

## Adaptabilidade

### Transcrição

[00:00] Vocês devem ter reparado, quando nós falamos dos motores de crescimento, que muitas das histórias que eu contei tinham um detalhe em comum. E basicamente esse detalhe é: nós erramos quando fazemos algumas suposições. Foi o caso da Hotmail, que achou que ia ter um crescimento recorrente e era viral. O caso da IMVU, que acreditou ter um crescimento viral, quando na verdade o modelo ideal para eles era um motor pago.

[00:30] Essas coisas acontecem com qualquer produto digital. Por isso, é importantíssimo nós termos adaptabilidade. Existem situações onde nós precisamos definir se vamos mesmo continuar naquele caminho ou se nós vamos, de repente, fazer alguma mudança, pivotar.

[00:50] Quem fez o curso de Agile Coaching, deve lembrar de uma história que eu contei sobre um rapaz que trabalhou na Toyota. Para quem não fez, vamos lá passar por essa história rapidinho.

[01:01] Mike é um americano que trabalhava numa fábrica de automóveis nos Estados Unidos e resolveu viajar até a Toyota para tentar compreender por que os carros lá funcionavam, tinham uma qualidade melhor. E ao chegar lá, o Mike foi instruído a parafusar retrovisores. E o chefe dele chegou e explicou um conceito bem simples, que era utilizado na linha de fábrica.

[01:27] Ele disse: olha Mike, toda vez que você tiver um problema, se você, por algum motivo, não conseguir parafusar ou danificar o carro, eu vou te pedir um favor: você vai apertar um botão. E quando você apertar aquele botão, vai tocar uma sirene, vai piscar a linha de produção inteira e todas as pessoas que estão aqui vão parar de trabalhar imediatamente para ver qual foi o erro que você cometeu. Bem intimidador para quem não está habituado a isso.

[02:01] Mas esse conceito da Toyota, que tem o nome de Andon Cord, uma corda que nós puxamos e para a produção, stop the line, é algo que ajuda muito, mesmo no mundo digital. E hoje em dia, ele é utilizado por pela maior parte das empresas de tecnologia, com uma pegada um pouco diferente, porque não é mais um botão que nós apertamos para mostrar o erro, mas nós vamos chegar lá.

[02:27] O Mike ficou um pouco assustado com essa ideia de apertar um botão e todo mundo assistir o erro dele. Mas ele continuou trabalhando, o trabalho dele era simples, tinha que parafusar um retrovisor. E eventualmente, se ele tivesse algum problema, ele ia apertar o botão.

[02:44] Um dia, a chave deslizou e o Mike acabou arranhando o retrovisor do carro. No que ele fez isso, ele apertou o botão. Criou coragem, apertou o botão. E aconteceu aquilo que o chefe dele falou que ia acontecer. Tocou a sirene, todo mundo parou de trabalhar e todo mundo se reuniu na frente do retrovisor que ele tinha arranhado, para tentar entender. Poxa, o que será que o Mike fez aqui?

[03:14] O Mike foi levado para uma reunião com a diretoria da Toyota e, lá chegando, as pessoas começaram a conversar em japonês, o Mike não conseguia compreender. E ao final, todos levantaram, bateram uma salva de palmas para ele e agradeceram. Falaram: Mike, muito obrigado. Por você ter nos contado um problema que aconteceu na nossa linha de produção, você vai nos ajudar agora a melhorar. Os carros da Toyota vão ser um pouco melhores graças a você.

[03:47] E o que eles fizeram foi algo muito simples, eles trocaram a chave que o Mike utilizava, e o parafuso, consequentemente, por um parafuso com aquele buraquinho em forma de cruz. Porque a chave desliza menos. E essa atitude do Mike ajudou a Toyota a melhorar todos os carros dela, porque daí para frente, todos os carros passaram a ter esse parafuso.

[04:11] Essa ideia de ter esse botão, stop the line, é fundamental por vários motivos. Eu quero destrinchar um pouco mais essa história. Quando o Mike apertou esse botão, nós paramos a produção, mas nós não tínhamos um lote de carros com parafuso já apertado. Quando nós usamos aquele one piece flow, que eu comentei na aula de small batches, a primeira do curso, você tem apenas um carro passando ali, você não tem estoque, você não tem que trocar o parafuso de todos os outros. Você só vai trocar dos próximos que estão vindo.

[04:48] Você consegue fazer esse processo de melhoria contínua muito mais rápido. Ainda entra outro detalhe muito interessante. Se você queria uma cultura onde as pessoas têm medo de comunicar o erro dentro da sua empresa, ninguém quer apertar o botão e parar a linha. Esse retrovisor, de repente, passa despercebido, chega ao cliente. E vamos imaginar, no caso da Toyota, exporta esse carro para a Índia ou para o Brasil, para Portugal. E de repente começa a ver que tem vários carros com retrovisor arranhado.

[05:24] Qual é o custo que a Toyota ia ter para descobrir a origem dos arranhões, e o que ela deveria fazer para corrigir esses arranhões. Nesse caso, muitas vezes, aplica-se uma solução sintomática. No final da linha de produção, nós passamos o esmalte no retrovisor, que geralmente retrovisor vem arranhado. E nós não atacamos a causa raiz. E é caro descobrir a causa raiz quando nós deixamos o erro passar e não alertamos sobre ele logo de cara.

[05:58] A Toyota tem um lema muito interessante sobre isso, ela diz: pare a produção para que a produção nunca tenha que parar. E é curioso quando nós trazemos isso para o mundo de produtos digitais, principalmente o desenvolvimento de software, nós temos servidores de integração contínua e testes automatizados. São boas práticas desde sempre para o desenvolvimento software.

[06:20] O teste automatizado é um teste, para quem não é da área, é um teste que é um código que testa o seu código. E um servidor de integração contínua é um computador que fica à disposição para, de tempos em tempos, rodar todos esses testes. Porque o código vai mudando, as pessoas vão alterando o código, e o servidor está ali repetindo o teste para te falar se aquela alteração quebrou o código ou não, se inseriu um bug ou não.

[06:47] Quando o servidor de integração contínua detecta um bug, é basicamente o mesmo processo da Toyota. Ele dispara um e-mail para a equipe de desenvolvedores e fala: olha, o código não está funcionando mais como deveria. O princípio é exatamente o mesmo. E isso nos ajuda a adaptar. Existe um caso curioso, que ficou bem famoso, de uma empresa canadense chamada Friendster.

[07:20] A Friendster era uma rede social, basicamente. E eles cresceram de uma forma exponencial, de uma forma muito rápida, e eles foram vítimas do próprio sucesso. Eles chegaram num ponto que tinha tanta gente, o crescimento foi tão exponencial, que eles começaram a enfrentar um gargalo técnico.

[07:49] A solução que eles tinham, o software que eles tinham e o hardware que eles tinham na época, não era capaz de sustentar a quantidade de usuários que eles passaram a ter. Curioso. Esse tipo de efeito, hoje em dia, é mais raro. Nós temos à nossa disposição a computação nas nuvens, é muito mais fácil, hoje em dia, escalar o software para vários usuários. Claro, depende da tecnologia que ele foi desenvolvido, mas é muito mais fácil do que era na época do Friendster.

[08:23] Mas nós temos também um gargalo pessoal. Porque quando nós começamos a crescer muito a empresa de uma vez, imagina: nós temos quatro funcionários sêniores tratado daquele produto digital, e de repente nós precisamos multiplicar. A equipe tem que virar uma equipe de quarenta pessoas para dar conta. Eu tenho quatro pessoas que conhecem bem a realidade da empresa, do produto, dos clientes. E trinta e seis que acabaram de entrar.

[08:50] Eu preciso fazer esse processo de onboard ser o mais acelerado possível. Tem que adaptar também o meu lado pessoal. Existem esses dois gargalos. Hoje em dia, o gargalo pessoal é muito mais complexo do que o técnico. O técnico, já existem soluções como umas boas práticas que você pode fazer para tentar. Não que não seja fácil, mas ele está mais encaminhado do que o pessoal. As pessoas, as empresas, elas têm mais dificuldade com essa escalabilidade de equipe.

[09:21] E eu vou falar sobre isso na próxima aula, vou ensinar uma técnica bem bacana para ajudar nessa parte. Sobre o a questão da causa raiz, que eu falei da Toyota. Não deixar esse probleminha, quando ele ocorre, ir passando e chegar ao cliente. Uma coisa que é bem legal de se fazer é executar os cinco porquês sempre que houver um problema, para nós tentarmos entender a causa raiz desse problema.

[09:51] Eu vou dar um exemplo real dos cinco porquês que aconteceu na minha empresa e foi uma coisa bem curiosa. Tudo começou quando um cliente ligou para minha empresa e reclamou que tinha um bug no sistema. O bug, ele cadastrou o ticket. Por que ele cadastrou o ticket? Nós temos o primeiro porquê. Ele cadastrou esse ticket, porque um feature tinha sumido do sistema. E por que esse feature tinha sumido do sistema?

[10:21] Alguém da equipe de desenvolvimento fez um rollback e, sem querer, tirou uma feature do repositório de código, voltou o código no tempo. Mas por que essa pessoa fez isso? Essa pessoa fez isso, porque ela acabou rodando um comando errado no Git. E o motivo dela ter rodado esse comando errado no Git. O Git é o programa que faz o gerenciamento do código. Por que ela rodou esse comando errado? Porque ela não teve treinamento quando ela entrou na empresa.

[11:02] No quarto porquê eu já estou começando a clarear bem as coisas, está vendo? Por que ela não teve treinamento quando entrou na empresa? Porque as pessoas que eram sêniores, que poderiam fazer esse treinamento, estavam todas viajando. O que nós começamos a perceber é que, de repente, nós não podíamos deixar todos os sêniores viajando. Nós tínhamos que deixar alguém ali.

[11:29] E não só não deixar viajar, mas separar um tempo para que a pessoa pudesse fazer um treinamento, elaborar algo para facilitar quem estava chegando. Nós não enxergávamos que o Git era algo que nós precisávamos dar um treinamento. E acabamos descobrindo que sim. Mas tudo começou com o registro de um bug do cliente. O cliente registrou um bug. E fazendo os cinco porquês, nós acabamos chegando à causa raiz.

[11:57] Nós temos que ter muito cuidado com os cinco porquês, ele não pode virar as cinco culpas, nós não podemos ficar: ah, eu fiz isso porque a outra pessoa fez isso. A outra pessoa fez isso. Não é para apontar dedo, é para nós tentarmos entender a origem do problema como organização, como equipe. E aí, sim, atacar para que nada disso volte a acontecer.