

Kano Model

Transcrição

[00:00] O segundo método de priorização que eu vou explicar para vocês é o modelo Kano, "Kano Model". Esse modelo é muito interessante porque ele parte de uma premissa que eu acho muito inteligente, em que o cliente só consegue te dar um pedaço da fórmula do sucesso do seu produto.

[00:20] A outra parte dessa fórmula nós precisamos descobrir juntos com o cliente. Então, ele tem ali essa noção, mas não é algo que ele naturalmente chega e conta. Tem um pedaço que nós temos que conversar um pouquinho para descobrir um pouco melhor.

[00:39] E no modelo de kano ele traça um gráfico que é bem interessante que é de Satisfação por Necessidade. No eixo de satisfação eu tenho o quão satisfeito eu estou com aquela funcionalidade e no eixo da necessidade, que está aqui na horizontal, eu tenho o quanto que a minha necessidade foi suprida.

[01:07] Então, eu dei o exemplo do login na aula anterior vamos seguir com ele para eu tentar explicar. Imagina que nós começamos nesse eixo da necessidade com um login só com usuário e senha. Eu tenho algum grau de satisfação quando eu tenho esse login só com usuário e senha.

[01:26] A medida que esse login vai sendo mais incrementado começa a ter login com redes sociais, login com autenticação em dois fatores, login com link mágico igual do Slack nós vamos tendo a necessidade cada vez mais suprida. Minha necessidade de logar no sistema vai ficando... eu tenho cada vez mais opções.

[01:48] Mas como é que fica o meu nível de satisfação a medida que essa necessidade ela vai sendo mais suprida? Então, o modelo de Kano ele trabalha com três curvas principais.

[02:01] A primeira nós chamamos de Musts que é essa ali de baixo. A do meio as Wants e a de cima os Exciters. O que é um Musts? É uma coisa que ela precisa ter. Se não tiver, eu não estou feliz. E o que a curva representa é bem isso. Enquanto essa necessidade não é suprida o meu grau de satisfação com o produto é muito baixo. Quando ela é suprida meu grau é Ok. É obrigatório ter isso se não tiver eu não to feliz.

[02:32] Esses Musts são algo que os clientes esquecem de mencionar muitas vezes porque é óbvio para eles que eu não preciso nem te contar que você deveria incluir isso no sistema.

[02:45] Então, na minha empresa eu já tive um problema uma vez no produto que o cliente esqueceu de me falar que ele tinha que funcionar offline e como era uma aplicação para minerador o técnico utiliza esse aplicativo no lugar onde não tem sinal de nada para eles eram muito óbvio que eles trabalham nesse ambiente todo dia. Eles esqueceram de mencionar que tinha que funcionar offline. Nós fomos descobrir um pouco tarde no projeto, deu para corrigir, mas poderia ter sido descoberto antes.

[03:17] Então, o que é um Must? Por exemplo, temperatura da comida no restaurante. Nós não temos que contar que queremos a comida quente, nós esperamos que ela venha quente, se ela vier fria nós achamos ruim. Se vier quente, não fizeram mais do que a obrigação.

[03:30] Em um produto digital nos dias de hoje seria, por exemplo, visualização em dispositivos móveis. Nós, às vezes, não contamos que queremos isso, mas já contamos como certo que vai ter que todo mundo vai acessar pelo celular, pelo tablet. Ao passo que as coisas que queremos que são as funcionalidades lineares, esses Wants, elas são funcionalidades que, quanto mais, melhor.

[03:55] Vamos supor, nós temos itens no cardápio, eu vou no restaurante e quanto mais coisas tiver no cardápio, mais satisfeito eu estou. Tem mais opções. Então, a medida que vai aumentando nós vamos ficando mais satisfeito. Por isso que chama funcionalidade linear, então, poderia ser, por exemplo, vamos supor, em um aplicativo de pedido de comida. O tempo com que a comida chega na sua casa. Quanto mais rápido, melhor. Se chegou em 5 minutos está ótimo, mas se chegar em quatro, melhor ainda. Nós vamos ficando mais felizes a medida que essa necessidade vai sendo suprida.

[04:36] Nós temos os Exciters, que são aqueles bônus, é um fator "uau" no produto. É aquela coisa que às vezes nós nem contávamos que ela fosse existir, mas já que ela existe ficamos feliz para caramba que tem. Então, vamos supor, no exemplo digital seria um aplicativo desses que você vai pedir a comida e descobre que você pode pagar com Bitcoin, por exemplo, para quem usa Bitcoin, nós não contamos com isso, mas se está lá faz uma diferença.

[05:07] Esse tipo de funcionalidade geralmente ela é esse fator uau. É aquela coisa que o cliente acaba lembrando quando ela é muito bem executada. Então, se pegamos, vamos supor, um exemplo de banco. "Ah, o banco tem o pagamento por QR Code". Algum tempo atrás não estávamos contando com isso de fazer a transferência ali rapidinho com QR Code, pagamento com QR Code. Então, ainda é um Exciter uma coisa que é um bônus, nem todo mundo oferece, então, é legal para caramba.

[05:41] Da mesma forma o restaurante que aceita esse pagamento com QR Code também cai nessa faixa. Uma coisa interessante das funcionalidades bônus nesses Exciters é que o tempo tende a tornar eles Musts, funcionalidades obrigatórias.

[06:02] Se pensarmos aí em uns quinze anos pra trás... quinze para vinte anos atrás em celulares com câmera. A câmera era um grande diferencial e era aquela qualidade terrível da foto, mas poxa o telefone tinha câmera. Você conseguia ter câmera no bolso e era um grande diferencial, celulares que tinham câmera eram até mais caros do que os outros e hoje em dia ninguém nem pensa em fazer um celular sem câmera, se alguém fizer cai naquela curva da necessidade do Must. Se não tem, eu estou infeliz e se tem eu estou ok, não está fazendo mais do que a obrigação.

[06:42] Mas o grande segredo do Kano Model é uma pesquisa que nós tentamos fazer com os usuários para identificar pra cada funcionalidade que pretendemos criar para o produto, qual que se encaixa em cada uma dessas curvas e tem mais duas curvinhas que eu não desenhei para não confundir, porque elas não são tão relevantes assim, que são as funcionalidades indiferentes e as invertidas.

[07:10] Então, a indiferente é aquela que honestamente não muda muito para o cliente a percepção dele de valor do produto.

[07:20] Vamos supor, você vai no restaurante, já que usamos muito aqui o exemplo do restaurante. E você tem um cardápio com a capa emborrachada, uma capa bonitona. É indiferente. É uma coisa que é legal ter, mas não vai te atrair mais clientes por conta disso, você não vai voltar no restaurante só para pegar aquele cardápio na mão.

[07:47] No caso de entrega de comida pensando um produto digital seria um relatório com o tempo médio das suas últimas entregas. Legal é, mas não é isso que vai fazer você usar aquele software mais ou menos.

[08:02] E tem o tipo invertido que é aquele que quanto mais funcionalidade tem, quanto mais aquela funcionalidade está elaborada dentro do produto mais insatisfeito nós ficamos. Então, são geralmente burocracias. Vamos voltar no exemplo do restaurante, seria o couvert artístico cobrado na hora que vamos no restaurante, principalmente quando ele é surpresa. Se te cobram R\$ 2,00 você fica um pouco chateado, mas se te cobram R\$ 50 sem te avisar você fica bastante chateado.

[08:36] Quanto mais tem, mais chateado ficamos. Um exemplo de produto digital seria, por exemplo, o excesso de notificações. Às vezes não queremos spam no e-mail, não queremos esse tipo de coisa, quanto mais mandam, mais chateado ficamos. Então, o nosso grau de satisfação ele desce a medida que essa funcionalidade é mais executada.

[09:01] Vamos voltar para pesquisa, como é que descobrimos onde uma funcionalidade se encaixa, em qual curva que ela está? O modelo de Kano define uma pesquisa muito interessante que se baseia em duas perguntas que é: como você se sente caso essa funcionalidade exista no sistema, em nosso produto, e como você se sente caso ela não exista no produto.

[09:26] Nós temos uma escala com cinco respostas. São as mesmas cinco respostas para as duas perguntas que são: Eu gosto que tenha/eu gosto que não tenha. Eu espero que tenha/Eu espero que não tenha. Eu não ligo se tem ou se não tem, tanto faz. Eu sobrevivo, então, assim eu consigo viver sem/eu consigo viver com isso me incomodando no produto. E a última que é eu não gosto que não tenha/eu não gosto que tenha.

[09:57] E aí quando as pessoas respondem a essas duas perguntas conseguimos fazer uma tabelinha, fazer um mapa de onde aquela funcionalidade se encaixa. O mais óbvio aqui são os dois extremos que é o eu gosto, eu não gosto. Então, se alguém fala que gosta que a funcionalidade exista e não gosta que a funcionalidade exista, nós não conseguimos concluir nada. Ficam duas respostas opostas e não concluímos.

[10:26] Agora quando a pessoa fala que gosta que a funcionalidade exista ou que ela não gosta que não exista já é uma funcionalidade linear. É uma coisa que está bem fácil de perceber o quão importante é para a pessoa. E da mesma forma que está na tabelinha aqui em cima, dá para ver que casando essas duas respostas nós conseguimos mapear, mais ou menos, nas nossas três curvas ali o que aquela funcionalidade representa para o cliente. Com só duas perguntas para cada funcionalidade, claro.

[11:03] Então, de novo, assim como para o modelo do Buy a Feature, é interessante limitarmos o número de funcionalidades para não ficar um formulário muito maçante também.

[11:15] Sabendo onde cada cada funcionalidade se encaixa fica mais fácil de começarmos a priorizar. Então, não adianta tentarmos fazer um produto que só tenha Exciter, só coisa bônus, se nós ainda não fizemos os Musts, as coisas que são obrigatórias, que vamos apresentar algo para o cliente e, por mais legal que sejam os Exciters, ele vai acabar ficando insatisfeito.