

INICIO

Entrevistado: Bom, legal. Parabéns pelo trabalho. Essa modelagem aí é incrível porque ela ajuda a ir conceituando. A notícia ruim é que, como qualquer outra modelagem, ela é uma simplificação da realidade. E aí falta informação, falta mais. Conforme você foi passando, falta isso, falta aquilo. E eu acho legal, a gente pode ir conversando aqui, eu vou ligando algumas coisas. Por qual ponto você acha melhor a gente começar aí? Porque aí eu vou trazendo um pouco do que eu tenho na vivência aqui. Existe um degradê de cenários. Tem desde o histórico de equipes tradicionais de projeto, equipes num cenário intermediário e equipes de produto, já matriciais. Uma coisa que eu senti falta é de, por exemplo, você tem ali o Product Team, você tem o Development Team, você modelou no Development Team o Operation e o Development, mas você não modelou Business Team ou a área de negócio. Por quê? Porque o Product Team não quebra só o gap entre Development e Operation, ele quebra o gap também com a área de negócio. Então, uma coisa que eu senti falta.

Entrevistador Isso você conectaria com o que o Manuel Pais cita sobre o **Stream Aligned**, que são times que estão alinhados com o objetivo da empresa ou...

Entrevistado: O Product Team já é um **Stream Aligned**. Mas eu entendo que você tem aqui o Product Team e você modelou um outro conjunto de conceitos, que é quando eu não tenho esse time de produto, eu tenho o time de desenvolvimento e operação. E você colocou aqui, conceitualmente, que times orientados a projetos, eles entregam para um time de desenvolvimento que tem uma fraca colaboração com o time de operação, porque são equipes diferentes e se fala um pouco. Show! Mas nesse sentido, falta esse outro ator que é o time de negócios. Uma organização que trata do time orientado a projetos, ela tem esses três grandes atores. Time de negócio, demandando projeto para o time de desenvolvimento, que ocorre por meio dos processos formais de engenharia de software, de análise de requisitos, elicitação de requisitos, detalhamento de requisitos. E esse time de desenvolvimento implementa e joga para o time de operação. Então, eu senti falta desse primeiro ator aqui, certo? Porque no time de produtos, que é o time **Stream Aligned**, nós não jogamos somente o time de desenvolvimento e de operação juntos, nós jogamos o time de business juntos. Sim. A grande, inclusive, mudança que nós estamos fazendo aqui no Banco do Brasil nos últimos quatro anos é exatamente essa fusão. O Gartner chama de Fusion Teams. São os times fundidos, vamos dizer assim. Então, você tem a área de business, de negócio, como um ator dentro desse time multidisciplinar, que é o Product Manager, o analista de produtos digitais e por aí vai, certo? Por isso que entra ali a disciplina Product Management. Show! Essa é a disciplina. Nós estamos formando os times de produto nessa disciplina. Não só os times de produto **Stream Aligned**, mas os times de plataforma, como o Manuel Pais também chama. Porque ele mesmo fala que os times de plataforma têm que ser conduzidos, desenvolvidos e evoluídos como produtos. Em produto. Internet, né? Vamos para o Bridge Team e o Enabling Team. Legal.

Entrevistador Só um parêntese. Que foi por causa do Manuel Pais que a gente fez essa conexão entre Product Management e o time horizontal, porque eles têm clientes internos. Então, foi bem interessante que você confirmou.

Entrevistado: Então, já está aqui ligado.

Entrevistador Beleza.

Entrevistado: Nós temos investido na capacitação das lideranças das nossas plataformas na disciplina de gestão de produtos digitais. Show! Aqui dentro do banco. O Bridge Team e o Enabling Team é o mesmo time. Na verdade, o que eu quero dizer com o mesmo time? Aí vem uma camada de complexidade. Eu vou expor ela aqui para você e aí você depois se divirta com ela para ver se isso muda alguma coisa conceitualmente ou cria um outro nível de abstração. O próprio Manuel Pais diz que os conceitos por trás do livro podem ser entendidos, diversos times, como montáveis, como bonecas russas. Você vai compondo em times maiores, como se fosse Scrums of Scrums e por aí vai. É o que ocorre em uma organização do tamanho do Banco do Brasil. Nós estamos falando de mais de mil times. E você compõe esses times em superestruturas. Uma plataforma, no nosso caso, não é composta de um time, mas de uma composição de diversos times.

Entrevistador Então vocês têm vários times para chegar em uma plataforma?

Entrevistado: Isso. A mesma coisa para o **Stream Aligned**. Nós temos vários times para montar um value stream para o cliente.

Entrevistador Entendi.

Entrevistado: Vamos exercitar isso para tangibilizar para você. Vamos para um exemplo de plataforma de desenvolvimento mobile. Desenvolvimento móvel. Eu tenho dentro do banco esta plataforma, esse time de plataforma que ele é composto de algumas squads, totalizando cerca de 150 pessoas. Eu tenho desde o time que cuida do processo de qualidade, se a ICD entra a parte de ALM, você modelou aqui, ALM, certo? Eu tenho o time que faz o papel de enabler. Eu tenho uma squad para fazer consultoria, treinamento, evangelização, cuida de guildas para disseminação de conhecimento, formatação de treinamentos, contratação de treinamentos, e formatação do que precisa complementar. Eu tenho times que cuidam dos frameworks, padronização dos frameworks, ciclo de vida dos frameworks, por exemplo. No Banco do Brasil eu tenho todos os times de **Stream Aligned** que desenvolvem coisas que estão no app, certo? Eles usam esta plataforma. E aí tem o framework, nós usamos React Native, e o time de plataforma é responsável por evoluir a versão do React Native, não só evoluir, mas garantir que ela está funcionando com o scanner da câmera, garantir que ele está entendendo um QR Code de um formato do Pix, garantir que ele está utilizando uma biblioteca de inteligência artificial para fazer reconhecimento de face. O time de plataforma elimina essa carga cognitiva, cuidando dessas partes mais baixo nível. Então, eu tenho várias squads dedicadas... Ah, eu tenho uma squad que cuida de componentes de segurança para jornada mobile. Então, esse time de plataforma é composto de pequenos subtimes. Vai desde o framework para codificar nativo, quando precisa codificar nativo, o framework multimobile que é em cima do React Native, até um framework low-code com uso de inteligência artificial para que os times de **Stream Aligned**, quando julgarem mais interessante, utilizarem até low-code para colocar coisas dentro do app. Então, a plataforma mobile é uma composição de várias squads que tem

o propósito de tornar tecnologicamente mais simples, rápido e fácil desenvolver soluções mobile do banco. Isso é o que nós chamamos de uma plataforma dentro do BB, e acaba sendo essa composição de um pouco mais de uma dezena de times. Legal.

----- interrompidos -----

Entrevistado: Então, vamos voltar lá para o... Vamos, deixa eu compartilhar aqui de novo. Ok. Então, eu estava exatamente no ponto que eu estava falando da composição das bonecas russas, certo? Esse diagrama todo dá uma boa ideia do que é um time, mas ele força uma simplificação da realidade, que é uma relação do conceito de plataforma e do conceito de time de produto como uma instância de um squad. E numa organização grande, e eu diria até de uma de médio porte, tá? Médio porte para cima, o conceito de time de produto ou time de plataforma não se modela um para um para a squad, é um para ele. Isso é uma das coisas que, conceitualmente, olhando, eu falei, faz falta, não dá para você instanciar isso, porque você começa a ter que modelar outras relações, que é, essas squads dentro da plataforma, elas têm um alinhamento, elas têm que ter um nível de colaboração mais intenso do que para fora, né? Então, bridge team e enabler team, eles existem dentro do conceito de uma plataforma.

Entrevistador Entendi.

Entrevistado: No caso do BP.

Entrevistador E são vários, e poderia ter outros aqui dentro também. E todos, talvez eu poderia, porque agora eu vou precisar refinar um pouco mais para tentar trazer essa ideia da complexidade, eu acho que eu não consigo trazer toda, mas com a nossa conversa eu consigo ajustar. São várias equipes dentro dessa ideia da boneca russa para eu conseguir prover um tipo de plataforma, no exemplo aqui, de desenvolvimento mobile.

Entrevistado: Sim, e a mesma coisa para um produto. Para eu fazer crédito de financiamento de veículos, eu preciso de 3, 4 squads diferentes. Dependendo do tipo do produto, da complexidade do produto, nós temos aqui Teams **Stream Aligned** com, por exemplo, CDC, Crédito Direto ao Consumidor e as suas diversas linhas de crédito. São 200 pessoas. Quando eu falo 200 pessoas, está incluído Business Team. Então, as squads são multidisciplinares, eu tenho Product Managers, Analistas de Produto e por aí vai. Todo o ecossistema de produtos, de business dentro, junto com o time de desenvolvimento, junto com o profissional de Operations, que aí é o SRE, o Engenheiro de Confiabilidade, atuando dentro desse time, profissional de Cyber, Cientista de Dados. Isso tudo é um time **Stream Aligned** de CDC, por exemplo.

Entrevistador: Mas chega a ser um time só com 200 pessoas ou são pequenos times?

Entrevistado: É uma composição de pequenos times. O conceito do **Stream Aligned**, que é alinhado ao fluxo de valor, é porque a composição e o que define o que estão esses pequenos times é a jornada do cliente. Então, você desenha qual é o funil de vendas, a jornada que o cliente segue para consumir o produto CDC. E aí você tem uma squad que é dedicada à etapa da jornada de descoberta de qual o tipo de empréstimo que eu preciso. E a missão dessa

squad é entender como as pessoas percebem a necessidade de um empréstimo e ajudar essas pessoas a descobrir qual o empréstimo é o melhor para elas. Certo? Fazendo simulações e por aí vai. Aí você vai... Cada time é responsável por uma etapa. É isso que é o **Stream Aligned**. É o velho Stream Aligned. Ele vai estar alinhado a esse fluxo de valor. Você faz o mesmo com o time de plataforma. Você tem que... No caso do time de plataforma, é qual é a jornada do dev ou qual é a jornada do time de produto que precisa usar a minha plataforma. Ah, ele precisa modelar a arquitetura da sua solução, ele precisa fazer um mock-up, ele precisa fazer um protótipo. E aí o time de plataforma tem a missão de facilitar esse processo. Então é um velho Stream, mas de engenharia de software, em torno do LM. Aí eu vou te dar a terceira dimensão do problema. Porque lembre-se, bonecas russas são infinitas. A gente trabalhou com duas dimensões, que é squads e as diversas squads formam uma plataforma. Ou um produto. Eu tenho um super set ainda acima disso. Vou te dar um exemplo. O ruim é que no livro do Manuel Pais, ele não tangibiliza isso claramente. Ele modela e faz as caixinhas ali e não vem exemplos. Eu tenho num nível bem baixo dentro da organização do Banco Brasil. Eu tenho uma plataforma de monitoração, que é responsável por... São vários squads que constroem todos os building blocks de observabilidade. Eu tenho uma... Eu tenho uma plataforma de cloud, que conecta todas as nuvens públicas e privadas para facilitar o processo de desenvolvimento usando o cloud. Sabe a plataforma de mobile que eu te dei o exemplo? Essa plataforma de mobile, ela usa a plataforma de cloud e a plataforma de observabilidade para facilitar o processo de desenvolvimento para mobile. Então, eu tenho um stacking, um empilhamento de plataformas em camadas de abstração, cada uma retirando carga cognitiva da camada seguinte. E quando eu digo camada seguinte, não é uma camada seguinte obrigatória. Eu posso ter um time de desenvolvimento que consome mobile e que também consome cloud. Então, desce um degrau abaixo, consome cloud. Desce outro degrau abaixo, consome diretamente observabilidade. Por N motivos, né? Então, eu colocaria aí essa terceira dimensão que é uma relação, uma modelagem entre esses supersets, que são essas plataformas. Porque elas também precisam colaborar entre si. E qual o desafio em escala organizacional? Estabelecer processos e protocolos de alinhamento entre essas superentidades, para que elas colaborem corretamente e não compitam, né? Ou não se defendam, senão eu posso só mudar o problema do muro do DevOps, o muro do business e da TI, para uma outra escala. Enfim, passeando ali pelo seu diagrama, eu falei, opa, beleza. Ele modela aqui essa camada de time, com esses conceitos, essas relações. Eu senti falta desses três passos de abstração. O primeiro que mostra que aquilo ali, isso ocorre de uma forma fractal, né? Ele vai subindo. No nosso caso, é bom eu parar nesse nível para você, porque tem um outro. Mas aí você começa a entrar em arquiteturas organizacionais. Business Architecture, Enterprise Architecture. É uma camada um pouco... É a camada onde a arquitetura de negócio da empresa influencia a arquitetura tecnológica ampla, uma influencia a outra.

Entrevistador Sim, que tem aquela ideia da forma que é a comunicação, a forma que eu estruturo o meu time, isso vai impactar no meu produto também, né? Isso, sim. Legal. Bom, muito interessante, muito obrigado por esses pontos gerais aqui. Acho que são insumos realmente que eu vou seguir trabalhando, eu tô nessa fase de refinar e...

Entrevistado: Aí você decide o que dá tempo, o que não dá tempo, né?

Entrevistador Não, mas dá tempo, porque eu tenho prazo para defender, eu tenho até 2027, eu tô super adiantado, eu tô no final.

Entrevistado: Ah, legal.

Entrevistador Então, acho que até janeiro, fevereiro, eu já consigo defender. Você tá em São Paulo, né? Não, agora eu tô em Brasília. Ah, você tá em Brasília? É, porque eu vou pra São Paulo, fiz todas as disciplinas, já fiz o meu sanduíche, já voltei, eu passei um ano em Madri, na Universidade Politécnica de lá. Ah, tu tá lá perto do Manuel Pais. Sim, eu entrevistei ele lá em Madri, inclusive. Ele tava aqui esses dias. Eu vim, eu fui no Banco do Brasil, falei com ele, tirei foto lá no evento que teve, né? Ah, legal. Fui assistir a palestra, tirei foto com ele. Mas em Madri, há uns dois anos atrás, eu entrevistei ele, a gente discutiu sobre o livro, eu mostrei esse diagrama, ele me trouxe algumas ideias também, foi bem interessante. Você já teve um artigo publicado com esse diagrama? Já, foi o primeiro diagrama.

Entrevistado: Ah, eu acho que eu li esse artigo. Porque você mostrou, falei, eu conheço essa diagramação, alguma coisa parecida, eu já vi. Sim.

Entrevistador Esse daqui já é uma versão mais refinada, aquela foi a primeira versão. A gente acabou de ter um segundo artigo aceito agora pro ICSE, que eu discuto um pouco como eu operacionalizo isso pra conseguir testar e trazer um pouco as próximas questões que eu vou te apresentar pra gente conseguir ver isso mais palpável no mundo real, né? E agora eu tô justamente na fase de teste, porque nessa minha tese, eu entendi que muitas vezes a gente acaba criando teorias que ficam no campo conceitual. E teorias na engenharia de software, por a engenharia de software ser uma área prática, ela precisa estar continuamente sendo refinada, atualizada, pra ela fazer sentido pra gente. Então, se eu deixasse ela, como era o caso de agora, muito superficial, sem entender essa complexidade que você me trouxe, ela em algum momento já não valeria mais. Não é uma teoria que vai seguir sendo discutida. Certo. Então eu tô propondo um método também de criar e refinar continuamente uma teoria.

Entrevistado: Legal.

Entrevistador Enquanto eu valido ela executando, aqui no caso do DevOps.

Entrevistado: Legal.

Entrevistador Então, em cima de tudo isso daqui, a gente tem alguns conceitos e relações entre esses conceitos que a gente chama de proposições.

Entrevistado: Só uma coisa, você não tá mais compartilhando tela, tá? Não sei se...

Entrevistador Ah, eu não tô?

Entrevistado: Não, caiu ela. Isso.

P1

Entrevistador: Mas tudo bem, agora eu vou seguir com uma outra parte, que eu posso só fazer umas perguntas. A primeira proposição que eu tenho, ela junta uma cultura de compartilhar conhecimento com colaboração. Então, por exemplo, se eu tenho equipes, vamos supor que equipes atualmente em silos, eu não sei se já foi o caso aí no Banco do Brasil, acredito que sim.

Entrevistado: Com certeza já foi. Toda empresa com TI já passou por essa etapa. E temos ainda equipes que estão presas em silos, que estão em processo de... São 12 mil pessoas para passar pelo processo de transformação, nós estamos no meio do caminho.

Entrevistador E você sabe quanto tempo já vocês estão nesse processo? Desde que surgiu o DevOps?

Entrevistado: Então, aí entra outra coisa que é a conceituação do DevOps e esses diversos níveis, certo? Quando a gente fala até do que surgiu com o livro do Manuel Pais em 2019 e tudo mais, é uma etapa muito recente. Mas a gente começou a implementar pipeline de CI e CD em 2009. Então, pipeline automatizado de CI e CD com continuous testing em 2009, certo? Agora, times integrados, que é onde você começa a quebrar a barreira cultural, foi há dois anos e meio. Eu diria para você que práticas de DevOps, ferramental de DevOps, tecnologias de DevOps, 15 anos. Mas a barreira cultural que depende de design organizacional é três anos. Dois para três anos.

Entrevistador E chegou a ter aí papéis com esse nome de DevOps, contratações? Ah, vamos contratar a DevOps agora. Isso nunca foi... Porque eu já observei muito em trabalhos que isso é... As pessoas se enganam muitas vezes, né? Acham que o DevOps é contratar uma pessoa que ele vai lá e resolver. É um papel, mas não é um papel.

Entrevistado: Na prática, nós sempre tivemos alguma coisa parecida com time de plataforma. Antes da definição clara do conceito ou popularização, vou dizer assim, porque plataforma é algo conceitualmente antigo até da indústria. Mas nós sempre organizamos times com esses papéis. De realizar uma etapa extremamente técnica da atividade que seria consumida por outros times. Nós fizemos isso na década de 90 para o desenvolvimento mainframe no banco. Então, nós temos uma, entre aspas aqui, plataforma mainframe que tem 30 anos que ela foi feita. Por que eu posso afirmar para você que é uma plataforma mainframe criada no Banco do Brasil e não é simplesmente a gente desenvolver no mainframe? Porque se você souber Cobol e você conhecer **CICS**, que é o servidor de Cobol da IBM, não é o suficiente para você desenvolver aplicações dentro do Banco do Brasil. Você tem que conhecer as ferramentas e a jornada da plataforma mainframe no banco.

Entrevistador Entendi. Ah, interessante. Então, o banco tem uma longa história e dá para ver bastante como foi a evolução do DevOps aí, né?

Entrevistado: É, exato. Então, acaba tocando. Essa parte mais técnica e ferramental do DevOps é bem antiga.

Entrevistador: Se você acredita que se eu tenho equipes, por exemplo, atualmente em **Silos**, se elas passarem a compartilhar mais ativamente a responsabilidade por um produto ou algo assim, se ela consequentemente, ou seja, se ela vai compartilhar o sucesso ou a falha de alguma coisa, se ela consequentemente passa a colaborar mais? Se isso vai permitir ou vai ser um habilitador para a colaboração?

Entrevistado: Somente mediante a alteração do design organizacional. Composição do time. Ou seja, não adianta eu ter um time de desenvolvimento e um outro time de operação e falar que as metas deles são as mesmas se eles estão em departamentos diferentes. Eu preciso gerar essa aproximação. Isso ocorre a nível de desenho organizacional. Porque existem inúmeros mecanismos dentro da empresa que afastam times que são diferentes.

Entrevistador Então, quando a gente promove isso de forma organizacional, eu consigo impactar tanto na frequência dessa colaboração quanto na qualidade dela também, então, né?

Entrevistado: Sim. Ao criar times de produtos, incluindo times de negócios, desenvolvimento e infraestrutura, o nível de colaboração tem aumentado muito. É notável.

P2

Entrevistador Ótimo. E agora sobre uma outra proposição, sobre redução. Se eu promovo uma colaboração, eu consigo reduzir os silos, os conflitos. Eu tenho algumas hipóteses que estaria um pouco ligado como as estruturas de silos organizacionais, elas podem ser caracterizadas com uma frequência eventual e uma baixa qualidade de colaboração. Então, esses times, eles não colaboram, né? Foi o que a gente comentou nessa anterior. Se eles não colaboraram, eu estou tateando, eu estou caminhando para um cenário de silos, né?

Entrevistado: Sim. Ou a colaboração nesse caso, no caso de silos, ela ocorre de maneira protocolar e com processos muito bem definidos. Se existe uma colaboração protocolar e com processos muito bem definidos, ela é ritualística e pobre.

Entrevistador: E seja por e-mail, né? Então, o e-mail abre um ticket para mandar uma solicitação.

Entrevistado: Abre um ticket, exato. Abertura de uma demanda, formalização de requisitos. Aquela engenharia de software clássica que no início do século foi implementada em escala e foi o erro de quem tentou implementar a **RUP**, certo? Levou a tudo isso. Protocolos... Se você lê o livro do Pressman de engenharia de software e fala vou implementar isso desse jeito, com essas etapas, com um profissional jogando para o outro, é isso.

Entrevistador: Uma cascata, né?

Entrevistado: São os silos. Nós tivemos isso.

Entrevistador: Essa foi uma das minhas primeiras disciplinas com a professora Carla. Primeiro, a gente teve que trabalhar nessa ideia de silos e depois mudou tragicamente para a gente ser um pouco ágil. E aí dá para a gente entender essas diferenças.

P4

Entrevistador: Hoje você trabalha e observa aí como você descreveria a cultura de **compartilhamento de conhecimento** na sua equipe ou nas equipes e nas estruturas que você tem. Essas informações, elas ficam disponíveis? Existe alguma **Wiki** que todo mundo pode compartilhar? **Ou existem ritos que facilitam essa informação?** Porque a gente tem a ideia de que isso também pode quebrar silos, né? **Ao compartilhar informações, se eu tenho um dashboard ou se eu sou um membro de uma equipe que eu preciso de uma informação crítica agora, o quanto fácil é para eu obter essa informação?**

Obs: vale observar que é uma informação necessária pra mim. Não estamos falando de informações gerais sobre o banco, pois sabendo que em um contexto financeiro podem haver fraudes.

Entrevistado: Legal. Tudo depende. Até porque, por exemplo, nós estamos falando de um banco. Informação pode ser utilizada para gerar fraudes financeiras, certo? **Então existem níveis e níveis de acesso à colaboração.** Vou falar o que ocorre do ponto de vista das camadas de plataformas tecnológicas, porque são elas que nós queremos que, através dos times de enablement, ajudem os times **Stream Aligned**. Nós temos um IDP, **Internal Developer Platform**. Essa é a última camada da boneca russa das plataformas. O desenvolvedor toca no IDP e o IDP é a composição de todas as plataformas. Então você consegue fazer toda a jornada, fim a fim, não importa qual perfil. Se você é UX designer, cientista de dados, desenvolvedor back-end, front-end, web, mobile, não interessa. **O IDP tem todas as jornadas. É nele aonde tem as Wikis, as documentações e inteligência artificial fazendo indexação com RAG e tudo mais para ajudar a tirar dúvidas e orientar o desenvolvedor.** Então tem tudo isso no que é a IDP do banco, que a gente chama de **TEC-BB**. Esse é o guarda-chuva. **Essa é a primeira parte, que é a questão da exposição de informação com roteiros, orientação de como trabalhar, melhores práticas e por aí vai.** Repositório de código-fonte. Nós fazemos uso do GitLab e estamos migrando para o GitHub. Beleza? As duas ferramentas semelhantes. Nós temos mais de 120 mil repositórios no Git, de acordo com o time. Então cada time tem ali a sua uma dúzia ou duas dúzias de repositórios e os times podem colaborar nos repositórios uns dos outros. Não significa que eles têm acesso livre e aberto. O time é que diz, esse componente ele é aberto para toda a organização. Esse componente, um time que quiser colaborar comigo, ele precisa pedir o acesso e aí ele pode fazer um pull request. Tem políticas a esse respeito por conta daquilo que eu falei. Nem toda fonte deve ser lida por qualquer pessoa, pois ele pode gerar fraudes. Vou falar de uma coisa pública, que eu posso falar porque foi público, se você procurar na internet. Recentemente um funcionário da TI do banco foi preso, porque foi identificado que ele vendeu a chave de acesso dele para uma quadrilha criminosa. Então assim, isso é tratado com bastante cuidado, um equilíbrio entre colaboração e as plataformas também, elas promovem guildas, meetups contínuos. Eu, antes de entrar aqui, estava acompanhando uma que estava sendo feita junto

com o pessoal da Google, para explicar toda a integração da nuvem da Google e dos modelos de IA que estavam sendo colocados na nossa plataforma interna de nuvem, promovido pelo time da plataforma de nuvem. Então, ali no **Enabling Team, eles cuidam dessa parte de DevEx, Developer Experience, que vai desde a jornada do desenvolvedor, métricas do desenvolvedor, technical writing para escrever coisas e roteiros para auxílio do desenvolvedor, curadoria de chatbots de LLM para tirar dúvidas do desenvolvedor e organização de meetups internos para disseminação de conhecimentos.** Então, tem tudo isso para poder promover o acesso à informação, com essa ressalva, dentro daquilo que é possível pelo nível de segurança que tem que ter dentro de um banco.

Entrevistador Legal.

Entrevistado: Falei demais, né?

Entrevistador Não, não. Eu sempre falo muito. Está ótimo. Que bom que está gravando. Tudo que eu precisava, porque eu queria entender se tem essa relação e é bem complexa essa relação no banco, principalmente pela questão da segurança. Então, vocês compartilham conhecimento de forma organizada para quem precisa, para quem não precisa e isso vocês entendem que promove a colaboração, que pode ajudar nas colaborações, não centralizar isso em alguém. Isso fica disponível, mas tem um nível de segurança ali também.

Entrevistado: Isso.

P5

Entrevistador Legal. E considerando que vocês têm agora um time cross-funcional, um time multidisciplinar, você acha que, considerando esses diferentes especialistas no time, essas pessoas têm facilidade de colaborar entre si ou isso pode acabar virando pequenos silos dentro da própria equipe? Ah, eu sou o desenvolvedor, então eu vou cuidar só da parte do desenvolvimento, você é da operação, você é do produto. Cada um fica dentro da sua microcaixinha ali dentro do time. Ou você acha que isso pode promover que essas pessoas trabalhem juntas numa mesma tarefa, não sei se chega a ter um pareamento entre um desenvolvedor e uma pessoa da operação, porque eles têm que resolver algum conjunto. Saberia dizer se tem, como que isso acontece?

Entrevistado: Eventualmente, vamos lá, estamos no processo e depende realmente de time para time. Sim. Então tem, e eu vou dizer para você a aposta nossa no ator crucial, gerência média. Então, cada squad nós colocamos dois atores como lideranças de referência responsáveis por promover essa integração, o Product Manager e o Tech Lead. Então, cada squad é liderada por essas duas pessoas e elas têm o papel de promover essa atuação integrada e colaborativa dos membros da squad. No Superset tem também uma dupla, que é o equivalente ao Product Manager da linha inteira, ou um Senior Product Manager, ou um GPM, Grupo Product Manager, o mercado tem vários, é isso? E um líder de tecnologia para toda essa tribo. No banco a gente chama de linha. Linha de plataforma, linha de negócio. Esses também são os responsáveis por fazer essa gestão integrada e colaborativa e promover a integração,

essa colaboração. O que parece que tem sido crucial, co-location. Fisicamente essas pessoas estarem compartilhando rotinas do dia. Por isso, o outro elefante na sala que é o trabalho remoto. Porque o trabalho remoto, ele deixa que as pessoas continuem com as suas rotinas, certo? E o colega da outra disciplina, ele só é alguém que de vez em quando eu estou trocando mensagem, protocolo ou uma vídeo num rito do ágio, mais um protocolo. Aquilo que eu falei, quanto mais protocolar é a interação entre esses indivíduos, mais pobre a colaboração. A maneira menos protocolar de ocorrer a interação, pessoas co-located, a mesma sala virando para o lado e falando, fulano, o que você acha disso, não sei o que lá? Zero protocolo, é uma integração muito mais íntima. Então nós temos, para você ter uma ideia, nós temos aqui em Brasília um prédio no final da Asa Norte, que é onde está a infraestrutura. O prédio no Setor Bancário Sul, que é aquele preto do Banco do Brasil, onde está o time de desenvolvimento. E aquele prédio azul que está no começo da Asa Sul, que chama Edifício Banco do Brasil, que é onde está o pessoal de negócio.

Entrevistador Ah, então...

Entrevistado: Então, Business, Dev e Ops, em prédios diferentes. Só que os times que estão nesse modelo matricial, a gente movimentou. Então, no prédio azul, assim como no prédio preto, estão aqui no centro de Brasília, eles têm gente de desenvolvimento, infraestrutura e negócio trabalhando juntos para quem já está nesse modelo. Mas o nosso silo clássico, físico, separado. Há 10 anos atrás, conversando com o pessoal de fora, eu falava que nós temos no banco... O muro do Devops é um problema de 10 milhas, que é a distância entre um prédio e outro. Nós temos... É um buraco.

Entrevistador Então, o maior do banco é esse, que fica a infra, é esse aqui da Asa Norte.

Entrevistado: Maior, você fala em distância, mas... Um quadro, por exemplo? Não, infra, hoje, ela é um terço da TI.

Entrevistador E você acredita que essa questão do home office, isso é benéfico um pouco para as pessoas? Mas isso acaba sendo uma dificuldade para o Devops, quando a gente fala de cultura e etc.

Entrevistado: Você tem visto aí, o Nubank acabou de anunciar o... Vamos chegar perto, gente, porque nós estamos tendo problema de cultura. É o tipo de coisa que, como indivíduo, hoje eu estou trabalhando de casa, hoje de manhã, à tarde eu vou estar no banco. Para o indivíduo, benéfico, para esse aspecto cultural que eu te demonstrei aqui, não tem ninguém para eu virar atrás e falar, tira uma dúvida aqui para mim, certo? A aproximação fica protocolar, então é um impacto negativo para a organização como um todo, para a engenharia de software como um todo.

Entrevistador Isso cria justamente isso que eu trouxe, esses mini silos. Eu tenho o meu silo aqui, porque eu estou na minha casa, estou sozinho, depois qualquer coisa eu mando a tarefa para você, ou protocolamente eu faço uma pergunta, uma comunicação.

Entrevistado: Diria para ti que é uma área de pesquisa que carece de muito investimento, pesquisar o impacto no trabalho remoto, na integração dos times de DevOps. Já tive discussões internas sobre isso, por exemplo, opa, a gente tem o GitLab, o GitLab é uma organização integralmente remota. Como que ela se preparou para ser integralmente remota? E aí tem um outro problema. Se ela fosse presencial e integrada, ela seria ainda melhor do que é hoje? Não dá para saber, porque você ou tem organizações híbridas ou integralmente remotas ou inteiramente presenciais, mas você não tem alguém que transitou, exceto durante a pandemia. E todo mundo que transitou durante a pandemia está identificando problemas e pedindo para ter um certo grau de presencialismo. E aí o modelo que tem sido estabelecido é o híbrido.

Entrevistador Que atende um pouquinho. É porque eu acho que muitas vezes os funcionários não entendem tão a profundo essa parte social, porque são pessoas construindo o software. É. Tem comunicação, colaboração, então não posso olhar só para esse outro lado.

Entrevistado: Ao longo dos últimos dois anos, eu tenho repetido continuamente aqui dentro do Banco do Brasil de que engenharia de software é uma ciência sociotécnica e nós negligenciamos às vezes o aspecto sócio.

Entrevistador: É o que eu mais discuto na minha pesquisa. Eu já começo as minhas apresentações trazendo esse ponto. Eu estou estudando o lado social. São pessoas. Eu quero engajar essas pessoas. Mas está ótimo.

P9

Pensando em times que não são multidisciplinares, vamos pensar só nos times de desenvolvimento, ou só o time... Tem lá todo mundo que é desenvolvedor Java em um time. É uma equipe clássica de infraestrutura. Nesses times, como você... A gente consegue observar esse senso de responsabilidade sobre o produto. No meu entendimento, é mais como transferência de trabalho. Fiz minha parte, agora vai para você. Ou você acha que pode ter um compartilhamento de responsabilidade de entrega entre eles, ou a responsabilidade é mais limitada naquela parte que eles tocam. **Aquilo que é importante para mim, é o que eu vou fazer, e vocês se viram com a parte de vocês. Ela é mais limitada.** Tipo, se der alguma coisa errada, a gente gasta mais tempo investigando do que sentando junto para tentar resolver, do que a culpa é de Dev. Não, a culpa não é minha.

Entrevistado: Existe isso, sim, e nós temos times ainda, como eu disse, metade ainda está desse jeito, separado, e é exatamente como você descreveu. Por exemplo, o time de desenvolvimento com relação ao negócio é eu implementei o software, se não está vendendo o produto, você que modelou o negócio errado, certo? Você que prospectou errado no mercado. E a mesma coisa quando ocorre um problema da aplicação em produção, a infraestrutura fala, você implementou errado, por isso que não escalou. Então, sim, esses times, eles tendem a olhar somente aquele pedaço e falar que a outra área gerou o problema e não se

responsabilizam pelo todo. Já os times que estão integralmente atuando juntos, eles têm a tendência, só que isso não é só colocar junto e já vira chave. É um processo de transformação de hábitos. Então, muda. Todo ser humano leva um tempo para mudar a maneira como enxerga o mundo. Então, nós temos times que estão em diversas etapas dessa jornada. A gente tem time que o time de desenvolvimento já começou a entender que é problema dele o market fit, e tem time que ainda está nesses pequenos silos internos. Foi colocado para trabalhar junto, só que a gente tem que fazer a gestão do processo de mudança de comportamento.

Entrevistador Sim. Voltando ali sobre as equipes de plataformas que a gente comentou, você acha que elas atuam mais como facilitadoras? Ou elas ensinam, promovem ferramentas, ajudam? Elas estão realmente querendo levar essa cultura do DevOps? Ou tem algumas delas ainda que têm o conhecimento de infraestrutura? Eles centralizam muito. Acredito que aí...

Entrevistado: Tem as duas coisas.

Entrevistador Tem as duas coisas, sim.

Entrevistado: Depende do nível da plataforma, da origem dela. Por exemplo, a de nuvem está muito perto da infraestrutura e ela tende a querer se defender do desenvolvedor. E tem outras que estão mais com a pegada de deixa eu facilitar. É um dilema da plataforma. Ela precisa facilitar e ao mesmo tempo ela precisa se defender do erro do time de desenvolvimento. Vou te dar um exemplo em larga escala e escala comercial. A AWS eventualmente cai um datacenter. Mas ela não é processada para cobrir os danos do business que ficou fora quando ela caiu. Sabe por quê? Porque ela tem serviços com N custos e alternativas e orientações de o que você precisa fazer para o seu serviço não cair quando o meu datacenter cair. E se você não comprou e não investiu nisso, seu business ficou fora porque você resolveu comprar o risco e não fazer esse investimento. Então por isso ela pode realmente ter um datacenter inteiro que caiu fora e ela sair **incólume** em termos de processo. Porque ela tem alternativas. Esse é um desafio de um time de plataforma. Nessa relação em quem consome. Então existe um nível de qual é a culpa se joga de um lado para o outro.

Entrevistador: E como que a gente lida com esse sentimento de dono ou de compartilhar o ownership de um produto por parte de desenvolvedor ou por parte de duas equipes diferentes? Uma de plataforma e uma **Stream Aligned**, uma de produto e etc. Vai ter um momento ali em que essa é só minha parte, essa é só sua, não é minha responsabilidade ou por ter uma equipe de plataforma eu também tenho esse senso de compartilhar.

Entrevistado: Qual é o produto? A plataforma tem um produto que é a plataforma. E realmente a plataforma não vai ter o senso de propriedade para os produtos do banco. Ela tem um

produto que é entregar performance de engenharia para os times de engenharia. Esse é o produto dela. **Sim. O senso compartilhado por um produto igual o CDC ele ocorre para os atores que estão ali dentro da linha de produto do CDC.** A plataforma realmente não entra nisso. Dá para notar isso. **Uma plataforma dentro da empresa não se sente dona de todos os produtos que estão dentro da empresa.**