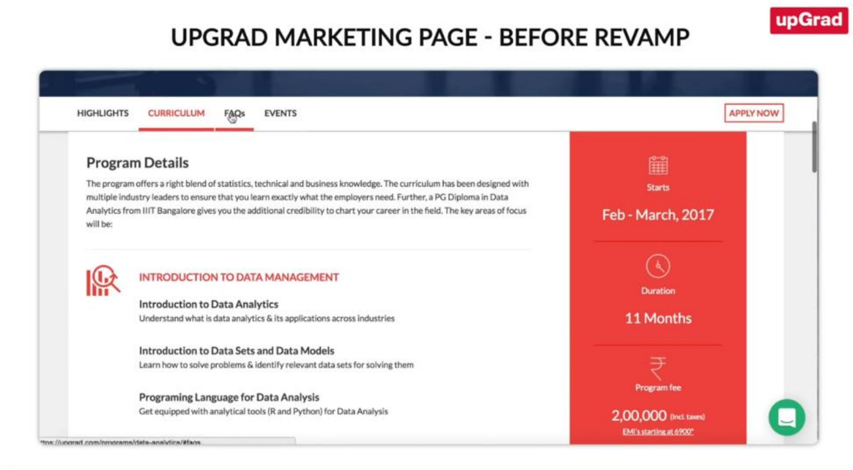


There are many online product management tools available like JIRA, Trello, Mingle, etc. Whatever might be a tool, a clear priority and estimation on each of the backlog items, which the team will work on soon should be clear.

In this video, you learned how affinity estimation can help in estimating many user stories at one go. You also saw how a product backlog is groomed with all the moving pieces, taking into consideration feedback from users and market. In the next video, let's see how a product backlog is created, taking any feature or product into consideration.

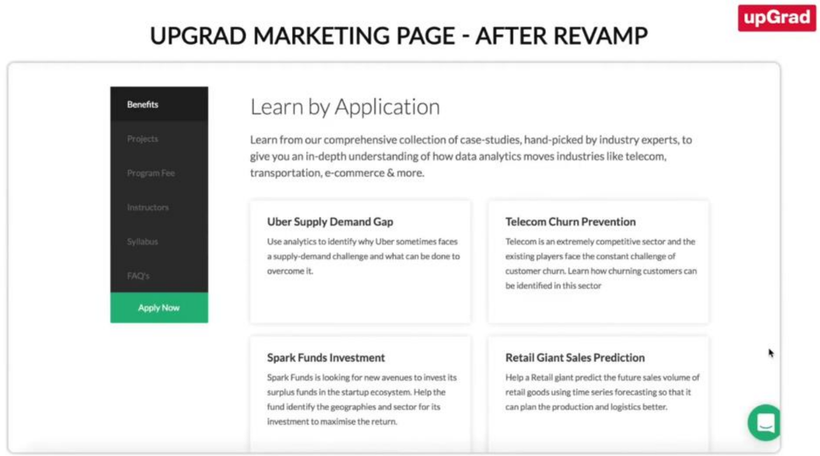
We have seen what product backlog is, how the backlog items are represented in terms of user stories and how we should go about prioritizing and estimating the items in a backlog. Now, let's see how a product backlog is created. What is the thought process behind it?

So, let's talk about this scenario where we recently revamped our course landing page on the marketing website, upGrad.com.



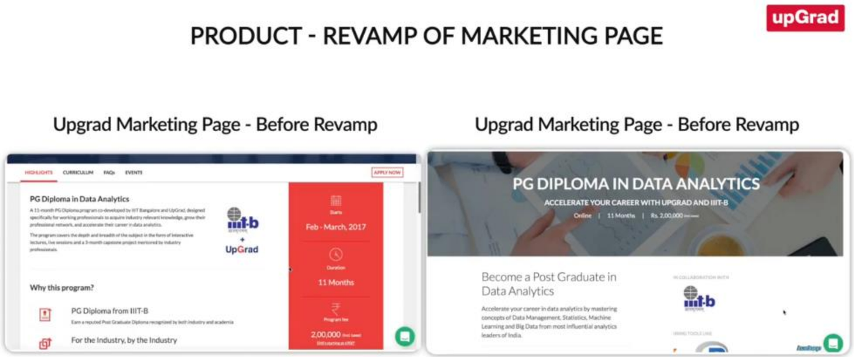
So, initially our course pages used to look like this. As you can see, we had different tabs for different sections of the information that we wanted to highlight.

The first step kind of had the major highlights of the program. And on the right, we had the vitals of the program in a ribbon. And any other sort of sections of information like the curriculum, like the FAQs, or the events that we were organizing for this program were in different tabular sections.

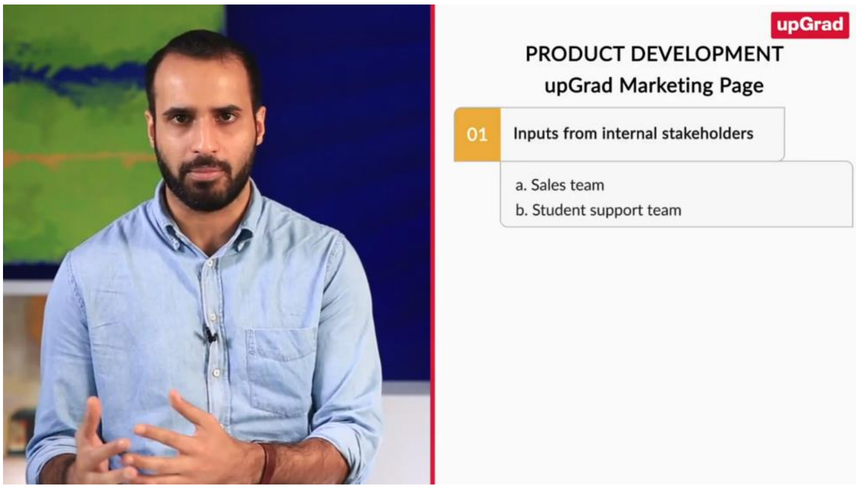


So, we did a major revamp recently, and this is what the new course page looks like. So, when we decided to revamp our course page from the old one to the new one that you see now, we have the following objectives in mind.

We wanted to properly highlight all the advantages on the services that will be provided in the program. We wanted to make the discoverability of information much easier, which is why we changed from the tabs to the infinite scrolled structure.



We wanted to give the students multiple call to actions, to highlight different levels of intent and not just have apply now as the only call to action for the students. We wanted to improve the credibility of the course. You know, given upGrad as a new brand, we wanted to sort of highlight the partners or certification that we're bringing out in the program as well to build that credibility.



So, let's talk about how we actually went about, you know, achieving this change. So, the first thing we did was we talked to our internal stakeholder teams, like the sales team and the student support team. Because these are the teams that are actually in touch with students and applicants every day. So, they can give us like really good information about what the students are thinking and what they're finding easy or difficult to find in this marketing page.

upGrad

INPUTS FROM THE SALES TEAM

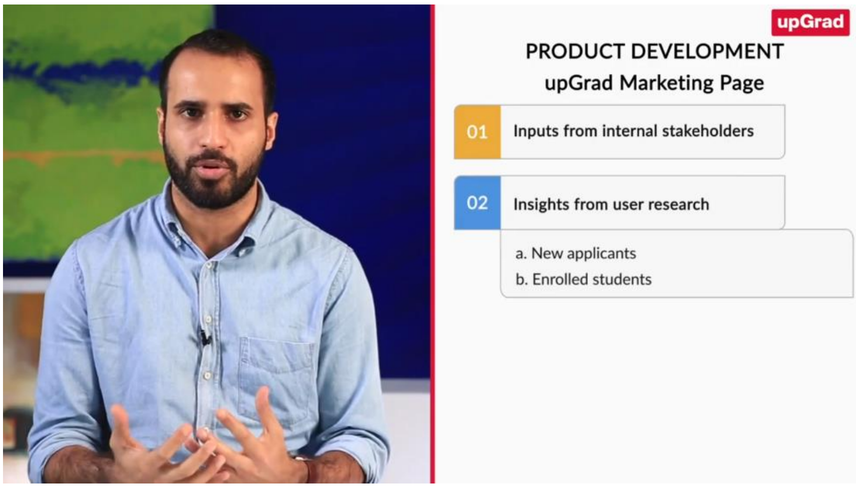
Inputs	Solution
Discoverability of program vitals is less	Highlighting program vitals on page and throughout application process
Only one CTA (Call to action) available	Multiple CTAs

So, let's first talk about what were the inputs that came from the sales team. I think the first, most important thing that came from the sales team was that they were getting a lot of applicants who actually did not even know the program vitals properly. They did not know what the duration of the program was. They did not know what the fee of the program was going to be.

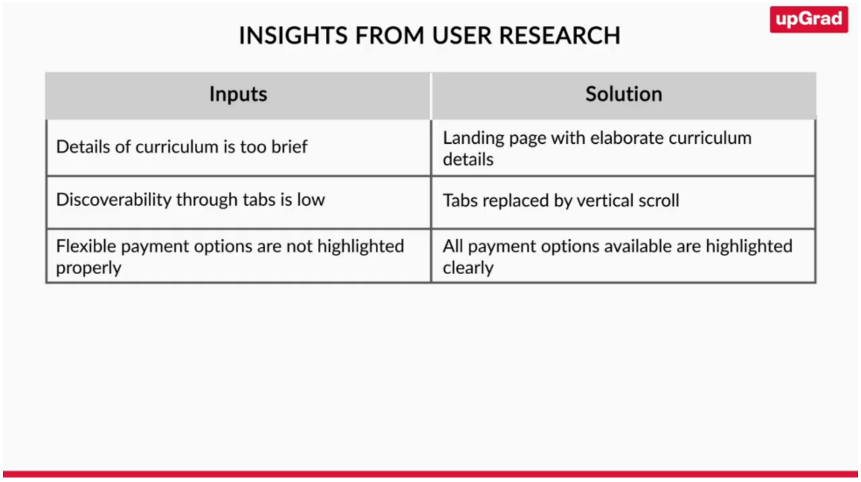
So, they were wasting a lot of time explaining all of this again and again on the call. So, we made it a point to highlight the program vitals like feed, duration, mode of teaching, etc., properly boldly on the page and also during the entire application process so that you get more qualified leads as a sales team.

The second input we got from the sales team was that we only had one call to action, which was apply now. Now, if you're a high intent user who's like really made up their mind to apply, this call to action makes sense. But you might want to do some other steps before actually making that commitment.

You might want to view the syllabus. You might want to attend an info session, etc., to show some intent, but you're not really there that you want to apply right now. And we were just missing out on that pool of students because we did not have enough actions or ways for them to show that intent.



Let's now move on to inputs that we got from user research. You know, talking to people who applied, talking to people who are enrolled in our programs, we understood more about their decision-making journey. And this is what we found out.



Inputs	Solution
Details of curriculum is too brief	Landing page with elaborate curriculum details
Discoverability through tabs is low	Tabs replaced by vertical scroll
Flexible payment options are not highlighted properly	All payment options available are highlighted clearly

The first thing we found out was that there wasn't enough information about what the curriculum of the program was going to be. You know, like what are the modules we're going to teach? What are the tools that are going to be taught in the program? What kind of assessments and assignments are going to be part of the program? These were key deciding factors. And a lot of this information was coming to students through sales calls. Kind of making this point of having a more detailed landing page redundant.

The second thing that students or we realized actually that the discoverability of the content was not easy. When we did actual user testing and contextual test sessions, we realized that a lot of people could not find the curriculum or the FAQs in the tabular structure that we'd made.

They would just sort of scroll through the first section of the page, which is kind of the normal behaviour now and completely miss out on that additional information, which could have been very, very helpful to them in making that decision.

The third thing we realized through these user testing sessions was that for this example, that we're mentioning a data analytics program, the 2-lakh fee upfront was a big sort of daunting commitment for someone. So, for a high price program, we needed flexible payment options.

Now we had flexible payment options, which the sales team would kind of highlight to the students, but it wasn't coming out properly on the page. And that could have sort of, you know, switch the balance in favour of someone going for the program or someone not going for the program. So, that's one important thing we realized as well.

upGrad

INSIGHTS FROM PRODUCT ANALYTICS

Inputs	Solution
Marketing video watched by prospective leads	Highlighting video on landing page

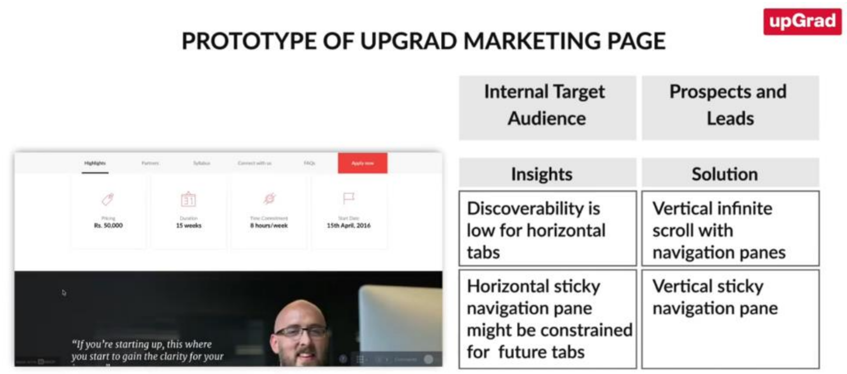
Another thing we realized, you know, looking at the data was that people who had watched the marketing video, were much more sort of likely to apply for the program. Again, I think, as there's enough research, you know video is a great form of content consumption.

In that two-minute video, you can really bring out the learning experience and highlights of the program. And I think we hadn't done a good enough job on that video or of highly highlighting the video on the page. And looking at the data, we realized that video could be a really good trigger to sort of get someone to apply for the program.

So, after we learned all of this and discovered all of this through inputs from our sales team, through the user research, then we went on to, you know, figure out solutions to address the problems.

We have seen how the team conceptualized the revamp of the market page and how they conducted user research, approaching target audience and figured out the user needs and pain points.

Let's see how Ravi and the team incorporated the inputs from user research into developing the product.



So, taking into account these inputs from our user research, as well as from the sales team, we went ahead and developed our first prototype. You can see the prototype on your screen now. What we then did was we took this prototype and we conducted usability testing sessions with this prototype.

We first found people internally in our company, because we had a lot of people from the demographic who would be applicants to a bunch of our programs. Then we also did usability testing sessions with other prospects and leads, who were interested in the program.

After doing a bunch of usability testing sessions on this prototype, we figured out further changes that needed to be made in the information flow, along with the sections that needed more detailing and more highlighting.

So, I think after a number of iterations to this prototype, these were the major changes we did between the prototype and the final version that we have right now. I think the first thing was making the navigation easier in the page.

Even though in the prototype, we had made the horizontal tabs kind of, much more visible, because our hypothesis earlier was that people were just not being able to view the tabs.

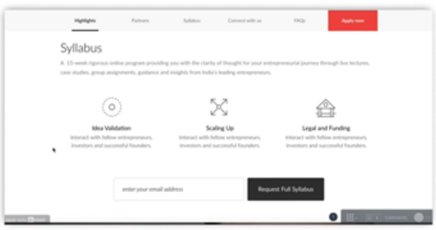
But I think as we did our testing sessions, we realized that, you know, the behaviour in the web has all moved towards one vertical infinite scroll. And I think just expecting different tabs within a page was not something that users were used to. And a lot of people were missing out on this sort of vital information as such.

So, what we decided was to move away with the horizontal tab structure completely, and just go to a long scroll, different vertical tabs, for different pieces of information.

We also changed the navigation structure to make it sticky, vertical on the left side. I think that kind of helped us as well because during this usability testing sessions, we realized that different programs needed different sections to be highlighted.

And having a horizontal kind of structure was really constraining us in terms of, you know, how could we categorize the information such that it made more sense to the user and the number of subsections we could have. I think having a vertical navigation just made it easier for us as well, to mark out the sections, which we felt would be most intuitive to the students and highlight the same, in the same manner.

PROTOTYPE OF UPGRAD MARKETING PAGE



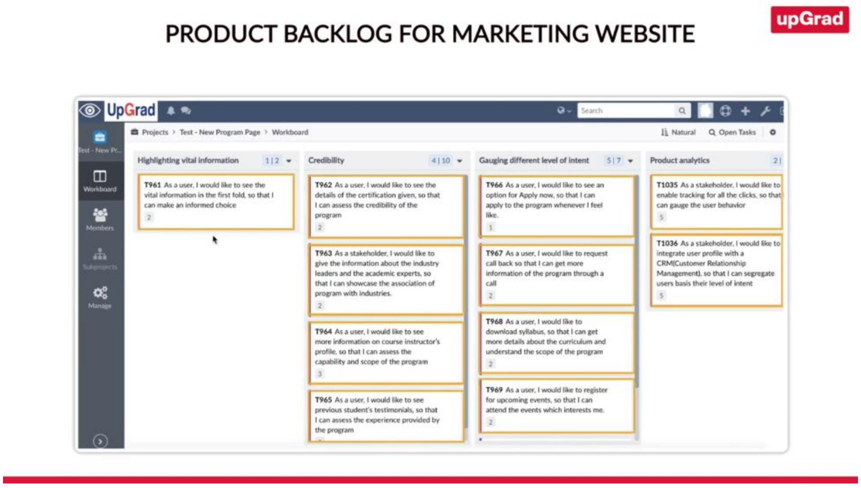
Internal Target Audience	Prospects and Leads
Insights	**Solution**
Discoverability is low for horizontal tabs	Vertical infinite scroll with navigation panes
Horizontal sticky navigation pane might be constrained for future tabs	Vertical sticky navigation pane
Highlighted syllabus is not detailed enough	Detailed syllabus highlighted upfront

I think the third thing that we realized also was that the way we had highlighted the syllabus was not detailed enough. You know, our hypothesis again, was that people don't want to get into the weeds of the syllabus.

The first thing they're looking at the program, we'll give them an overview and then an option to sort of, you know, download the detailed syllabus if they want. What we realized was that actually even in the first view, this was a big decision-making criterion for students, that they really wanted precise details on the syllabus who was teaching, which module, etc., because that gave them the sense of comfort and credibility and sense of exhaustiveness in the program.

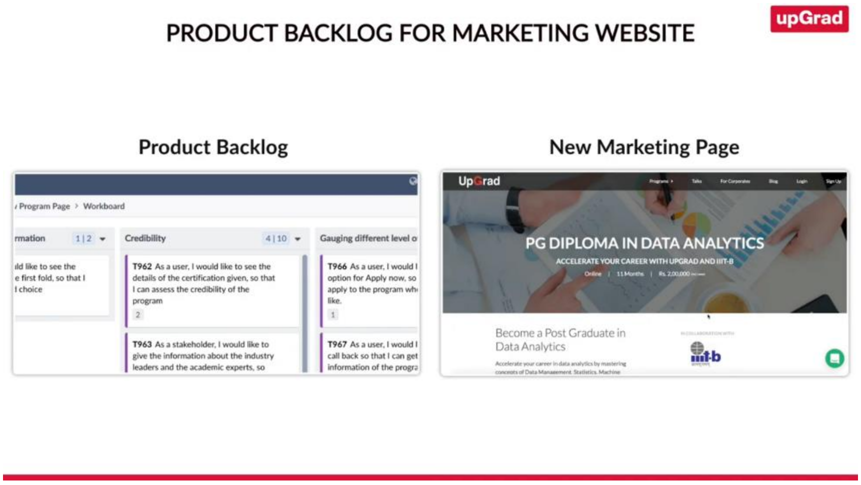
So, I think these were the three major things that we realized in the user testing sessions with our prototype and, you know, accordingly we made those changes into the final version.

So, now all we needed to achieve, you know, this redesign page revamp, we placed it all in our product backlog. I think you can see the product backlog we had for our marketing page revamp right now on your screens.



Let me just talk about a couple of epics and their user stories so you get a better idea. I think the first Epic we'll pick is information flow. So, the most important user story there is that as a user, I would like to see all the vital information about the program in the first fold itself, so that I can make an informed choice, whether I want to sort of go ahead with researching further or not.

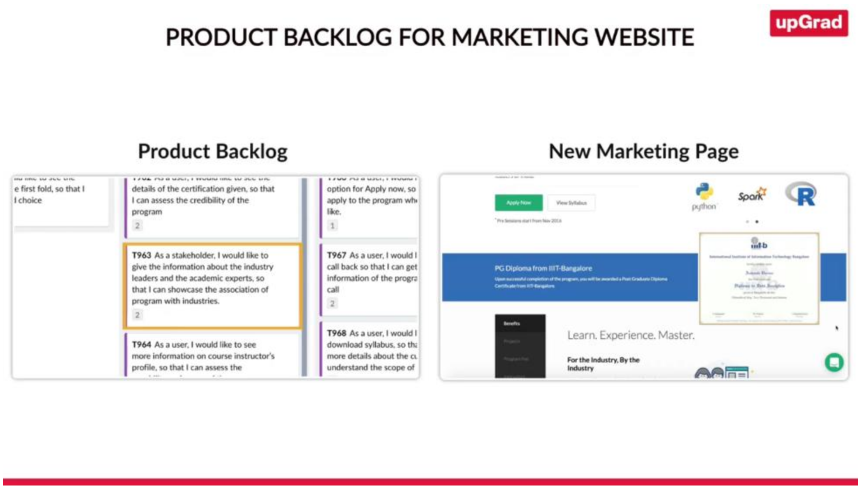
I think kind of breaking this into layman terms, what it meant was the important information, like duration, the price, and who is the sort of certification granting body of the program, we kind of wanted to put it in the first fold.



Let's pick up the next Epic, which is credibility of the program, and talk about a couple of user stories there. So, the first one there is as a user, I would like to see the details of the certification given so that I can assess the credibility of the program.

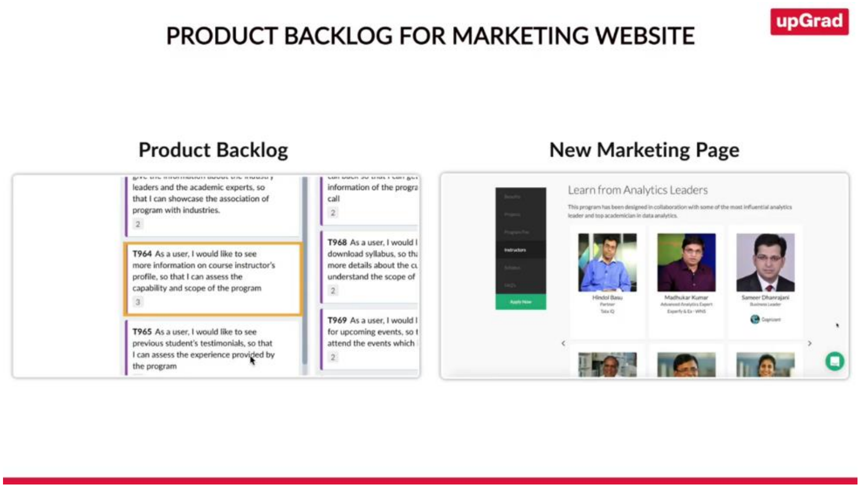
Again, kind of diving deeper into it, upGrad was a new brand. You know, we were still building our credibility in the market, especially for our data analytics program as you see on the screen, you know, it's a long 11-month program actually, and a pretty high price program.

We were doing this in collaboration with a university, which is IIIT Bangalore. So, I think we really wanted to highlight this to the users that you're getting a full postgraduate diploma program. And then the way we did it, as you see, is by having visual of the certificate. That just gives a sense of credibility to the user that, okay, this is what I'm going to get out of the program at the end.



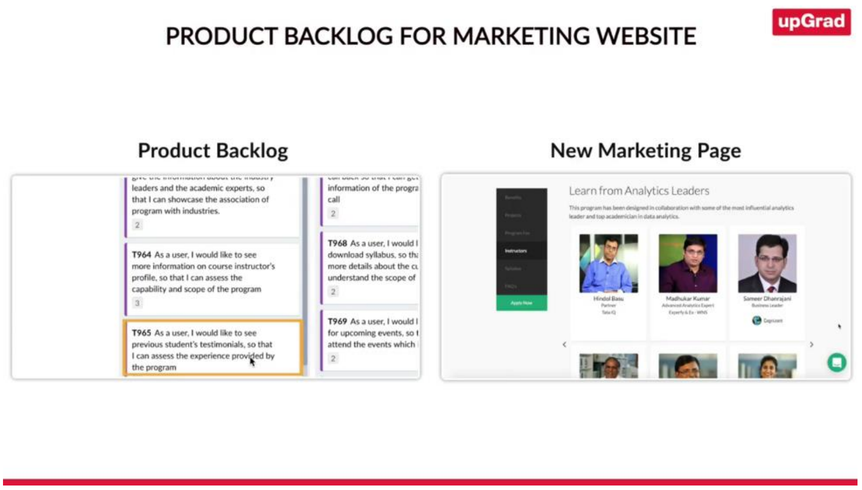
As a stakeholder, I would like to give more information about the industry leaders and the academic experts who are involved in the program, so I can showcase their association, you know, with our program. Now you notice how we wrote this from a stakeholder's perspective and not from a user's perspective.

So, a user is interested in getting to know more about the credibility of the program. We know from our research and past experience that the kind of industry and academic experts you're associated with, goes a long way in establishing that credibility with the user. So, as upGrad, as the stakeholder, we wanted to make sure that we are highlighting this, even though the user might not be explicitly looking for it in the first thing.



Another story in this Epic was that as a user, I would like to see more information on the instructor profile so that I can assess the capability and the scope of the program.

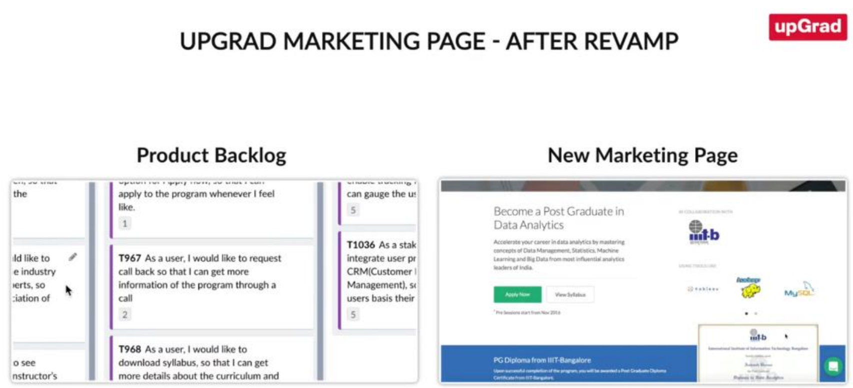
Now you will notice it's a little different from the previous point, right? That was more about the industry experts. Here, from a user's perspective, what we realized was that users really interested in knowing that what is the profile of the people who are actually teaching. I think that was something that was important to the users and that's something we put it as a story that we want to highlight.



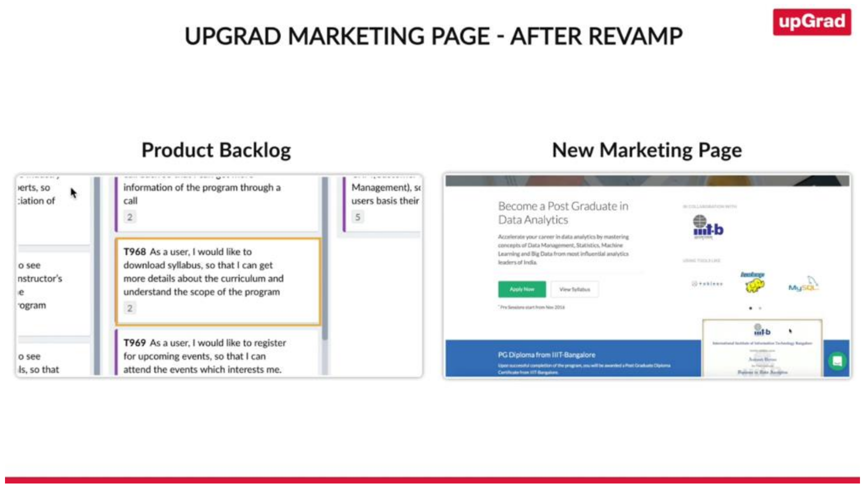
Another story in this Epic was that as a user, I would like to see previous students' sort of testimonials or reviews, so that I can get some social proof about the experience provided by the program. Again, I think all of you will know there's enough research that giving social proof, giving sort of third-party validation, that's what a lot of users look for and react positively to. And that's why we had put that, you know, as part of one of our stories.

We have seen how developing a prototype and conducting usability testing on that help to figure out whether the assumptions were right or wrong, instead of developing the entire product. So, here the prototype acted like an MVP. We have also seen how user requirements I mentioned in terms of user stories or epics.

Let's see how the backlog is created and how it is prioritized.

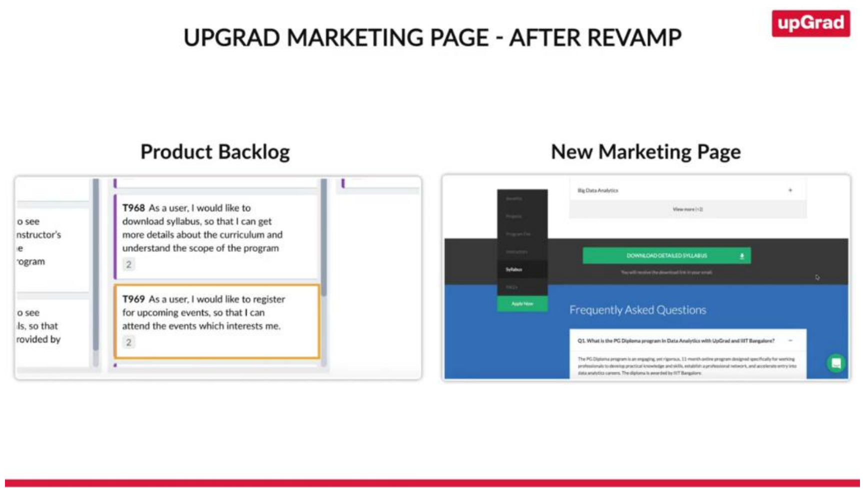


Let's talk about another Epic, which was gauging different levels of intent. So, the first story there was, as a user, I would like to see an option for apply now so that I can apply to the program whenever I feel like. I think that's kind of the most intuitive and the most common, right. I am a user I'm interested in the program. I just want to go and be able to apply. That's why I need a prominent call to action for apply now.

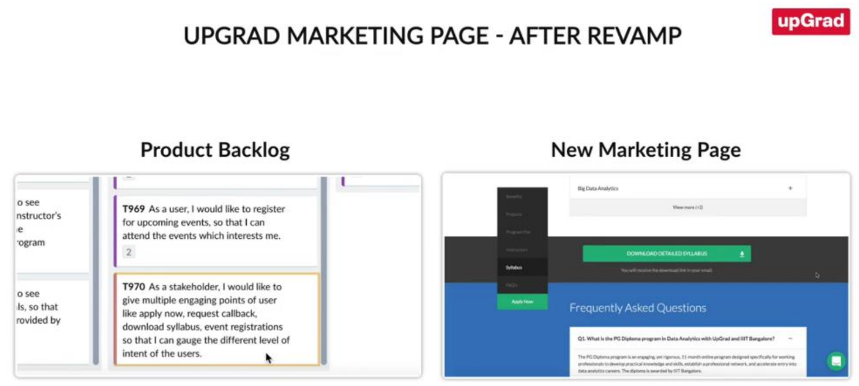


The next story there is, as a user, I would like to request call-backs so that I can get more information of the program through a call. Note, how this relates back to, you know, what we talked in the beginning of the session that people have different levels of intent.

There is someone who wants to apply directly, but there are a lot of people who still want to get more information. And I think the most comfortable way or the way that people are most comfortable with in doing is just getting on a call and talking with the program team. So, we wanted to give the users the ability to do so easily.

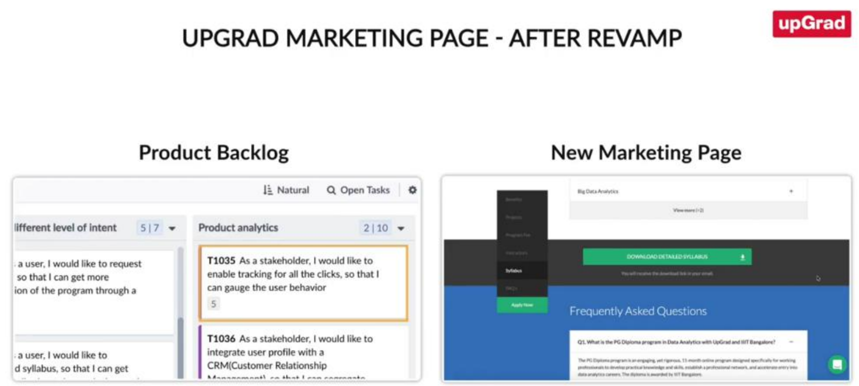


Another story in this Epic is that as a user, I would like to download the syllabus so I can get more details about the curriculum and understand the scope of the program. Again, sort of going back to, you know, what had come out in our user research, that the depth of the syllabus, the curriculum who's covering, it was a major decision-making factor. And we wanted to give an easy option for our users to be able to see that in more detail.



Another one in this event is that as a user, I would like to register for upcoming events so that I can attend the events, which interest me. Think again, I'm someone who's not really there and applying now, but I want to know more, I want to interact more with upGrad, with the program team to understand, you know, what you guys are all about.

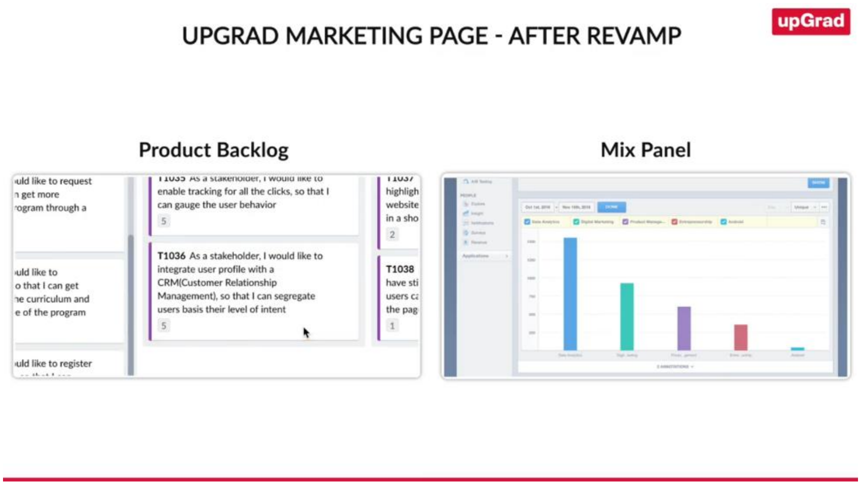
And for that we used to organize, we still organize some online, some offline events. And I think user just wanted to have a clear picture of what are the events you're organizing so that I can go and apply to them.



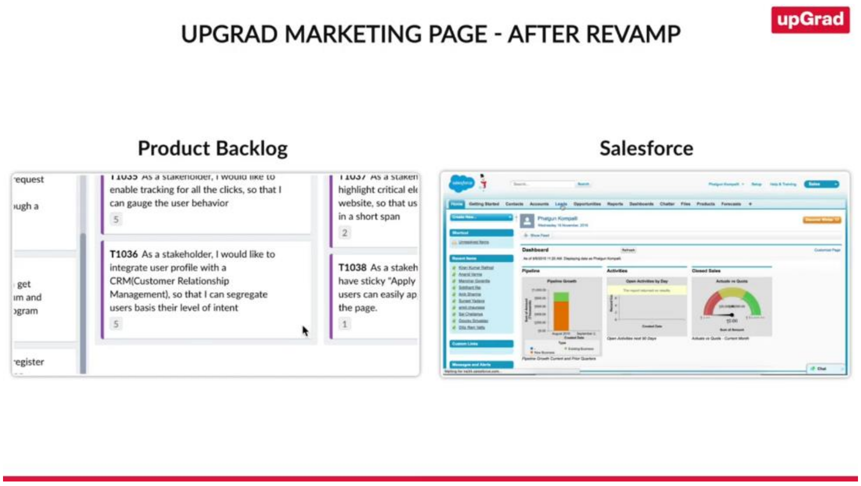
Now synthesizing all of this from our perspective, the story says that as a stakeholder, I would like to give multiple intent points or call to actions for users like apply now, like request call-back, like download syllabus, like event registration, so that I can gauge the different level of intent of the users.

So, for any sales team, it's important to know whether, you know, this particular leader, this particular prospect, where are they in the priority order or in the intent order.

So, all of this were strong indications for us that who is interested to what level in the program. Someone who's making the commitment to come to an online or an offline session, is showing more intent than just someone who's requesting a call back. And I think that's how it helped us kind of prioritize who should we engage with first.



So, another thing that we kept into consideration while building the backlog for this change was what kind of analytics and tracking, we wanted to include in the page. So, the first thing we did was to enable tracking of all the events on the page, you know, in accordance with the naming framework of the tracking tool that we use. At upGrad, we were using Mixpanel. And so, all the kind of user events or the user properties were tagged with that framework.



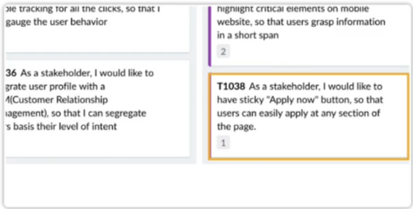
The second thing that we wanted to do was integrate all the call to actions with our CRM. So, CRM, for those of you who don't know is a customer relationship management tool. And at upGrad, we use Salesforce. It's used by companies to keep a track of all their prospects and leads, and also the level of intent that they've shown.

So, if someone would kind of view syllabus or someone would, you know, register for a webinar, we would kind of tag them as Salesforce, as high intent leads. If someone would actually start an application, that was like super high priority lead.

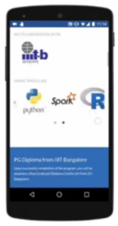
If someone would have just viewed the page and just kind of just signed up, but not really shown any of these intent actions, we would show them as an interested prospect. So, we had different levels of tagging in our CRM and that sort of, that score or that tag had to be auto determined based on what actions the user is taking on the page.

UPGRAD MARKETING PAGE - AFTER REVAMP

Product Backlog



Mobile Experience



I think the third thing was the mobile experience. I think, what we realized was we had just made a responsive version of this page, but it was really making our mobile site very heavy. So, what we had to do was we really took out non-essential information from our mobile site, we had the data that our mobile session duration was sort of 30 to 35% less than our desktop session duration.

And I think we just had to take that into account that people are spending less time on our mobile site. We should take out all nonessential information and just show people the critical elements of the program, you know, so that the mobile experience is kind of smooth.

And the fourth thing we did in the mobile experience was having the call to action button sticky throughout, which is like, it's always visible. Because I think in mobile, sometimes it becomes harder to navigate on long pages. So, just having that call to action button sticky, made it easier for users wanted you to display that intent to kind of go in the next step of the funnel.

So, let me just give you a quick example also on how prioritization was done, you know, in some of these stories in the backlog. So, I think you would have learned in the prioritization section, that first thing to do is sit together as a team, all the stakeholders who are involved in executing the story and assign some velocity or some difficulty points to it, every firm follows their own convention in that.

PRIORITISATION OF PRODUCT BACKLOG

Product Backlog Items	Tasks
As a stakeholder, I would like to highlight academic and industry experts' profiles, so that I can showcase their association with the program	Separate section on the page Vital details of instructors
As a stakeholder, I would like to have expert pages with instructor profiles, past experience, etc., so that I can gain more credibility for the program	Separate page for instructor profiles
As a user, I would like to see past student experiences, so that I can see third party perspective about the program	Student testimonials Involves collaboration of marketing team, student support team, production team and students

But let me talk about three kinds of stories here. The first story was that as stakeholders, we wanted to highlight the academic and instructor sort of experts who were involved with us in the program. So, that involves showing the profiles in a section, in a separate section on the page, you know, showing kind of vital details of those people who are involved.

The second was detailed instructor profiles. And I think the way we had kind of tried to execute this was having a different page for each of the instructor, which talks in depth about what have they done, in their past sort of experiences and how are exactly they are involved in the program.

And the third thing was student testimonials, where, you know, we wanted to highlight student stories, like how has this program actually impacted someone? Where were they in the beginning of the program? How did their career change during the program and where are they now?

PRIORITISATION OF PRODUCT BACKLOG

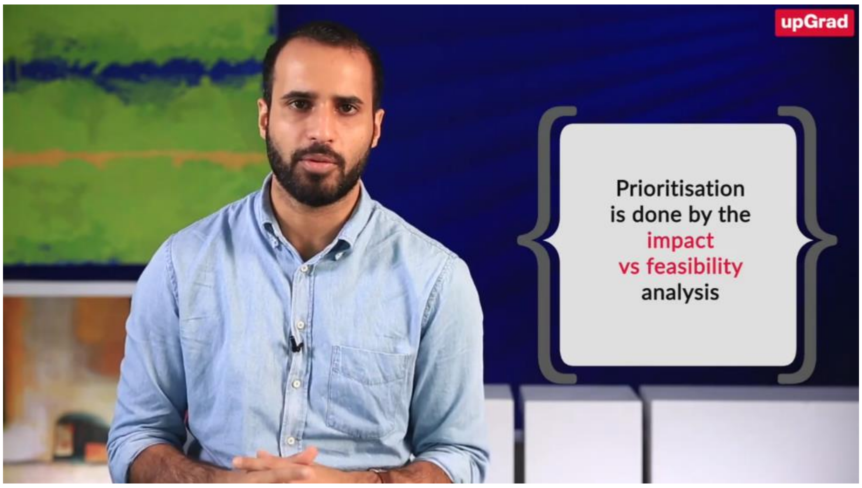
Product Backlog Items	Tasks	Feasibility	Impact	Priority
As a stakeholder, I would like to highlight academic and industry experts' profiles, so that I can showcase their association with the program	Separate section on the page Vital details of instructors	Less complex	Same	1
As a stakeholder, I would like to have expert pages with instructor profiles, past experience, etc., so that I can gain more credibility for the program	Separate page for instructor profiles	Intermediate	Same	2
As a user, I would like to see past student experiences, so that I can see third party perspective about the program	Student testimonials Involves collaboration of marketing team, student support team, production team and students	Most complex	Same	3

So, this third actually involved inputs from the marketing team, from the student support team, etc. It involved collaboration with the production team to actually go and shoot, you know, because we wanted to put out video testimonials. So, I think in terms of feasibility, this was definitely the most complex story out of these three.

However, in terms of impact, we still believed that it would have equivalent impact. So, I think that's why clearly for us, this story got to a later release because we needed important work from all these different stakeholders. I think between the first two, very clearly the first one went into the first release as a higher priority because, you know, we were creating a section there that, where we wanted to highlight our academic and industry partners.

And all that was needed was to put out their image, their LinkedIn profile, their current title, because the objective in the story was to give people a snapshot of the kind of varied industry and academic experts we had on board.

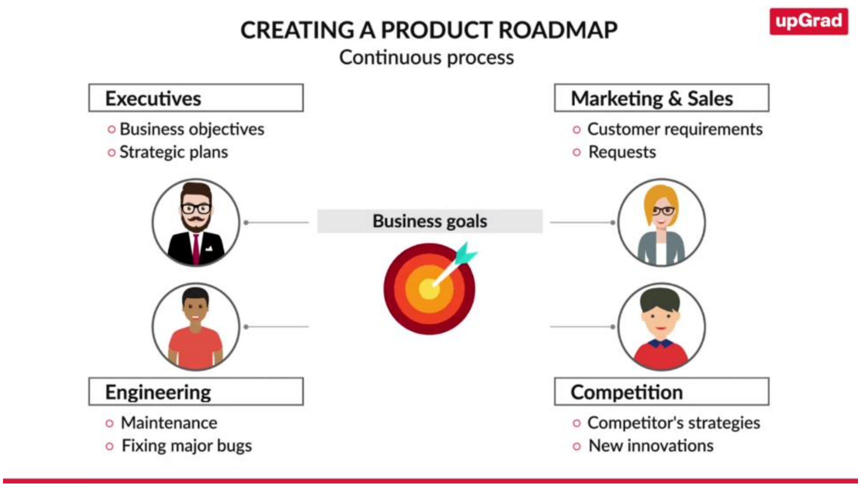
The second story involved, actually creating a new page. Thinking about a new template for a new page, where we would highlight how this instructor is involved in the program. And that kind of required more input from the design side. That's why, in terms of complexity, this story at higher points associated with it. In terms of impact, we believed it would be equivalent.



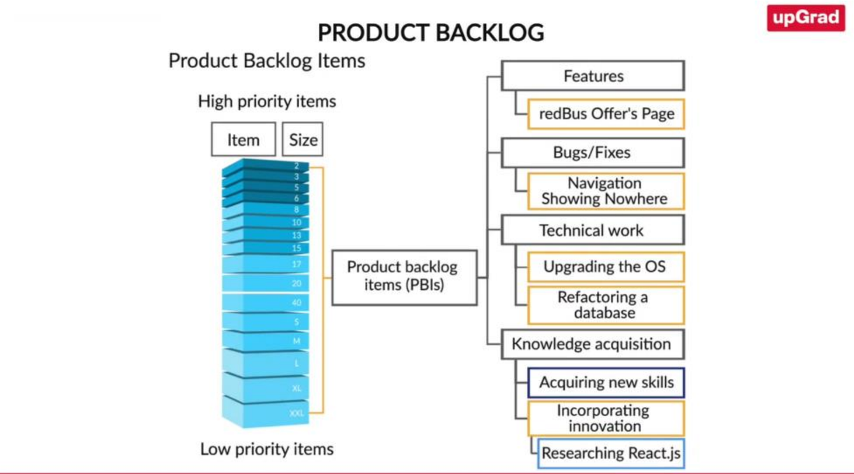
So, we basically used this, you can see that this was not very, you know, complicated or ultra-scientific, this is a basic, you know, impact versus feasibility kind of an analysis in which we prioritize these three stories as one, two and three. And that's how we went about shipping them.

We have seen how epics are placed and how they're further broken down into user stories in a backlog. We have also seen how prioritization is done based on the impact feasibility analysis. So, this is how a product backlog is created. Next up, a quick summary of this session.

That was a very insightful session. Let's do a quick recap. Firstly, you have learned a roadmap is the timeline of features for a product, and also understood how it conveys strategies to various stakeholders. Next, you learned that roadmaps are created differently by different companies and are often subject to change. Later, you have discovered that roadmap health in strategizing your goals and strategizing the features to achieve the vision.

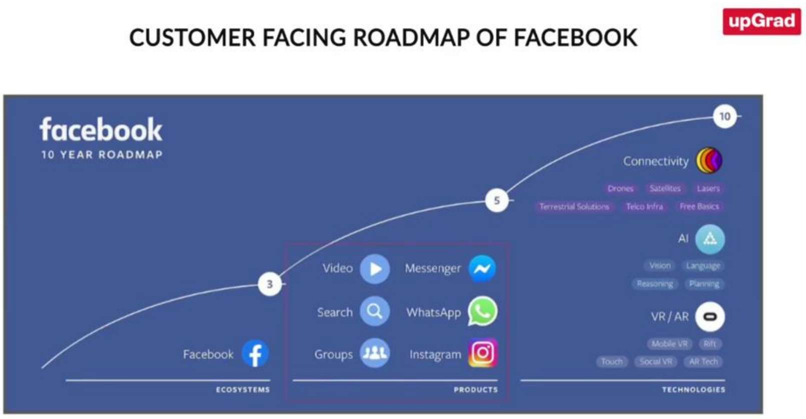


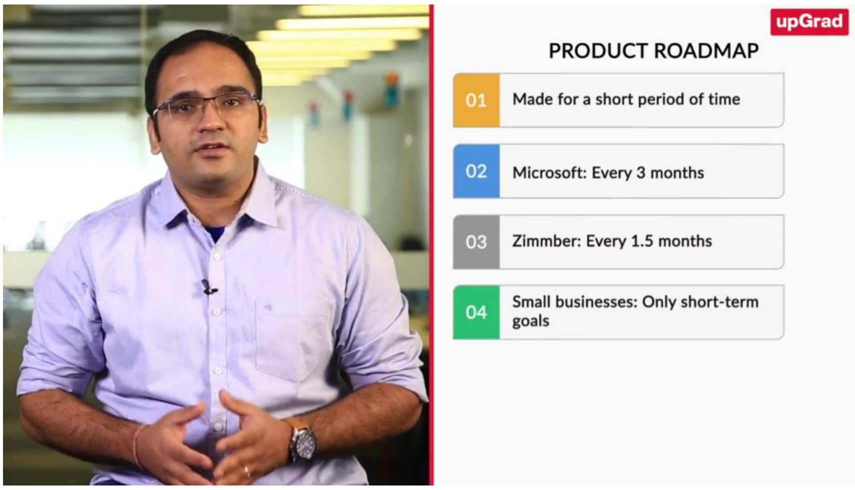
Next, you learned that for creating a roadmap, inputs from various stakeholders, markets, competitors, launches, etc., considered along with the strategy for achieving the vision. Further, we have seen that different companies follow different cycles of roadmaps.



Coming to the product backlog, you have learned that backlog items can be either feature bugs, technical work, or knowledge acquisition work. You also saw how the backlog items could be represented as either user stories or use cases. Later, you understood that not all requests are requirements. So, you need to dig deeper to understand the core problem, rather than just developing whatever stakeholders are asking for.

You saw the Moscow method to prioritize the backlog items and how various other parameters like impact, effort, feasibility, business value, etc., help in prioritizing the backlog. Further, you saw how estimation of backlog items are done in terms of story points, using various methods like planning poker, or affinity estimation. You also saw how the velocity of a team is figured out after some sprints.

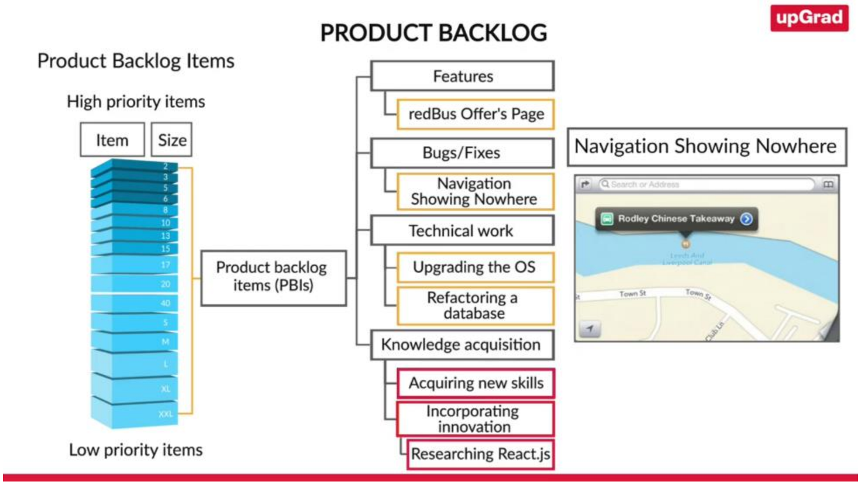




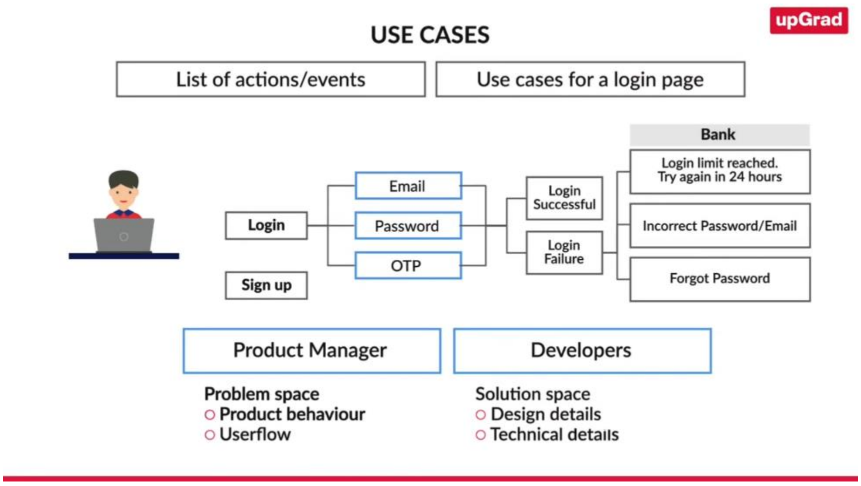
Now that you've gone through the concepts covered in this session, let's recap the examples you saw to understand these concepts better. First, we saw the customer facing roadmap of Facebook and how it looks different from an internal roadmap. Later to realize that a roadmap adapts to change, we had seen how the roadmap of a crowdsourcing company changed when it decided to enter the US market.

We also took the example of the merger of Microsoft and LinkedIn as an occasion where roadmaps must have changed. Next, you saw how roadmap is created at Microsoft and Zimmber and how prioritization is done there.

You also saw that Microsoft creates a roadmap for every three months and small companies like Zimmber creates a roadmap for every one and a half month.



Further, to understand different types of backlog items, you have seen the offer's page of Red Bus as an example of a feature and a map showing nowhere as an example of a bug. Technical work can be upgrading all developers, machines and knowledge acquisition can be figured out whether a programming language works for our system or not.

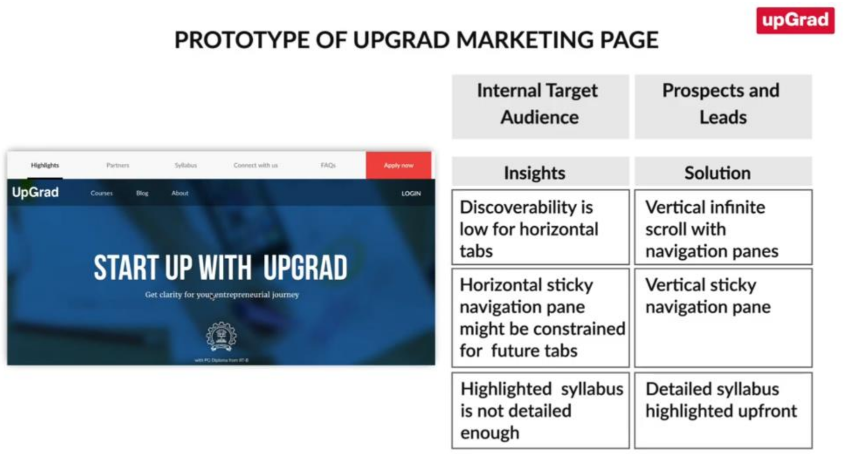


We also saw how these items could be represented as user stories. Next to understand the use cases, you saw the example of the login page and several use cases for the feature. To distinguish user requests from user requirements, you saw the example of the request for an additional graph on CRM page. But how the requirements there would be a filter for selecting variables to create graphs and reports.

To understand Moscow method, which is used to prioritize the backlog, we saw the example of a critical bug as a must have. An informative video as a should have, personalised welcome as it could have. And some minor changes in terms and conditions as a would like, but won't have.

Lastly, we have seen how a product backlog is created for a product and the entire process to arrive at creating a product backlog. You must have understood that whenever you go about building any product or adding any feature in an existing product, you have basic objectives and assumptions in your mind.

Then you should go and conduct in depth user research in various ways like surveys or interviews of the target audience and validate your assumptions. It also helps in finding out user requirements and creating personas of your users. In case you're adding any new feature to the existing product, you can conduct usability testing and figure out the actual need for the feature.



Then you can create a prototype and conduct usability testing on the same before going into developing the final product. This will help in figuring out the user requirements and understand their behaviour.

PRIORITISATION OF PRODUCT BACKLOG

Product Backlog Items	Tasks	Feasibility	Impact	Priority
As a stakeholder, I would like to highlight academic and industry experts' profiles, so that I can showcase their association with the program	○ Separate section on the page ○ Vital details of instructors	Less complex	Same	1
As a stakeholder, I would like to have expert pages with instructor profiles, past experience, etc., so that I can gain more credibility for the program	○ Separate page for instructor profiles	Intermediate	Same	2
As a user, I would like to see past student experiences, so that I can see third party perspective about the program	○ Student testimonials ○ Involves collaboration of marketing team, student support team, production team and students	Most complex	Same	3

Later, we have seen how those requirements are translated into backlog items in terms of user stories or epics, and then finally prioritize based on the impact and feasibility of implementation. This is all about product backlog. In the next video, you will learn about how to create product requirement documents. Until then, keep exploring and have fun.