

How To Use the Kano Model?

WHAT IS THE KANO MODEL?

The Kano model is a framework used to prioritize features on the roadmap based on how likely they will satisfy or delight users. Your team should pull together a list of features to be considered, and plot them on a chart that visualizes satisfaction versus functionality. It points towards how desired or needed a feature can be, or if users are simply indifferent to it.

WHEN AND WHY SHOULD YOU USE THE KANO MODEL?

Whenever your team is considering the list of features to work on for your next release, and would like to figure out what mix is the best. This will also result in the best allocation of limited resources and time.

Kano allows you to work out the right combination of:

- Minimum basic features that must be included
- Performance features to start working or investing in
- Which delight features will impress users the most

HOW DOES IT WORK?

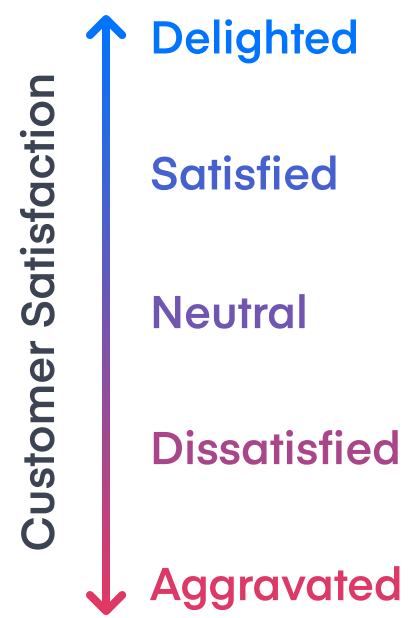
Here's where things get interesting; to identify how satisfied or even delighted users will be with a product, we have to consider the two dimensions (or plotted axes), satisfaction versus functionality. How will users react to the level of functionality of our features?

! Please note

This doesn't always work as a linear scale, as you'll come to learn in the following section, and it's impossible to always stay at the top of the scale.

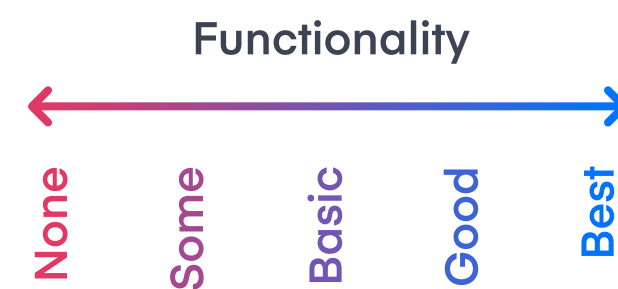
Satisfaction (Y-axis):

The vertical axis measures the level of satisfaction, and it ranges from frustration (or complete dissatisfaction), to delight (or complete satisfaction).



Implementation (X-axis)

Also called investment or sophistication by some, this represents how well we've implemented a certain feature, how much has been invested in its development, or how much of a particular feature the user gets. It ranges from None to Best (or Very Well Done).



THE FOUR “FEATURE BUCKETS”

Looking at the graph we can quickly identify and classify the features into four buckets:

Basic features

Also known as must-be features. These are expected by your users but they won't satisfy them more. Without them, they won't even consider your product. For example, we expect our email to be able to import or look up contacts, or a messenger app to send messages. If they don't have this or the feature doesn't work, users will simply go elsewhere.

We expect these features to be there and work, and therefore we see that as our product team puts more effort or money into making them more functional, our satisfaction grows. Yet it will never reach the positive quadrant. Once it reaches its maximum potential, the team can stop investing effort into it.

Performance features (one-dimensional)

The more of these we give users, the more satisfied they'll be, therefore moving in a linear direction. As we increase functionality, so does our investment. Examples could be: storage space on your cloud service, or faster internet from your provider.

Indifferent

Certain features don't make much of a difference. The user feels indifferent towards their presence or absence. Therefore you should avoid working on these.

Attractive features

Also known as delighters. These are pleasant surprises that the user isn't expecting, but like the name suggests, once introduced they immediately generate excitement. Introducing these features can be what differentiates you. Think about the time Apple introduced Apple Pay from your Iphone, or the first time you were able to collaborate on Google Docs.

These are the kind of features that make you go, “Wow!How cool!”, and if plotted on the graph, it's easy to see how the slightest increase of functionality will rapidly increase satisfaction.

FEATURES CATEGORIES CHANGE

Categories change over time in a dynamic environment, just like user expectations. Our users will change their perception on product attributes in the future. What now seems to be a game-changer to you, might become a standard or expected a year from now. That's why Attractive features will eventually transform into Performance and Basic features in time.

What our customers feel about some product attribute now is not what they'll feel in the future. Attractive features turn into Performance and Must-be features as time goes by.

THE QUESTIONNAIRE

In order to discover customer insights about your product’s features, we must deploy a Kano questionnaire followed by an evaluation of the different combinations.

The questionnaire consists of questions about the feature we’d like to assess, and the questions are termed functional or dysfunctional forms:

- If you have this feature, how do you feel?
- If you don’t have the feature, how do you feel?

The possible answers for these questions are:

- I like it
- I expect it
- I am neutral
- I can tolerate it

! Consider this

In practice, you should also consider adding an extra question, asking how important they consider a certain feature to be.

This answer allows you to distinguish which features are most important to users. It’s a tool to differentiate big from small features and the impact they have on your user’s view on the product.

	Minimal	Low	Medium	High	Massive
Functional					
How would you feel if the product had...?					
How would you feel if there was more of...?					
Dysfunctional					
How would you feel if the product did not have...?					
How would you feel if there was less of...?					

EVALUATE AND PLOT

The beauty of the Kano model is that when we pair up our functional and dysfunctional answers, we uncover how much a feature is wanted, needed or if our users are indifferent to it.

We use an evaluation table to uncover in which categories our features fit, by pairing functional answers with dysfunctional ones in rows and columns.

! Before we proceed

You'll notice that there are two new categories in the table - Questionable and Reverse.

- Questionable suggests that someone didn't quite understand the questions or feature being described.
- Reverse suggests that what we suggest is the opposite of what the user wants.

		Dysfunctional (Feature absent)				
		Like it	Expect it	Don't care	Live with	Dislike
Functional (Feature present)	Like it	Questionable	Attractive	Attractive	Attractive	Performance
	Expect it	Reverse	Questionable	Indifferent	Indifferent	Must-be
	Don't care	Reverse	Indifferent	Indifferent	Indifferent	Must-be
	Live with	Reverse	Indifferent	Indifferent	Questionable	Must-be
	Dislike	Reverse	Reverse	Reverse	Reverse	Questionable

WHAT TO DO WITH OUR RESULTS?

First you need to fully understand the table and what each pairing means in terms of categories.

Questionable: they are contradictory so you'll always see them in a diagonal pattern across the table.

Performance features: these are features that users like having and dislike not having. The more performance features, the better.

Must-be (basic): Users dislike not having them, and when present they range from tolerating to expecting them.

Attractive: Users like having features they don't expect having. The dysfunctional range will therefore go from 'expect it', to 'tolerate it'.

Indifferent features: Will always be in the middle of the table (as in the graph), as users are always neutral or can tolerate them, in both their functional and dysfunctional answers.

THE FINAL STEP

Now you know how each pairing is categorized you'll want to get all your answers together and organize your data to see where the features go. The two approaches to organizing your data are called **discrete** and **continuous analysis**.

If you are new to this and don't have much time, we recommend using the simpler approach- **discrete**.

The discrete approach gathers all of your respondents' answers from the evaluation table. You should then count the total responses per category or demographic, and designate a feature's category based on the most recurrent response. This will allow you to also place them on the quadrant.

In order to prioritize your results, use the following order:

Must-be , Performance, Attractive
Indifferent.

You'll end up with a table like this:

Feature	M	P	A	I	R	Q	Total	Category
Feature 1	10	4	2	1	-	3	20	M
Feature 2	4	13	1	2	-	-	20	P
Feature 3	14	1	4	1	-	-	20	M
Feature 4	5	3	10	-	2	-	20	A



Why we love it

It highlights market fit: fully customer centric, it allows immediate identification of product advantages and weaknesses through its features.

It tailors a product to the needs of current and target users. This allows for predictions about features and audiences based on expectations.



A few downsides

Subject to inherent limitations caused by survey delivery

It solely focuses on customer opinion: this means it fails to account for a level of knowledge about the product and individual bias.

Prone to delayed time-to-market : delays are due to surveying, data collection and processing time.



Centralize feedback and make customer-driven product decisions with airfocus. Join thousands of teams who use our flexible product management platform to build products that matter. [Try for free](#)