



# DevOps

Para  
**leigos**

**Emily Freeman**

PREFÁCIO DE Nicole Forsgren



**ALTA BOOKS**

EDITORA

Rio de Janeiro, 2021

## DevOps Para Leigos\*

Copyright © 2021 da Starlin Alta Editora e Consultoria Eireli. ISBN: 978-85-508-1573-2

*Translated from original DevOps For Dummies®. Copyright © 2020 by John Wiley & Sons, Inc. ISBN 978-1-119-55222-2. This translation is published and sold by permission of John Wiley & Sons, Inc., the owner of all rights to publish and sell the same. PORTUGUESE language edition published by Starlin Alta Editora e Consultoria Eireli, Copyright © 2021 by Starlin Alta Editora e Consultoria Eireli.*

Todos os direitos estão reservados e protegidos por Lei. Nenhuma parte deste livro, sem autorização prévia por escrito da editora, poderá ser reproduzida ou transmitida. A violação dos Direitos Autorais é crime estabelecido na Lei nº 9.610/98 e com punição de acordo com o artigo 184 do Código Penal.

A editora não se responsabiliza pelo conteúdo da obra, formulada exclusivamente pelo(s) autor(es).

**Marcas Registradas:** Todos os termos mencionados e reconhecidos como Marca Registrada e/ou Comercial são de responsabilidade de seus proprietários. A editora informa não estar associada a nenhum produto e/ou fornecedor apresentado no livro.

Impresso no Brasil — 1ª Edição, 2021 — Edição revisada conforme o Acordo Ortográfico da Língua Portuguesa de 2009.

### Produção Editorial

Editora Alta Books

### Produtor Editorial

Thiê Alves

### Equipe de Marketing

Livia Carvalho  
Gabriela Carvalho  
marketing@altabooks.com.br

### Editor de Aquisição

José Rugeri  
j.rugeri@altabooks.com.br

### Gerência Editorial

Anderson Vieira

### Gerência Comercial

Daniele Fonseca

### Coordenação de Eventos

Viviane Paiva  
comercial@altabooks.com.brw

### Equipe Editorial

Ian Verçosa  
Illyabelle Trajano  
Luana Goulart  
Maria de Lourdes Borges  
Raquel Porto

### Rodrigo Ramos

Thales Silva

### Equipe de Design

Larissa Lima  
Marcelli Ferreira  
Paulo Gomes

### Equipe Comercial

Daiana Costa  
Daniel Leal  
Kaique Luiz  
Tairone Oliveira  
Vanessa Leite

### Tradução

Samantha Batista

### Revisão Gramatical

Jana Araujo  
Thaís Pol

### Revisão Técnica

Amanda Pinto  
Especialista em DevOps

### Diagramação

Lucia Quaresma

### Copidesque

Alessandro Thomé

Publique seu livro com a Alta Books. Para mais informações envie um e-mail para [autoria@altabooks.com.br](mailto:autoria@altabooks.com.br)

Obra disponível para venda corporativa e/ou personalizada. Para mais informações, fale com [projetos@altabooks.com.br](mailto:projetos@altabooks.com.br)

**Erratas e arquivos de apoio:** No site da editora relatamos, com a devida correção, qualquer erro encontrado em nossos livros, bem como disponibilizamos arquivos de apoio se aplicáveis à obra em questão.

Acesse o site [www.altabooks.com.br](http://www.altabooks.com.br) e procure pelo título do livro desejado para ter acesso às erratas, aos arquivos de apoio e/ou a outros conteúdos aplicáveis à obra.

**Suporte Técnico:** A obra é comercializada na forma em que está, sem direito a suporte técnico ou orientação pessoal/exclusiva ao leitor.

A editora não se responsabiliza pela manutenção, atualização e idioma dos sites referidos pelos autores nesta obra.

**Ouvidoria:** [ouvidoria@altabooks.com.br](mailto:ouvidoria@altabooks.com.br)

#### Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD

F855d Freeman, Emily  
DevOps Para Leigos / Emily Freeman ; traduzido por Samantha Batista. - Rio de Janeiro, RJ : Alta Books, 2021.  
368 p. : il. ; 17cm x 24cm.  
Tradução de: DevOps For Dummies  
Inclui índice.  
ISBN: 978-85-508-1573-2  
1. Ciência da Computação. 2. DevOps. I. Batista, Samantha. II. Título.  
2020-3153 CDD 004  
CDU 004

Elaborado por Wagner Rodolfo da Silva - CRB-8/9410



Rua Viúva Cláudio, 291 — Bairro Industrial do Jacaré  
CEP: 20.970-031 — Rio de Janeiro (RJ)  
Tels.: (21) 3278-8069 / 3278-8419  
[www.altabooks.com.br](http://www.altabooks.com.br) — [altabooks@altabooks.com.br](mailto:altabooks@altabooks.com.br)  
[www.facebook.com/altabooks](https://www.facebook.com/altabooks) — [www.instagram.com/altabooks](https://www.instagram.com/altabooks)



ASSOCIADO



# Sumário Resumido

<b>Prefácio</b> .....	xxi
<b>Introdução</b> .....	1
<b>Parte 1: Desmistificando o DevOps</b> .....	5
<b>CAPÍTULO 1:</b> Apresentando o DevOps .....	7
<b>CAPÍTULO 2:</b> Projetando Sua Organização .....	17
<b>CAPÍTULO 3:</b> Identificando o Desperdício .....	33
<b>CAPÍTULO 4:</b> Persuadindo Colegas a Experimentar o DevOps .....	45
<b>CAPÍTULO 5:</b> Considerando Sua Organização .....	61
<b>Parte 2: Estabelecendo um Pipeline</b> .....	73
<b>CAPÍTULO 6:</b> Acolhendo o Novo Ciclo de Vida do Desenvolvimento .....	75
<b>CAPÍTULO 7:</b> Planejando com Antecedência .....	85
<b>CAPÍTULO 8:</b> Projetando Recursos de uma Perspectiva DevOps .....	101
<b>CAPÍTULO 9:</b> Desenvolvendo Código .....	119
<b>CAPÍTULO 10:</b> Automatizando Testes Antes do Lançamento .....	137
<b>CAPÍTULO 11:</b> Implementando um Produto .....	147
<b>Parte 3: Conectando o Circuito</b> .....	167
<b>CAPÍTULO 12:</b> Implementando a Iteração Rápida .....	169
<b>CAPÍTULO 13:</b> Criando Ciclos de Feedback Acerca do Cliente .....	185
<b>CAPÍTULO 14:</b> DevOps Não É um Time (Mas às Vezes É) .....	199
<b>CAPÍTULO 15:</b> Capacitando Profissionais de Tecnologia .....	217
<b>Parte 4: Praticando o Kaizen, a Arte da Melhoria Contínua</b> .....	229
<b>CAPÍTULO 16:</b> Aceitando Bem o Fracasso .....	231
<b>CAPÍTULO 17:</b> Preparando-se para os Incidentes .....	241
<b>CAPÍTULO 18:</b> Realizando Revisões Pós-incidente .....	263
<b>Parte 5: Instrumentalizando Sua Prática DevOps</b> .....	273
<b>CAPÍTULO 19:</b> Adotando Novas Ferramentas .....	275
<b>CAPÍTULO 20:</b> Gerindo Sistemas Distribuídos .....	289
<b>CAPÍTULO 21:</b> Migrando para a Cloud .....	309

<b>Parte 6: A Parte dos Dez</b> .....	321
<b>CAPÍTULO 22:</b> As Dez (ou Mais) Principais Razões da Importância do DevOps. . . . .	323
<b>CAPÍTULO 23:</b> As Dez Principais Armadilhas do DevOps .....	331
<b>Índice</b> .....	339

AMOSTRA

1

# Desmistificando o DevOps

AMOSTRA

## NESTA PARTE. . .

Entenda os valores e as prioridades do DevOps, que focam pessoas, processos e tecnologia.

Planeje a cultura de sua organização com o DevOps em mente, encorajando o trabalho em equipe, reduzindo as divisões e aceitando as falhas.

Identifique o desperdício e localize os gargalos no decorrer do ciclo de vida de seu desenvolvimento de software para estabelecer as vitórias mais fáceis (e mais imediatas) para sua transformação DevOps.

Convença seus colegas, de executivos a desenvolvedores, dos benefícios de DevOps para uma organização de tecnologia.

Meça seu trabalho e acompanhe seus sucessos DevOps, permitindo que todos vejam as melhorias incrementais.

- » **Avaliando os principais dogmas de DevOps**
- » **Compreendendo os valores de DevOps**
- » **Vendo como sua organização se beneficia**

## Capítulo **1**

# Apresentando o DevOps

O DevOps transformou a maneira como os times de desenvolvimento e operações colaboram e compartilham conhecimentos entre si, a fim de criar e entregar software. É uma filosofia vasta e abrangente que inspira diversas implementações por todas as empresas de tecnologia.

Eu defino o DevOps como uma cultura organizacional de colaboração, senso de propriedade e aprendizagem com o propósito de acelerar o ciclo de vida de desenvolvimento de software, da concepção à produção. O DevOps possibilita a redução de atrito interpessoal, elimina gargalos, melhora a colaboração entre as pessoas, aumenta a satisfação no trabalho por meio da capacitação dos times envolvidos e acelera a produtividade deles. Ele não faz milagres, mas pode ter um grande impacto na sua organização e em seus produtos.

Neste capítulo, enfatizo a importância da cultura sobre o processo e o instrumental, discuto os princípios e valores do DevOps e mergulho em como a organização se beneficiará de uma abordagem DevOps.

# O que É DevOps?

Este livro não tem uma prescrição DevOps exata para você — porque isso não existe. O DevOps é uma filosofia que prioriza pessoas a processos e processos a ferramentas. Ele constrói uma cultura de confiança, colaboração e melhoria contínua. Como uma cultura, vê o processo de desenvolvimento de modo holístico, levando em consideração todos os envolvidos: desenvolvedores, testadores, o pessoal de operações, segurança e os profissionais de infraestrutura. O DevOps não coloca nenhum desses grupos acima dos outros e nem classifica a importância de seus trabalhos. Em vez disso, uma empresa que pratica DevOps trata todo o time de tecnologia como crucial para garantir que o cliente tenha a melhor experiência possível. (Você pode encontrar mais sobre cultura corporativa no Capítulo 2.)

## O DevOps evoluiu do Ágil

Em 2001, 17 desenvolvedores de software se encontraram e publicaram o “Manifesto Ágil”, que declara os 12 princípios da gestão de projetos ágeis (veja o box “As Origens do Ágil”, no Capítulo 7, para mais detalhes). Esse novo fluxo de trabalho foi uma resposta à frustração e à inflexibilidade dos times de tecnologia que trabalhavam em um processo em cascata (waterfall). Trabalhando com princípios ágeis, os times de tecnologia não precisam aderir aos requerimentos originais ou seguir um fluxo de trabalho de desenvolvimento em cascata (waterfall), no qual cada time entrega o trabalho para o próximo. Em vez disso, são sempre capazes de se adaptar às necessidades de mudanças constantes do negócio e do mercado, e às vezes até à mudança de tecnologia e ferramentas.

Embora o Ágil tenha revolucionado o desenvolvimento de software de várias maneiras, ele não conseguiu resolver o conflito entre os times de desenvolvimento e o time de operações. Os silos ainda se desenvolveram em torno de conjuntos de habilidades técnicas e especialidades, e os times de desenvolvimento ainda entregavam código ao time de operações para que implementassem e dessem suporte.

Em 2008, Andrew Clay Shafer conversou com Patrick Debois sobre suas frustrações com o constante conflito entre os times de desenvolvimento e de operações. Juntos, eles lançaram o primeiro evento DevOpsDays, na Bélgica, para criar uma forma melhor e mais ágil de abordar o desenvolvimento de software. Essa evolução do Ágil ocorreu e, desde então, o DevOps permite que empresas de todo o mundo desenvolvam softwares com mais rapidez e qualidade (e normalmente por um preço mais baixo). O DevOps não é modinha: é uma filosofia organizacional amplamente aceita.

## O DevOps se concentra nas pessoas

Qualquer pessoa que lhe diga que DevOps diz respeito a ferramentas, com certeza deseja lhe vender alguma coisa. Acima de tudo, o DevOps é uma filosofia que se concentra nos times de desenvolvimento e operações e em como eles podem trabalhar melhor juntos para produzir ótimos softwares. Você poderia gastar milhões em várias ferramentas de DevOps do mundo e ainda não chegaria nem perto do seu “nirvana”. Em vez disso, concentre-se no bem mais importante da tecnologia: as pessoas. Pessoas felizes desenvolvem ótimos softwares. E como deixá-los felizes? Bem, crie um ambiente de trabalho colaborativo no qual o respeito mútuo, o conhecimento compartilhado e o reconhecimento do trabalho duro possam prosperar. Veja os Capítulos 2 e 15 para mais informações sobre como criar times de desenvolvimento e operações felizes e capacitados que incorporam um mindset de crescimento e se orgulham de seu trabalho.

## A cultura empresarial é a base do DevOps

Sua empresa tem uma cultura, mesmo que tenha sido desenvolvida por inércia. Essa cultura tem mais influência em sua satisfação no trabalho, na produtividade e na velocidade do time do que você pode imaginar.

A cultura da empresa é mais bem descrita como as expectativas, o comportamento e os valores implícitos de uma organização. A cultura é o que diz aos funcionários se a liderança da empresa está aberta a novas ideias. É o que informa a decisão de um funcionário de levar um problema a público ou varrá-lo para debaixo do tapete.

A cultura deve ser projetada e refinada, e não deixada ao acaso. Embora a real definição varie de uma empresa para outra e de uma pessoa para outra, o DevOps é essencialmente uma abordagem cultural, uma mentalidade.

Uma cultura empresarial tóxica matará sua jornada DevOps antes mesmo de ela começar. Mesmo que seus times de desenvolvimento e operações adotem um mindset DevOps, as atitudes e os desafios da empresa mais ampla contaminarão seu ambiente.

Com o DevOps, evitamos a culpa, aumentamos a confiança e focamos o cliente. Damos aos times de desenvolvimento e operações autonomia e os capacitamos para fazer o que fazem de melhor: criar soluções. Ao começar a implementar DevOps, damos aos times tempo e espaço para se ajustarem a ele, permitindo que tenham oportunidades de se conhecer melhor e construir conexões com times de especialidades diferentes. Além disso, medimos seu progresso e recompensamos realizações. Nunca culpe indivíduos por alguma falha. O time deve melhorar junto e continuamente, e as realizações devem ser celebradas e recompensadas.

## **Você aprende observando seu processo e coletando dados**

Observar o fluxo de trabalho sem expectativas é uma técnica poderosa que deve ser usada para ver os sucessos e os desafios do fluxo de trabalho de modo realista. Essa observação é o único modo de encontrar a solução correta para as áreas e os problemas que criam gargalos no processo. Assim como no software, jogar Kubernetes (ou outra nova ferramenta) em um problema não necessariamente o corrigirá. É preciso saber onde os problemas estão antes de tentar corrigi-los. Ao prosseguir, deve-se coletar dados — não para medir o sucesso ou o fracasso, mas para acompanhar o desempenho do time. Determinamos o que funciona, o que não funciona e o que devemos tentar da próxima vez. No Capítulo 3, você aprenderá a identificar seus gargalos no processo de desenvolvimento.

## **A persuasão é o segredo da adoção do DevOps**

Vender a ideia de DevOps para seus líderes, colegas e funcionários não é fácil. O processo nem sempre é intuitivo para os times de desenvolvimento e operações também. Uma grande ideia não deveria se vender sozinha? Ah, se fosse fácil assim! Contudo, um conceito-chave a se manter sempre em mente ao implementar o DevOps é que ele enfatiza pessoas. As chamadas “soft skills” da comunicação e colaboração são cruciais para a transformação DevOps. Persuadir outras pessoas do time e dentro da empresa a adotar o DevOps requer uma grande prática e habilidade no exercício de comunicação. As primeiras conversas com colegas sobre DevOps podem prepará-lo para o sucesso no futuro — especialmente quando atingir uma lombada inesperada.

## **Pequenas mudanças incrementais não têm preço**

O aspecto do DevOps que enfatiza fazer pequenas mudanças incrementais tem sua origem na manufatura lean, que abraça o feedback acelerado, a melhoria contínua e um prazo de lançamento mais rápido. Quando falo de transformações DevOps, gosto de usar a água como metáfora. A água é um dos elementos mais poderosos do mundo. A não ser que as pessoas fiquem olhando a água de uma enchente subir à sua frente, elas a acham relativamente inofensiva. O Rio Colorado esculpia o Grand Canyon. Lentamente, ao longo de milhões de anos, a água cortou a pedra para expor quase 2 bilhões de anos de solo e rocha.

Você pode ser como a água. Seja a mudança lenta e implacável em sua organização. Veja aquela famosa citação de uma entrevista de Bruce Lee para se inspirar (<https://www.youtube.com/watch?v=cJMwBwFj5nQ>) [conteúdo em inglês, legenda a seguir]:

*Seja disforme, amorfo, como a água. Coloque água em um copo, ela se transforma no copo. Coloque água em uma garrafa, ela se transforma na garrafa. Coloque-a em um bule, ela se transforma no bule. Agora, a água pode fluir ou pode colidir. Seja como a água, meu amigo!*

Fazer mudanças incrementais significa, por exemplo, que você pode encontrar um problema e corrigi-lo. Depois, corrigir o próximo. Assim, não assume muitas demandas ao mesmo tempo e não escolhe todas as batalhas ao mesmo tempo para lutar. Você compreende que algumas batalhas não valem a energia e o capital social que podem lhe custar.

## Beneficiando-se do DevOps

Este livro inteiro aborda como você e seu time podem se beneficiar da implementação de DevOps em sua organização. Além do componente humano, que possibilita entrega mais rápida, com qualidade, segurança e inovação, o DevOps tem benefícios técnicos.

A integração e a entrega contínuas (CI/CD) estão fortemente alinhadas com o DevOps. A entrega contínua de software remove muitos dos gargalos que geralmente vemos em times que implementam esporadicamente. Ao criar pipelines automatizados que passam novos códigos por um conjunto de teste robusto, pode-se ter mais confiança nas implementações. (Falo mais sobre CI/CD no Capítulo 11.)

DevOps também possibilita uma recuperação mais rápida de incidentes. Em algum ponto, você inevitavelmente experimentará uma interrupção de serviço que impactará os clientes, não importa o quanto seu código tenha sido bem testado. Mas os times que já usam o DevOps como sua rotina de trabalhos encontram resoluções mais rápido por intermédio de uma maturidade maior que os times possuem, acessibilidade mais aberta, aprendizado compartilhado e melhor monitoramento de desempenho de todos os processos, performance e riscos.

Os times de tecnologia não são os únicos do lado da organização que se beneficiam do DevOps. O lado dos negócios de sua empresa verá menos reclamações dos clientes, uma entrega mais rápida de novos recursos e confiança melhorada em serviços existentes.

O DevOps possibilita que se realize mais com os recursos existentes. Ele aceita a realidade das restrições e mostra como ter sucesso dentro de seu ambiente único.

# Mantendo a CALMS

Ao começar a se familiarizar com DevOps, você provavelmente se deparará com um modelo chamado CALMS. Ele é um acrônimo para cultura, automação, lean, medição e compartilhamento [*sharing*, em inglês], e é uma estrutura útil para compreender os princípios DevOps e avaliar seu sucesso e maturidade ao aplicá-los em sua organização.

## Cultura

Sua cultura precisa ser colaborativa e centrada no cliente, o que significa que seus times de desenvolvimento e operações compreendem que o propósito da tecnologia é facilitar a vida dos clientes. Se eles não encontrarem valor no produto, ele fracassará. A tecnologia é secundária a esse objetivo. As melhores culturas DevOps são extremamente colaborativas e multifuncionais, com pessoas de diferentes times e diversos conjuntos de habilidades trabalhando juntas para criar um produto melhor. Ouvir é um componente crucial da comunicação, e um teste decisivo e fácil da cultura é escutar conversas. As pessoas estão constantemente falando ao mesmo tempo? Se sim, aposto que você tem chances de grandes melhorias culturais à frente.

## Automação

Tarefas repetitivas são o pior pesadelo de um time de tecnologia, não só porque são chatas, mas porque são ineficazes. Os times de tecnologia falam a língua dos computadores para coordená-los a realizar o trabalho que as pessoas não querem fazer. Geralmente as tarefas mais fáceis nas melhorias de automação são os builds de código, os testes automatizados, as implementações e o fornecimento de infraestrutura. Aprofundo a identificação dessas tarefas no Capítulo 3.

## Lean

Lean não se refere apenas à manufatura lean. Aplica-se mais amplamente à natureza dos times de DevOps, que são ágeis e obstinadas. Times lean evitam atividades de baixo impacto porque elas não fornecem valor ao cliente. Outro aspecto do lean é como ele mantém o objetivo da melhoria contínua. Todos abraçam o mindset de crescimento e querem melhorar verdadeiramente.

## UM CONTO DE TECNOLOGIA: O QUE ME LEVOU AO DEVOPS

Quero contar um segredinho. Cheguei ao DevOps por acidente. Pois é! Foi totalmente acidental. Mas acho que minha história fala muito sobre o poder do movimento e da comunidade DevOps.

Eu era uma desenvolvedora Java back-end em uma pequena empresa com um time de desenvolvimento de engenharia tradicional. O time consistia de uma dúzia de desenvolvedores e dois funcionários do time de operações. (Parece a proporção normal, não é?)

O código tinha um bug. Atualizei o código que selecionava imagens de pré-visualização na aplicação. Ainda assim, a página inicial não exibia as mudanças, e os ops me culpavam. Investiguei e concluí que era um problema de CDN (Content Delivery Network). Devido a restrições de acesso dos desenvolvedores em meu time, não conseguimos mitigar o problema. Precisávamos do time de operações.

O especialista ops achava que era um problema de código e se recusou a me ajudar. Fomos e voltamos três vezes antes de eu entrar em um armário e digitar um resumo furiosamente. *Humpty Dumpty: A story of DevOps gone wrong* ["Humpty Dumpty: Uma história de DevOps que deu errado", em tradução livre] foi minha primeira palestra tecnológica e foi inspirada em minhas experiências e frustrações pessoais com desenvolvedores enfrentando o pessoal de operações.

Naquela empresa, e em muitas outras, o time de operações era um gargalo. Ele me impedia de fazer meu trabalho. Mas não era sua culpa. As pessoas envolvidas destacavam o problema, mas a própria questão era um problema de processo.

Minha experiência naquele emprego me levou ao DevOps, que despertou meu interesse. No decorrer do aprendizado de DevOps, encontrei um grande alívio na descoberta de que os problemas que eu havia enfrentado não eram apenas meus! Eu não era uma desenvolvedora ruim. Era apenas humana, e os outros profissionais como eu sentiam as mesmas frustrações em seus empregos. Meu maior desejo é que este livro possa tranquilizá-lo de que sua experiência é válida e comum e também mostrar algumas abordagens que podem ajudar a tornar seu trabalho um pouco mais incrível.

## Medição

Os dados são cruciais para DevOps. A medição do progresso por meio dos dados informará quase todos os aspectos da transformação em sua organização. Mas lembre-se de que esse progresso nunca deve ser atribuído ao desempenho individual. Pense nisso como acompanhar seu progresso no decorrer de uma maratona sem fim, em vez de um modo de saber quando terá “terminado”. Em DevOps, você nunca termina. Ninguém termina! A melhoria é contínua.

Em vez de considerar os dados coletados como uma medida do quanto está se saindo mal, pense neles como uma estimativa de sua melhoria. Celebre as vitórias. Essa abordagem sustenta todo o seu time de tecnologia e mantém os profissionais felizes, motivados e produtivos. Garanto que você estará fazendo algumas coisas bem, e destacá-las é importante. No Capítulo 5, falo sobre o que pode ser medido.

## Compartilhamento

O DevOps foi fundado porque os times de operações e o desenvolvimento tinham alguns conflitos. Eles careciam de interesses em comum e eram incentivados com base em padrões e necessidades diferentes. O time de operações geralmente era medido pela confiança e disponibilidade de uma aplicação, assim como o time de desenvolvimento é, com bastante frequência, incentivado a criar novos recursos para a aplicação. (Falo mais sobre como as operações e o desenvolvimento são medidos na próxima seção.) Você sabe qual é a maior ameaça do tempo em uma atividade? As implementações. Os times de desenvolvimento iniciam as implementações com novas liberações de código. Por isso dizem no mundo de tecnologia que o pessoal de operações “odeia” os desenvolvedores. Normalmente não é algo tão sombrio assim, mas há um pouco de verdade nisso. O atrito praticamente impossibilita a solução de problemas e transforma tudo em um jogo de culpa. O DevOps busca mudar completamente essa atmosfera e criar um ambiente em que ambos os times ensinem um ao outro e se sintam capacitados — construindo, assim, um único time no qual todos contribuem.

## Resolvendo o problema dos interesses conflitantes

Em times tradicionais de tecnologia, os desenvolvedores (aqueles que escrevem o código) e a operação (aquela que implementa os sistemas e mantém a infraestrutura de TI) estão em lados opostos de uma guerra sem fim. Tá, não é exatamente isso. Mas eles não se dão bem, e isso porque são avaliados por critérios diferentes.

Os desenvolvedores normalmente são medidos pelo número de recursos que lançam ou pela quantidade de bugs que corrigem. (Avaliar desenvolvedores por linhas de código escritas é uma péssima ideia. Muitas vezes, os melhores desenvolvedores deletam mais linhas do código do que adicionam.)

Infelizmente, a qualidade e a confiabilidade do código normalmente não são medidas. Como consequência, os desenvolvedores naturalmente priorizam o trabalho que os fará parecer produtivos. Não passam um tempo refatorando código para torná-lo mais legível ou para pagar a dívida técnica acumulada do último grande avanço do produto.

Em comparação a como os desenvolvedores são medidos, o time de operações geralmente é medido pela confiabilidade e pelo tempo ativo das aplicações/ambiente disponibilizado. Você provavelmente já ouviu falar dos cinco 9: 99,999% de disponibilidade. Os cinco 9 significam que seu site só pode ficar offline cinco minutos por ano. Cinco minutos... *por ano*. Isso é pedir demais! Também é caro de manter, por causa da quantidade de armazenamento e de recursos computacionais necessários à disposição. Isso sem falar do impacto pessoal que tem nos indivíduos de operações que recebem a tarefa de manter a disponibilidade nesse nível. É solicitado ao time de operação que realizem esforços heroicos e respondam a problemas independentemente do dia, horário, cargas de trabalho existentes ou obrigações pessoais.

Para esclarecer o conflito: em organizações tradicionais de tecnologia, os desenvolvedores devem implementar um novo código para liberar melhorias ou novos recursos. Mas as implementações são as ações mais comuns que iniciam interrupções no serviço e indisponibilidades das aplicações no ambiente.

Dois problemas surgem dessa situação:

- » **A responsabilidade é isolada.** Os desenvolvedores não sabem como lançar ou dar suporte ao seu código e carecem de conhecimento de sistemas que o capacitem a entender os requisitos de infraestrutura. A maioria dos desenvolvedores não sabe (ou se importa) como seu código realmente executa. Seu trabalho já foi terminado.
- » **Os objetivos e incentivos estão em oposição.** Os desenvolvedores jogam códigos no time de operações e esperam que ela os implemente e garanta que rodem perfeitamente. O time de operações é incentivado pelo tempo de atividade, disponibilidade e confiabilidade. Geralmente supõe que o código foi mal escrito e levará bronca (ou será despedido) por um incidente que não é culpa dele.

Entendeu por que ouvimos suspiros audíveis quando os times de desenvolvedores e de operações interagem? O DevOps busca eliminar tanto os desafios criados pela responsabilidade isolada quanto os objetivos contrários. Ao alinhar os incentivos, compartilhar conhecimento, remover barreiras e respeitar os diferentes papéis, o DevOps pode melhorar drasticamente a comunicação interpessoal e a cooperação de seu time.