

Entrevistador: Pronto. Está gravando. Então, de novo, eu queria te agradecer por essa disponibilidade. A ideia é que a gente faça mais uma discussão. Eu vou compartilhar aqui minha tela. Bom, Leo, eu vou contar para você mais ou menos o que eu já trabalhei.

Entrevistador: No primeiro artigo, chegamos nesse modelo de UML, que harmoniza todo aquele conhecimento. Ainda, durante o desenvolvimento desse método, já executamos parte da operacionalização, ou seja, de pegar a teoria conceitual e colocá-la na prática, para fazer um processo sistemático de teste. Então, a gente já tinha conceitos e proposições.

Entrevistador: Aqui na operacionalização, só precisou criar os constructos e hipóteses. E aí, como tem funcionado? Eu já tive, já fiz uma rodada de validação. A ideia é que eu faça essas rodadas de validações com experts no tema, então, você, eu conversei com o Manoel Paes naquela época, e outros dois experts do mercado.

Entrevistador: A partir da primeira rodada de testes, eles me recomendaram alguns refinamentos, como, por exemplo, nos times horizontais, que têm o Enabler e o Bridget. Eles me recomendaram adicionar o Customer Focus, porque eles têm um foco no cliente, mas é um cliente interno. Esses times de DevOps, horizontais, e principalmente o de plataforma, que é o Enabler. E então eles têm uma parte de management também, dessa atividade de *manage*.

Entrevistador: E o que acontece? Aí a gente chega na versão atualizada, que eu tenho dessa teoria. E aí eu queria, na primeira fase, discutir com você. Então, primeiro eu vou apresentar. E aí eu discuto com você, mais ou menos, o que você acha dessa dinâmica, essa forma de representar, desde as instâncias que eu faço também.

Entrevistador: Então, só para te ajudar a lembrar. Tudo bem até aqui?

Entrevistado (Leo): Aham. Sim.

Entrevistador: Tá. Então, lá quando a gente harmonizou, fez aquele artigo, a gente disse que o software, um sistema de software, ele é definido por uma ou várias atividades. Dessas atividades, a gente tem automações do ciclo de vida ou da infraestrutura. Outra atividade entra também o *management*, que eu consigo falar de *Product Management* ou de *Project Management*.

Entrevistador: O *Product Management*, ele está mais relacionado ao time de produto. E agora também ele está um pouco relacionado ao time horizontal, que aí é que entra os dois times de DevOps, que eu identifiquei na época. Um com mais informações e mais forte, e o outro um pouco menos, mas os dois, na época, a gente decidiu que entraria.

Entrevistador: E o, o essas atividades, quem executa, são atores, que no caso são os nossos times. E depois disso eu tenho as especificações desses times. E o que é interessante a partir disso daqui? Eu consigo instanciar, por exemplo, um *Product Team*. Então, o *Product Team*, eu consigo dizer que ele tem uma autonomia, que ele normalmente não tem um *blame*, que o

alinhamento entre Dev e Ops dele é *Product Thinking*. Então, o foco deles é no produto, não é otimização local, como acontece em outros.

Entrevistador: O nível de responsabilidade e compartilhamento dele é *medium* ou completo. Eles fazem, eles têm uma colaboração. Uma evolução importante em relação à colaboração é que a colaboração, ela tem frequência, qualidade, só que ela também precisa ter uma motivação, uma duração, uma definição de pronto. Que, segundo o Manoel Paes, se eu só falo, ah, que um time, ele colabora, mas sem ter um objetivo, a colaboração, ela é muito cara para eu deixar ela aberta, sem um critério de parada, ou até quando a gente vai. Porque, normalmente, eu vou estar tirando um pouco do que cada time deveria estar focando, porque agora eles estão colaborando para resolver alguma coisa, ou principalmente, automatizar.

Entrevistador: O time também agora, ele tem uma carga cognitiva, porque isso é importante, porque algumas coisas podem reduzir essa carga cognitiva, como, por exemplo, se eu proponho, ou se eu tenho um time de plataforma, que dá um serviço de plataforma para o time de produto, isso, de certa forma, vai diminuir a carga cognitiva dele, porque ele vai ter que lidar com menos informações ou menos contexto.

Entrevistador: Então, por exemplo, se tem uma plataforma que auxilia na parte operacional, o time de produto, ele pode usar isso como um serviço. E, então, ele não precisa. Ele sabe como aquele ele funciona, mas ele não precisa aprofundar e desenvolver tudo aquilo. Então, isso reduz um pouco a carga cognitiva dele. Ele tem uma comunicação mais frequente, e etc.

Entrevistador: Outra estrutura que tem é o Dev e o Ops. Como você chama de silos?

Entrevistado (Leo): É, volta, volta no primeiro, eu não tinha sacado. Todo o primeiro é o quê?

Entrevistador: Todo esse primeiro é a instanciación do *Product Team*. O time de produto, ou aquele time que ele tem acesso a tudo o que é necessário, ele é independente e multidisciplinar, então vai ter gente de Dev, vai ter gente de Ops, vai ter toda uma equipe que é necessária para ele ser independente e desenvolver o produto de ponta a ponta e ficar responsável por isso.

Entrevistador: Tá. E aí dessa mesma forma, Leo, a intenção foi instanciar as quatro estruturas comuns que a gente identificou. Então, depois, os times em silo, aqui já na segunda versão atualizada, que a gente tem o time de Dev transferindo o trabalho para o time de Ops. Eles têm uma colaboração, motivação e etc. Eles têm uma comunicação também, mas a comunicação deles é mais baixa. Aqui eu coloco, é rara ou pobre, entre aspas.

Entrevistador: Aí a gente entra no *Enabler Team*. Que o *Enabler Team*, ele nada mais é do que um time horizontal. Quando eu falo time horizontal, ele é cross-funcional, e ele é mais ou menos um time interno que foca no cliente interno, ele não oferece serviços para o cliente final. E o *Enabler Team*, além de promover uma plataforma, ele vai realizar consultoria, treinamento e evangelização. Então, ele é o time de plataforma que a gente conhece, ou o SRE da Google

também. Ele trabalha com essas automações, tanto da infraestrutura quanto do ciclo de vida. E tudo isso daqui volta para prover o serviço de plataforma.

Entrevistado (Leo): Só uma observação, não sei se é pertinente ou não, mas talvez você fale aí um *label* aí na associação do *Enabler Team* para a plataforma.

Entrevistador: Ah, esse aqui.

Entrevistado (Leo): É. Você falou "promove" ou coisa assim.

Entrevistador: Boa, obrigado.

Entrevistador: Bom. E o último, que é o *Bridget Team*, eu identifiquei esse daqui a princípio em dois artigos. Teve uma pessoa que comentou sobre ele também, é um time que pode trabalhar/colaborar no time de desenvolvimento e às vezes no time de operação.

Entrevistador: Então, ele leva esse nome porque ele faz a ponte entre esses dois. O foco dele não é tanto em promover consultoria, treinamento, evangelização, mas é ajudar a automatizar e fazer essa relação entre esses dois times. Então, ele ajuda muito aqui o time de operação, mas ele também provê um serviço de plataforma que faz essas automações, tá?

Entrevistador: O que é, e aí, o que eu, o que você acha desse modelo?

Entrevistado (Leo): Me surpreendeu ter a plataforma também no *Bridget Team*. Ele também é um cara que eu não vi muito. Até esperava ver na minha pesquisa, mas não vi. Mas aí ele apareceu aqui com *platform service*, só porque aí que me surpreendeu.

Mas, mas eu saquei a ideia, é muito boa. Tem todo um modelo com propriedades e opções para essas propriedades. E aí a instanciação é, seria, aquelas estruturas, mas aí você caracteriza bem formalmente cada estrutura, usando a mesma régua para medir elas, o mesmo *template*, o gabarito, e aí vai ajustando os valores. Então, faz, faz muito sentido.

Entrevistado (Leo): E, e no geral, parece bem, bem corretas as as propriedades aí, as relações e tal.

Entrevistador: Se vamos discutir um pouco mais sobre o *Bridget Team*. Você acha que ele não provê serviço de plataforma? Isso fica única e exclusivamente do *Enabler* focando mais nisso?

Entrevistado (Leo): É, não, não que eu acho que não proveja. Mas não esperaria. Porque esse *Bridget Team*, do jeito que eu imaginava, era algo mais do início do DevOps. O pessoal tentando aproximar os dois mundos, então criava um time, tinha até o Jazz Humble que tinha crítica a esse time. E aí, quando eu fui fazer a minha pesquisa, eu não achei, ou pelo menos eu não interpretei que não achei, que talvez um caso ou outro poderia, talvez, até ter forçado para classificar assim, mas acho que seria um pouco de qualquer forma.

COMENTARIO:

Entrevistado (Leo): Eu enxergava isso como uma coisa mais do início do DevOps. Enquanto, eu vejo a plataforma como o ápice, até o momento, o fim da história do DevOps. Então, fiquei um pouco surpreso, mas, de novo, você que viu os casos concretos, que levaram a essa modelagem. E esse *Bridget Team* acabou até porque eu não vi ficando de fora lá, lá do meu modelo. Então, eu não tô dizendo que eu acho que é errado, eu tô dizendo que fiquei um pouco surpreso.

Entrevistador: Tá. Então, eu gosto do questionamento sobre isso, porque a princípio a gente teve poucas referências para chegar, que falavam sobre o *Bridget Team*. Se eu não me engano, de 10 documentos, dois ou três citavam essa estrutura. Então, é bom você discutir isso comigo, porque eu consigo levar isso em uma nova versão, ou uma nova entrevista, e ver se isso se mantém, ou se isso é algo que realmente ficou no passado, e vale a pena eu ajustar a teoria e remover essa parte, esse time, ou não dar tanta importância para ele.

Entrevistado (Leo): Uhum.

Entrevistador: Bom, uma coisa que é importante dessa estrutura, para mim, é que a gente consegue, eu consigo dar uma visão para as pessoas do mercado, de que elas consigam instanciar, identificar em qual estrutura elas estão aqui agora, e identificar, principalmente, quais as nuances que ela precisa fazer para ir para o próximo nível. Ou quais práticas eu tenho que mudar. Se eu preciso, eu quero ir para o próximo nível, então eu preciso mudar a autonomia do meu time, e trabalhar para conseguir ajustar essas variáveis do meu time, e aí sim, eu conseguir identificar ele no, no outro nível. Então, se eu tenho um modelo só, que eu consigo acompanhar ou me ver aqui, depois eu, ah, e ter um caminho mais direto do que eu tenho que fazer para chegar no outro nível, a gente acha que isso ajudaria.

Entrevistado (Leo): Uhum. Sim.

Entrevistado (Leo): Eh, é, no caso, o conceito de nível, não sei se eu entendi. Mas, pela minha interpretação, que talvez não seria bem nível, mas eu enxerguei que eu me encaixo, eu sou um *Enabler Team*. Então, do *Enabler Team*, vamos ver as propriedades. Então, dá um zoom aí, por favor, no time. Então, a autonomia devia ser *self-organizável*. Ah, o meu é, beleza. *Blame*, vixe, aqui é *true*. Então, dá *check* em tudo, mas esse, só esse cara ficou diferente. Então, esse *diff*, do mundo ideal, pelo menos aqui formalizado, e da minha realidade, onde tem *diff*, é onde talvez, provavelmente, eu deveria pensar em trabalhar.

Entrevistador: Exato. Exatamente essa é a ideia.

Entrevistado (Leo): Uhum. Isso. Então, só para concluir, porque com o nível eu até no primeiro instante, pensei, será que está falando que cada uma dessas coisas aqui seria o nível? Mas não é, é isso aí mesmo que a gente fala.

Entrevistador: É, é tipo o nível de maturidade. Ah, eu sou um *Enabler Team*, mas ainda tem coisas que dá para eu melhorar a maturidade do meu time, ou deixar o time mais redondo. Por exemplo, tem *blame* no meu time, então eu posso ajustar porque o melhor dos casos, no *Enabler Team*, é melhor que não tenha *blame*.

Entrevistado (Leo): Entendi. Mas, de novo, talvez cuidado com a palavra "níveis" e "maturidade". Porque tem toda a coisa dos níveis de maturidade de CMMI, que já é bem consolidado na cabeça das pessoas, de escadinha, então que você teria degraus de referência, que não é o caso aqui. Então, talvez só, depois que eu, que eu entendi a ideia, mas talvez melhor colocar com outras palavras para as pessoas.

Entrevistador: Tá. Tá bom. Obrigado. Eh, bom, a minha intenção inicial era isso, era uma contextualização, e a gente discutir por cima essas relações, esses movimentos. A partir disso daqui, na operacionalização, é quando eu consigo chegar em proposições, e as proposições são relações entre dois conceitos que eu consigo descrever.

Entrevistador: E quando eu falo isso, é para facilitar na hora de testar. Eu consigo ir em nuances ou pontos muito específicos de relações, como no caso desse daqui. Um time que tem uma cultura baseada em compartilhar responsabilidade e conhecimento, isso vai facilitar esse time a colaborar.

Entrevistador: Então, eu identifiquei essa relação em todos os trabalhos anteriores, e a ideia é que eu consiga validar se essa proposição é verdade ou não. E eu gero as hipóteses dela com base em hipóteses não causais. Porque, nesse sentido, é difícil ficar testando causalidade. Porque, por exemplo, ah, eu sei, aqui no, mais genérico, que se um time, ele tem mais responsabilidade, isso vai habilitar a colaboração, mas não é só isso que habilita a colaboração.

Entrevistador: Então, eu gero as hipóteses não causais, porque, com as hipóteses não causais, eu consigo ver, por exemplo, que um time que é *full sharing*, ou seja, ele compartilha muita responsabilidade, isso não é um problema para eles, torna-se possível eu ter uma relação com uma colaboração de frequência diária e de alta qualidade.

Entrevistador: Então, eu consigo ver essas coisas nas nuances. Então, só para você entender mais ou menos como que funciona o meu teste, ele é para ir a pequenas partes da minha operacionalização, ou do meu teste em si, tá?

Fase 2:

Entrevistador: Nesta segunda fase, vou fazer algumas perguntas que são relacionadas a essa relação entre conceitos, chamamos de proposições.

Pergunta (P3):

Em sua experiência, você já observou a relação entre **automated lifecycle management (ALM)** e a **colaboração entre times**? Existe uma influência entre eles?

Entrevistador: E a primeira é tentar relacionar o conceito de que, se eu tenho um ciclo de vida automatizado, então, por exemplo, isso daqui, se eu, na minha estrutura de time, eu consigo ter

uma infraestrutura automatizada, um *management Automated Lifecycle Management*, no caso, desculpa, esse daqui, isso vai facilitar ou tem alguma influência na hora que dois times vão colaborar. Porque, se eu tenho algo automatizado, isso facilita que o time de operação e o time de desenvolvimento tenham mais facilidade na hora que eles vão colaborar. Você já viu alguma coisa sobre essa relação? Você acha que pode aumentar ou pode diminuir a necessidade de colaborar ou a frequência também?

Entrevistado (Leo): Uhum. Então, primeiro vamos ver se eu entendi o que é o *Automated Application Life Cycle*. Quando você mandou o seu arquivo, eu até lembro que eu li, entendi, mas agora, nem sei se eu lembro. Mas eu entendi que é coisas do tipo, ah, usar o Git, versionar as coisas, por aí? Ou, como é que você complementaria?

Entrevistador: É, ou propor automações, por exemplo, fazer CI/CD, ter um monitoramento contínuo, dos testes também. Então, são ferramentas que facilitam essa automação do ciclo de vida, desde o desenvolvimento até ele em produção, manter isso em produção e etc. Se isso facilitaria que dois times colaborem para resolver algum problema, etc.

Entrevistado (Leo): Tá. Aí eu vou aproveitar e vou colocar outra coisa que eu fiquei com uma pulga da primeira coisa que você mostrou lá do documento, que seria a definição de colaboração. E, e até para provocar um pouco, volta lá no documento.

Entrevistador: Esse daqui?

Entrevistado (Leo): É, tem que, é, aí, aí mesmo. Tem que, tem que entender bem aqui o conceito de colaboração. Porque, se não, talvez seja meio tautológico esse P1 aí. Porque, se você tá compartilhando, um conhecimento... é, porque o que eu tô validando agora é a dois.

Entrevistador: Opa, não, desculpa, é a três, proposição três.

Entrevistado (Leo): Ah, tá. Mas vamos ficar no um mesmo. Volta lá no um. Porque, se você compartilha alguma coisa, você já estaria colaborando, né? Se eu estou compartilhando *ownership*, eu já, por definição, eu já não estaria colaborando? Então, por isso a dúvida da definição de colaboração, até para não ficar tautológico isso aí.

Entrevistador: Tá. A colaboração para mim é entre dois times. Então, aqui, por exemplo, o time, ele colabora com outro time. E se esse time, ele já tem o costume de compartilhar conhecimento internamente, para ele vai ser mais fácil ter uma colaboração, e isso vai facilitar na hora de colaborar, que eles já estão mais abertos a trocar conhecimentos e etc.

Entrevistado (Leo): Entendi. Era compartilhar o conhecimento internamente, não entre si.

Entrevistador: É, como uma cultura interna do time.

Entrevistado (Leo): Uhum. Beleza. Então, a resposta é, eu acho que sim, facilita a colaboração. Porque colaboração não é só uma conversa, não é só você ficar conversando tudo. A ideia é que, quanto mais coisas você tem registradas no Git, no CI, mais respostas eu e

no meu time eu tenho sem precisar estar ligando e falando com a outra pessoa. Eu consigo ter mais contexto quando necessário. E aí, se, eventualmente, eu tenho que ir para colaboração, eu até já vou com mais contexto, com insumos para perguntas.

Entrevistado (Leo): E, sei lá, eu tô lembrando agora de uma ferramenta de monitoração que tem na empresa, que a gente pede para monitorar, e até a anterior, a gente, desenvolvedor, fazer o pedido para monitoração e a gente não tinha acesso à interface da ferramenta. E era bem ruim, isso dificultava a colaboração, essa colaboração, ou essa interação, pelo menos, não sei se é colaboração, com o outro time. Porque, sei lá, eu vou pedir uma coisa, mas eu nem sei o que tem cadastrado, porque eu não consigo ver as coisas cadastradas, eu só sei o que eu pedi.

Entrevistado (Leo): Então, isso até, às vezes, pode deixar o meu pedido até mais pobre ou impreciso. Então, ter essas ferramentas e, claro, essas ferramentas terem, tem uma, pelo menos, visão compartilhada com as duas partes que estão interagindo. Porque, se for só para um, aí não adianta. Mas, aí assumindo, que tem o *automated life cycle*, e que o que tem de *automated life cycle* é visível para as duas partes colaborando, então isso facilita a colaboração, sim.

Entrevistador: Então, você diria que não facilita só a colaboração, mas a comunicação também.

Entrevistado (Leo): Vixe, aí você vai ter que me dizer qual é a diferença.

Entrevistador: Como foi no seu caso, a comunicação eu não preciso trabalhar junto com o time, mas eu preciso ter uma informação que o outro time só tem para ele. Se eu tenho uma automação, e que essa automação é visível, melhora a comunicação pontual ali, que um time pode precisar com o outro.

Entrevistado (Leo): Uhum. Isso.

PERGUNTA PT2. E O INVERSO?

Entrevistador: E, no contrário, você acha que, se eu tenho uma colaboração, mais ou menos boa, de qualidade e com uma frequência OK, isso, com isso eu consigo automatizar melhor o meu ciclo de vida? É uma relação de mão dupla? Se eu tenho algo automatizado, facilita a colaboração, mas, ao mesmo tempo, se eu tô apto a colaborar, e se eu faço colaboração, isso automaticamente vai implicar, porque eu vou gerar automações, e facilitar o fluxo e o ciclo de vida do produto.

Entrevistado (Leo): Diria que não, com asterisco. Porque, não no caso geral. Por um lado, até olhando para a história do DevOps, talvez a colaboração veio primeiro, e com essa colaboração nasceram as ferramentas, nasceu o *life cycle*. Então, olhando para a história, o que aconteceu, parece que sim, mas aconteceu aqui numa partezinha do mundo e foi se expandindo. Mas, em vários outros lugares, às vezes, você até tinha uma boa colaboração, sei lá, com a gente, lá do servidor, às vezes não gostava, mas às vezes gostava do cara lá do

banco, interage bem com ele, mas estava todo mundo num *mindset* que as coisas eram daquele jeito e não necessariamente disso ia brotar evoluções de automação.

Entrevistado (Leo): Em algum lugar brotou, mas em outros, não. Então, não necessariamente uma coisa leva a outra. Talvez seja não uma condição. Para que essas ferramentas pudessem ter se desenvolvido, a colaboração foi necessária. Mas seria mais de ser uma condição necessária do que uma suficiente ou que tenderia a levar. Então, no sentido de causa, não diria que causa, diria que seria condição.

Pergunta (P15):

Você acredita que ao ter AUTOMATED INFRASTRUCTURE MANAGEMENT facilita a ter uma ou melhorar responsibility/ownership sharing?

Entrevistador: Você acha que, se eu tenho uma infraestrutura compartilhada, ou um *Automated Infrastructure Management*, se isso facilita a que dois times possam compartilhar responsabilidade, ou o *ownership* entre eles? Se eu tenho uma automação disso no meio?

Entrevistado (Leo): Diria que sim, tentando elaborar, mas é muito parecido com o outro. Se você não tem essa automação, a visão, um lado faz uma coisa, é como que o outro vai saber que fez? Se não tem uma coisa automatizada. Então, um dos lado que fez, vai ter que botar isso num, num documento, numa documentação, numa planilha, e eventualmente, vai esquecer, vai ficar defasado e isso vai, vai empobrecer aí a colaboração. Então, é isso aí.

Entrevistador: Tá. O que eu tava identificando com isso é que, se eu tenho um gerenciamento dessa automação da infraestrutura aqui no centro, eu já teria algo que é transparente para os dois lados, e isso poderia facilitar que eles compartilhem conhecimento, mas não diretamente, então? Isso pode ajudar, mas não é uma ajuda tão forte, no seu ponto de vista, e que depende de outras coisas também.

Entrevistado (Leo): Não, certamente. Porque, às vezes, o pessoal da Infra fez lá uma automação, consegue ver, mas, o pessoal do desenvolvimento não interessa, eles não querem ver, que é só continuar pedindo. Mas, normalmente, eu acho que, se você quiser trazer valor, vai acabar querendo ver. Mas acho que a tendência é que é, sim, que uma coisa acabe ajudando, promovendo a outra, sim.

Entrevistador: Boa.

Pergunta (P16):

E se eu expando isso para automatizar todo o ciclo de vida da aplicação, você acha que isso impacta no compartilhamento de responsabilidade entre os times?

Entrevistador: E aí, do próximo, então você já comentou também, que se eu automatizar todo o ciclo de vida, isso também vai facilitar a compartilhar a responsabilidade entre esses times.

Pergunta (P30):

Manoel e outros entrevistados me informaram que uma colaboração de alta qualidade entre dois times, foca em resolver ou automatizar um gargalo entre dois times. Você concorda que ao colaborar eles reduzem essa dependência e deixam o time mais autônomo?

Entrevistador: A terceira, que ela se repete ali embaixo também, é que o Manoel e outros entrevistados, eles me informaram que uma colaboração de alta qualidade entre dois times, deixa eu voltar para cá pra gente falar aqui. Uma, então, uma colaboração que ela tem uma qualidade alta entre dois times, ela tá mais focada quando esses times querem resolver um gargalo entre eles.

Entrevistador: Tem dois times que está com um gargalo, tá tendo transferência de tarefa, eles querem automatizar isso. Você concorda, então, que, quando eles automatizam, ou quando eles colaboram, eles reduzem essa dependência que um time tem do outro? Se isso é algo direto que pode ser um fruto da colaboração? Porque, para o Manoel, ele é muito incisivo, a colaboração, ela só deve ser feita para eu resolver o problema entre dois times. Qualquer outro tipo de colaboração é inútil. Porque, um time que, ah, eu vou colaborar com outro time para o resto da minha vida. Para ele, isso gasta muito dinheiro e não vale a pena.

Entrevistado (Leo): Eh, só que, bom, a segunda vez que você falou, deu mais *fit* aí para mim. Mas, por que problema? Assim, qualquer problema? Porque, do jeito que você falou a primeira vez, até pensei num contraexemplo. Eu tenho meu time que consome um outro serviço, que pode ser de plataforma ou outro serviço mesmo de outro time de produto. E o outro produto está com problema lá. Então, eu vou colaborar, interagir com eles, para resolver esse problema. Por que o meu serviço não está conseguindo chamar o outro? Por que está dando problema nessa situação?

Entrevistado (Leo): E aí, é um problema, é uma colaboração pontual, como o *Team Topology* defende que seja. Mas não, aí, nesse caso, necessariamente, no sentido de reduzir a dependência de um para o outro. Então, essa coisa que toda colaboração seria para diminuir a dependência dos times.

Entrevistador: E deixar eles mais autônomos, assim? Porque, se eu já automatizei essa parte que tava sendo um problema, e eu tinha que estar sempre falando com esse outro time, se a gente junta e automatiza isso, eu consigo voltar e focar nas minhas coisas, gastar a minha carga cognitiva com os produtos para o cliente que eu preciso entregar, por exemplo.

Entrevistado (Leo): É que, aí, eu tô supondo que não é um problema de falta de automação. É só um problema de, de repente, eu comecei a usar o serviço dele de um jeito que eles não

esperavam ser usado. Encontrei um *bug*, ou está instável. Comecei a usar com uma carga alta que eles não esperavam. Provavelmente, alguma dificuldade que antes funcionava entre os nossos serviços, agora não mais. A gente vai ter que resolver de um jeito, vai ter que conversar para ver como resolve. Uma vez, então, a gente tem um nível de acoplamento tal. Surgiu um problema, a gente teve que conversar, se juntar nessa missão para resolver, então, colaboração por tempo determinado que eles defendem. Mas, uma vez resolvido esse problema, o nível de acoplamento depois é o mesmo de antes, de surgir o problema. Então, a gente não colaborou para automatizar, mas para diminuir o acoplamento.

Entrevistador: Entendi. Então, o acoplamento, ele se mantém, faz muito sentido. O acoplamento, ele se mantém. Então, a colaboração, nesse caso, foi para resolver um problema pontual, que estava atrapalhando um dos times. Mas eles, isso não aumenta a independência dele.

Entrevistado (Leo): Isso.

Entrevistado (Leo): E aí, nesse sentido, os times, essa coisa de eles colaborarem para sempre, não é que eles estão continuamente colaborando para sempre, mas para sempre eles estão, tem um interligado um com os outros, de forma que, quando um tiver o problema, um vai ter que interagir com o outro. Então, eles têm um acoplamento, que é, entre aspas, eterno, mas não quer dizer que eles têm que estar colaborando sempre, eles só vão colaborar, quando tiver algum tipo de problema e que um possa ajudar o outro a resolver.

COMENTÁRIO: Me parece que precisa ter uma noção geral ou um alinhamento geral, que não é ligada a ficar colaborando. Mas sim ter uma boa comunicação entre os times. Aqui o leo pode ter confundido os conceitos deveria ter explicado melhor.

Entrevistador: Beleza. Não é algo tão à ponta da letra, como o Manoel leva, de que, ah, eu primeiro tenho que definir um contrato, porque, sem definir um contrato, nem vale a pena colocar esses dois times para trabalhar.

Entrevistado (Leo): É, bom, é, não, tão formalmente. Você tem, tem, é bom, eu gosto da visão deles, você tem que ter um objetivo claro da colaboração. Por que está colaborando? O contrato parece meio formal demais.

Entrevistador: Sim. Tá. Bom, beleza.

Pergunta (P5):

Você acha que se eu tenho um time cross-functional/skills em uma equipe influenciam o nível de colaboração (frequência e qualidade)?

Entrevistador: E você acha, deixa eu mostrar um exemplo aqui que a gente tem, um time cross-funcional. Esse daqui. Bom, se eu tenho um time que é cross-funcional, em uma equipe,

isso vai influenciar quando ele for colaborar, na frequência, com o outro time, e esse outro time, independente se ele é cross-funcional ou se não é cross-funcional? Você acha que, se por o time ser cross-funcional, ter várias pessoas com vários conhecimentos amplos ali, isso facilita na hora de colaborar, por exemplo, com um time que é só de desenvolvimento?

Entrevistado (Leo): Não sei. Tô achando que não, que vai ser algo muito caso a caso aí.

Entrevistador: Se esse time, ele é cross-funcional, significa que eles têm várias *expertises* ali dentro. E quando ele for precisar resolver algum outro problema, ele já tem essa *expertise* de ponta a ponta do que eles fazem. Talvez, isso pode melhorar ou facilitar a colaboração, ele chegar do outro lado mais rápido, ou algo assim.

Entrevistado (Leo): Vou colocar outro lado que diria o contrário. Tem um colega lá da, da minha equipe, que, ah, eu tenho duas soluções, A ou B. A solução A a gente resolve aqui, a B tem que falar com alguém de fora da equipe. Quero a A, não quero falar com alguém de fora da equipe. Então, às vezes, o time cross-funcional, ele é autossuficiente. E ele é um time já, que era um caso antigo, bem estável, é uma família. A gente é uma família lá, a gente se conhece, entende bem. Então, tem uma super fluência entre a gente.

Entrevistado (Leo): E o cross-funcional é bem isso, ele é autossuficiente. Então, ele está acostumado a se resolver por si só. Então, quando tem que ir para fora, é uma coisa diferente, é um custo diferente, é um gasto de energia a mais, que eu tô acostumado aqui a conversar com os coleguinhas, com a família aqui, eu vou ter que sair de casa lá para falar com o outro. Então, talvez, até deixaria as pessoas mais acomodadas a não ter que interagir com desconhecidos, desconhecidos, entre aspas, mas pessoas menos frequentes. Então, poderia ser até um fator para inibir a colaboração.

Entrevistador: Entendi. Iria para o lado oposto, então.

Entrevistado (Leo): É. Então, não, não colocaria uma relação.

Pergunta (P17):

Na sua experiência, você já identificou que equipes CROSS-FUNCTIONALITY/SKILLS tem mais facilidade em compartilhar conhecimento?

Entrevistador: Tá bom. E você acha que esses times, que ele é cross-funcional, ele tem mais facilidade em compartilhar conhecimento?

Entrevistado (Leo): Também não colocaria a relação. Porque, se ele é cross-funcional, olha só, a ideia é que justamente ele tem vários *roles* dentro. Você tem o, sei lá, dependendo do caso extremo, você tem, o time já tem um cara de banco, mas daí, provavelmente, vai ser um cara de banco só. O time já tem um UX, mas vai ser um UX só. Então, se o cara de banco é o

único da equipe, ele não precisa compartilhar tanto o conhecimento, talvez. Se a pessoa de UX é a única da equipe de UX, talvez não precisa compartilhar tanto o conhecimento técnico específico.

Entrevistador: Me, e se algum dia acontece, não, então, acaba, quando eu tenho um cross-funcional, eu acabo não tendo essa necessidade de, ah, todo mundo sabe de tudo, todo mundo está integrado de tudo. Se algum dia eu não conseguir estar aqui, as outras pessoas conseguem resolver o problema por mim, sei lá, fiquei doente, alguma coisa assim. Quando a gente tem um cross-funcional, isso pode acabar limitando um pouco também, então?

Entrevistado (Leo): Pois é, é um desafio do cross-funcional. Porque, se não cross-funcional, dependendo do que você entende por cross-funcional, que precisa ter no time, mas, se você quer ter essa diversidade grande de papéis, já é um desafio garantir que tem uma pessoa na equipe que mande de cada um dos assuntos, quanto mais dois, para um cobrir as férias do outro. Então, então, é.

Entrevistador: É uma relação que você não faria também, não, então?

Entrevistado (Leo): É, também não arriscaria fazer.

Pergunta (P18):

Você diria que os times Enabler (Platform) é caracterizado também por ser um time cross-funcional/skills?

Entrevistador: Você diria que os times Enabler plataforma, ele também é caracterizado por ser um time cross-funcional?

Entrevistado (Leo): Sim, sim, faz sentido. Porque é aquela coisa, a plataforma é um produto. Você precisa desenvolver, você precisa operar, você precisa gerenciar. Precisa ter UX, precisa ter banco. Então, faz sentido, sim.

Pergunta (P24):

Podemos afirmar que em uma estrutura de times que possui um Enabler (Platform) team, esses time promove self-organization e autonomia aos times de produto por exemplo?

Entrevistador: Tá. Beleza. E ainda um pouquinho sobre o Enabler, então, eu poderia afirmar que, se eu tenho uma estrutura de time que tem o Enabler, esse time, ele vai promover *self-organization* e ajudar a dar autonomia para um time de produto, por exemplo? Um dos objetivos dele é garantir que o outro time, ele tenha autonomia. E eu até diria que ajuda na carga cognitiva desse time.

Entrevistado (Leo): Sim, mas um pouco assim, abusando da palavra tautológico, pela própria definição que está de *Enabler Team* aí. De você ter o *consulting, training, evangelization*, provê a plataforma. Então, todas essas coisas são sobre isso, ajudar o outro a ter autonomia. Então, até pela definição que está assim, tem que ser isso.

Pergunta (P29):

Você acredita que prover Serviços de Plataforma (ou a Equipe de Plataforma) impacta a carga cognitiva das equipes de desenvolvimento ou produto? Ela ajuda a reduzir?

Pergunta (P31):

Ainda no tema das equipes mais “avançadas” do DevOps. Elas ajudam a reduzir a carga cognitiva das equipes?

Entrevistador: Ótimo. Tá bom. E aí, eu estendo isso de novo, que, se eu promovo, ou se eu coloco um serviço de plataforma para um time de produto, isso automaticamente reduz a carga cognitiva dele. Porque eu peguei parte da tarefa dele, que seria colocar em produção, tô fazendo e disponibilizando como serviço. Então, também, aproveitando para abusar da palavra, é tautológico da definição. Isso impacta diretamente, com certeza.

Entrevistado (Leo): Depende do ponto de partida. Em relação, de novo, tem aquilo lá, em relação ao que. Porque, se é a relação de antes é de algo que o produto, o time tinha que cuidar da infra, e tal, talvez reduza a carga. Mas, se for pensar do pessoal do silo, que antes, ah, eu só codava, eu só desenvolvia, e depois, para implantar, eu entregava lá, e o cara, mandava pela parede o pacote, e o pessoal de Ops, que se virava. Nesse caso, talvez, vai ser, mais coisa para aprender. Talvez vá, vá aumentar, até.

Entrevistador: Que ele teria que aprender a usar a plataforma e mais isso, que é, é mais responsabilidade para ele.

Entrevistado (Leo): Uhum.

Entrevistador: Bom, bem pontuado. Obrigado. Eh, bom, essa ideia desses equipes de DevOps, que elas ajudam a reduzir a carga cognitiva, beleza. E aí, eu, eu acho que serve, porque, para o Enabler, sim. E o que eu queria trazer é para o time horizontal, tanto o Enabler quanto o Bridget. Se esses dois, eles ajudam também a reduzir essa carga cognitiva. Aí, eu acho que dá para a gente voltar naquilo que você comentou sobre o Bridget antes, que ele é justamente só essa ponte. Então, talvez ele não reduz essa carga.

Entrevistado (Leo): Acho que não, colocaria isso para, para o Enabler. É, pelo contrário, é um cara que tá querendo mudar a cultura, trazendo novidades, que tá empurrando mais, mais coisa aí para a realidade das pessoas. Então, acho que não.

Entrevistador: Sim. É, o Enabler, ele empurra também, mas, ao mesmo tempo, ele ajuda o time, dependendo do time que ele tá ajudando. Eu queria, inclusive, aproveitar e fazer um parênteses. Você, nos seus estudos, você viu algum time Enabler, por exemplo, ajudando um time, ou em um contexto de uma empresa que ele tem só times de desenvolvimento, e ele tá começando a ajudar, ou essa plataforma, ela é especificamente direcionada para times de produto, que já são cross-funcionais, são responsáveis por lidar com os seus produtos de ponta a ponta? Ou você acha que essa plataforma, ou você já viu, deveria ter uma conexão aqui também, com o time de desenvolvimento? Porque, com a operação, ela já tem. Ele já ajuda o time de operação, de certa forma.

Entrevistado (Leo): Hum, difícil. Eu diria que sim, mas é que, talvez, o que era um time de desenvolvimento, conforme ele passa a usar o time de plataforma, ele vai se tornando cada vez mais um time de produto, eu diria, talvez.

Entrevistador: Tá. Mas, então, não necessariamente, por exemplo, se eu chego aqui, nessa estrutura de time de plataforma, o resto da minha equipe, o resto da minha empresa já tá organizado em time de produto. Não, necessariamente.

Entrevistado (Leo): Ah, tá, boa. É, porque tem a contra, eu tava pensando em produto, assim, no sentido de que não é DevOps. É, o cara já consegue operar, então é o time de produto. Mas, não sei se estaria pensando em contraposição a projeto. Você tem time de projeto, em vez de time de produto. Que time de produto é mais normal, mas, às vezes, você ainda tem.

Entrevistador: É. O time de projeto, eu relacionaria com o Dev aqui, por exemplo, que ele tem o *Product Project Management*, no *management*. E aí, ele tá ligado aqui. E aí, o *Product Management*, que tá ligado ao time de produto.

Entrevistado (Leo): Ah, tá, é, para ser, você tem, o desenvolvimento seria o time de projeto. É, ele pode usar a plataforma também.

Fase 3:

H2.2	Teams with eventual collaboration are associated with fewer organizational silos.
H2.4	Teams with low-quality collaboration are associated with fewer organizational silos.

Entrevistador: Bom, agora eu tenho algumas hipóteses que eu discuti no anterior, mas eu queria um segundo ponto de vista. E a primeira é, se eu tenho um time que, normalmente, ele tem só uma colaboração eventual. Então, ele não colabora muito. É eventual o tipo de

colaboração que ele tem no dia a dia. Então, ele não colabora. Isso tá associado com a redução de silos? Se um time colabora muito pouco?

Entrevistado (Leo): Tô pensando. É, porque quando você tem silos, a ideia é que eles colabo... é que eles, talvez, de colaborar muito pouco, eles colaboram, mas é uma colaboração atritosa. Talvez é mais a qualidade da colaboração do que a quantidade que caracteriza o silo. Aí, fala de novo a hipótese.

Entrevistador: A hipótese é, se um time com uma colaboração eventual, ele tá associado com redução de silos organizacionais.

Entrevistado (Leo): É, porque colaboração é, não diria. Tô achando que são umas coisas meio separadas, aí. Porque, também, colaboração com quem?

Entrevistador: Com o outro time. Então, tem vários times que estão acostumados, sei lá, a colaborar. A minha questão aqui é que eu acho que isso é uma hipótese falsa. Deixa eu te dar um *spoiler* assim, e eu só queria ver, ter um segundo ponto de vista. Porque eu acho que, se um time, ele tá acostumado, ou times que estão colaborando com colaborações muito eventuais, não é algo muito frequente, de qualidade, isso não tá associado à redução de silos organizacionais. Isso tá associado a silos organizacionais, não à redução, entendeu?

Entrevistado (Leo): Sim. No sentido, se eu colaboro pouco, é porque eu quero ficar na minha, eu fiquei que quer ficar na minha. Tô construindo um silo aqui. É.

Entrevistador: Será que dá, então, para eu inferir que essa é a relação, então? É, de associação?

Entrevistado (Leo): É, diria que, talvez, seja essa relação, ou não tem relação. Mas, o outro lado que você apresentou aí realmente não é. Porque, às vezes, pode ser que a colaboração é pouca, não por questão de silo, mas, às vezes, por essa questão que está tudo bem automatizado, já está tudo muito redondo, já os processos. Então, poderia ser isso, ou poderia ser pelo silo. Mas, de novo, talvez o que caracterize a mais o silo, pensando agora, não é nem tanto a quantidade de comunicação, de colaboração, mas, a qualidade dessa colaboração.

Entrevistador: Uma coisa que eu discuti muito com o Manoel é que a frequência da colaboração, ela vai ser muito definida pela motivação, a duração, o tempo que eles têm, e o critério de pronto. Então, a qualidade tem outros, algumas outras coisas que impactam, mas a frequência, eles vão definir de acordo com o que eles precisam. Se é algo urgente, a colaboração, a frequência da colaboração vai ser diária, até eles resolverem. Se não é tão urgente, pode ser mais espaçado. Faz sentido isso?

Entrevistado (Leo): É, mas, bom, vou, vou pensar aqui no silo tradicional, lá, do, tem o Dev, tem o Ops. Eles têm uma colaboração assim, a frequência é baixa, no sentido que você não está todo dia interagindo, mas é uma frequência com uma periodicidade firme. Cada vez que eu vou fazer um *deploy*, eu preciso de alguma coisa, então, eu vou estar sempre falando com eles. Não vou estar falando todo dia, mas é garantido que em algum momento eu vou precisar. Então, tem uma frequência baixa, mas é uma frequência que está lá, firme. Então, então, não sei o quê.

Entrevistador: A gente está, relacionando essa colaboração eventual ou baixa com a redução, ou se isso está ligado aos silos.

Entrevistado (Leo): Isso, então, eu tenho o silo. Então, eu tenho uma frequência baixa, mas persistente é um silo. Só que a qualidade, geralmente, é baixa. Porque, aí, tem aqueles atritos e tal. E a motivação seria o procedimento. Que para fazer tal coisa, eu preciso do outro. E aí, ah, mas a duração. Então, o que está, seria a frequência baixa, mas com uma duração eterna. A gente está sempre precisando.

Entrevistado (Leo): Então, talvez, assim, fique difícil pegar um desses atributos da colaboração e falar se isso leva ou não para o silo. E, na minha cabeça, talvez seja mais fácil pegar o contrário. Pegar o silo e dizer, ah, o silo tem uma colaboração assim, assado, com frequência baixa, qualidade baixa, duração alta. Quer dizer, o silo, é que o silo também pode ter, outro silo, no caso, mas pensando no silo no silo do Dev e do Ops, que um precisa do outro.

Entrevistador: Em, mas pode ser nesse exemplo do Dev e do Ops, mesmo. Aí, que a minha intenção, com esses pequenos nuances, é identificar se, realmente, tem uma associação. Se os times, com silos organizacionais, entre o Dev e o Ops, por exemplo, eles vão ter uma colaboração eventual, mas, como você comentou, constante, e pontual ali para resolver aquela coisinha e pronto, acabou aqui, eu não quero mais saber de você. De resto, a gente segue mandando tarefa um para o outro.

Entrevistado (Leo): Isso. Isso. Mas, eu acho que sim, que dá para caracterizar, essa colaboração entre silos, usando esses atributos. Mas, o que, talvez, não dê, que acho que era a hipótese, pegar um atributo só daqui e, desse atributo, traçar, se isso tende ou não para o silo, só com o atributo sozinho, talvez.

H2.4	Teams with low-quality collaboration are associated with fewer organizational silos.
------	--

Entrevistador: Tá bom. Ótimo. A segunda de, ah, cinco, tá? A segunda é, que, ah, é a mesma coisa, só que discutindo a baixa qualidade da colaboração. Então, se eu tenho o time de Dev e

Ops, que eles têm uma qualidade muito baixa de colaboração, isso tá associado com a redução de silos, ou tá associado a um silo organizacional em si? Que eu entendo da sua resposta é que, sim, que isso está associado com o silo. Poderia fazer essa associação, entendendo que isso não é a única coisa que vai caracterizar o silo, mas isso também ajuda a caracterizar.

Entrevistado (Leo): Uhum. Isso. Uhum.

H4.10	A team culture based on full knowledge sharing is associated with low quality collaboration between team members.
-------	---

Entrevistador: Tá. A próxima é, se eu tenho um time, que a cultura é baseada em compartilhar conhecimento, isso está associado com uma baixa qualidade na colaboração entre times?

Entrevistado (Leo): Não.

H9.2	Teams not characterized by cross-functionality/skills are associated with full responsibility/ownership sharing.
------	--

Entrevistador: Times que não têm cross-funcionalidades, eles estão associados com muito compartilhamento de responsabilidade?

Entrevistado (Leo): Sim, porque, se não é a ideia que não é cross-funcional, é porque não tem todas as habilidades necessárias no time. Se não tem no time, em algum momento, algumas coisas, eles vão precisar pedir ajuda de fora.

H10.1	Teams characterized by full responsibility/ownership sharing are associated with organizational silos.
-------	--

Entrevistador: Tá. Ótimo. E, por fim, times caracterizados por muito compartilhamento de responsabilidade e *ownership*, eles estão associados com silos organizacionais. Tem como fazer essa associação?

Entrevistado (Leo): Não. Porque o silo, ele estaria associado com, ter pouco compartilhamento.

Fase 4:

Siloed departments

lack of proper test automation		The model did not fit. Obs: Team could point to Culture_Values_Best_Practices, and then we could say about a team with Practices without continuous_testing.
--------------------------------	--	---

Entrevistador: Tá. Beleza. A gente já está chegando na terceira fase, então, a gente passou por esses. Agora eu vou colocar aqui o que a gente discutiu. E a gente entra na parte que, lembra que lá atrás, você e o Paulo pegaram para tentar instanciar aquela teoria que a gente gerou, lá no início? Aí, agora é hora de eu pegar aquela discussão que a gente fez e eu trazer para refinar a teoria, para adequar com os pontos que a gente não identificou. Então, eu peguei as questões que vocês levantaram. Então, por exemplo, sobre o departamento de silos, uma das suas propriedades é que "lack of proper test automation". E o modelo, ele não tinha isso. Mas, segundo você, a gente poderia atender essa propriedade se eu fizesse uma conexão entre o time e as culturas. Só voltar aqui para você lembrar. Então, tem uma conexão aqui do time, porque o time também teria várias culturas e valores culturais e boas práticas. Você acha que isso se mantém?

Entrevistado (Leo): Esse cara é meio difícil. Porque assim, na essência da coisa, não precisaria ser assim. Poderia ser silo e ter bastante automação de teste. Poderia ser silo e todo mundo valorizar as melhores práticas, cada um da sua função. Mas, é que isso foi uma coisa mais observada. Então, no universo que a gente observou, a gente observou que isso tende a para isso. Mas isso, talvez, no nosso contexto, era mais uma nota, uma observação marginal que a gente achou interessante, e que talvez, principalmente nesse seu caso, que é um modelo mais formal, não necessariamente precisaria entrar no modelo. Mas, se entrar no modelo, aí teria que deixar claro que os valores, na instanciação, é uma coisa *grounded*, com base no que foi visto, não que, necessariamente, precisaria ser assim.

Entrevistador: Beleza. Bom, então, talvez isso não, se eu adicionar é com aspas e descrevendo isso bem, talvez aqui na relação, alguma coisa aqui.

Entrevistado (Leo): Mas, o meu palpite é que, talvez, não, talvez, não valeria a pena.

Collaborating departments:

- Not enough to achieve high delivery performance (nossa teoria nao mostra e discute sobre delivery performance, talvez seja interessante adicionar. Onde o Leo adicionaria isso?)

infra as development collaborator	Team.responsibility/owners hip_sharing = full sharing	But it would be more precise with an additional optional "method" for Operation_team: contributes_to_the_product (or something like this)
-----------------------------------	--	---

Entrevistador: Tá. Beleza. Outra coisa é nos times de *Collaborating Departments*, que você discute muito *delivery performance*. Tanto é que o seu trabalho leva esse termo no título. "Infra as a developer collaborator". Então, você recomendou adicionar algum opcional no time de operação, que ele vai contribuir com o time de produto também, ou algo assim?

Entrevistado (Leo): Mas, talvez não precise. Porque, no fundo, esse "infra as a developer collaborator" é meio que o SRE. E, pelo menos, o tanto que você falou para mim, você encaixou o SRE lá no *Enabler Team*. Foi. A gente não tinha enxergado porque eu vi o *Enabler Team*, como o propósito dele é, é fazer o Enabler. Nesse caso, do Infra, o propósito do cara é Infra, mas ele vai, também, adicionalmente, tá lá colaborando. Então, por isso que, para a gente, eram coisas diferentes.

Entrevistado (Leo): Agora, se você diz que teria que voltar lá para justificar, mas que de alguma forma essa ideia do SRE está incluso no *Enabler Team*, então já estaria contemplado. Mas é que, de novo, no caso do SRE, o cara que é SRE do Google, ele tem, realmente, o objetivo de instanciar e cuidar da Infra. Ele vai lá e mexe no produto também. Não é exatamente o objetivo da vida dele, o *consulting, training, evangelization*, mas é coisas que ele, tem os caras da Infra, os caras que são SRE fazem isso a mais. É sempre um desafio, nesse *framework*, como encaixar o tal do SRE.

Platform teams

Product teams are fully accountable for the non-functional requirements of their services	Product_Team with access_to_all_necessary_information()	But this is just part. We could have something as Product_Team with operations_responsibility

Entrevistador: Sim. Beleza. De plataforma *team*, que você comentou que esses aqui é só uma parte, que o "product team is fully accountable for the non-functional requirements". Que, se eu tenho um *product team* com acesso a todas as informações necessárias, talvez isso já seja suficiente, mas talvez eu possa refinar? Porque esse acesso aqui, a descrição dele é meio que superficial, pelo seu ponto de vista que você teve lá. Então, talvez, eu, e você recomendou que o *product team* com *operationalizations* ou *operations responsibility*. E aí, o que eu vejo de possível alteração, ou algo assim, é fazer ajustes nesse carinha aqui.

Entrevistado (Leo): Hum. É, porque isso, para a gente, lá, estava bem no cerne da definição das estruturas, essa coisa. Até no último, da conclusão, mesmo, da tese, eu faço um quadrinho, assim, de resumo de comparação. E, nesse quadrinho, é bem chave essa questão da operação. Com quem que está a operações. Então, se eu fosse formalizar, em termos de atributo, que nem você fez, talvez teria bem menos atributo, mas, para mim, esse seria um atributo bem chave. Quem é responsável pela operação.

Entrevistador: Entendi. É, eu entendo hoje que ele é, é sim, responsável. Ele usa a plataforma para facilitar a vida, mas se o sistema cai, alguma coisa assim, é, eles que vão ter que lidar. Esse daqui é só um suporte básico.

Entrevistado (Leo): Sim, certamente, é assim.

The platform itself handles many NFR concerns	Platform with continuous_monitoring and recovery_automation()	But not so precise. Maybe it could be something like Platform coping_with_nfr
---	---	---

Entrevistador: "The platform itself handles many non-functional requirements concerns". Então, a plataforma com *continuous monitoring* e *recovery*, mas isso está pouco. Eu preciso, talvez, especificar um pouco mais e adicionar mais nuances, como o "coping with non-functional requirement" ou algo assim.

Entrevistado (Leo): Mas, também, não sei se precisaria estar no seu modelo. Porque isso é meio que já está na definição do que é a plataforma. Porque, se, se voltar no seu modelo, você tem o time. O time tem diferentes instâncias. Que vai variar as propriedades. A plataforma não tem isso. Não tem instâncias da plataforma.

Entrevistador: Não, eu coloco ela muito mais ligada a essas atividades de automação, tem um interface, alguma coisa assim.

Entrevistado (Leo): Isso. Tanto que não tem atributo aí na plataforma. Qual que é a diferença?

Entrevistador: E aqui é que esses times, eles constroem essa plataforma, que vai ser provida como um serviço. Eu imagino que aqui é só um serviço que, dentro desse "*platform servicing*", eu tenho tudo isso daqui. Uma interface, uma ALM interface, uma *automated infrastructure interface*, e tudo isso é desenvolvido aqui para realizar essas tarefas.

Entrevistado (Leo): Uhum. Entendi. De qualquer forma, esse plataforma, *platform service*, eles não tem atributos. Porque aí, no, no seu conceito, não é instanciável que nem o time, que você vai ter várias coisas. A plataforma é plataforma. Então, aqui, o lado dos NFR, é só uma descrição do que é essa plataforma. Então, não enxergaria que precisaria estar no seu modelo.

The infrastructure team is no longer requested for operational tasks	The Bridge_Team automates_infrastructure_creation/deployment/management.	But it do not create/deploy/manage_infrastructure as the Operation_Team would do.
--	--	---

Entrevistador: Tá. Beleza. Eh, tá. Acabando. "The infrastructure team is not longer request for operational tasks." Então, o *Bridget Team*, ele automatiza isso, mas ele não cria, faz *deploy*, faz *management*, assim como o *operational team* deveria fazer.

Entrevistado (Leo): Eh, peraí que embolou um pouco. Porque, o que que são as colunas, mesmo? É porque também surgiu o *Bridget Team* aí no.

Entrevistador: Não, é que esse daqui é o, o que você usa para descrever o time. Esse daqui é tentando instanciar no meu modelo, como representar isso daqui lá, e esse último foi o comentário.

Entrevistado (Leo): Ah, tá. É, como eu falei, o *Bridget Team*, talvez eu não, não usaria o *Bridget Team*. Como, eu não vi na prática, lá, na minha pesquisa, eu não instanciaria, provavelmente, nada usando o *Bridget Team*. Mas, talvez, pensando aqui, talvez, isso impacte, seja a questão do tipo de colaboração. Você tem o time de produto, ou de desenvolvimento, e o time de infra, que vira o time de plataforma. O tipo de relação que eles têm, vai, vai mudar. Que os atributos da colaboração vão mudar.

Entrevistador: É que a responsabilidade dele não é mais criar isso, como era a responsabilidade do time de operação.

Entrevistado (Leo): Isso. Quer dizer que eles têm até uma frequência menor, mais pontual, só quando tem problema. Então, eu, talvez, isso aí, nas instanciações que você já fez dos vários tipos, eu imagino que já contemple essa questão. Porque, até já, já mudou o nome. Já vira um *Enabler Team*. Então, eu acho que seja contemplado no modelo.

avoids the need for product teams to have infrastructure specialists	Procut_Team with cross-functionality/skills = false	here I would prefer to say "low" rather than "false"
--	---	--

Entrevistador: Tá. Beleza. Eh, tô acabando. "Avoids the need for product team to have infrastructure specialists". Então, o meu *product team*, ele tem, não tem *cross-functionality skills*, é falso. E você não, acha que é melhor, talvez, eu tirar do booleano e colocar outros níveis, assim, para eu descrever esse caso? Que eu não tenho tanto especialista. Então, eu não teria só *cross-funcional*. É forte dizer que é falso, é melhor falar que é pouco. E aí, o que eu fiquei pensando é, então, se eu tiro do binário, eu poderia colocar um, a princípio, evoluir para ser três, um pouco, médio ou alto. E, conforme dá para evoluir a teoria, ajustando essas variáveis.

Entrevistado (Leo): Só tô em dúvida se o *low* que a gente falou é *low* por causa de, ah, o cara não precisa saber tanto de infra, mas tem que saber um pouquinho. O time tem que saber um pouquinho, então é *low*. Talvez seja isso que a gente quis dizer. E aí, sim. Mas tem que tomar cuidado, porque aí a gente tá discutindo essa *cross-funcionalidade* de níveis, em relação à infra. Então, tá, acho que tudo, talvez, já pressuponha isso. Porque, daí, a *cross-funcionalidade* em relação, sei lá, design, de banco de dados, UX, aí é outra história, a gente não tá falando. É *cross-funcionalidade* em relação à infra. Aí, talvez, o *low* seria por causa disso. Para dizer que talvez alguma coisinha tem que saber, mas talvez seja uma, se for isso, uma preciosidade, talvez, em termos de modelo, falso, isso, seja suficiente, também.

Entrevistador: Tá. Eh, eu no meu, a gente, eu não falo muito de *performance* de *delivery*. Você acha que é alguma coisa que é crucial que eu volte, que eu tente inserir no meu modelo, e isso, ou talvez dá para entender que isso é algo meio implícito?

Entrevistado (Leo): É, acho que não precisa. Porque isso aí é, talvez mais uma consequência que a gente procura observar do que uma, parte da caracterização. Que acho que, talvez, o modelo só seria o foco, em caracterizar o que que são as coisas.

Entrevistador: Então, isso poderia ser coisas que eu leve para discussão. Eu não preciso colocar no modelo em si, mas é importante eu discutir na minha tese. Assim como você fez também.

Entrevistado (Leo): É, também não necessariamente é importante. Acho que, talvez, para você é uma coisa opcional aí, se quiser incluir essa coisa de discussão, mas, talvez, não, não seria essencial.

Team cognitive load limits cross-functionality

Entrevistador: Tá bom. Eu vou fazer só mais uma, e aí a gente pode encerrar, que essa daqui eu achei interessante. Nas discussões que eu tive com o Manoel, e aí isso apareceu também com outro especialista, ele me disse que um ponto crucial sobre essa carga cognitiva é que ela pode sabotar a multifuncionalidade de uma equipe. A ideia de ter um especialista para cada área, um desenvolvedor, um operação, um designer, etc., faz que cada um opera num silo, como a gente comentou, do *cross-funcional* e tal. Mesmo estando em um grupo. Então, eu acabo tendo um, e aí o que eu trago de limite de capacidade cognitiva do time. Vai chegar em um momento que é impossível eu ter um especialista para cada coisa que eu preciso. É difícil manter, não dá para manter uma equipe, sei lá, com 30 pessoas, a comunicação vai ser ruim e etc. E aí, o que eu entendi é que a carga cognitiva do time, isso vai limitar a *cross-funcionalidade* dela. Não no booleano, se é sim ou se não, mas vai limitar a quantidade de perfis que eu consigo ter ali e fazer com que todo mundo lide com eles. Mas, até então, de acordo com essas pessoas também, eu tava entendendo que, mesmo em times *cross-funcionais*, essas pessoas, elas compartilham um pouco de responsabilidade, vai aprendendo um pouquinho aqui, um pouquinho ali, não que fiquem cada um no seu silo. Que é o que ele chama aqui de antipadrão. Que, se eu tenho cada um muito fechadinho na sua caixinha, isso pode passar que, ah, a responsabilidade da outra pessoa aqui não fez, isso atrapalha um pouco, entendeu? Não só a carga cognitiva, mas tudo isso que está por trás.

Entrevistado (Leo): É, boa parte das coisas, aí, concordo muito, mas acho que me perdi um pouco em algumas questões. Principalmente, do ponto central aí, que a carga cognitiva limita a *cross-funcionalidade*. Essa frase ainda não consegui digerir.

Entrevistador: Ah, sim. Ela limita na quantidade que eu não vou conseguir ser 100% *cross-funcional*. Mas isso parte de um ponto, que eu acho que você não concorda também, que o time *cross-funcional*, ele não vai, não vai dar. Cada um, realmente, vai estar na sua caixinha, e a ideia de ter o time *cross-funcional* é isso. Que cada um faça a sua coisa, e ali dentro do time, cada um com a sua responsabilidade, consegue chegar do outro lado sem depender de outro time, que aí piora. Então, você concorda que seriam silos, que são pequenos silos dentro de uma equipe.

Entrevistado (Leo): Tenderia a concordar. Vou dar um exemplo aí da, do lugar que eu tô agora. Quando entrei, é um sistema, que o sistema tem vários módulos. Então, tem várias equipes, mas é um banco de dados só, e é um DBA só. Então, que que acontecia? O DBA, que cuidava de todos os *scripts* lá. E aí, tinha que falar, eu consegui audiência com o DBA, para a coisa andar. Então, é ele que, que ia lá e realmente efetivava a alteração. E, e com isso, por mais que tenha algum nível de compartilhamento, de conhecimento, assim, eu sei que, ah, é ele que vai criar o *script*, então, quem tem que saber lá do SQL que cria é ele, dos padrões, da nomenclatura, do banco. Então, praticamente, só ele que sabia. Agora, a gente tá mudando isso. Comecei lá, colocar o *Flyway* lá, a gente também começou a surgir a ideia que, ah, esse cara, talvez, vá, vá sair, fazer outras coisas. Então, a gente tem que pensar nisso. Então, agora, tá criando uma equipe de DBAs, então, em vez de ter um, pelo menos ter uma equipe.

Entrevistado (Leo): Mas, eu acho que a experiência com ele reforça essa ideia aí que a *multifuncionalidade*, o time *cross-funcional*, tende a ter mais esse aspecto de silos internos. Foi até que eu falei lá atrás, que, acaba que não tem, necessariamente, tanto compartilhamento de informação, porque cada um tem a sua coisa. Mas, tudo isso, beleza, mas, onde a carga cognitiva entrou, aí, eu fiquei um pouco confuso, ainda.

Entrevistador: Ah, sim. A carga cognitiva é mais que vai chegar um momento que é impossível ter todas essas pessoas. É uma carga cognitiva, mas é uma carga cognitiva porque, até então, eu não tinha esse ponto de vista seu que, se é um time *cross-funcional*, eles vão compartilhar responsabilidade, conhecimento entre eles. Só que, se a gente entende que não, que formam silos entre eles, eles não necessariamente compartilham responsabilidade. Então, isso não vai limitar a carga cognitiva. Porque, quanto mais gente que entra, teoricamente, é mais informação, mais conhecimento que aquele time vai compartilhar ali. Só que não é muito bem o que se está dizendo, entendeu?

Entrevistado (Leo): Isso. Não diria isso porque, estaria com o pressuposto, talvez, alguns papéis, como desenvolvedor, vai ter vários. Mas, tirando o desenvolvedor, os outros papéis que entra, entra um ou dois, porque está precisando daquele papel, que é um conjunto à parte de corpo de conhecimento, que não vai ser, totalmente disseminado, só uma coisinha ou outra. Então, se o time *cross-funcional* for crescendo nessa pegada de ir incorporando, porque precisa dessa *skill* que não tinha, então é isso, essa, esse *skill* novo não vai aumentar muito a carga cognitiva, só, talvez, um pouquinho. E, a ideia de trazer uma pessoa para lá, justamente, para que o resto da equipe não tem que se preocupar com esse *concern* e manter. A ideia é que, em vez de, ter a plataforma que automatiza e assim, baixa a carga cognitiva, você tem mais uma pessoa no seu time que manja daquilo, e aí eu não preciso me preocupar em, em aprender e aumentar a minha carga cognitiva. Então, diria que a *cross-funcionalidade* feita desse jeito, ela também poderia ser uma, talvez uma estratégia pior, menos efetiva, mais cara, mas seria também uma estratégia de baixar a carga cognitiva, não do time, como um todo, mas das pessoas no time.

Entrevistador: Boa. Ótimo. Ótimo. Ótimo. Eu teria outros dois pontos aqui, mas é de boa se a gente não discutir, dependendo do seu tempo.

Entrevistado (Leo): Não, mas eu tô tranquilo.

AUTOMATION brings from experience

Entrevistador: O primeiro é que a automação, ela surge da experiência. Então, a automação surge do momento em que a gente está ali, tem várias dificuldades, e eu tô, tá um, manda tarefa para lá, joga o pacote de volta, e etc. Se a gente vai identificando isso como um problema, isso a gente vai ter que ficando repetindo várias vezes, vai chegar num momento que eu falo, não, isso tem que automatizar isso daqui. Estamos perdendo muito tempo, estamos gastando, precisamos automatizar. Então, surge essa ideia de que a automação, ela vem da experiência, através de, adquirida em diferentes contextos, e isso introduz soluções

inovadoras que aumentam a eficiência dessa, dessa equipe, ou desse produto que eles querem.

Entrevistado (Leo): Sim, sim, faz sentido. É uma, pelo menos, pensando em produto, um tipo de automação, ele nasce em algum momento disso. Embora, talvez, uma vez que alguém criou, ah, daí criou, uma ferramenta, sei lá, o *Ansible* ou alguma coisa, aí, em outro contexto, às vezes, alguém pode querer trazer essa automação por *hype*. Mas, mesmo que, ah, eu tô querendo trazer por *hype*, porque eu vi falar que o *Kubernetes* é *hype* e eu quero trazer, mas, o *Kubernetes* nasceu, é, porque, alguém, passou por isso antes.

Entrevistador: Bom, acho que fechamos, Leo. Muito obrigado pelo seu tempo. Deixa eu parar.