RAPHAEL FONTES · E-BOOK

Criando equipes ágeis

Quem precisa estar num time de software para ele entregar de verdade, e como conduzir esse time depois de montado.

SEGUNDA EDIÇÃO · 2026

SOBRE ESTA EDIÇÃO

O que mudou desde a primeira versão

A primeira edição deste material é de 2021. Reescrevi porque parte dela envelheceu de um jeito que não dava para corrigir com retoque, e prefiro dizer exatamente o que mudou a publicar uma versão nova fingindo que a anterior continuava válida.

> O eixo da primeira edição eram os papéis do Scrum. Essa palavra saiu do Guia do Scrum em novembro de 2020, poucos meses antes de eu publicar.

Não foi troca de sinônimo. A categoria deixou de existir: hoje são **responsabilidades**. Junto com ela saiu o Time de Desenvolvimento, que virou **Developers**, acabando com a ideia de um time dentro do time. É uma mudança de estrutura, não de vocabulário, e um material que ensina a montar equipe não pode carregar a estrutura errada.

O QUE É NOVO

Um capítulo sobre inteligência artificial dentro do time, e uma parte inteira sobre conduzir a equipe depois de montada. A primeira edição terminava na composição, e composição é metade do problema.

SUMÁRIO

O caminho deste e-book

As partes 1 a 5 respondem quem precisa estar no time. A parte 6 responde o que fazer depois, que é onde a maioria dos times bons se perde.

PARTE 1

Por que a pergunta mudou

Montar time de software em 2026 não é o mesmo exercício de cinco anos atrás, e o motivo não é moda.

PARTE 1

O que ficou barato, e o que não ficou

Com inteligência artificial dentro do ciclo de desenvolvimento, a parte executável do trabalho técnico ficou mais rápida e mais barata. Escrever a primeira versão de um código, gerar teste, produzir documentação: tudo isso custa uma fração do que custava.

O que não ficou mais barato foi **decidir o que construir**, **avaliar se o que voltou presta** e **responder pela consequência**.

FICOU MAIS BARATO

- Escrever o código da primeira versão
- Gerar teste e documentação
- Traduzir entre linguagens e formatos

NÃO FICOU

- Decidir o que vale construir
- Revisar o que voltou, e reprovar
- Integrar no que já existe e opera
- Responder pela consequência

O QUE ISSO MUDA NA COMPOSIÇÃO

O gargalo saiu de quem escreve e foi para quem decide e revisa.

Time montado só com capacidade de execução produz mais volume e a mesma quantidade de valor. O que passou a faltar primeiro é gente com contexto suficiente para dizer que aquilo não deveria ser construído.

> A IA não tirou ninguém do time. Ela mudou onde o time trava, e quem monta equipe precisa saber disso antes de abrir a próxima vaga.

PARTE 1

O efeito prático na hora de montar o time

Isso empurra o valor do time para cima, na direção de julgamento. E produz um erro de composição que eu tenho visto com frequência: montar equipe pensando só em capacidade de execução, porque execução é o que se mede fácil.

> Time assim produz mais volume e a mesma quantidade de valor. Às vezes menos, porque agora existe mais coisa construída para manter.

O que falta primeiro nesse cenário não é gente que escreva. É gente com contexto suficiente para dizer que aquilo não deveria ser construído, e com autoridade para sustentar o não.

Por isso este e-book começa por composição e termina em condução. Quem monta um time com as habilidades certas e depois conduz errado desperdiça as duas coisas.

UMA RESSALVA HONESTA

Nada aqui depende de você usar IA. As sete habilidades da Parte 3 valiam antes e continuam valendo. O que a IA muda é o peso relativo entre elas, e a velocidade com que um erro de decisão vira código em produção.

PARTE 2

Responsabilidade não é cargo

É a distinção que sustenta o resto do e-book, e a que mais gera confusão dentro das empresas.

PARTE 2

Cargo é da empresa, responsabilidade é do time

Cargo é a posição da pessoa na hierarquia da empresa. Carrega senioridade, faixa salarial e plano de carreira. Analista desenvolvedor sênior de Java é um cargo: ele diz o nível e a especialidade de quem ocupa.

Responsabilidade é aquilo por que a pessoa responde dentro daquele time, para aquele objetivo. Não é ordenável, não aparece no contracheque, e muda de time para time.

> A mesma pessoa pode ter um cargo sênior e responder por uma parte pequena do produto. Isso não é rebaixamento, é divisão de trabalho.

Essa distinção não é preciosismo de vocabulário. Ela resolve um problema real: quando o time se organiza por cargo, a conversa vira comparação vertical, e a pergunta que domina a sala passa a ser quem é mais sênior. Ao se organizar por responsabilidade, a pergunta muda para quem responde pelo quê, que é a única que ajuda a entregar.

NA PRÁTICA

Se você quer reduzir disputa de ego num time de especialistas, comece por aqui. Trocar cargo por responsabilidade muda mais comportamento do que qualquer conversa sobre a importância de colaborar.

PARTE 2

O que o Guia do Scrum mudou em 2020

Uso o Scrum como referência de vocabulário porque é o framework mais conhecido, o que facilita para quem está começando. E ele mudou de um jeito que ainda não chegou em boa parte do material que circula.

ATÉ 2019

Scrum Team

Product Owner | Scrum Master

Development Team (um time dentro do time)

DESDE O GUIA DE 2020

Um Scrum Team, três responsabilidades

a palavra papel saiu, e com ela o time dentro do time

o time é autogerenciável: decide quem faz o quê, quando e como

POR QUE CADA UMA RESPONDE

Product Owner RESPONDE PELO VALOR DO PRODUTO
prioriza, mantém o backlog, decide o que não entra. Precisa de mandato para dizer não.

Scrum Master RESPONDE PELO PROCESSO
remove impedimento e facilita. Cuida do processo, não das pessoas.

Developers RESPONDE PELO INCREMENTO
constroem algo utilizável a cada Sprint. Multifuncional é o time, não a pessoa.

> A mudança de 2020 não foi só de vocabulário. Acabar com o time dentro do time desmonta a dinâmica de um lado pedir e o outro executar.

O time também deixou de ser descrito como auto-organizável e passou a ser **autogerenciável**: ele decide internamente quem faz o quê, quando e como. E surgiu o **Product Goal**, um objetivo de produto que serve de alvo para o time planejar.

PARTE 2

As três responsabilidades, sem idealização

RESPONDE PELO VALOR

Product Owner

Mantém o backlog, prioriza e decide o que não entra. Trabalha com as áreas interessadas para entender o que o produto precisa resolver.

RESPONDE PELO PROCESSO

Scrum Master

Facilita e remove impedimento. Cuida do processo, não das pessoas. Faz a ponte com as áreas que travam o time por fora.

RESPONDE PELO INCREMENTO

Developers

Constroem algo utilizável a cada ciclo. Multifuncional é o time, que reúne as habilidades necessárias, e não cada pessoa.

Duas observações que material de certificação costuma omitir, e que decidem se isso funciona na sua empresa.

Product Owner sem mandato vira gerente de pedido. Se a pessoa não pode dizer não, ela não está priorizando, está transcrevendo. Isso é problema de estrutura organizacional, e nenhum treinamento resolve.

O Scrum Master é a responsabilidade mais banalizada das três, porque virou a porta de entrada mais comum. Ter a credencial diz pouco. Saber conduzir uma sala onde duas áreas discordam diz tudo.

PARTE 3

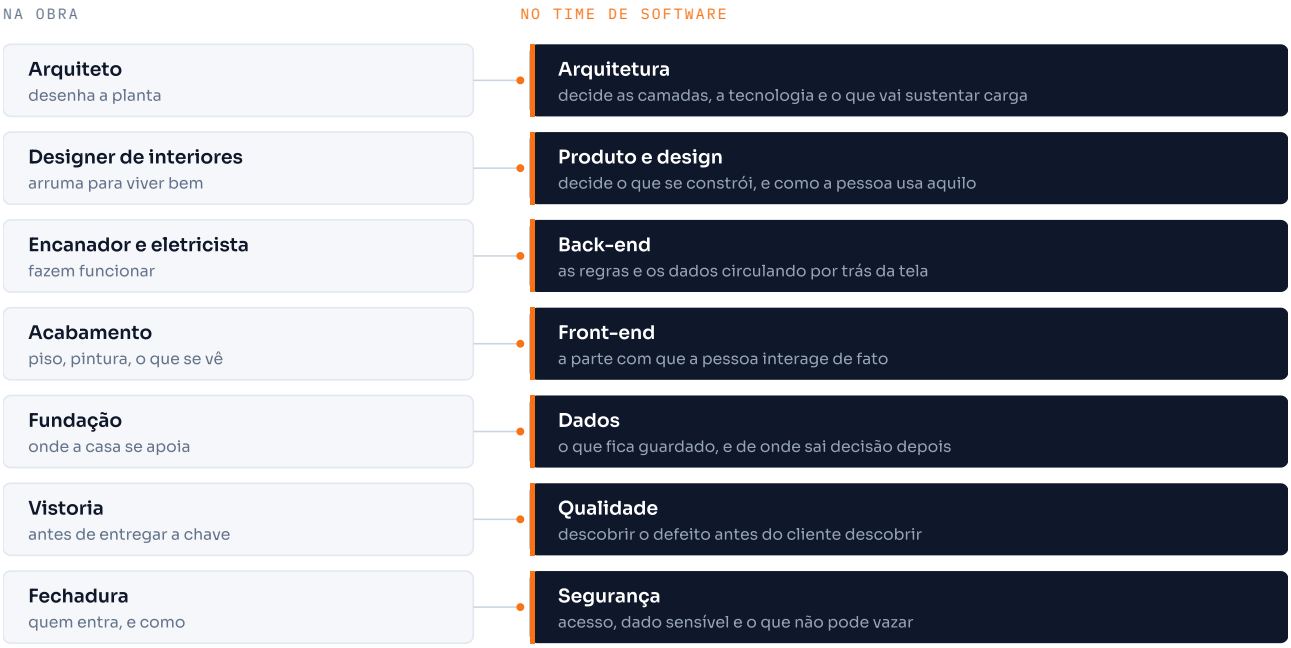
As habilidades que o time reúne

Sete habilidades, explicadas pela construção de uma casa. Sete habilidades, não sete pessoas.

PARTE 3

Por que a casa explica melhor que o organograma

Uso essa comparação há anos, com gente que não é de tecnologia, e ela funciona porque todo mundo já viu uma obra dar errado. Ninguém acha estranho que uma casa precise de arquiteto, eletricista e vistoria. Mas muita empresa acha perfeitamente normal montar um produto digital só com programador.



> Sete habilidades, não sete pessoas. Um time pequeno acumula várias delas na mesma pessoa, e isso é normal. O que não dá é fingir que alguma não precisa existir.

PARTE 3

O erro mais comum: achar que full-stack resolve

Empresa inexperiente contrata dois ou três desenvolvedores full-stack e acredita que está coberta, porque no papel aquelas pessoas fazem front e back. É o equivalente a construir uma casa só com pedreiro, sem arquiteto, sem engenheiro e sem mestre de obras.

A casa fica de pé. Você percebe o problema quando entra nela.

> Full-stack cobre duas das sete habilidades. As outras cinco não desaparecem por não terem dono: elas viram retrabalho.

E o custo não aparece no primeiro mês. Aparece quando o produto precisa crescer e ninguém sabe por que aquela decisão de arquitetura foi tomada, quando o cliente reclama de algo que ninguém testou, ou quando o dado que a diretoria pede não existe porque ninguém pensou em guardar.

COMO EU LEIO ISSO HOJE

Não é que todo time precise de sete pessoas. É que todo time precisa saber quais das sete habilidades ele não tem, e decidir de propósito o que vai fazer a respeito. O perigo não é a lacuna, é a lacuna que ninguém enxerga.

PARTE 3

O que mudou nas habilidades desde 2021

A primeira edição separava WebDesign de UX, e tratava banco de dados como a única responsabilidade ligada a dado. Os dois envelheceram.

Design virou uma coisa só, e subiu. Não é mais escolher paleta e tipografia de um lado e organizar a tela do outro. É participar da decisão do que se constrói, antes de existir tela para desenhar.

Dado deixou de ser só armazenamento. Manter o banco funcionando continua necessário, mas a habilidade que falta na maioria dos times é outra: pensar o que precisa ser registrado hoje para existir decisão amanhã. Isso se decide quando a funcionalidade é construída, não depois.

Segurança virou habilidade de time, não de auditoria. Enquanto for algo que alguém de fora verifica no fim, ela vai continuar sendo descoberta tarde e cara.

E A QUALIDADE

Continua sendo a habilidade mais desvalorizada das sete, e continua sendo a que mais aparece na conta. Com IA gerando volume maior de código, ela ficou mais importante, não menos: revisar passou a ser o gargalo.

> Habilidade sem dono não desaparece. Ela vira retrabalho, e o retrabalho aparece no trimestre seguinte, sem etiqueta dizendo de onde veio.

PARTE 4

Inteligência artificial dentro do time

O capítulo que não existia na primeira edição, e sem o qual este material não faria sentido hoje.

PARTE 4

A IA não tirou ninguém do time

Ela mudou onde o time trava. E quem monta equipe precisa saber disso antes de abrir a próxima vaga, porque a vaga certa mudou.

Na prática eu vejo três deslocamentos.

Revisar ficou mais caro que escrever. Quando a primeira versão de algo sai em minutos, o tempo do time migra para avaliar se aquilo resolve o problema certo, se cabe na arquitetura que existe e se não introduziu um risco silencioso. Time sem gente experiente o suficiente para reprovar produz rápido e acumula dívida.

Contexto virou o insumo escasso. A ferramenta não conhece o seu cliente, o seu contrato, a sua regra de negócio esquisita que existe por um motivo histórico. Quem carrega isso são as pessoas, e isso valoriza quem está há mais tempo, ao contrário do que muita gente previu.

A fronteira entre as habilidades ficou mais permeável. É mais viável hoje alguém de produto montar um protótipo funcional, ou alguém de back-end resolver uma tela. Isso é bom, e tem um efeito colateral: aumenta a chance de alguém construir algo fora da sua área sem o julgamento de quem entende dela.

> O QUE ISSO MUDA NA PRÓXIMA VAGA
>
> Antes de abrir uma vaga de execução, vale perguntar onde o seu time trava hoje. Se o gargalo é revisar, aprovar e integrar, contratar mais alguém para escrever aumenta a fila em vez de diminuir. Nesse cenário o perfil que destrava é senioridade com contexto, não capacidade de produzir volume.

PARTE 4

O que isso não resolve

Vale dizer com clareza, porque a conversa sobre IA costuma andar em dois extremos e nenhum dos dois ajuda quem precisa montar um time na segunda-feira.

- Não substitui conhecer o próprio negócio. Resposta boa depende de pergunta boa, e pergunta boa vem de contexto.
- Não substitui responsabilidade. Quando algo dá errado em produção, quem responde é uma pessoa, com nome.
- Não conserta time mal composto. Acelera o que já existe, inclusive o erro.
- Não substitui a decisão de não construir, que continua sendo a decisão mais valiosa de um time de produto.

> A tecnologia acelerou a execução. A responsabilidade continuou sendo humana.

Na minha visão, o time que se dá bem nesse cenário não é o que adotou mais ferramenta. É o que tem gente capaz de julgar, e que combinou como esse julgamento acontece antes de a coisa chegar em produção.

PARTE 5

Compondo na prática

Tamanho, acúmulo de responsabilidade, e as combinações que não podem acontecer.

PARTE 5

Quantas pessoas, e quem acumula o quê

A referência atual do Guia do Scrum é de **dez pessoas ou menos**, e vale para o time inteiro, não só para quem constrói. Na primeira edição eu escrevi de três a nove, que era a formulação antiga e se referia só à equipe de desenvolvimento.

Acumular responsabilidade é normal e esperado em time pequeno. A mesma pessoa pode responder por produto e design, ou por back-end e dados, e isso costuma funcionar bem porque as habilidades são vizinhas.

> O que não funciona é acumular responsabilidades que precisam discordar uma da outra.

Product Owner que também constrói tende a priorizar o que é gostoso de construir. Scrum Master que também é o chefe formal não consegue ser o lugar seguro para alguém dizer que está travado. E quem constrói validando o próprio trabalho enxerga o que esperava enxergar.

Não é desconfiança de caráter. É que julgamento independente é uma função, e ela some quando as duas pontas são a mesma pessoa.

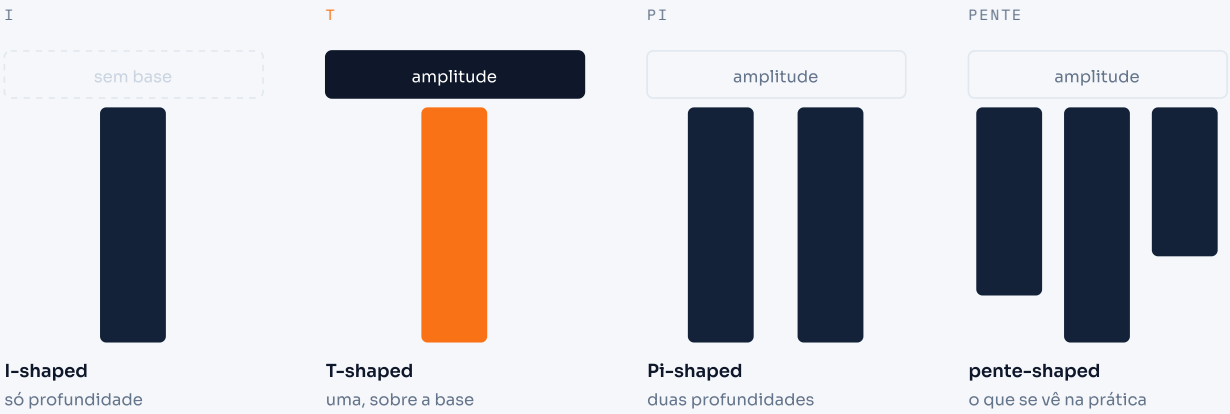
UM TESTE SIMPLES

Liste as sete habilidades da Parte 3 e escreva ao lado de cada uma o nome de quem responde por ela hoje. As que ficarem sem nome são as suas lacunas reais. As que tiverem o mesmo nome em três linhas ou mais são o seu risco de dependência de pessoa.

PARTE 5

Que perfil eu procuro para cada vaga

Quando me perguntam se eu prefiro especialista ou generalista, a resposta honesta é que os dois extremos me dão trabalho. Especialista puro trava na fronteira, e generalista puro tem dificuldade de responder por algo.



I-shaped
só profundidade

T-shaped
uma, sobre a base

Pi-shaped
duas profundidades

pente-shaped
o que se vê na prática

EIXO VERTICAL = ESPECIALIDADE · EIXO HORIZONTAL = ÁREAS VIZINHAS

> O T é o modelo mais usado para conversar sobre perfil. O pente é o que costuma aparecer quando você olha a carreira real de alguém experiente.

Na composição, o que importa não é ter muitos perfis em T. É que as **verticais sejam diferentes**. Cinco pessoas com a mesma especialidade formam um time especialista com aparência de multidisciplinar.

PARTE 6

Depois de montado, conduzir

A primeira edição terminava na composição. Composição é metade do problema, e não é a metade em que os times bons se perdem.

PARTE 6

Conhecer quem já está no time

A gente acha que conhece as pessoas com quem trabalha. Sabe o nome dos filhos, sabe do time de futebol, e não faz ideia do que aquela pessoa considera um trabalho bem feito.

Apliquei uma prática do Management 3.0 chamada **Personal Maps** com o meu time, cada um desenhando um mapa de si, e saí da sessão com informação que dois anos de reunião de acompanhamento não tinham produzido.

O que mais me marcou foi perceber que eu estava, sem saber, tomando decisões contra o valor declarado de algumas pessoas. Quem valoriza processo sofre quando um prazo curto corta o teste automatizado. Quem é movido por entrega topa numa boa. Os dois são bons profissionais, e a mesma decisão afeta cada um de um jeito.

DETALHE QUE FEZ DIFERENÇA

Apresentei o meu mapa primeiro. Eu sabia que ninguém ia querer começar, e quem abre define a profundidade que os outros vão seguir.

O maior risco dessa prática é fazer a sessão, coletar tudo isso, e no mês seguinte agir exatamente como antes. Aí ela vira teatro, e time percebe teatro rápido. O valor não está no evento, está no que você faz com a informação depois.

> Liderar sem conhecer as pessoas é possível. É só caro.

PARTE 6

Acompanhar de perto sem sufocar

Microgerenciamento é o estilo de gestão em que o gestor controla como o trabalho é feito, em vez de conduzir o resultado. Ele parece funcionar, e é isso que o mantém vivo: controle intenso entrega previsibilidade no trimestre, e cobra a conta depois.

MESMA INTENSIDADE, OBJETO DIFERENTE

Acompanhar de perto

OLHA

risco
dependência entre frentes
resultado combinado

a decisão de método fica com quem executa

Microgerenciar

OLHA

como cada um está fazendo
cada decisão, antes de sair
o detalhe da execução

a decisão de método sobe para o gestor

O TESTE

Tire férias. Se as decisões esperam por você, o gargalo do projeto é você.

> Time novo e situação crítica pedem acompanhamento próximo, e isso não é microgerenciamento. O que muda não é a frequência do olhar, é onde ele se apoia.

Time novo e situação crítica precisam de acompanhamento próximo, e isso não é microgerenciamento. O que muda não é a frequência do olhar, é onde ele se apoia.

PARTE 6

Fazer um time de especialistas colaborar

Time formado pelos melhores profissionais individualmente costuma render menos que a soma das partes. Não é defeito de caráter: especialista sênior chega com repertório, convicção e a expectativa legítima de ser reconhecido pelo que sabe.

O que resolveu isso comigo não foi conversa sobre colaboração. Foi mudar o desenho do trabalho.

- **Trocar cargo por responsabilidade**, para tirar a comparação vertical do centro da conversa.
- **Desenhar as entregas de forma interligada**, de modo que ninguém consiga concluir a própria parte sem o trabalho de outra pessoa.

A segunda foi a que mudou comportamento. Não é armadilha: a dependência já existia, ela só não estava declarada, então cada um tratava a própria parte como se fosse um produto completo.

> Quando a dependência fica visível, pedir ajuda deixa de ser admissão de fraqueza e vira parte do procedimento.

E tem um efeito de reciprocidade que eu não previa: quem precisou de ajuda fica bem mais disponível quando alguém precisa dele.

PARTE 6

Formar quem vai liderar depois

Empresa que treina bem a capacidade técnica e não forma quem vai conduzir acaba promovendo por antiguidade ou por desempenho individual. São os dois piores critérios disponíveis, e os mais usados.

Promover o melhor técnico trata uma competência como prova de outra: a empresa perde o melhor executor e ganha um gestor sem preparo, muitas vezes infeliz com o novo trabalho.

As competências que faltam são justamente as que trabalho técnico não exercita: comunicar decisão para quem não é da área, mediar conflito entre duas pessoas parcialmente certas, dar retorno difícil, decidir com informação incompleta e priorizar entre coisas legítimas.

Isso não se resolve com treinamento de dois dias. Resolve com exposição gradual a responsabilidade real, com acompanhamento: conduzir uma frente pequena, participar da negociação junto, escrever a comunicação difícil e receber revisão antes de enviar.

COMO EU IDENTIFICO POTENCIAL

Observando quem as pessoas procuram quando têm dúvida, sem que isso esteja no organograma. E vale lembrar: nem todo mundo com esse perfil quer liderar, e insistir com quem não quer é ruim para os dois.

PARTE 6

Se o time for remoto

Time distribuído perde duas coisas que o escritório dava sem ninguém pedir: saber quem está disponível agora, e conversar sem marcar reunião. A reunião diária resolve alinhamento, e não resolve nenhuma das duas.

O efeito que mais preocupa é silencioso. No presencial, uma dúvida de trinta segundos custava trinta segundos. No remoto ela virou um ritual: mandar mensagem, esperar resposta, combinar horário, entrar em chamada. Quando o custo sobe, as pessoas param de perguntar e decidem sozinhas, com menos contexto.

> Erro por falta de pergunta é caro e não aparece em nenhum indicador.

Ferramenta de escritório virtual ajuda, e não faz milagre. Ela remove o atrito de agir sobre a confiança que já existe. Se o time não tem abertura, o que aparece é um mapa bonito com todo mundo em status ocupado.

Por isso a ordem importa: primeiro deixar explícito que interromper é permitido, combinar o que é assíncrono e o que é síncrono, e a liderança aparecer disponível de fato. Depois disso, ferramenta amplifica. Na ordem inversa, ela só expõe.

> Ferramenta não cria confiança. Ela remove o atrito de agir sobre a confiança que já existe.

Vale uma ressalva sobre custo, porque ela mudou desde 2022: várias dessas ferramentas acabaram com o plano gratuito. Hoje a conversa deixou de ser "vale experimentar" e passou a ser quanto custa, por pessoa, contra quanto custa o mal-entendido que o time tem toda semana.

PARA CONTINUAR

Onde cada assunto está aprofundado

Cada parte deste e-book tem um artigo mais longo no site, com o caso real por trás dela.

Profissional T-shaped

raphaelfontes.com.br/artigos/profissional-t-shaped/

Personal Maps: conhecer o time de verdade

raphaelfontes.com.br/artigos/personal-maps/

Microgerenciamento: como identificar

raphaelfontes.com.br/artigos/microgerenciamento/

Por que um time de especialistas não colabora

raphaelfontes.com.br/artigos/colaboradores-individuais/

Como formar novos líderes técnicos

raphaelfontes.com.br/artigos/desenvolvendo-novos-lideres/

Comunicação em time remoto

raphaelfontes.com.br/artigos/equipes-remotas/

Qual certificação Scrum escolher

raphaelfontes.com.br/artigos/certificacoes-scrum/

SOBRE O AUTOR

Raphael Fontes

Head of Technology e CTO em exercício na Sesatech, empresa do grupo Oplium. Lidera uma estrutura de mais de 100 pessoas em engenharia, dados, produto, consultoria e suporte, atendendo uma operação de 400 pessoas entre a empresa e o grupo.

São 18 anos em tecnologia, começando como desenvolvedor em 2008, com sete anos de consultoria em desenvolvimento de software, arquitetura de software e dados na everis, hoje NTT DATA, e oito anos na Sesatech, da gestão de projetos até a liderança executiva.

Coautor do livro *Jornada Ágil de Liderança*, da coleção Jornada Colaborativa. MBA em Technology Leadership CTO/CIO pela FIAP, especialização em Gerência de Projetos de Software pela PUC-Rio, e certificações em gestão de projetos e agilidade, entre elas PMP, SAFe e Scrum.

> Tecnologia só gera valor quando conecta pessoas, processos e dados ao negócio.

ONDE ME ACHAR

raphaelfontes.com.br
linkedin.com/in/raphaelsfontes