

2022 ACCELERATE

Relatório State of DevOps



Patrocínio

Índice

01	Resumo executivo	03	05	Surpresas	55
02	Sua equipe está à altura?	08	06	Informações demográficas e firmográficas	59
03	O que fazer para melhorar?		07	Considerações finais	67
	Introdução	19			
	Nuvem	21	08	Agradecimentos	68
	SRE e DevOps	26			
	Capacidade técnica de DevOps	29	09	Autores	69
	Cultura	37			
04	Por que a segurança da cadeia de suprimentos é importante	42	10	Metodologia	73
			11	Leitura adicional	76

01

Resumo executivo



Derek DeBellis



Claire Peters

Nos últimos oito anos, elaboramos o relatório Accelerate State of DevOps com base em entrevistas com 33 mil profissionais. O foco da nossa pesquisa é examinar como capacidades e práticas indicam os resultados que consideramos centrais para DevOps:

- Desempenho da entrega de software: as quatro métricas principais do desempenho da entrega de software, que são frequência de implantação, tempo de lead para alterações, taxa de falha nas alterações e tempo de restauração do serviço.
- Desempenho operacional: a quinta métrica principal, que é a confiabilidade.
- Desempenho da organização: se a organização consegue cumprir as metas de desempenho e de lucratividade.

Também analisamos os fatores que levam a outros resultados, como esgotamento e a probabilidade de os funcionários recomendarem as próprias equipes.



Como proteger a cadeia de suprimentos de software

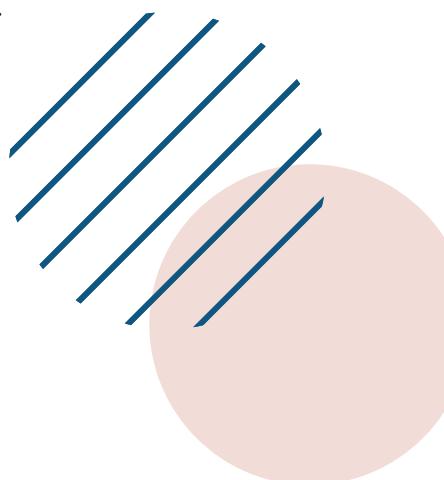
Em 2021, descobrimos que proteger a cadeia de suprimentos de software é essencial para alcançar muitos resultados importantes.

Neste ano, nos aprofundamos mais na segurança da cadeia de suprimentos de software, que é o tema principal da nossa pesquisa e relatório. Usamos o framework [Supply Chain Levels for Secure Artifacts \(SLSA\)](#) para analisar práticas técnicas que auxiliam no desenvolvimento dessa segurança. Também usamos o [Secure Software Development Framework \(NIST SSDF\)](#) do Instituto Nacional de Padrões e Tecnologia para analisar atitudes, processos e práticas não técnicas relacionados à proteção da cadeia de suprimentos de software.

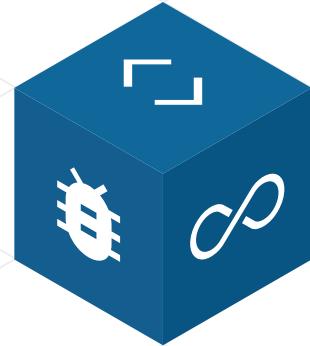
Descobrimos que o maior fator para prever as práticas de segurança de desenvolvimento de aplicativos de uma organização era cultural e não técnico.

As culturas com alta confiança, baixa culpa e foco em desempenho adotam práticas de segurança emergentes em frequência 1,6 vez maior que culturas com baixa confiança e alta culpa concentradas em regras. Também encontramos evidências iniciais de que a verificação de segurança antes da implantação é eficaz em identificar dependências vulneráveis, o que reduz as vulnerabilidades no código de produção.

A adoção de boas práticas de segurança no desenvolvimento de aplicativos está relacionada a outros benefícios. Constatamos que as equipes que estabelecem essas práticas de segurança reduziram o esgotamento dos desenvolvedores. Entre as equipes com baixos níveis de práticas de segurança, a probabilidade de altos níveis de esgotamento é 1,4 vez maior que nas equipes com altos níveis de segurança.¹ As equipes que estabelecem práticas de segurança têm probabilidade muito maior de recomendar a própria equipe aos demais. Além disso, as práticas de segurança relacionadas a SLSA sugerem um bom desempenho tanto da organização quanto da entrega de software, mas é preciso implantar sólidos recursos de integração contínua para que isso ocorra.



¹Nessa estatística, consideramos "altos níveis" como desvio-padrão ≥ 1 na pontuação (por exemplo, segurança), e "baixos níveis" como desvio-padrão ≤ -1 .



Fatores que impulsionam o desempenho organizacional

Além das práticas de segurança mencionadas acima, as principais variáveis que afetam o desempenho organizacional costumam estar em uma destas categorias:

Cultura organizacional e da equipe

As culturas com alta confiança e baixa culpa, segundo a definição de [Westrum](#), costumam ter maior desempenho organizacional. Da mesma forma, as organizações com equipes que se sentem apoiadas por financiamento e patrocínio da liderança costumam ter melhor desempenho. A estabilidade e a percepção positiva da equipe (probabilidade de recomendá-la) também costumam levar a níveis mais altos de desempenho organizacional. Por último, as empresas que oferecem condições de trabalho flexíveis costumam ter níveis mais altos de desempenho organizacional.

Confiabilidade

As práticas que associamos com a engenharia de confiabilidade (como metas claras, métricas salientes etc.) e a extensão em que as pessoas cumprem as expectativas de confiabilidade são bons indicadores de altos níveis de desempenho organizacional.

Nuvem

Descobrimos que o uso da nuvem é um indicador do desempenho organizacional. As empresas com software criado na nuvem e para a nuvem costumam ter melhor desempenho organizacional. O uso de nuvens privadas, públicas ou híbridas (ou uma mistura delas) corresponde a um melhor desempenho organizacional do que o uso apenas de servidores no local. Entre as organizações que usam nuvens públicas, a probabilidade de desempenho acima da média é 1,4 vez maior que as demais.

O uso da nuvem também parece afetar o desempenho organizacional por meio de outros fatores no nosso conjunto de dados. Um dos exemplos é a segurança da cadeia de suprimentos. Constatamos que as organizações que usam nuvens públicas costumam implementar práticas de SLSA, talvez porque os provedores de nuvem ofereçam componentes para muitas práticas de SLSA, como a automação de builds e implementações.^{2,3} A questão principal é que, ao usar plataformas de nuvem, as equipes têm acesso a muitos recursos e práticas que costumam melhorar o desempenho organizacional.

²Jung, Sun Jae. "Introduction to Mediation Analysis and Examples of Its Application to Real-world Data." Journal of preventive medicine and public health = Yebang Uihakhoe chi vol. 54,3 (2021): 166-172. doi:10.3961/jpmph.21.069

³Carrión, Gabriel Cepeda, Christian Nitzl e José L. Roldán. "Mediation analyses in partial least squares structural equation modeling: Guidelines and empirical examples." Partial least squares path modeling. Springer, Cham, 2017. 173-195.

O contexto é importante

Há muito tempo, DORA considera que os efeitos dependem do contexto geral da equipe. Acreditamos que é importante entender as características da equipe (processos, pontos fortes, limitações e metas) e o ambiente em que o trabalho é feito. Por exemplo, um recurso técnico que é vantajoso em um contexto pode ser prejudicial em outro. Este ano, criamos um modelo claro dessas condições hipotéticas na forma de interações. Muitas das hipóteses foram confirmadas pelos dados deste ano:

- O alto desempenho na entrega de software só beneficia o desempenho organizacional quando esse último desempenho também é alto. Não adianta entregar rapidamente quando o serviço não tem a confiabilidade que os usuários esperam.
- A implementação de controles de segurança da cadeia de suprimentos de software, como aqueles recomendados pelo framework SLSA, têm impacto positivo sobre o desempenho da entrega quando a integração contínua está firmemente estabelecida. Sem a integração contínua, pode haver conflito entre o desempenho da entrega de software e os controles de segurança.
- As práticas de engenharia de confiabilidade do site (SRE) têm impacto não linear sobre a capacidade da equipe de cumprir as metas de confiabilidade. A SRE só auxilia na confiabilidade quando uma equipe alcança certo nível de maturidade nessa prática. Antes de a equipe
- atingir essa maturidade, não detectamos relação entre SRE e o cumprimento de metas de confiabilidade. No entanto, conforme cresce, a adoção da SRE atinge um ponto de inflexão em que o uso dela se torna um forte indicador da confiabilidade. Por sua vez, essa melhora na confiabilidade afeta o desempenho organizacional.
- Capacidades técnicas dependem umas das outras. A entrega contínua e o controle de versão amplificam a capacidade mútua de melhorar o desempenho da entrega de software. A combinação de entrega contínua, arquitetura acoplada com flexibilidade, controle de versão e integração contínua promove um desempenho na entrega de software maior que a soma das partes.



As condições de que a entrega depende e a necessidade de entender o contexto de uma equipe nos levam a uma conclusão semelhante a este trecho de 2021:

"Para fazer melhorias significativas, as equipes precisam adotar uma filosofia de melhoria contínua. Use os comparativos de mercado para medir seu estado atual, identificar restrições com base nas capacidades investigadas pela pesquisa e testar melhorias para aliviar essas restrições. A experimentação envolverá uma combinação de vitórias e falhas, mas em ambos os cenários as equipes poderão realizar ações significativas como resultado das lições aprendidas."

De fato, este ano descobrimos um efeito alinhado a essa filosofia geral: as equipes que reconhecem a necessidade de melhorar continuamente costumam ter melhor desempenho organizacional do que as demais.

Ou seja, as equipes precisam se adaptar continuamente e testar novas práticas de desenvolvimento de software.

Sabemos disso porque, em geral, as equipes que fazem isso têm melhor desempenho organizacional. Nem sempre (o que funciona para uma organização não funciona necessariamente para outra), mas na maioria do tempo sim. Ao testar práticas de DevOps, não se preocupe com eventuais falhas enquanto você descobre o que funciona para sua equipe.

Este ano, também descobrimos algumas [surpresas](#) nos dados, mas você vai precisar continuar lendo para saber.

As equipes que reconhecem a necessidade de melhorar continuamente costumam ter melhor desempenho organizacional do que as demais.

02

Sua equipe está à altura?



Derek DeBellis

Quer saber como sua equipe se compara às outras empresas do setor? Esta seção inclui a avaliação de comparativo de mercado mais recente do desempenho de DevOps. Vamos examinar a maneira como as equipes desenvolvem, entregam e operam sistemas de software. Depois vamos dividir as respostas em grupos que indicam as combinações mais comuns de desempenho de DevOps.

Este ano, incluímos dois métodos de agrupamento. O primeiro é baseado no precedente histórico. Neste método, criamos grupos com base em quatro métricas que indicam o desempenho da entrega de software: tempo de lead, frequência de implantação, tempo de restauração do serviço e taxa de falha nas alterações. Há um resumo de cada uma abaixo. O objetivo é quantificar o desempenho atual da sua equipe para que você possa compará-lo a outras equipes.

O segundo método inclui uma quinta métrica, confiabilidade, que usamos para entender o desempenho operacional. Por que adicionar uma métrica nova à análise de agrupamento? Porque temos visto constantemente a importância dela. Temos evidências que sugerem que o desempenho da entrega pode prejudicar o desempenho organizacional quando não está associado a um sólido desempenho operacional. Ao contrário do nosso método tradicional de agrupamento, este é um exercício descritivo, que tenta mostrar maneiras comuns em que as equipes operam entre o desempenho da entrega e o operacional. Por isso, não é sempre óbvio qual grupo é melhor.

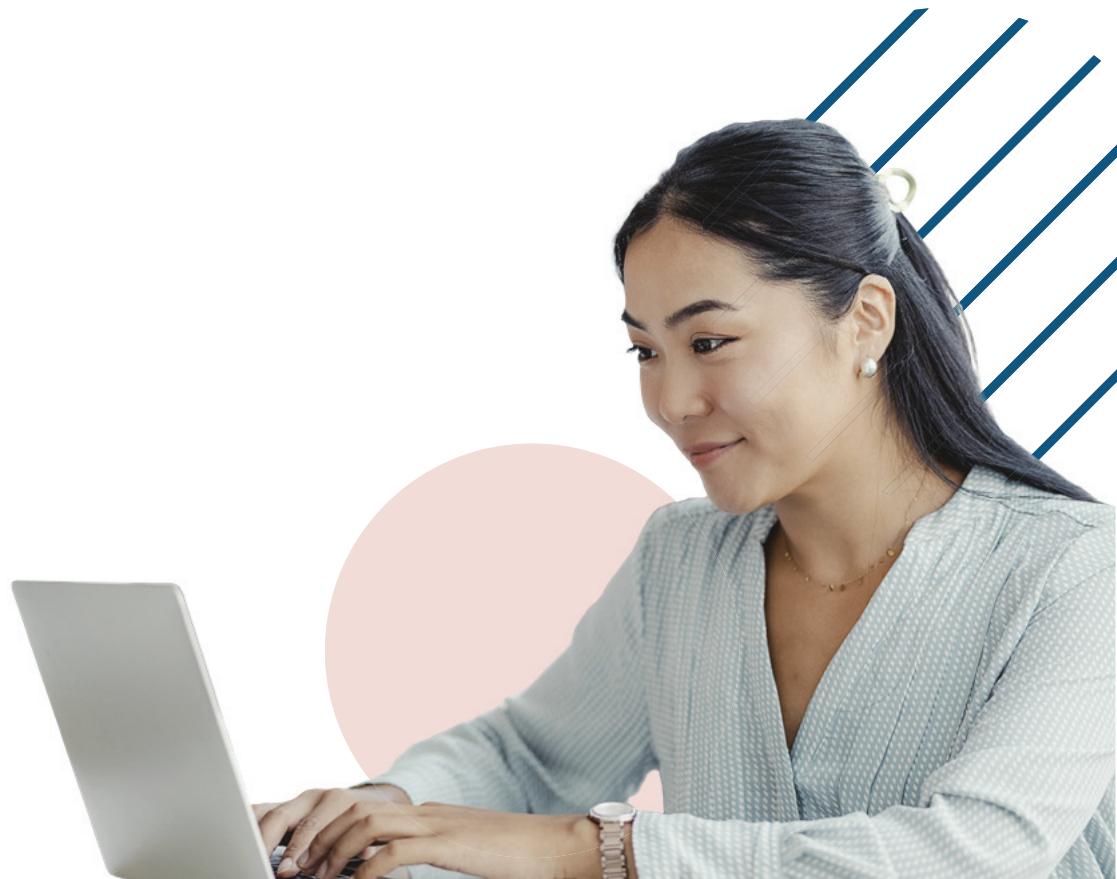
Primeiro vamos ver uma breve visão geral das cinco medidas que usamos para entender o desempenho operacional e da entrega de software.



Desempenho operacional e da entrega de software

Para atender à demanda dos setores em evolução, as organizações precisam entregar e operar software de maneira rápida e confiável. Quanto mais rápido as equipes puderem fazer mudanças no software, mais rapidamente você conseguirá agregar valor aos seus clientes, realizar experimentos e receber feedback. Com oito anos de coleta e pesquisa de dados, desenvolvemos e validamos quatro métricas que medem o desempenho da entrega de software. Desde 2018, incluímos uma quinta métrica

para indicar a capacidade operacional. As equipes que se destacam nas cinco medidas demonstram um desempenho organizacional excepcional. Chamamos essas cinco medidas de **desempenho operacional e da entrega de software (SDO, na sigla em inglês)**. Essas métricas estão relacionadas a resultados no nível do sistema. Isso evita as armadilhas comuns das métricas de software de rastreamento, que podem colocar funções umas contra as outras e fazer otimizações locais ao custo do resultado geral.

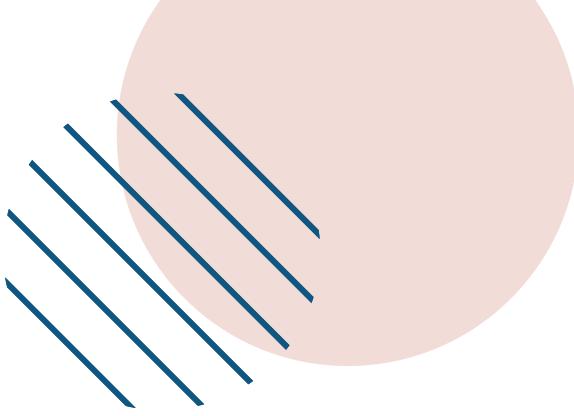


Cinco métricas do desempenho operacional e de entrega

As quatro métricas de desempenho da entrega de software podem ser consideradas em termos de capacidade e estabilidade. Medimos a capacidade usando o **tempo de lead das alterações de código** (ou seja, o tempo da confirmação do código até o lançamento na produção) e a **frequência de implantação**. Medimos a estabilidade usando o **tempo para restaurar um serviço** após um incidente e a **taxa de falha nas alterações**.

A quinta métrica, que representa o **desempenho operacional**, é uma medida de práticas operacionais modernas. Baseamos o **desempenho operacional** na **confiabilidade**, que significa verificar se os serviços atendem às expectativas dos usuários, como disponibilidade e desempenho. Antes nós medíamos a disponibilidade em vez da confiabilidade. Mas, como a disponibilidade é um foco específico da engenharia de confiabilidade, expandimos para medir a confiabilidade em 2021. Dessa forma, representamos melhor disponibilidade, latência, desempenho e escalonabilidade. Especificamente, pedimos aos participantes para avaliar a capacidade deles de atingir ou exceder as metas de confiabilidade. Constatamos que as equipes com graus variados de desempenho da entrega chegam a resultados melhores (por exemplo, menos esgotamento) quando também priorizam o desempenho operacional.





Método de agrupamento histórico: agrupando o desempenho da entrega

Neste ano, ao avaliar a solução com quatro grupos que usamos desde 2018, percebemos que os dados mostravam claramente um grupo com baixo desempenho e outro com alto desempenho. No entanto, os dois grupos que costumávamos usar para demarcar o desempenho médio e o desempenho alto não eram diferenciados o suficiente para justificar uma divisão. Além disso, os vários índices usados para escolher a melhor solução de agrupamento

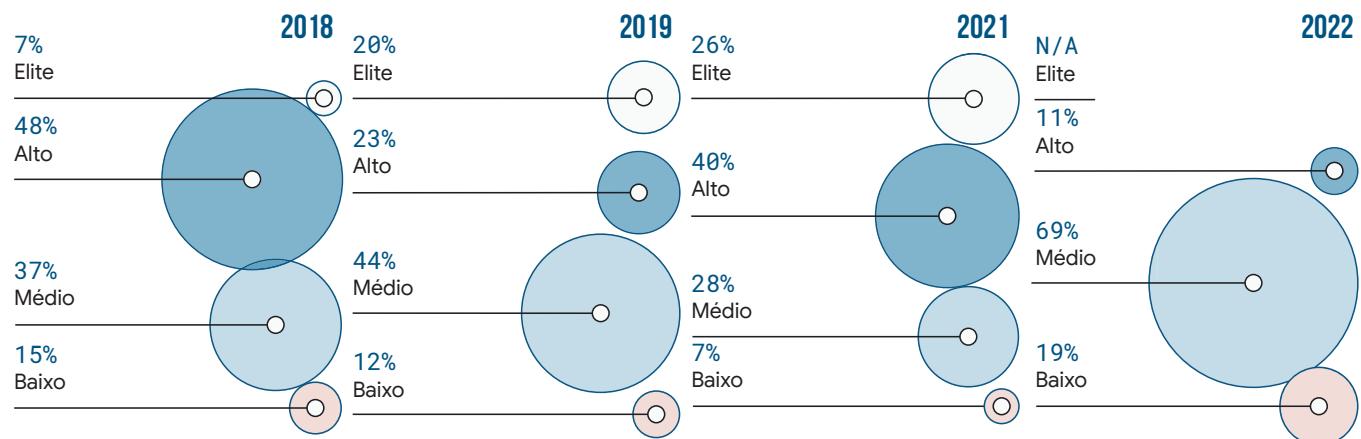
indicavam sempre que três grupos capturavam melhor os dados, seja qual fosse a técnica aplicada. A tabela abaixo mostra as características do desempenho da entrega de cada grupo.

A diferença surpreendente em relação ao ano passado é que não consideramos nenhum grupo como elite desta vez. O grupo alto deste ano é uma mistura dos grupos alto e elite do ano passado. Decidimos omitir o grupo de elite porque o grupo com melhor desempenho não indica características suficientes da elite do ano passado.

Métrica de desempenho na entrega de software	Baixo	Médio	Alto
Frequência de implantação No caso de aplicativos ou serviços principais em que você trabalha, com que frequência sua organização implanta códigos na produção ou os libera para usuários finais?	Entre uma vez por mês e uma vez a cada seis meses	Entre uma vez por semana e uma vez por mês	Sob demanda (várias implantações por dia)
Tempo de lead para alterações No caso de aplicativos ou serviços principais em que você trabalha, qual é o tempo de lead para alterações (ou seja, quanto tempo leva para ir da confirmação do código à execução do código na produção)?	Entre um mês e seis meses	Entre uma semana e um mês	Entre um dia e uma semana
Tempo de restauração do serviço No caso de aplicativos ou serviços principais em que você trabalha, quanto tempo geralmente leva para restaurar o serviço quando ocorre um incidente ou um defeito que afeta os usuários (por exemplo, interrupção não planejada ou deficiência do serviço)?	Entre uma semana e um mês	Entre um dia e uma semana	Menos de um dia
Taxa de falha nas alterações No caso de aplicativos ou serviços principais em que você trabalha, qual porcentagem de alterações na produção ou liberadas para os usuários resulta em queda no serviço (por exemplo, leva à deficiência ou interrupção do serviço) e, posteriormente, exige remediação (por exemplo, exige um hotfix, reversão, correção de encaminhamento, patch)?	46% a 60%	16% a 30%	0% a 15%

Isso sugere que a amostra não representa equipes ou organizações com funcionários que sentem que evoluíram. Uma hipótese possível (que não podemos afirmar como fato por falta de dados) é que houve redução da inovação no desenvolvimento de software, em termos de práticas, ferramentas e compartilhamento de informações. A redução pode ser resultado da atual pandemia, que dificulta o compartilhamento de conhecimentos e práticas em equipes e organizações. Isso teria reduzido as oportunidades para se reunir e aprender uns com os outros, o que, por sua vez, tornaria a inovação mais lenta. Esperamos analisar melhor os dados para entender as causas dessa constatação.

Dito isso, quando comparamos os grupos baixo, médio e alto deste ano com os do ano passado, vemos um leve aumento no desempenho da entrega. Parece que os grupos deste ano estão entre dois dos grupos do ano passado. O grupo alto de 2022 está entre o alto e o elite de 2021. O grupo baixo de 2022 parece estar entre o baixo e o médio de 2021. O aumento no grupo de baixo desempenho sugere que, embora o teto do desempenho na entrega tenha subido, o piso desceu.

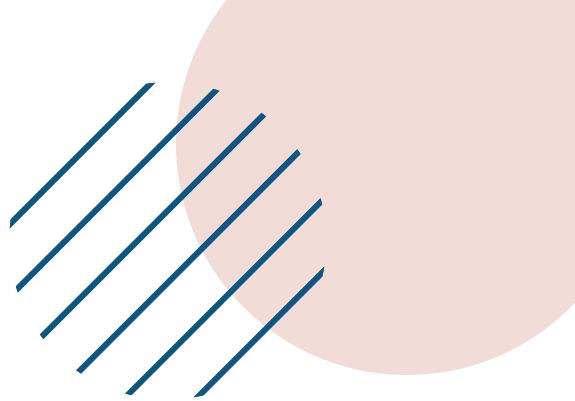


Os números na tabela acima mostram que a porcentagem de participantes com alto desempenho é a mais baixa em quatro anos. Já a porcentagem de participantes com baixo desempenho teve um aumento drástico de 7% em 2021 para 19% este ano. Mais de dois terços dos participantes deste ano estão no grupo médio. A evidente queda nos grupos com desempenho alto e de elite pode indicar que os participantes deste ano estão em organizações ou equipes que ainda não estabeleceram (ou estão estabelecendo agora) uma cultura de DevOps, que vemos surgir em várias equipes modernas.

Talvez estejamos falando demais sobre as diferenças entre 2021 e 2022 em vez de destacar as semelhanças. Há muitas características semelhantes nos grupos dos dois anos, incluindo uma enorme separação entre participantes com desempenho alto e baixo. Por exemplo, estimamos que os participantes com alto desempenho¹ tenham 417 vezes mais implantações que aqueles com baixo desempenho.

¹ Consulte a seção "Metodologia" para entender como a estimativa é feita





Agrupamento do desempenho na entrega e do desempenho operacional

Decidimos fazer uma análise por grupos nas três categorias que as cinco métricas representam:

capacidade de processamento (composto por tempo de lead das alterações no código e frequência de implantação), **estabilidade** (composto por tempo de restauração do serviço e taxa de falhas nas alterações) e **desempenho operacional** (confiabilidade). O motivo é o papel essencial que o desempenho operacional tem

em nossos modelos. Nas organizações que não demonstram um desempenho operacional sólido, a capacidade de processamento e a estabilidade têm menor impacto no desempenho organizacional. Achamos que descrever o cenário do desempenho de DevOps sem considerar o desempenho operacional significa deixar uma parte essencial de fora.

Ao analisar os dados, chegamos a uma solução com quatro grupos. Confira os números dos quatro grupos e os respectivos nomes:

Grupo	Estabilidade		Desempenho operacional	Capacidade de processamento		% participantes
	Tempo de restauração do serviço	Taxa de falha nas alterações		Confiabilidade	Tempo de lead	
Início	Entre um dia e uma semana	31% a 45%	Às vezes atende às expectativas	Entre uma semana e um mês	Entre uma vez por semana e uma vez por mês	28%
Fluidez	Menos de uma hora	0% a 15%	Geralmente atende às expectativas	Menos de um dia	Sob demanda (várias implantações por dia)	17%
Desaceleração	Menos de um dia	0% a 15%	Geralmente atende às expectativas	Entre uma semana e um mês	Entre uma vez por semana e uma vez por mês	34%
Desativação	Entre um mês e seis meses	46% a 60%	Geralmente atende às expectativas	Entre um mês e seis meses	Entre uma vez por mês e uma vez a cada seis meses	21%

No entanto, se você visitasse duas equipes do mesmo grupo, elas pareceriam muito diferentes, e nossa história não captura tudo que você veria. A história que apresentamos para cada grupo acima é uma tentativa de usar nossa experiência trabalhando com várias equipes para tornar inteligíveis esses padrões nos dados. Além disso, se você visitasse a mesma equipe em dois momentos diferentes, é possível que ela não ficasse no mesmo grupo. Um motivo para isso é que a equipe melhorou ou piorou. Outra possibilidade é que a equipe migrou para um padrão de implantação mais apropriado ao estado atual do aplicativo ou serviço dela. Por exemplo, no início do desenvolvimento de um aplicativo ou um serviço, o foco da equipe era a exploração (grupo Início). Mas, conforme foi encontrando o nicho dela, a equipe mudou o foco para a confiabilidade (grupo Fluidez ou Desaceleração).

Cada grupo tem as próprias peculiaridades e é responsável por uma proporção substancial das respostas.

O grupo **Início** não tem desempenho bom nem ruim nas nossas dimensões. Talvez o grupo esteja na fase inicial do desenvolvimento do produto, recurso ou serviço. Talvez esteja menos preocupado com a confiabilidade porque o foco é receber feedback, entender se o produto é adequado ao mercado e, em geral, explorar.

O grupo **Fluidez** tem bom desempenho em todas as características: alta confiabilidade, alta estabilidade, alta capacidade de processamento. Apenas 17% dos participantes estão nesse estado.

Os participantes do grupo **Desaceleração** não implantam com frequência, mas geralmente têm êxito quando o fazem. Mais de um terço dos participantes está nesse grupo, sendo o maior e mais representativo da amostra. Este padrão é característico (mas não exclusivo) de situações em que as equipes estão melhorando aos poucos, mas elas e os clientes estão satisfeitos com a situação atual do aplicativo ou produto.

Por fim, o grupo **Desativação** é como uma equipe que está trabalhando em um serviço ou um aplicativo que ainda tem valor para ela e os clientes, mas que não está mais em desenvolvimento ativo.

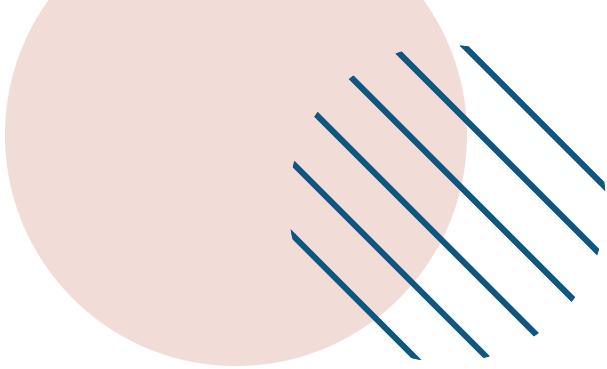
As principais diferenças entre os grupos são as práticas e as capacidades técnicas. Devido ao alto nível de desempenho operacional e na entrega de software do grupo **Fluidez**, decidimos examinar as diferenças entre ele e os outros grupos nesses aspectos. Constatamos que, em relação aos demais grupos, o grupo **Fluidez** dedica mais atenção a estes itens:

- Arquiteturas com acoplamento flexível: na medida em que as equipes podem fazer grandes mudanças no design do sistema sem depender das mudanças de outras equipes em outros sistemas
- Aumentar a flexibilidade: a flexibilidade da empresa em relação ao trabalho dos funcionários
- Controle de versões: a forma de gerenciar mudanças no código e configuração do aplicativo, na configuração do sistema etc.
- Integração contínua (CI): a frequência de integração das ramificações na linha principal
- Entrega contínua (CD): capacidades para colocar mudanças em produção de maneira segura, sustentável e eficiente

² (Westrum)

Algo curioso é que o grupo **Fluidez** costuma se preocupar menos com a documentação. Ano passado, constatamos que a documentação é essencial para o desempenho operacional e da entrega (SDO, na sigla em inglês). Como o grupo **Fluidez** consegue bom desempenho de SDO sem grande dedicação à documentação? Em primeiro lugar, há muitos caminhos para um bom desempenho de SDO fora da documentação. Além disso, é possível que o grupo **Fluidez** refatore constantemente o código para criar um processo com documentação automática. Assim, a necessidade de documentos como descrevemos na pesquisa seria menor.

O grupo **Desaceleração**, que forma a maior proporção dos participantes, é formado geralmente por pessoas de organizações maiores, com menos recursos nativos da nuvem do que os demais. Imagine uma empresa muito madura, com processos calcificados que, no fim das contas, ainda proporcionam experiências estáveis e confiáveis (e valiosas) aos usuários finais. O grupo mostra uma cultura generativa e orientada ao desempenho.² Uma das características mais interessantes do grupo Desaceleração é a combinação incomum de baixa capacidade de processamento e alta cultura de trabalho positiva (uma cultura "generativa" segundo Westrum). É mais comum ver capacidade de processamento e cultura de trabalho proporcionais (ambas altas ou baixas). Esperamos pesquisar mais para entender melhor a relação entre os dois fatores.



Também comparamos os grupos em três resultados: esgotamento, desempenho organizacional e trabalho não planejado. As constatações foram contrárias às nossas expectativas. O grupo **Desativação** teve maior desempenho organizacional que os demais. Analisando as características do grupo (baixa estabilidade e baixa capacidade de processamento), isso parece contrariar as constatações anteriores de DORA. Mas, em vez de considerar isso uma anomalia aleatória (grande probabilidade), queremos analisar algumas possíveis explicações.

Para entender essas constatações, é importante ter em mente outra constatação complementar. O grupo **Desativação** consegue alto desempenho organizacional a um grande custo: as equipes do grupo têm os maiores índices de esgotamento, se sentem mais suscetíveis a erros e recebem mais trabalho não planejado. Ao mesmo tempo, os resultados sugerem que a confiabilidade é o suficiente para alcançar alto desempenho organizacional, mas, sem velocidade e estabilidade, a equipe vai pagar um alto custo em esgotamento e trabalho não planejado.

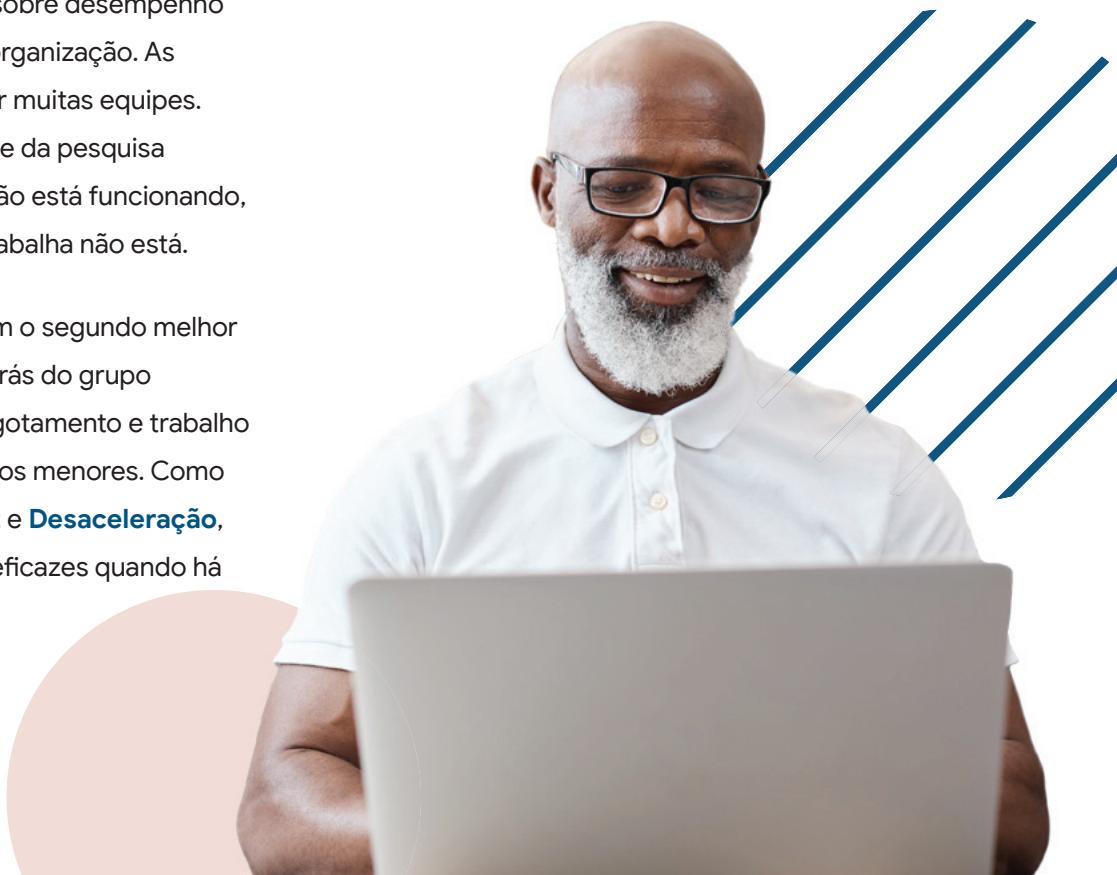
Temos outras hipóteses para explicar por que o grupo **Desativação** tem melhor desempenho organizacional que os demais, especialmente o grupo **Fluidez**. Confira os pontos a seguir.

- Existem atributos relacionados aos quatro grupos que não estão nos nossos dados. Por exemplo, o tamanho da organização pode ser um bom indicador da maturidade. O grupo **Fluidez** é formado principalmente por empresas pequenas, o que pode indicar que os produtos delas estão na fase inicial.
- A maioria dos integrantes do grupo **Fluidez** são de empresas menores, que podem estar menos limitadas por processos históricos e infraestrutura. Por isso, elas teriam processos de DevOps mais sofisticados. Dito isso, os dados mostram uma correlação positiva entre o tamanho da organização e o desempenho organizacional, por motivos que podem não ter ligação com tecnologia.
- O grupo **Fluidez** costuma ignorar os princípios descritos na cultura generativa de Westrum. Vimos que isso geralmente prejudica o desempenho organizacional.
- As organizações de cada grupo podem ter definições diferentes sobre expectativas de confiabilidade e maneiras diferentes de monitorá-las. O mesmo vale para a definição das metas de desempenho organizacional.

- O grupo **Desativação** tem alto desempenho organizacional no curto prazo, mas como será o desempenho no longo prazo? O esgotamento vai causar rotatividade dos funcionários? As organizações vão conseguir escalar os processos?
- Estamos fazendo as perguntas em vários níveis. Perguntamos sobre capacidades técnicas (como arquitetura com acoplamento flexível) no nível da equipe, mas perguntamos sobre desempenho organizacional no nível da organização. As organizações costumam ter muitas equipes. É possível que o participante da pesquisa reconheça que a organização está funcionando, mas a equipe em que ele trabalha não está.

Além disso, o grupo **Fluidez** tem o segundo melhor desempenho organizacional, atrás do grupo **Desativação**, e os níveis de esgotamento e trabalho não planejado dele estão entre os menores. Como demonstram os grupos **Fluidez** e **Desaceleração**, filosofias de DevOps são mais eficazes quando há confiabilidade.

Estamos interessados em continuar explorando novas maneiras de descrever as variações do setor. A partir de agora, queremos continuar incluindo o desempenho operacional como uma dimensão relevante para entender as variações. Também queremos evitar grupos muito prescritivos e avaliativos (como Elite). Em vez disso, o foco será o simples exercício descritivo de identificar constelações comuns do desempenho de SDO.



03

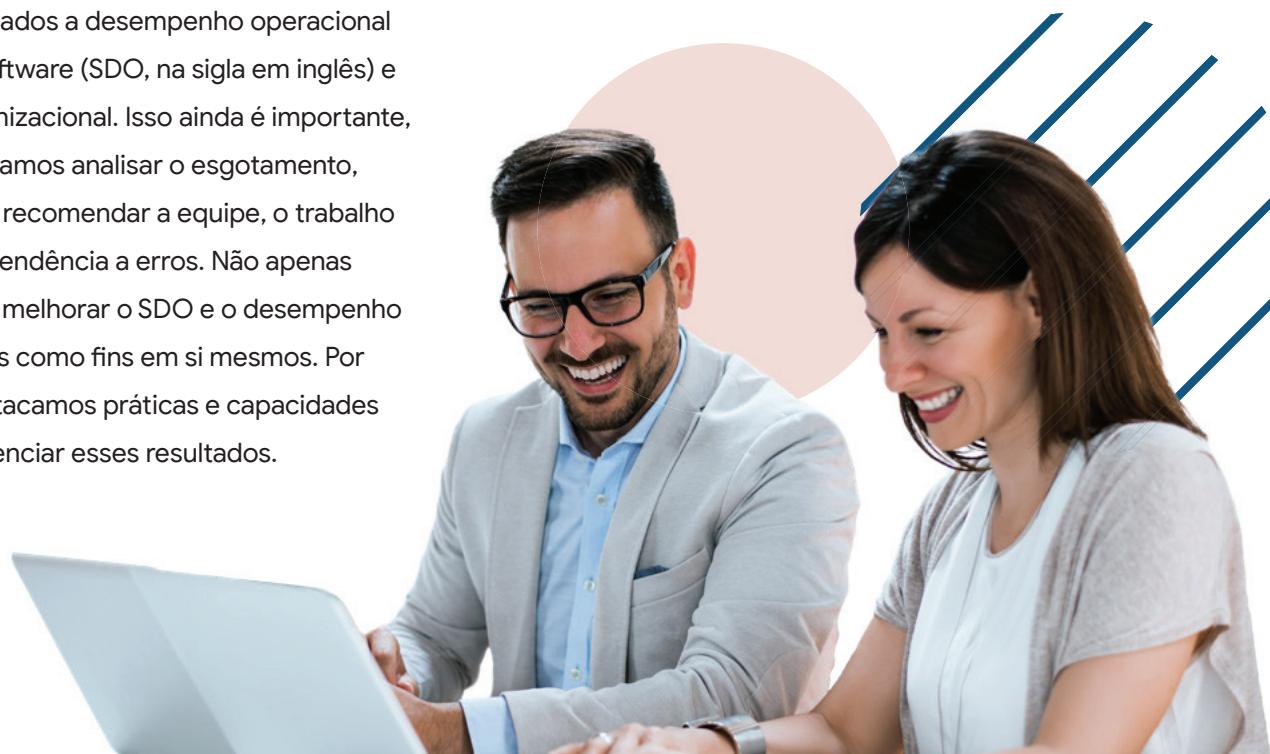
O que fazer para melhorar?

O que fazer para melhorar os vários resultados?

O objetivo do relatório State of DevOps é apresentar orientações baseadas em evidências, que ajudem sua equipe a se concentrar em práticas e capacidades de DevOps e chegar aos resultados importantes para vocês. Neste ano, expandimos nossa investigação para a segurança e o conjunto de resultados que as equipes querem. Antes nosso foco eram os resultados relacionados a desempenho operacional e da entrega de software (SDO, na sigla em inglês) e desempenho organizacional. Isso ainda é importante, mas também queríamos analisar o esgotamento, a probabilidade de recomendar a equipe, o trabalho não planejado e a tendência a erros. Não apenas como maneiras de melhorar o SDO e o desempenho organizacional, mas como fins em si mesmos. Por isso, este ano, destacamos práticas e capacidades que parecem influenciar esses resultados.



Derek DeBellis





O modelo de pesquisa deste ano mudou para refletir melhor uma teoria em que DORA é baseado: não existe uma abordagem única para DevOps. Na prática, constatamos que, para fazer recomendações, é preciso entender o contexto geral da equipe. Uma prática que é benéfica para uma equipe pode ser prejudicial para outra. Por exemplo, há tempos formulamos a hipótese de que as capacidades técnicas (como arquitetura com acoplamento flexível, desenvolvimento com base na linha principal, controle de versões e integração contínua) têm maior impacto positivo sobre o desempenho da entrega de software quando existe entrega contínua. Este ano, modelamos essa e outras interações de forma explícita. A meta é ampliar nosso conhecimento de "o que tem efeito sobre o quê?" para "em que condições estes efeitos existem, são amplificados ou atenuados?". Entender essas condições foi uma tarefa complicada e confusa, mas estamos felizes de mostrar algumas constatações iniciais a você.

Os modelos de pesquisa deste ano e dos anos anteriores estão disponíveis no [nossa site](#).

Além das quatro chaves

Como as métricas de DORA melhoraram o desempenho do desenvolvimento e das operações? Uma equipe multifuncional de engenheiros de software da Liberty Mutual Insurance analisa o desempenho regularmente usando as [métricas "quatro chaves"](#) de DORA. Por exemplo, Jenna Dailey, chefe de scrum sênior da Liberty Mutual, contou que uma equipe usou a pesquisa de DORA para identificar um gargalo, migrar para uma abordagem de desenvolvimento conduzida por testes e melhorar o desempenho geral.

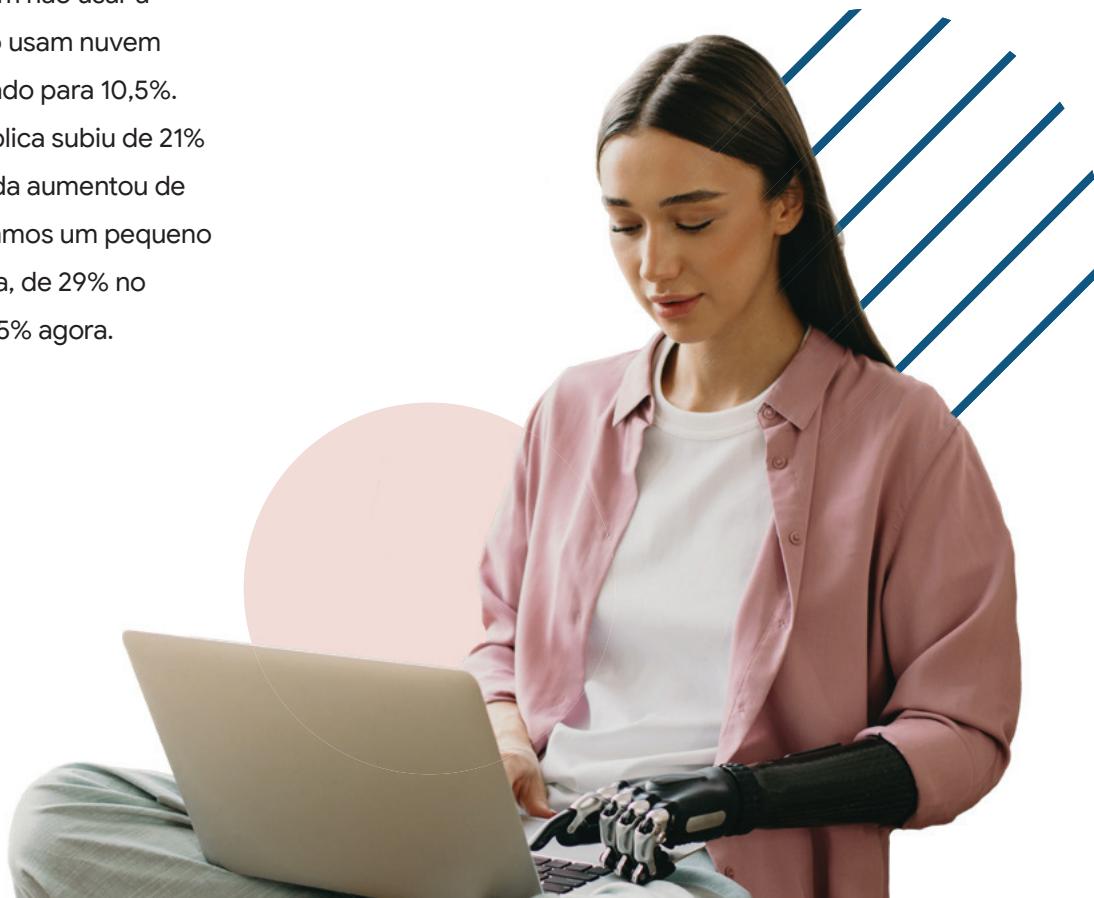
Saiba mais sobre a abordagem da Liberty Mutual para usar dados e métricas DORA e melhorar a qualidade e a entrega de software no recente [Tomorrow Talks](#).



Eric Maxwell

Nuvem

Continuando o movimento que vimos nos últimos anos, o uso da computação em nuvem segue crescendo. De fato, a porcentagem de pessoas que afirmam usar a nuvem pública, incluindo várias nuvens, aumentou de 56% em 2021 para 76% agora. O número de pessoas que afirmam não usar a nuvem, incluindo aquelas que não usam nuvem privada, caiu de 21% no ano passado para 10,5%. O uso de mais de uma nuvem pública subiu de 21% para 26%, e o uso da nuvem híbrida aumentou de 25% para 42,5%. Também registramos um pequeno aumento no uso da nuvem pública, de 29% no relatório do ano passado para 32,5% agora.



O uso da computação em nuvem tem impacto positivo no desempenho organizacional em geral. Os participantes que usam a nuvem têm **probabilidade 14% maior de alcançar** as metas de desempenho organizacional do que os demais que não a usam.

Como mostramos nos anos anteriores e continuamos a validar neste relatório, o uso da computação em nuvem tem um impacto positivo sobre o desempenho organizacional em geral. Os participantes que usam a nuvem têm **probabilidade 14% maior de alcançar** as metas de desempenho organizacional do que os demais que não a usam. Segundo nossa pesquisa, com a computação em nuvem, as equipes se superam em aspectos como segurança da cadeia de suprimentos de software e confiabilidade, fatores que levam ao desempenho organizacional.

Algo surpreendente é que usuários de todos os tipos de nuvem (pública, privada, híbrida e multi) mostraram uma associação negativa com a taxa de falhas nas alterações, ou seja, um aumento

da taxa. Esse fato pede uma maior investigação. Em vez de especular quanto aos motivos, vamos investigar melhor em pesquisas futuras. Mas, com poucas exceções, o uso de aplicativos nativos da nuvem (que foram criados e arquitetados para a nuvem) se destacou com sinais positivos em tudo que pesquisamos.

O uso de qualquer plataforma de computação em nuvem, pública ou privada, contribui positivamente para os resultados de cultura e de ambiente de trabalho (por exemplo, cultura generativa, menos esgotamento, mais estabilidade e maior satisfação dos funcionários). Os usuários da nuvem tiveram pontuação 16% maior nesses resultados culturais.

O uso da nuvem continua acelerando ano a ano

	2022	Variação desde 2021
Nuvem híbrida	42,47%	25%
Uma ou várias nuvens públicas	76,08%	36%
Nuvem privada	32,55%	12%
Sem nuvem	10,55%	-50%



O uso da nuvem híbrida e de várias nuvens (bem como privada) parece ter impacto **negativo** sobre os indicadores de desempenho da entrega de software (MTTR, tempo de lead e frequência de implantação), a menos que os participantes tenham alto nível de **confiabilidade**.

Nuvem híbrida e várias nuvens aumentam o desempenho organizacional

Continuamos vendo sólidas indicações de que o uso da nuvem híbrida e de várias nuvens públicas tem impacto positivo sobre a organização. Os profissionais que usam várias nuvens têm desempenho organizacional 1,4 vez maior que aqueles que não usam a nuvem. No entanto, o uso de nuvem híbrida e de várias nuvens (bem como privada) parece ter um impacto negativo sobre vários indicadores de desempenho da entrega de software (MTTR, tempo de lead e frequência de implantação), a menos que os participantes também tenham alto nível de confiabilidade. A constatação também demonstra

a importância de uma prática robusta de SRE e a importância da confiabilidade na entrega de software.

Em 2021, pedimos que os participantes contassem o motivo *principal* para usar várias nuvens públicas. Já em 2022, pedimos que os participantes contassem todos os benefícios advindos do uso de vários provedores de nuvem. A disponibilidade foi o benefício mais citado, o que coincide com a atenção que percebemos no setor em relação à confiabilidade: para serem confiáveis, os serviços precisam estar disponíveis. Mais de 50% dos participantes afirmam usar os benefícios específicos de vários provedores de nuvem.

Benefícios da adoção de vários provedores de nuvem

Disponibilidade	62,61%
Usar os benefícios específicos de cada provedor	51,59%
Divisão da confiança entre vários provedores	47,54%
Recuperação de desastres	43,48%
Compliance jurídico	37,97%
Tática de negociação ou requisito de compras	19,13%
Outros	4,06%



O uso das cinco características da computação em nuvem é o início de uma longa cadeia causal que leva ao desempenho organizacional.

As cinco características da computação em nuvem

Mantendo nossa abordagem de pesquisa, procuramos descobrir não apenas se os participantes usam tecnologias de computação em nuvem, mas como eles usam as tecnologias. Para isso, perguntamos sobre as cinco características essenciais da computação em nuvem, segundo a definição do Instituto Nacional de Padrões e Tecnologia (NIST, na sigla em inglês).

Autoatendimento sob demanda: os consumidores podem provisionar recursos de computação quando necessário, automaticamente, sem interação humana do provedor.

Amplo acesso à rede: os recursos têm ampla disponibilidade, e os consumidores podem acessá-los por vários clientes como smartphones, tablets, laptops e estações de trabalho.

Pooling de recursos: os recursos do provedor são agrupados em um modelo multilocatário, com recursos físicos e virtuais atribuídos dinamicamente sob demanda.

O cliente geralmente não tem controle direto sobre o local exato dos recursos fornecidos, mas pode especificar a localização em um nível de abstração mais alto, como país, estado ou data center.

Elasticidade rápida: é possível provisionar e liberar a capacidade de forma elástica, para aumentar ou reduzir a escala rapidamente com a demanda. Os recursos disponíveis para provisionamento parecem ser ilimitados e podem ser apropriados em qualquer quantidade a qualquer momento.

Serviço mensurado: os sistemas de nuvem controlam e otimizam recursos automaticamente por meio da medição em um nível de abstração apropriado ao tipo do serviço, como armazenamento, processamento, largura de banda e contas de usuário ativas. É possível monitorar e controlar o uso de recursos, além de gerar relatórios sobre a transparência no uso.

Este relatório valida a pesquisa DORA dos três anos anteriores, concluindo que a presença dessas cinco características em uma organização tem impacto positivo sobre o desempenho operacional e da entrega de software. Também constatamos que as características melhoram o desempenho organizacional porque iniciam processos que afetam a organização de maneiras positivas. Demonstrar as cinco características da computação em nuvem é a primeira etapa de uma longa jornada que aumenta o desempenho organizacional.

Em 2022, percebemos que as equipes aproveitam os diferenciadores da computação em nuvem cada vez mais. Pelo quarto ano consecutivo, houve crescimento na adoção das cinco características. O pool de recursos teve o maior aumento, de 14%. A elasticidade rápida, que era o segundo atributo mais usado ano passado, teve o menor aumento, de 5%.



NIST	2021	2022	Alteração percentual
Acesso amplo à rede	74%	80%	8
Elasticidade rápida	77%	81%	5
Autoatendimento sob demanda	73%	78%	7
Serviço mensurado	78%	83%	7
Pool de recursos	73%	83%	14



14%
de aumento no pool
de recursos

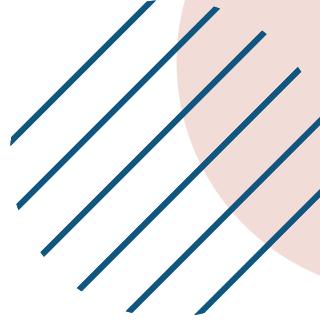


Dave Stanke

SRE e DevOps

As equipes de tecnologia bem-sucedidas fazem mais pela organização do que entregar código. Mais até do que entregar código de qualidade. Elas também garantem que os serviços entregues estejam disponíveis, tenham bom desempenho e sejam consistentes com as expectativas dos usuários ao longo do tempo. A confiabilidade é uma medida multifacetada da capacidade da equipe que cumprir esses compromissos. Este ano, continuamos nossa análise sobre a confiabilidade como fator na entrega de software e nas operações.

A engenharia de confiabilidade do site (SRE) é uma influente abordagem de operações. Ela teve origem no Google e hoje é praticada em muitas organizações. A SRE prioriza aprendizado empírico, colaboração multifuncional, amplo uso de automação e o uso de técnicas de medição como objetivos de nível de serviço (SLOs). Outras práticas modernas de operações usam métodos semelhantes, mas com nomes diferentes. Por isso, para avaliar a extensão das práticas da forma mais objetiva possível, tivemos o cuidado na pesquisa em usar uma linguagem neutra e descritiva no texto apresentado aos participantes. Também coletamos dados sobre os resultados da engenharia de confiabilidade: na medida em que as equipes conseguem cumprir as metas da área. O modelo preditivo inclui as entradas e os resultados (práticas de SRE e resultados de confiabilidade), bem como outras capacidades de DevOps.



A confiabilidade é essencial

A adoção da SRE está disseminada nas equipes que entrevistamos: a maioria dos participantes usa uma ou mais das práticas que mencionamos. Nessas variadas equipes, os dados revelam uma relação sutil entre a confiabilidade, a entrega de software e os resultados: quando há baixa confiabilidade, o desempenho da entrega de software não é um bom indicador do sucesso organizacional. No entanto, com uma confiabilidade melhor, começamos a ver uma influência positiva da entrega de software sobre o sucesso comercial.

Sem a confiabilidade, o desempenho da entrega de software não é um bom indicador do sucesso organizacional.

O investimento na SRE melhora a confiabilidade, mas apenas depois de um certo limiar de adoção.

Esse fenômeno é consistente com o uso do framework "margem de erro" da SRE: quando um serviço não é confiável, não há vantagem para os usuários em enviar código rapidamente nesse contexto frágil.

Como os engenheiros de confiabilidade do site dizem há tempos, a confiabilidade é o "recurso" mais importante de qualquer produto. Nossa pesquisa sustenta a observação de que cumprir as promessas aos usuários é necessário para que a melhoria na entrega de software gere benefícios para a organização.

Observe a curva J

Quais desafios estão à espreita no caminho para a confiabilidade? Na publicação "Enterprise Roadmap to SRE" da O'Reilly,¹ James Brookbank e Steve McGhee, colaboradores da pesquisa DORA, refletem sobre a experiência deles na implementação da SRE em organizações estabelecidas e recomendam "observar a curva J da mudança". Descrita no relatório State of DevOps de 2018, a "curva J" é um fenômeno em que as transações organizacionais costumam mostrar sucesso no início, seguido por períodos de retorno diminuído

ou até mesmo regressões. Mas aqueles que persistem diante desses desafios alcançam níveis sustentados e renovados de conquistas elevadas.

Nossa pesquisa este ano mostra um padrão de curva J nas equipes de tecnologia que estudamos. Quando as equipes fazem menos práticas de engenharia de confiabilidade (o que sugere que estão no início da jornada de adoção da SRE), essas práticas não são bons indicadores de melhor confiabilidade. No entanto, conforme adotam mais a SRE, as equipes chegam a um ponto de inflexão em que o uso dela passa a ser indicador da confiabilidade e, por sua vez, do desempenho organizacional.



¹ <https://sre.google/resources/practices-and-processes/enterprise-roadmap-to-sre/>

Equipes confiáveis criam serviços confiáveis: a cultura de equipe generativa indica melhor confiabilidade.

As equipes que estão começando a jornada para a SRE precisam estar preparadas para revezes ao longo do caminho. É uma estrada longa, que exige o realinhamento da cultura, dos processos e das ferramentas aos novos princípios norteadores. Mas as equipes podem ter certeza de que, com tempo e a continuação do investimento, há alta probabilidade de sucesso.



As equipes que persistem além do início da adoção da SRE alcançam melhores resultados de confiabilidade.

Investimento em pessoas, processos e ferramentas

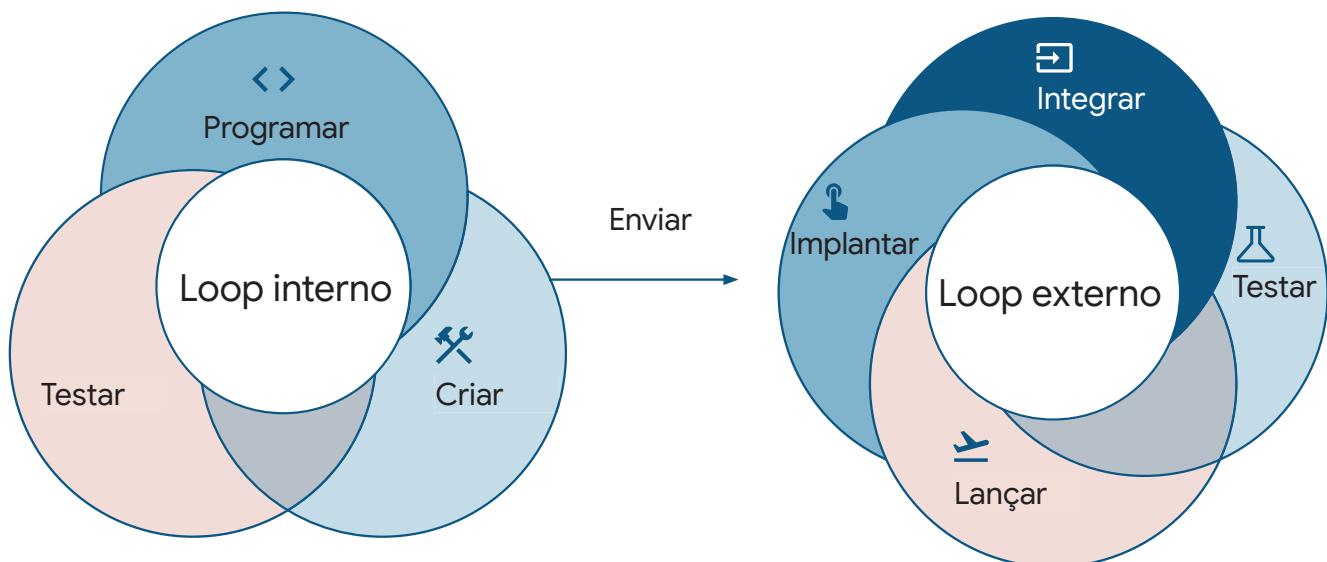
A confiabilidade é um esforço humano, e a SRE é exemplo disso em muitas formas. Um dos princípios da abordagem é que a percepção dos usuários é a verdadeira medição da confiabilidade, em vez de dados internos de monitoramento. Então não é surpresa que a confiabilidade seja conduzida por uma dinâmica positiva de equipe. Constatamos que as equipes com cultura generativa, que exibe confiança e colaboração, são mais propensas a praticar a SRE e a alcançar bons resultados de confiabilidade. As equipes estáveis, com membros consistentes ao longo do tempo, também oferecem maior confiabilidade aos serviços voltados ao usuário. E, como no DevOps de forma geral, incrementar o esforço humano com processos e ferramentas ajuda na engenharia de confiabilidade. Práticas como o uso da computação em nuvem e da integração contínua são bons indicadores de melhores resultados de confiabilidade.

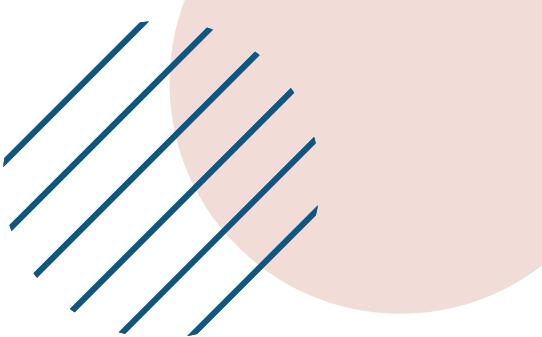


Eric Maxwell

Recursos técnicos de DevOps

Este ano, analisamos várias capacidades técnicas para entender os resultados gerados por cada prática. Consideramos duas fases amplas do desenvolvimento de software: o "loop interno", que inclui tarefas de desenvolvedores como programação, teste e envio ao controle de versões, e o "loop externo", que inclui atividades como mesclagem de código, análise de código automatizada, execução de teste, implantação e lançamento.





Equipes com alto desempenho que cumprem as metas de confiabilidade têm probabilidade 1,4 vez maior de usar CI.

Nossa pesquisa mostra que as empresas que se sobressaem no desenvolvimento dos loops interno e externo conseguem enviar código de maneira mais rápida e mais confiável. As capacidades que mais contribuem para o alto desempenho são controle de versões, integração contínua, entrega contínua e arquitetura com acoplamento flexível.

As equipes com alto desempenho que cumprem as metas de confiabilidade têm:

33%

maior probabilidade de usar o controle de versões

39%

maior probabilidade de praticar a integração contínua

46%

maior probabilidade de praticar a entrega contínua

40%

maior probabilidade de ter sistemas baseados em uma arquitetura com acoplamento flexível

De fato, os participantes que usam as capacidades acima mais do que a média têm **desempenho organizacional 3,8 vezes** melhor que aqueles que não as usam.

Integração contínua

A integração contínua (CI) faz parte do processo de desenvolvimento do loop externo. Ela cria automaticamente um artefato e executa vários testes automatizados para cada confirmação, para verificar se o código está pronto para implantação. O processo retorna feedback rápido e automatizado aos desenvolvedores, que podem então trabalhar com maior confiança. A CI é um fator essencial para levar o código da estação de trabalho do desenvolvedor até a produção. Como nos anos anteriores, constatamos que a CI aumenta o desempenho da entrega. **As equipes com alto desempenho, que cumprem as metas de confiabilidade, têm probabilidade 1,4 vez maior de usar CI que as demais.**

Neste ano, analisamos melhor a outra parte do processo de desenvolvimento do loop externo, a entrega contínua, que vamos discutir em um próximo capítulo. Mas antes vamos ver mais de perto um componente complementar da integração contínua: o desenvolvimento baseado na linha principal.

Desenvolvimento baseado em linha principal

O desenvolvimento baseado em linha principal é a prática de mesclar código continuamente na linha principal e evitar ramificações de recursos de longa duração. Há anos já se constatou que a prática, considerada um complemento da integração contínua, acelera a velocidade da entrega de software.

Devido à mudança demográfica neste ano em termos de tempo de experiência profissional, percebemos que a experiência é importante na implementação do desenvolvimento baseado em linha principal. No ano passado, 40% dos participantes afirmaram ter mais de 16 anos de experiência profissional, categoria que representou apenas 13% este ano. Continuando com o tema "a entrega depende", vemos que as pessoas com menos experiência têm menos resultados positivos com o desenvolvimento baseado em linha principal. Constatamos que elas têm estas características:

- ▼ **Menor** desempenho geral da entrega de software
- ▲ **Maior** quantidade de trabalho não planejado
- ▲ **Maior** propensão a erros
- ▲ **Maior** taxa de falhas nas alterações

As pessoas com mais de 16 anos de experiência que usam o desenvolvimento baseado em linha principal alcançam os benefícios da prática e têm estas características:

- ▲ **Maior** desempenho geral maior da entrega de software
- ▼ **Menor** quantidade de trabalho não planejado
- ▼ **Menor** propensão a erros
- ▼ **Menor** taxa de falhas nas alterações

O provável motivo para isso é que a implementação do desenvolvimento baseado em linha principal exige mais práticas. Quando as equipes não têm regras rígidas contra o abandono de linhas principais quebradas, não usam ramificações de código restritas e fazem reversões automáticas de código que quebram a linha principal, fica mais difícil seguir essa prática.

No entanto, a presença do desenvolvimento baseado na linha principal mostra um impacto positivo sobre o desempenho organizacional em geral.



Frank Xu

Entrega contínua

A entrega contínua (CD) é uma prática de desenvolvimento de software que:

1. Permite que a equipe implante software na produção ou para os usuários finais a qualquer momento
2. Garante que o software esteja pronto para implantação em todo o ciclo de vida, mesmo durante a criação de recursos
3. Cria um loop de feedback rápido para que a equipe verifique se o sistema tem boa qualidade e está pronto para implantação, além de priorizar a correção de problemas que bloqueiam o desenvolvimento

A entrega contínua não implica necessariamente em implantação contínua, a prática em que cada build do software é implantado automaticamente. É preciso apenas que o build esteja pronto para implantação a qualquer momento.

No ano passado, analisamos as capacidades técnicas de DevOps que indicam a prática de CD pela equipe. Constatamos que fatores como arquitetura com acoplamento flexível e testes e integrações contínuos estão entre os indicadores mais fortes. Neste ano, além de analisar os fatores que motivam o uso da CD, analisamos e identificamos os efeitos da CD sobre os resultados de desenvolvimento, tanto sozinha quanto na interação com outras capacidades de DevOps.

A CD melhora o desempenho da entrega de software

Assim como nas constatações do ano passado, o uso da CD indica melhor desempenho na entrega de software, tanto sozinha quanto combinada a outras capacidades de DevOps. As equipes que usam muito a CD têm maior probabilidade de ter alta frequência de implantação de código na produção e menor tempo de lead para mudanças e para restauração de serviço.

As equipes que combinam o controle de versões e a entrega contínua têm probabilidade 2,5 vezes maior de ter alto desempenho da entrega de software do que aquelas que usam apenas uma das práticas.

Além disso, a probabilidade de os participantes terem maior desempenho da entrega de software é 2,5 vezes maior quando a equipe deles também adota o controle de versões.

A CD pode aumentar o trabalho não planejado

Os dados sugerem que a entrega contínua faz os desenvolvedores gastarem mais tempo com retrabalho ou com trabalho não planejado. Uma hipótese para explicar isso é que os desenvolvedores costumam criar aplicativos de maneira iterativa quando os loops de feedback são mais integrados. Por isso, eles podem considerar que algumas mudanças iterativas são trabalho não planejado na mesma parte do software. Esse trabalho pode ser, ao mesmo tempo, não planejado e motivado pelo feedback de uma implantação anterior.

Práticas técnicas e CD

Nossa pesquisa mostrou regularmente que há um amplo conjunto de capacidades técnicas que apoiam a CD. Neste ano, analisamos o que acontece quando algumas dessas capacidades individuais são usadas com a CD. Constatamos que o desenvolvimento baseado em linha principal e a arquitetura com acoplamento flexível, quando juntas com a CD, podem prejudicar o desempenho da equipe. Por exemplo, vimos evidência de que as equipes que adotam as arquiteturas com acoplamento flexível e a CD em conjunto têm probabilidade 43% maior de antecipar mais erros do que a média, em comparação às equipes que só adotaram a CD. Esses erros incluem, por exemplo, falhas temporárias de produtos, vulnerabilidades de segurança e redução significativa de desempenho em servidores. Esses efeitos, que exigem maior investigação, apontam a existência de alguma fricção nas equipes que estão melhorando. A fricção pode estar relacionada à curva J da transformação, em que as equipes têm melhorias no início, mas esmorecem depois que passam dos itens mais fáceis. É preciso comprometimento para alcançar todo o potencial da melhoria. Ao melhorar qualquer capacidade, como a CD, observe os efeitos sobre a equipe e o desempenho em geral.



David Farley

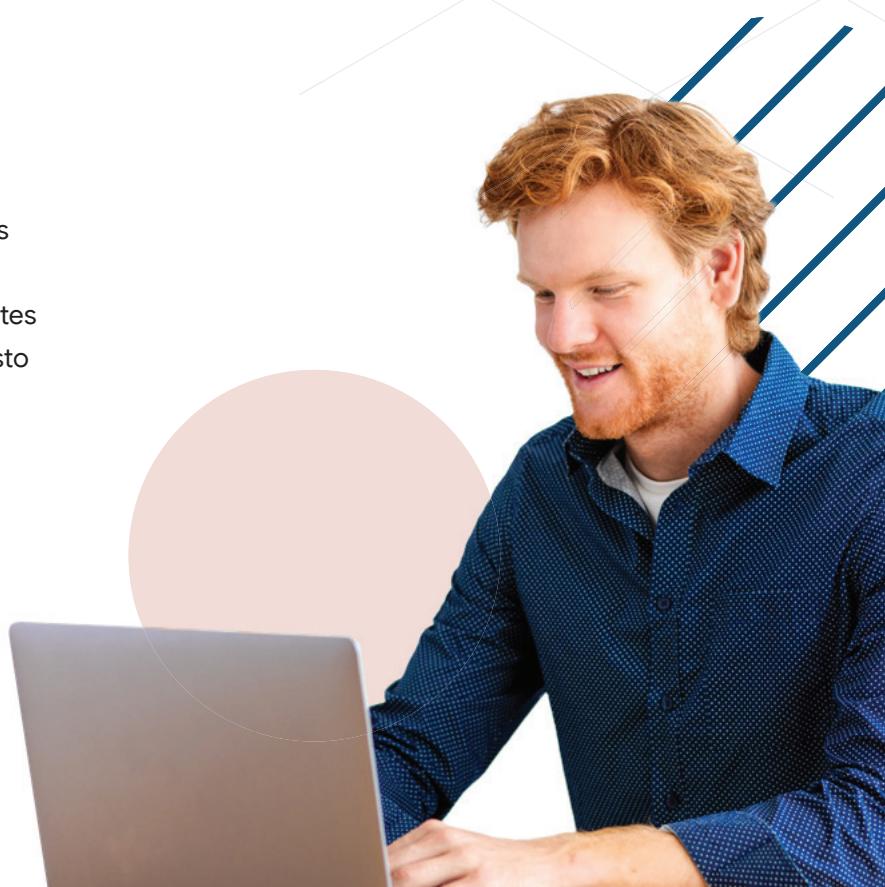
Arquitetura com acoplamento flexível

Os sistemas com acoplamento flexível são importantes para a eficácia de equipes e organizações. Isso não está restrito a sistemas baseados na nuvem ou em microsserviços. Tem a ver com a capacidade de mudar da organização. A facilidade com que uma organização consegue mudar um software de maneira segura e confiante é indicador da qualidade do software.

Com uma arquitetura com acoplamento flexível, as equipes conseguem fazer o seguinte:

- Fazer grandes mudanças no design do sistema sem esperar que outras equipes mudem outros sistemas
- Receber feedback mais rápido por meio de testes independentes e sob demanda com menor custo de coordenação
- Implantar código com tempo de inatividade irrisório

No relatório deste ano, perguntamos aos participantes se o software criado por eles é baseado em uma arquitetura com acoplamento flexível. Os resultados são intrigantes e mostram diversas associações positivas entre presença desse tipo de arquitetura e o desempenho das equipes em várias dimensões.



Os benefícios de uma arquitetura com acoplamento flexível

As equipes que criam software com arquiteturas com acoplamento flexível estão mais bem posicionadas para ter bom desempenho em estabilidade, confiabilidade e capacidade de processamento. Essas equipes também têm maior probabilidade de recomendar a empresa delas a um amigo ou colega.

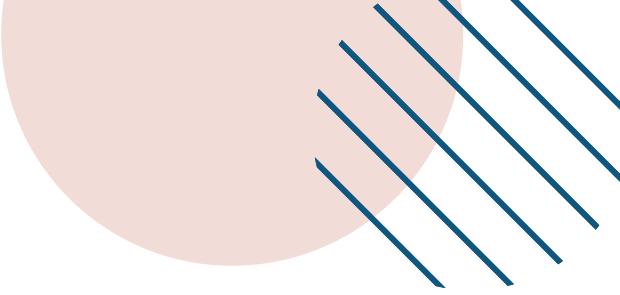
É comum que softwares que usam uma arquitetura de acoplamento flexível sejam criados por equipes que implementam na nuvem e usam arquitetura de microserviços.

As equipes que criam software com arquiteturas com acoplamento flexível estão mais bem posicionadas para ter bom desempenho em estabilidade, confiabilidade e capacidade de processamento.

gerenciando centenas de serviços. Mas o acoplamento flexível é mais do que uma simples medição da contagem de serviços no sistema. Os componentes de uma arquitetura com acoplamento flexível podem ser implantados de maneira independente. Com essa independência, as equipes conseguem desenvolver, testar e implantar serviços sem os altos gastos de coordenação entre as equipes.

No mundo real, o acoplamento flexível não está restrito a um estilo de arquitetura. Trata-se da capacidade de fazer mudanças em uma parte do sistema sem que elas afetem outras partes. Dessa forma, as organizações podem dividir o trabalho para que cada equipe progride sem precisar coordenar com as demais.

Na nossa experiência, as equipes que precisam de testes de integração profundos com outros serviços para confiar no software antes da implantação ainda não alcançaram o acoplamento flexível. Para isso, as equipes precisam melhorar as interfaces e o isolamento entre os sistemas. Uma maneira eficaz de melhorar as interfaces e o isolamento é melhorar a capacidade de teste dos serviços e dos componentes. Quando o design permite testar serviços isoladamente, a interface tem acoplamento flexível por definição.



Também constatamos que as equipes coesas e estáveis que usam arquitetura com acoplamento flexível são mais propensas a usar práticas de desenvolvimento de software que incentivam e apoiam a melhoria contínua. Por exemplo, práticas de SRE como definir metas de confiabilidade para priorizar o trabalho ou revisar regularmente essas metas com base em evidências auxiliam na arquitetura com acoplamento flexível.

Nas organizações com essas arquiteturas, é mais fácil adicionar funcionários, porque as equipes não precisam se coordenar e podem aumentar de forma independente.

Em resumo, o impacto do acoplamento flexível dos serviços de software vai além do aspecto técnico. Ele também afeta os aspectos sociotécnicos do desenvolvimento. O acoplamento é a base da Lei de Conway, a ideia de que os sistemas de design de uma organização espelham a estrutura de comunicação dela. Sistemas com acoplamento mais flexível indicam organizações com acoplamento mais flexível, com desenvolvimento mais distribuível e maleável.



Surpresas nos resultados

A pesquisa deste ano revelou que a arquitetura com acoplamento flexível pode contribuir para o esgotamento nas equipes. Essa é uma surpresa que contraria as constatações dos anos anteriores. Nossa análise mostra que as equipes estáveis com fluxo livre de informações têm menor nível de esgotamento. A cultura generativa de Westrum e a estabilidade da equipe promovem a arquitetura com acoplamento flexível e diminuem o esgotamento, então esta é uma clara contradição. É preciso mais pesquisa para chegar a conclusões definitivas.

Ao mesmo tempo, quando uma organização consolidada de segurança define e controla os requisitos da área, as equipes podem ter maior dificuldade para desacoplar o software delas das demais equipes. O fato demonstra mais ainda o benefício de transferir a segurança para a equipe mais responsável pelo aplicativo (confira também: [Por que a segurança da cadeia de suprimentos é importante](#)). Essa é uma das formas mais sutis de acoplamento nas organizações. Os dados que coletamos dizem respeito à segurança, mas a constatação provavelmente vale também para outras funções centralizadas. Quando transferem as decisões sobre segurança e outras funções centralizadas às equipes, as organizações se aproximam dos benefícios do uso da arquitetura com acoplamento flexível.



Daniella Villalba

Cultura

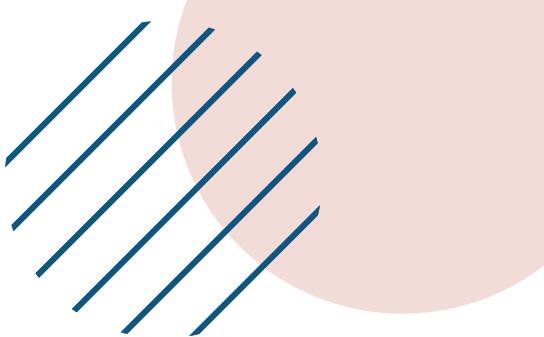
"Foi assim que sempre fizemos as coisas aqui."

Provavelmente as pessoas já repetiram essa frase inúmeras vezes, em diversos setores, para descrever como a organização delas lida com desafios e oportunidades.

Cada organização tem uma cultura própria, e nossa pesquisa mostrou que a cultura é fundamental para o sucesso da organização e o bem-estar dos funcionários.

Também é um aspecto essencial do DevOps, já que esse método trata de ferramentas, práticas **ecolaboração entre pessoas** para desenvolver e entregar software de maneira rápida, confiável e segura. Entender os fatores que afetam a cultura da organização ajuda a liderança a enfrentar os desafios nessa área. Por isso, promover uma cultura saudável deve ser uma prioridade para as organizações. Se forem ignorados, os desafios da cultura talvez impedirão a disseminação das práticas de DevOps.





Neste ano, continuamos usando a tipologia organizacional de Westrum para medir a integridade da cultura das organizações. Além disso, expandimos nossa definição de cultura, medindo rotatividade da equipe, condições flexíveis de trabalho, percepção da adoção organizacional e esgotamento.

Os dados da pesquisa deste ano convergem com as constatações anteriores de que o **desempenho organizacional** é afetado pelo tipo de cultura em uma organização. Em específico, uma **cultura generativa** está associada a melhor desempenho organizacional em relação às organizações com cultura burocrática

ou patológica. Os funcionários das organizações com uma cultura generativa têm maior propensão a pertencer a equipes estáveis, produzir documentação de alta qualidade e gastar o tempo com trabalho importante.

Rotatividade da equipe

Investigamos a rotatividade da equipe e constatamos que há maior proporção de **equipes estáveis** (cuja composição não mudou muito nos últimos 12 meses) nas organizações com alto desempenho. A rotatividade constante pode prejudicar a produtividade e o moral, já que os novos membros da equipe precisam de tempo para a integração. E aqueles que ficam

Cultura organizacional de Westrum

Patológica Voltada ao poder	Burocrática Voltada a regras	Generativa Voltada ao desempenho
Baixa cooperação	Cooperação modesta	Alta cooperação
Os mensageiros são atacados	Os mensageiros são ignorados	Os mensageiros são treinados
Evita responsabilidades	Limita responsabilidades	Compartilha responsabilidades
Conciliações são desencorajadas	Conciliações são toleradas	Conciliações são incentivadas
Falhas levam à busca de um bode expiatório	Falhas levam à justiça	Falhas levam a investigações
Novas ideias são destruídas	Novas ideias causam problemas	Novas ideias são implantadas



precisam se adaptar às mudanças na carga de trabalho e na dinâmica da equipe. Além disso, nossa pesquisa mostrou que as equipes estáveis afirmam produzir mais documentação de qualidade do que aquelas com alta rotatividade. Para equipes que lidam com mudanças constantemente, é difícil manter as práticas que levam à documentação de qualidade.

As organizações com alto desempenho costumam ter condições de trabalho flexíveis.

Condições flexíveis de trabalho

Considerando que várias organizações adotaram condições flexíveis de trabalho desde o surgimento da pandemia de COVID-19, investigamos se a liberdade dos funcionários de escolher entre trabalho remoto,

presencial ou híbrido está ligada ao maior desempenho organizacional. Constatamos que as organizações com maior **flexibilidade para os funcionários** têm melhor desempenho organizacional do que aquelas com condições mais rígidas. É uma evidência de que dar liberdade aos funcionários para modificar as condições de trabalho quando necessário gera benefícios diretos e tangíveis para as organizações.

Esgotamento

O esgotamento é um sentimento de medo, apatia ou cinismo em relação ao trabalho. Quando sofrem esgotamento, as pessoas não estão apenas desmotivadas ou exaustas. Elas também costumam ter menor satisfação no trabalho, o que pode aumentar a rotatividade. O esgotamento está associado a vários problemas de saúde psicológica e física, como maior risco de depressão e ansiedade, doença cardíaca e pensamentos suicidas.¹

No ano passado, medimos o esgotamento no contexto da pandemia de COVID-19 e constatamos que a cultura generativa está associada a um índice menor do fenômeno.

¹ Maslach C, Leiter MP. Understanding the burnout experience: recent research and its implications for psychiatry. World Psychiatry. 2016 Jun;15(2):103-11. doi: 10.1002/wps.20311. PMID: 27265691; PMCID: PMC4911781.

Os modelos flexíveis de trabalho estão associados a menor índice de esgotamento e aumento na probabilidade de o funcionário recomendar a equipe como um bom lugar para trabalhar.

Neste ano, replicamos essa constatação e expandimos nossa compreensão sobre o assunto, mostrando que equipes estáveis e condições flexíveis de trabalho também estão associadas a menos esgotamento. Também medimos o Net Promoter Score (NPS) das equipes, que indica se as pessoas recomendam as equipes a amigos ou colegas. Constatamos que o NPS das equipes está associado à percepção da adoção da liderança. Em um reflexo das constatações sobre o esgotamento, descobrimos que, quando há cultura generativa, equipe estável e condições flexíveis de trabalho, as pessoas estão mais propensas a recomendar as equipes aos demais.

A percepção dos funcionários sobre a organização

Por último, investigamos a percepção da adoção da liderança. Para isso, pedimos que as pessoas estimassem o grau de apoio que as equipes delas devem receber nos próximos 12 meses. Os resultados mostram que a maior percepção da adoção da liderança (por exemplo, mais apoio financeiro, mais alocação de recursos, patrocínios) está associada a organizações com alto desempenho.

Também pedimos que as pessoas estimassem a probabilidade de haver uma violação de segurança ou uma interrupção completa nos próximos 12 meses. Segundo os resultados, as pessoas que trabalham em organizações com bom desempenho estão menos propensas a estimar a ocorrência de erros graves. Elas têm uma perspectiva mais positiva da organização. Também constatamos que as pessoas que trabalham em organizações com alto desempenho de software e de entrega **não costumam** achar que as práticas atuais delas precisam de mudanças para melhorar os resultados comerciais.

Algumas palavras sobre representação

Constatamos que os funcionários de grupos sub-representados afirmam gastar mais tempo com trabalho não planejado, quer estejam em organizações com alto ou baixo desempenho. Também constatamos que esses funcionários têm maior nível de esgotamento do que aqueles que não pertencem a grupos sub-representados. Os líderes de equipe precisam ter cuidado com o risco desse desequilíbrio e alocar o trabalho por igual entre os membros.

Em conjunto, as constatações ressaltam a importância de criar um ambiente saudável e inclusivo para os funcionários, tanto na organização quanto na equipe.

Continuamos a enfatizar a importância da cultura, mas sabemos que mudar ou mesmo melhorar a cultura de uma organização não é uma tarefa fácil. Recomendamos que as organizações tentem primeiro entender as experiências dos funcionários para depois investir recursos para solucionar problemas culturais como parte da transformação do DevOps.



04

Por que a segurança da cadeia de suprimentos é importante

Em novembro de 2020, poucos profissionais de tecnologia suspeitavam que uma crise na segurança da cadeia de suprimentos de software estava se aproximando. A [Open Source Security Foundation](#), sucessora de iniciativas anteriores, havia sido fundada para cuidar da segurança de softwares de código aberto. Havia alguns [pontos importantes](#) para lidar com o problema, mas o tópico não estava nas manchetes dos principais jornais. Um grande ataque, [SolarWinds](#),

mudou tudo isso. Quando alguém consegue invadir milhares de grandes empresas e redes governamentais sem chamar a atenção, usando atualizações de software do tipo cavalo de troia, os tempos mudam rapidamente.

Hoje entendemos que o tópico da segurança na cadeia de suprimento de software é essencial. Talvez não no jantar em família, mas certamente na sala do conselho.



John Speed Meyers Todd Kulesza



Há várias iniciativas, e grande parte do setor de software se comprometeu a reformar as próprias práticas de segurança nessa área e a melhorar a segurança dos elementos comuns em código aberto.

Neste capítulo, vamos falar sobre duas iniciativas: Supply Chain Levels for Software Artifacts ([SLSA](#), se pronuncia "salsa") e NIST Secure Software Development Framework ([SSDF](#)). Ambas oferecem várias medidas de defesa para impedir que invasores adulterem os processos de produção de software e passem pelas defesas da rede com atualizações maliciosas.

Mas qual é o grau de adoção das práticas de segurança da cadeia de suprimentos de software da SLSA e da SSDF? Quais práticas precisam de ajuda para aumentar a adoção e quais já são amplamente usadas? Até hoje, não havia respostas sistemáticas para essas perguntas. Apresentamos algumas respostas iniciais derivadas de entrevistas com centenas de profissionais de software sobre as práticas que eles usam nessa área. Há quatro constatações que se destacam:

01 A adoção já começou: as práticas de segurança da cadeia de suprimentos de software da SLSA e da SSDF já têm uma modesta adoção, mas ainda há espaço para mais.

02 As culturas mais saudáveis saem na frente: a cultura organizacional é um motivador importante para as práticas de segurança de desenvolvimento de software. Com maior confiança, as culturas "sem culpa" estão mais propensas a estabelecer práticas de SLSA e SSDF do que as culturas organizacionais com baixa confiança.

03 Existe um ponto de integração crucial: a adoção dos aspectos técnicos da segurança da cadeia de suprimentos de software parece depender do uso de CI/CD, que geralmente traz a plataforma de integração para muitas dessas práticas.

04 Há benefícios inesperados: além da redução do risco, as práticas de segurança melhores têm outras vantagens, como redução do esgotamento.

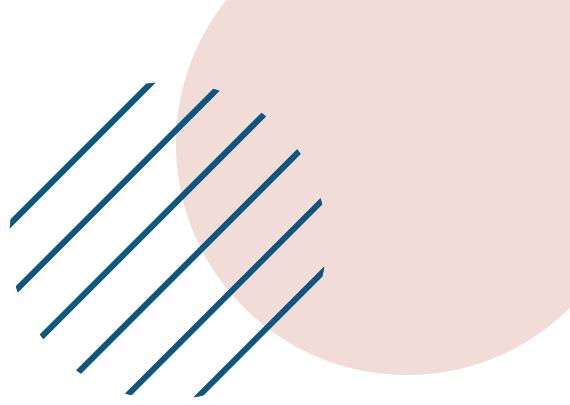
O que as empresas fazem hoje para evitar as vulnerabilidades de segurança

Para entender melhor o que as organizações fazem hoje para identificar e corrigir vulnerabilidades de segurança no software, adicionamos mais de 20 perguntas à pesquisa deste ano. Elas se dividem em duas categorias:

- Perguntas em que o participante precisa concordar ou discordar com uma afirmação. Por exemplo, "Minha organização tem um método eficaz para corrigir ameaças à segurança" ou "Eu tenho acesso às ferramentas necessárias para realizar testes de segurança".
- Perguntas para medir se práticas de segurança estão bem estabelecidas nas organizações. Por exemplo, "Os builds são definidos apenas por scripts e nada mais" ou "As versões de produção são criadas em um sistema centralizado de CI/CD, nunca na estação de trabalho de um desenvolvedor". Usamos a escala "estabelecido/não estabelecido" porque testes iniciais mostraram que os participantes tendem a concordar com algumas perguntas relacionadas à segurança. Mas as perguntas sobre SSDF foram formuladas de forma mais natural, com a escala de resposta concordo/discordo.

O **framework SLSA**, na versão 0.1 atualmente, descreve uma série de práticas de integridade da cadeia de suprimento de software, associadas a "níveis". Os níveis mais altos correspondem a maior garantia de segurança. Perguntamos aos participantes sobre várias das práticas associadas à SLSA. A pergunta era: "As práticas a seguir estão bem estabelecidas para a aplicação ou o serviço principal em que você trabalha?". A tabela 1 mostra as definições das práticas de SLSA usadas na pesquisa.

A **SSDF**, atualmente na versão 1.1, é um conjunto de práticas que ajudam as organizações a distribuir software com menos vulnerabilidades, bem como minimizar o impacto potencial das vulnerabilidades restantes. Em vez dos "níveis" da SLSA, as práticas da SSDF são divididas em quatro categorias: preparação da organização, proteção do software em desenvolvimento, produção de software protegido e resposta eficaz às vulnerabilidades descobertas. Perguntamos aos participantes o quanto eles concordam (ou discordam) de afirmações que escrevem as práticas da SSDF. A tabela 2 mostra um resumo das perguntas.



Prática da SLSA	Definição da pesquisa
CI/CD centralizadas	As versões de produção são criadas em um sistema centralizado de CI/CD, nunca na estação de trabalho de um desenvolvedor
Preservação do histórico	As revisões e os respectivos históricos de alterações são preservados por período indefinido
Script de build	Os builds são totalmente definidos por scripts e nada mais
Isolamento	Os builds são isolados e não podem interferir em builds simultâneos ou posteriores
Arquivos de texto de build	As definições e as configurações de build são definidas em arquivos de texto, armazenados em um sistema de controle de versões.
Metadados de parâmetros	Os metadados de build (por exemplo, dependências, processo, ambiente) referentes a um artefato contêm todos os parâmetros de build
Metadados de dependências	Os metadados de build (por exemplo, dependências, processo, ambiente) referentes a um artefato documentam todas as dependências
Geração de metadados	Os metadados de build (por exemplo, dependências, processo, ambiente) são gerados pelo serviço de build ou por um gerador de metadados que lê os dados daquele serviço
Bloqueio de entradas	No build, as etapas não podem carregar entradas dinamicamente (ou seja, todas as fontes e dependências necessárias são coletadas antes)
Sem edição dos usuários	Os usuários dos serviços de build não podem editar os metadados de build (por exemplo, dependências, processo, ambiente)
Disponibilidade dos metadados	Os metadados de build (por exemplo, dependências, processo, ambiente) estão disponíveis para quem precisa deles (por exemplo, em um banco de dados central) e são entregues em um formato aceito
Análise por duas pessoas	Duas pessoas de confiança precisam analisar cada alteração individual no histórico de uma revisão antes do envio
Assinatura dos metadados	Os metadados de build (por exemplo, dependências, processo, ambiente) que descrevem a produção de um artefato são assinados pelo serviço de build

Tabela 1: perguntas sobre SLSA na pesquisa

Observação: os participantes tinham cinco opções para cada pergunta: não está estabelecido, um pouco estabelecido, moderadamente estabelecido, muito estabelecido e totalmente estabelecido.

Prática da SSDF	Definição da pesquisa
Avaliações de segurança	Uma avaliação de segurança é realizada para todos os recursos principais em que eu trabalho
Análise/teste contínuo do código	Realizamos continuamente análises e testes manuais ou automatizados do código de todas as novas versões com suporte, para identificar ou confirmar a presença de vulnerabilidades que não foram detectadas
Teste de segurança inicial	Eu ou outra equipe realizamos testes de segurança no início do processo de desenvolvimento de software
Lidar com ameaças efetivamente	Minha organização tem um método eficaz para lidar com ameaças à segurança
Integração à equipe de desenvolvimento	Os papéis de segurança são integrados à nossa equipe de desenvolvimento de software
Exigência de documentos	Nossa organização tem processos para identificar e documentar todos os requisitos de segurança do software que desenvolvemos ou adquirimos (incluindo de terceiros e de código aberto)
Revisão regular dos requisitos	Os requisitos de segurança são revistos a intervalos regulares (uma vez por ano ou menos se necessário)
Geração de metadados	Os metadados de build (por exemplo, dependências, processo, ambiente) são gerados pelo serviço de build ou por um gerador de metadados que lê os dados daquele serviço
Integração ao ciclo de desenvolvimento	Na minha empresa, o protocolo de segurança de software faz parte do processo de desenvolvimento
Processo padrão em todos os projetos	Na minha empresa, temos um processo padronizado para lidar com a segurança do software em diferentes projetos
Monitoramento dos relatórios de segurança	Monitoramos permanentemente as informações de fontes públicas sobre possíveis vulnerabilidades no software que usamos e nos componentes de terceiros nesse software
Ter as ferramentas necessárias	Tenho acesso às ferramentas necessárias para executar testes de segurança

Tabela 2: perguntas sobre SSDF na pesquisa

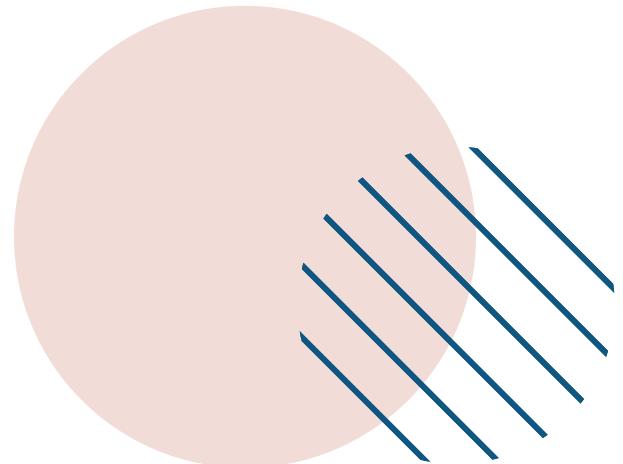
Observação: Os participantes tinham sete opções para cada pergunta: discordo totalmente, discordo, discordo um pouco, não concordo nem discordo, concordo um pouco, concordo e concordo totalmente.

No geral, constatamos uma adoção relativamente ampla das práticas emergentes do setor, mas com muito espaço para um maior estabelecimento. Por exemplo, 66% concordaram com a afirmação "Na minha empresa, o protocolo de segurança de software faz parte do processo de desenvolvimento", mas apenas 18% concordaram totalmente. As figuras 1 e 2 mostram as respostas dos participantes às perguntas sobre segurança.

Constatamos que o uso de sistemas de integração contínua/entrega contínua (CI/CD) para versões de produção é a prática mais bem estabelecida: 63% dos participantes concordam que ela está "muito" ou "completamente" estabelecida. O fato de que a CI/CD está no topo da lista está alinhado a pesquisas anteriores sobre segurança, que constataram que [a maioria das organizações implementa a verificação de segurança no nível do aplicativo como parte do processo de CI/CD](#). Além disso, entrevistas qualitativas sobre segurança, feitas em separado, mostraram que a maioria dos desenvolvedores não consegue executar ferramentas no local durante o desenvolvimento. Da mesma forma, o framework SLSA usa os sistemas de CI como ponto de integração central para a segurança da cadeia de suprimentos. Segundo nossa análise de modelo, que descrevemos na próxima seção, a presença de CI em

uma organização é um bom indicador da maturidade das práticas de segurança. Por isso, acreditamos que, sem essa peça essencial da infraestrutura, é muito difícil para uma organização garantir a execução consistente de verificações, linters e testes nos artefatos de software criados.

Além da CI/CD, outras práticas bem estabelecidas são a preservação indefinida do histórico do código (60%), a definição de build apenas por meio de scripts (58%), o isolamento dos builds (57%) e o armazenamento das definições de build no controle de versões (56%). Por outro lado, as duas práticas menos estabelecidas são a exigência de aprovação de dois ou mais analistas para cada mudança no código (45%) e a assinatura dos metadados de build para evitar ou detectar adulterações (41%).



Além das perguntas sobre as práticas estabelecidas, também perguntamos se os participantes concordam ou discordam com algumas afirmações sobre a segurança na organização deles. A declaração com maior nível de concordância (81%) foi: "Monitoramos permanentemente as informações de fontes públicas sobre possíveis vulnerabilidades no software que usamos e nos componentes de terceiros nesse software". Por outro lado, a declaração com a menor proporção de concordância (56%) foi sobre os impactos negativos das práticas de segurança no desenvolvimento de software. A afirmação era: "Os processos de segurança de software que existem na minha empresa tornam mais lento o processo de desenvolvimento dos aplicativos em que trabalho".

É ótimo que a afirmação tenha o menor nível de concordância, mas o fato de que a maioria dos participantes afirma que os processos de segurança atuais tornam o desenvolvimento lento sugere que há muito espaço para melhoria nas ferramentas e abordagens da segurança. Nossa análise de modelo apoia justifica essa interpretação, mostrando um efeito misto (porém pequeno) sobre o desempenho da entrega de software.



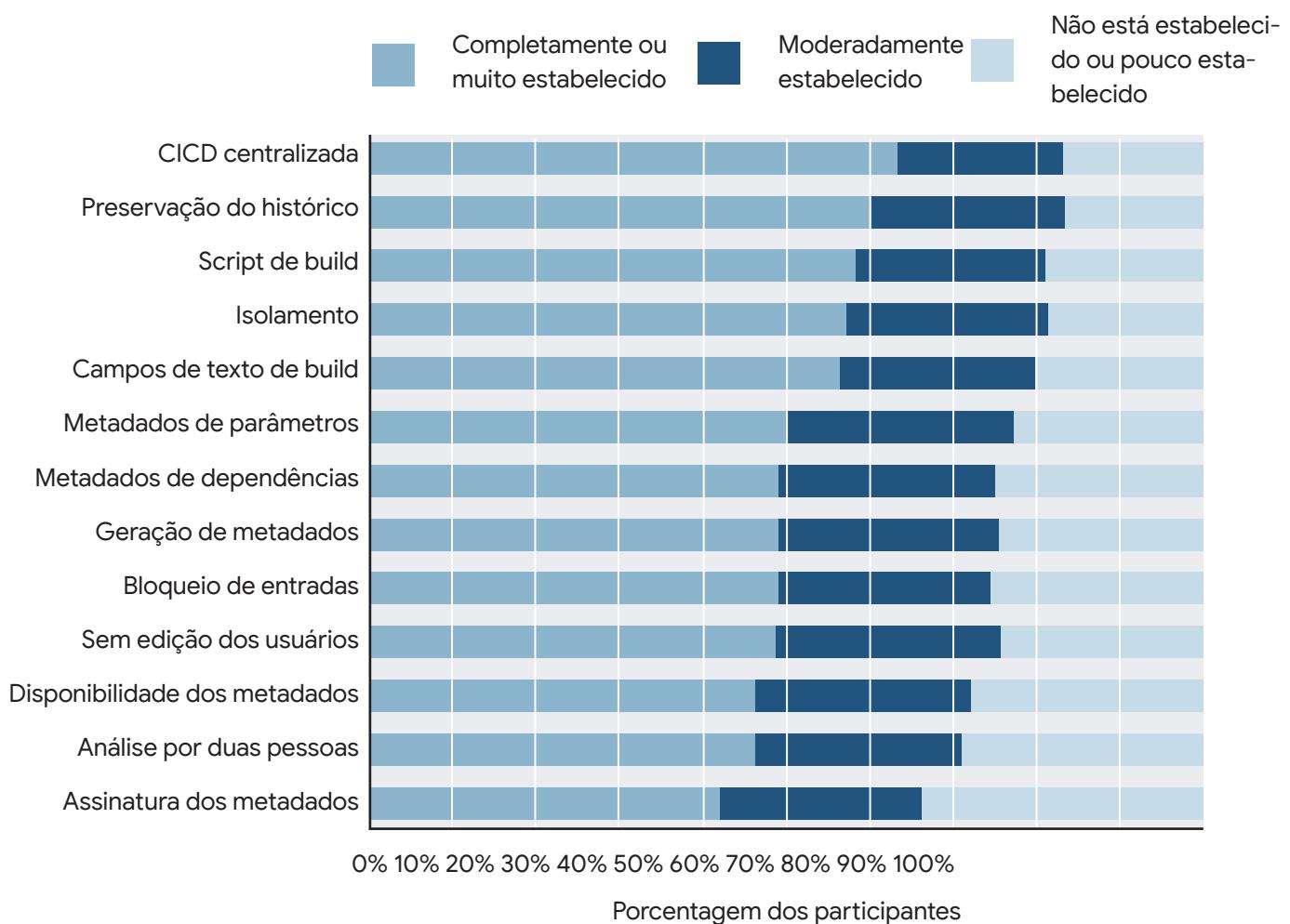


Figura 1. Estabelecimento das práticas da SLSA

Respostas na pesquisa sobre o estabelecimento das práticas da SLSA. A maioria dos participantes indicou certo grau de estabelecimento de todas as práticas, mas poucos afirmaram que elas já estão "completamente" estabelecidas.

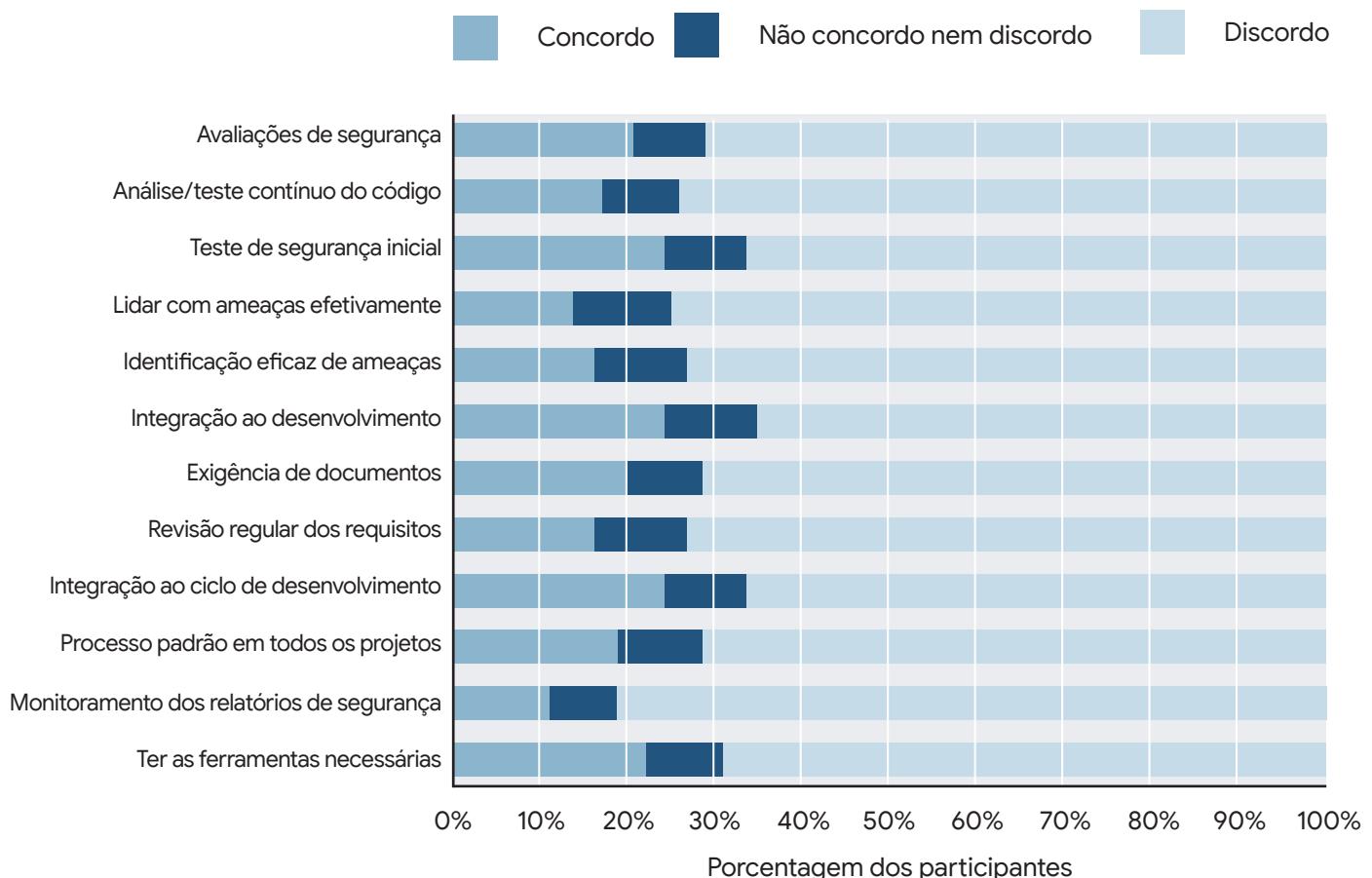


Figura 2. Estabelecimento das práticas da SSDF

Respostas na pesquisa sobre o estabelecimento das práticas da SSDF. Assim como a SLSA, a maioria dos participantes concordou que as organizações deles seguem todas as práticas.

O que ajuda as empresas a seguir boas práticas de segurança?

A segurança do aplicativo é apenas um aspecto do desenvolvimento de software, uma das várias demandas sobre o tempo e a atenção dos desenvolvedores. Tratar a segurança com alta fricção pode gerar frustração nos desenvolvedores e ser ineficaz, já que as pessoas evitam os pontos de fricção. Por exemplo, segundo um conjunto de entrevistas com engenheiros de software profissionais, o contato deles com as equipes de segurança era limitado ao início ou ao fim de um projeto e havia dificuldade na interação entre eles. Nas palavras de um participante: "Temos uma equipe de segurança do aplicativo, mas eles nunca analisaram meu código... Eu geralmente os evito, como a maioria dos engenheiros".

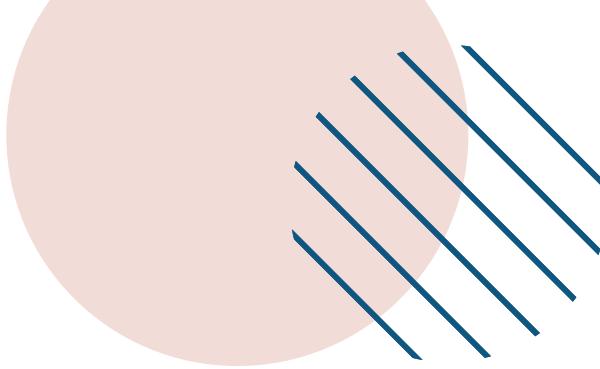
Uma maneira de melhorar a segurança de software é reduzir as barreiras que impedem o seguimento das práticas. Os desenvolvedores com quem conversamos querem fazer a coisa certa. Eles até expressaram frustração com a prioridade que o lançamento de recursos tem sobre a correção de possíveis problemas de segurança. Por exemplo, um participante de outra pesquisa sobre segurança disse que o maior desafio era: "Tornar isso uma prioridade. Não é sexy, não vende produtos [e] não é um problema até que se torne um problema".

¹ Um fator interessante é que essas mesmas pessoas não concordaram mais que as outras com as perguntas da NIST SSDF. A SLSA e a SSDF tratam de aspectos diferentes da segurança de desenvolvimento de aplicativo, mas esperávamos que houvesse sobreposição entre esses conjuntos de perguntas. Como mencionamos, é possível que haja um viés para as respostas "concordo" na escala da SSDF, o que explicaria a diferença.

Os dados da nossa pesquisa sugerem que há vários fatores que tornam mais fácil para os engenheiros fazer o certo pela segurança.

O principal fator que constatamos não era técnico, mas sim cultural: as organizações próximas do grupo de cultura "generativa" de Westrum são mais propensas a estabelecer práticas de segurança conforme a definição do framework SLSA¹. Alguns aspectos das culturas generativas são alta cooperação, compartilhamento de riscos e responsabilidades e aprender com os erros. Nossa hipótese é que essas características geram várias práticas saudáveis de segurança. Por exemplo, incentivar os engenheiros de software a serem mais proativos sobre a segurança da cadeia de suprimentos, recompensar pessoas de qualquer papel por atividades de segurança, ou reduzir o risco que as pessoas veem em denunciar possíveis problemas de segurança.

Em termos de tecnologia, três dos mais importantes fatores da segurança estão relacionados à infraestrutura. Isso faz sentido: se a infraestrutura facilita a realização de tarefas como verificação de vulnerabilidades ou análises manuais de código, é mais provável que os engenheiros as usem. Constatamos especificamente que sistemas de controle de código-fonte, integração contínua e



entrega contínua estão associados a práticas da SLSA mais firmes. Uma parte importante disso é o momento em que os problemas de segurança chamam a atenção dos desenvolvedores, que é durante a CI segundo outra pesquisa. Geralmente a CI vem logo antes das revisões de código e inclui a execução de verificações de vulnerabilidade e de outras ferramentas, o que garante a aplicação dos mesmos requisitos de segurança a todas as confirmações. Quando não há um sistema de build centralizado, a verificação fica muito mais difícil. Por sua vez, a ausência do controle de código-fonte torna difícil ter um sistema de build centralizado.

Mas a verificação de segurança na CI/CD pode não ser cedo o suficiente para os engenheiros de software. Em entrevistas sobre segurança com desenvolvedores de aplicativo, eles disseram que a verificação de segurança nas estações de trabalho deles pouparia tempo e esforço. Duas situações muito citadas: 1) saber de antemão se uma dependência com vulnerabilidades conhecidas está sendo usada, para reavaliar se é necessário usá-la antes do build; e 2) evitar o longo tempo de espera da CI, às vezes várias horas, apenas para confirmar se as mudanças atuais corrigiram um problema de segurança. Em ambos os casos, os engenheiros de software disseram que uma "interrupção" de CI é necessária, mas poder executar as mesmas

ferramentas de segurança no computador local tornaria o trabalho mais rápido e eficiente.

Os fatores culturais e tecnológicos que mencionamos acima são os principais para a segurança, mas não os únicos. Alguns outros fatores importantes:

- Flexibilidade nas condições de trabalho (por exemplo, a organização aceita o trabalho de casa?)
- Uso da nuvem, seja pública ou privada
- Trabalho com um aplicativo ou serviço nativo da nuvem
- Sentir que a empresa valoriza a equipe e investe nela
- Baixa rotatividade na equipe
- Tamanho da organização (as maiores têm pontuações de segurança mais altas)

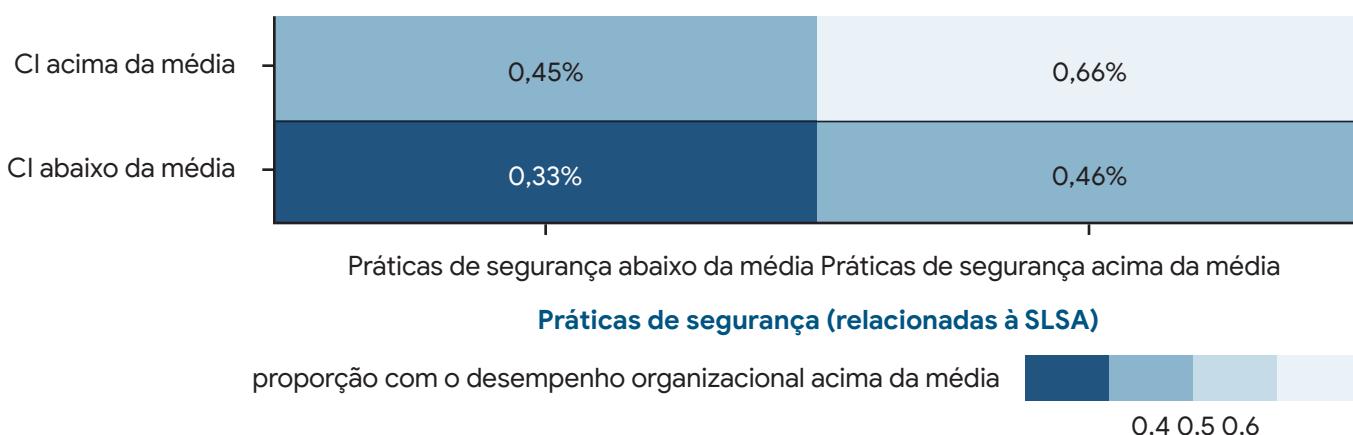
Mas esses fatores parecem estar correlacionados à cultura generativa de Westrum (por exemplo, condições flexíveis de trabalho, sentimento de valorização pela organização ou baixa rotatividade da equipe) ou com o uso da CI/CD (por exemplo, trabalhar em aplicativos nativos da nuvem ou em uma grande organização). Por esses dados, acreditamos que a cultura organizacional e os processos modernos de desenvolvimento (como a integração contínua) são os principais fatores para a segurança de desenvolvimento de aplicativo. As organizações que querem aumentar a segurança devem começar por esses fatores.

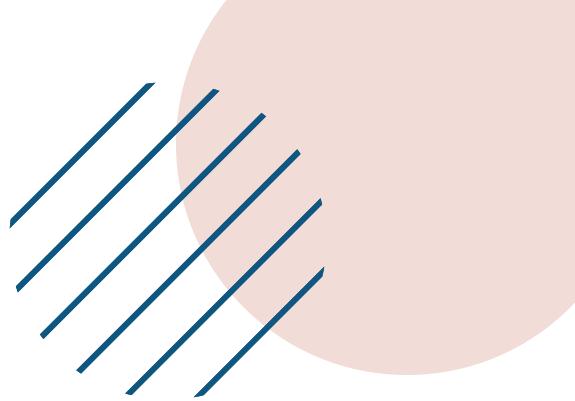
Quais são os resultados das boas práticas de segurança?

Quais benefícios as organizações alcançam quando melhoram as práticas de segurança do desenvolvimento de software? Os dados da nossa pesquisa mostram que os participantes esperam **menor chance de violações de segurança, interrupções de serviço e degradação de desempenho quando as empresas estabelecem mais práticas de segurança da cadeia de suprimentos**. Da mesma forma, realizamos outra pesquisa na primeira metade de 2022 e constatamos que o uso de ferramentas como verificações de vulnerabilidade durante a CI aumenta a probabilidade de identificar vulnerabilidades nas dependências de software. Os participantes que usam essas ferramentas têm o dobro de probabilidade de identificar uma vulnerabilidade de segurança no próprio código ou nas dependências. Resumindo: parece que as práticas da SLSA e da SSDF funcionam bem. Não afirmamos que elas

eliminam as ameaças à segurança, mas nossas evidências sugerem que elas reduzem o risco à segurança das organizações.

As práticas de segurança também têm impacto positivo sobre os resultados de desempenho, mas com uma surpresa: a CI tem papel essencial. Quando a CI não é implementada, as práticas de segurança não mostram efeito algum sobre o desempenho da entrega de software. Mas, quando a CI é implementada, as práticas têm um forte efeito positivo sobre o desempenho. Isso significa que a CI é necessária para que as práticas de segurança tenham efeito positivo sobre o desempenho da entrega de software. Além disso, as práticas de segurança geralmente têm efeito positivo sobre o desempenho organizacional, que é amplificado quando a CI está bem estabelecida. O gráfico abaixo mostra esse efeito.





Além da redução na percepção dos riscos de segurança, os participantes também relataram **menos esgotamento** entre os membros da equipe e maior disposição de recomendar a organização como **bom lugar para trabalhar**. Essas constatações mostram como a segurança é uma tarefa a mais para os engenheiros de software. As ferramentas e os processos que ajudam a incorporar práticas seguras no fluxo de trabalho atual de desenvolvimento (em vez de trabalho não planejado ou pânico quando uma ameaça é descoberta) são uma forma de reduzir os riscos e aumentar a satisfação dos desenvolvedores.

Em conjunto, as evidências sugerem que **equipes saudáveis e com bom desempenho tendem também a ter boas práticas de segurança**, amplamente estabelecidas. Mas, como já notamos, ainda há espaço para melhoria. Seguir frameworks como a SLSA ou a SSDF, por si só, não melhora as métricas de cultura e de desempenho que medimos. Mas está claro que a segurança não precisa vir em troca de outras prioridades no desenvolvimento.

As ferramentas e os processos que ajudam a incorporar práticas seguras no fluxo de trabalho atual de desenvolvimento (em vez de trabalho não planejado ou pânico quando uma ameaça é descoberta) são uma forma de reduzir os riscos e aumentar a satisfação dos desenvolvedores.

05

Surpresas



Derek DeBellis

Embora o tema de cada relatório seja as respostas da pesquisa naquele ano, procuramos entender as constatações no contexto de todo o catálogo de relatórios State of DevOps e pesquisas adjacentes (por exemplo, sobre esgotamento e cultura). Testar a confiabilidade desses efeitos por meio da replicação é um aspecto essencial do programa de pesquisa. Dessa forma, podemos ajustar nossas teorias de acordo com os dados e entender a evolução ou o surgimento de tendências.

Este ano, encontramos algumas surpresas. Há muitos motivos possíveis para isso. Na amostra deste ano, há mais pessoas no início da carreira do que em relatórios anteriores. Uma interpretação é que há mais pessoas diretamente responsáveis por implementar as práticas e as capacidades técnicas, em vez de pessoas

responsáveis por coordenar ou orientar a implantação das práticas. Outra possibilidade é que houve uma mudança no setor ou no mundo. O que funcionava ontem não necessariamente vai funcionar amanhã. Por exemplo, forças macroeconômicas e mais um ano à sombra da pandemia de COVID-19 podem ter mudado a física do DevOps. Por último, mudanças sutis no que incluímos no nosso modelo podem ter alterado as relações entre as variáveis.¹



¹ "Book of Why", de Judea Pearl, e "Statistical Rethinking", de Robert McElreath, apresentam exemplos incríveis que mostram como a inclusão ou a retirada de fatores dos modelos estatísticos afeta o resultado.

Uma constatação inesperada ou fora da hipótese traz dificuldades para os pesquisadores na hora de escrever o relatório. Existe o risco de que a constatação seja espúria ou, pelo menos, não estabeleça ainda (ou mesmo contrarie) a evidência empírica de vários estudos. Por isso, o mais responsável é realizar outras pesquisas para tentar replicar os resultados e entender a causa deles. Ao destacar as surpresas, os pesquisadores também correm o risco de menosprezar a quantidade de efeitos que surgiram com frequência ao longo dos anos de pesquisa. Nós fazemos uma análise estatística de mais de cem caminhos para cada relatório State of DevOps. Assim corremos o risco de gerar resultados espúrios por mero acaso. Tentamos compensar isso com a replicação todos os anos.

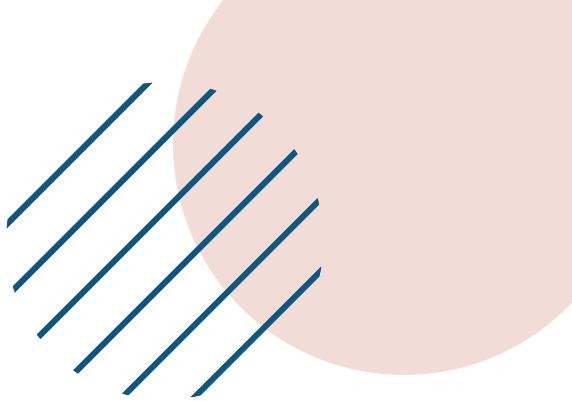
Por outro lado, quando não publicamos a constatação, há o risco de criar um efeito de "armário de arquivo" em que o que é esperado ou palatável se torna conhecido, e o que é inesperado ou difícil de aguentar fica escondido. Nós buscamos um equilíbrio: não queremos sensacionalizar constatações recentes, mas achamos essencial divulgá-las. Aqui estão os fatos que mais nos surpreenderam e nossa interpretação deles:

01 Observamos com frequência que o desenvolvimento baseado em linha principal tem impacto positivo sobre o desempenho da entrega de software. De fato, observamos isso todos os anos do estudo desde 2014. As capacidades de desenvolvimento baseado em linha principal tiveram comportamento incomum neste ano. Por exemplo, elas tiveram um impacto negativo sobre o desempenho da entrega de software. Nos relatórios anteriores, ocorreu o contrário. Como esta constatação é tão inusitada, queremos ver se ela será replicada em pesquisas posteriores e se a comunidade tem alguma explicação.

02 Descobrimos que o desempenho da entrega de software só aumenta o desempenho organizacional quando o desempenho operacional também é alto. Mas muitos participantes não apresentaram alto desempenho operacional. Isso contraria nossas pesquisas anteriores, em que a conexão entre o desempenho da entrega de software e o organizacional estava muito mais clara.

² Kerr, N. L. (1998). HARKing: Hypothesizing after the results are known. *Personality and social psychology review*, 2(3), 196-217.

³ Rosenthal, R. (1979). The file drawer problem and tolerance for null results. *Psychological bulletin*, 86(3), 638.



03 As práticas de documentação prejudicam o desempenho da entrega de software. Isso contraria os relatórios anteriores. Uma hipótese é que a documentação está ficando cada vez mais automatizada, especialmente em equipes com alto desempenho. Até coletarmos mais dados, temos poucas evidências para apoiar ou refutar esse argumento.

04 Algumas capacidades técnicas (como desenvolvimento baseado em linha principal, arquitetura com acoplamento flexível, CI e CD) parecem ser indicadores de esgotamento. Como já mencionamos, muitos dos participantes desta amostra estão em uma fase anterior da carreira em relação aos participantes dos anos anteriores. Por isso, é possível que os participantes estejam implementando o recurso e não sejam os responsáveis por criar ou supervisionar a iniciativa. O processo de implantação pode ser mais desafiador que a supervisão. Queremos pesquisar mais para entender melhor a constatação.

05 Práticas de engenharia de confiabilidade prejudicam o desempenho da entrega de software. Uma explicação é que não há relação causal entre os dois fatores. Neste ano observamos, em uma nova análise por grupos (consulte "Sua equipe está à altura?"), que um subconjunto de grupos se dedica mais à confiabilidade e ignora o desempenho da entrega de software. Acreditamos que os fatores não estão ligados, ou seja, é possível fazer um sem fazer o outro. Mas, para que o desempenho da entrega de software tenha impacto no desempenho organizacional, é necessário haver confiabilidade.

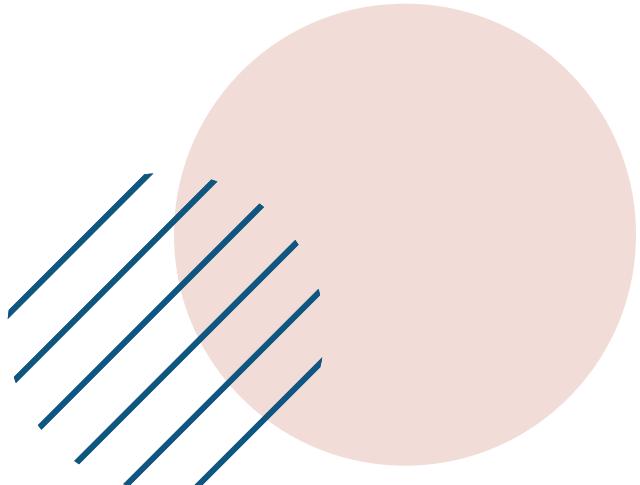
06 Adicionamos práticas da SLSA para entender se as equipes estão adotando essas abordagens para proteger a cadeia de suprimentos de software. Havia a expectativa de uma associação entre a implementação de práticas de segurança e o desempenho, como uso de capacidades técnicas, melhor desempenho da entrega de software e melhor desempenho organizacional. Mas ficamos surpresos em ver que as práticas de segurança são o mecanismo com o qual os recursos técnicos afetam ambos os desempenhos.



A inclusão das práticas da SLSA parece ser responsável pelo efeito da integração contínua, do controle de versões e da entrega contínua no desempenho da entrega de software e no organizacional. Em outras palavras, parece haver uma cadeia causal nos dados: várias capacidades técnicas têm impacto positivo nas práticas da SLSA e, por meio disso, passam a ter impacto positivo também sobre essas formas de desempenho. Usamos análises de mediação para detectar esse resultado.^{4,5} Isso nos incentiva a analisar se a medição das práticas da SLSA está identificando outros atributos da equipe (como o desempenho geral) e como as práticas de segurança melhoraram o desempenho da entrega de software e o organizacional.

Queremos estudar esses efeitos novamente no próximo ano, para ver se conseguimos reproduzir e explicar os novos padrões ou se devemos ignorá-los como pontos fora da curva (que também devemos explicar). Como sempre, apreciamos o feedback da comunidade.

Acesse <http://dora.community> (em inglês) e participe da Comunidade DORA para continuar a conversa sobre estas surpresas e outras constatações do relatório deste ano.



⁴ Jung, Sun Jae. "Introduction to Mediation Analysis and Examples of Its Application to Real-world Data." Journal of preventive medicine and public health = Yebang Uihakhoe chi vol. 54,3 (2021): 166-172. doi:10.3961/jpmph.21.069

⁵ Carrión, Gabriel Cepeda, Christian Nitzl e José L. Roldán. "Mediation analyses in partial least squares structural equation modeling: Guidelines and empirical examples." Partial least squares path modeling. Springer, Cham, 2017. 173-195.

06

Informações demográficas e firmográficas



Derek DeBellis

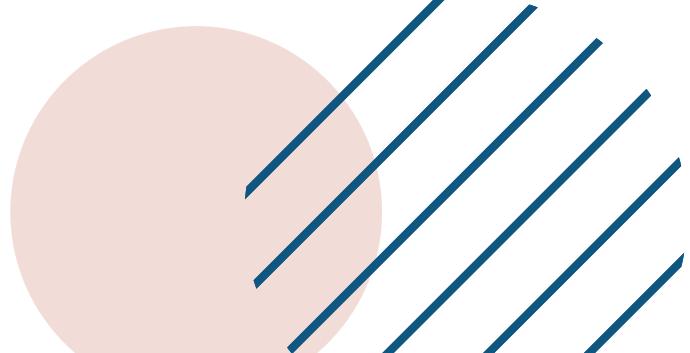
Agradecemos sua colaboração com nossa pesquisa e com o setor!

Quem respondeu à pesquisa?

Com oito anos de pesquisa e respostas de mais de 33.000 profissionais do setor, o relatório State of DevOps mostra as práticas de desenvolvimento de software e de DevOps que mais dão sucesso às organizações. Este ano, mais de 1.350 profissionais que trabalham em vários setores ao redor do mundo compartilharam experiências para ajudar a entender melhor os fatores que geram maior desempenho. Agradecemos sua colaboração com nossa pesquisa e com o setor! Em resumo, a representação entre informações demográficas e firmográficas permaneceu incrivelmente consistente.

Assim como nos anos anteriores, coletamos informações demográficas de cada participante da pesquisa. As categorias incluem gênero, deficiência e grupos sub-representados.

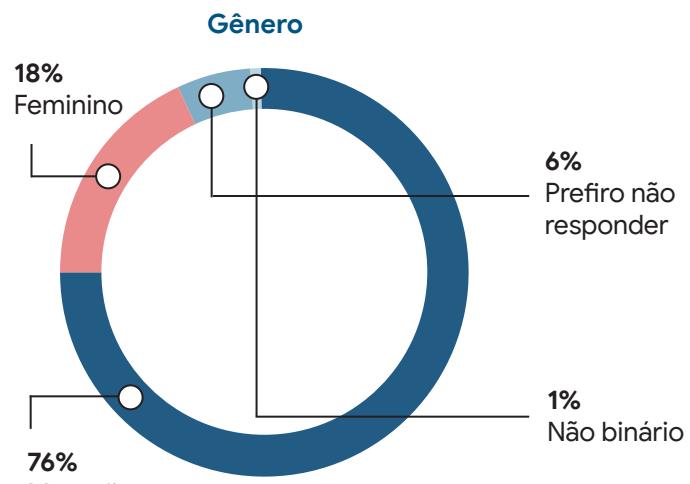
Neste ano, observamos uma representação consistente com os relatórios anteriores em categorias firmográficas, incluindo tamanho da empresa, setor e região. Mais de 60% dos entrevistados trabalham como engenheiros ou gerentes, e um terço trabalha no setor de tecnologia. Além disso, vemos a representação do setor de serviços financeiros, varejo e empresas industriais/de manufatura.



Informações demográficas

Gênero

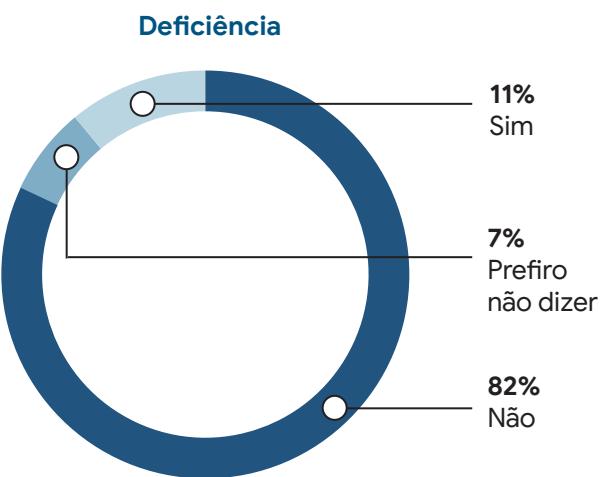
A amostra deste ano tem uma maior proporção de mulheres do que 2021 (18% contra 12%). A proporção de homens (76%) é menor que em 2021 (83%). Os participantes indicaram que as mulheres representam 25% das equipes deles, idêntico a 2021 (25%).



Porcentagem de mulheres: **mediana de 25%**

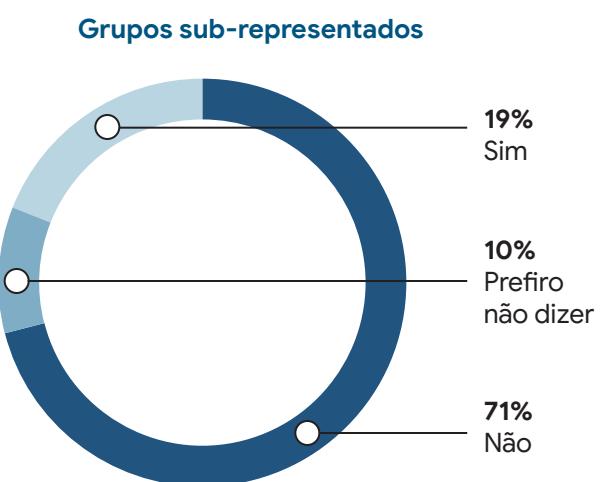
Deficiência

Identificamos deficiências em seis dimensões que seguem a orientação do [Washington Group Short Set](#). Este é o quarto ano que perguntamos sobre deficiências. A porcentagem de pessoas com deficiência é igual à do relatório de 2021, 11%.



Grupos sub-representados

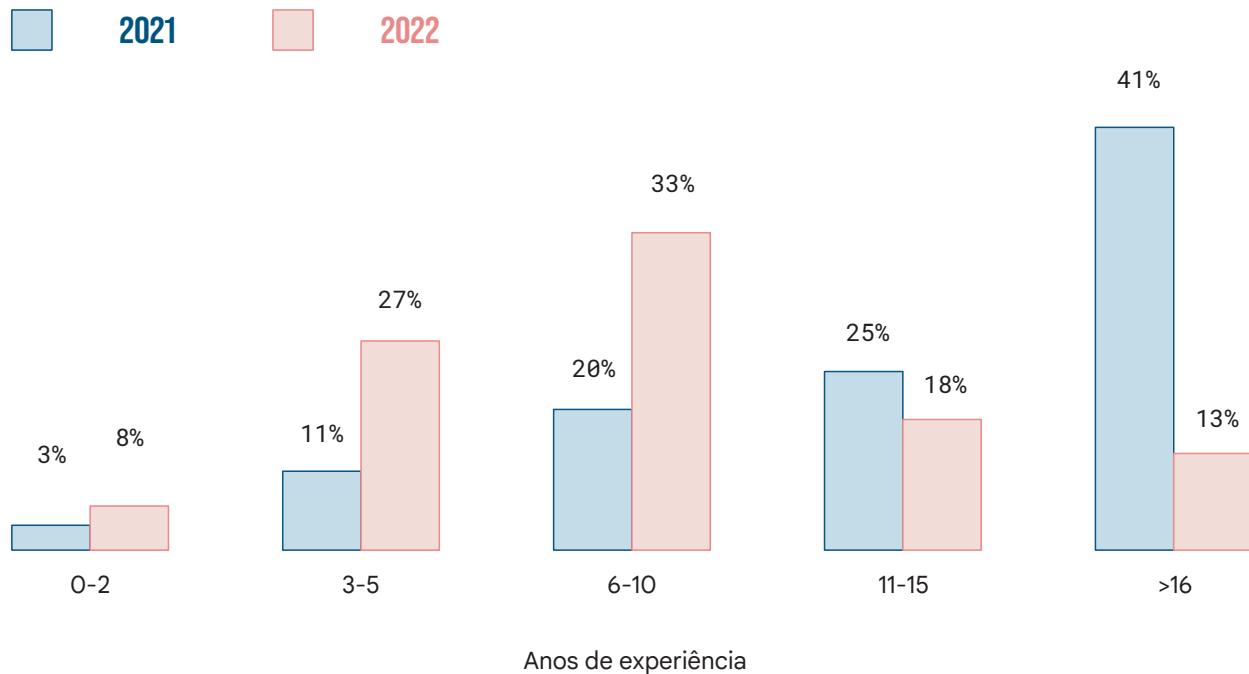
Identificar-se como membro de um grupo sub-representado pode se referir a raça, gênero ou outra característica. Este é o quinto ano que perguntamos sobre a falta de representatividade. A porcentagem de pessoas que se identificam como sub-representadas aumentou um pouco, de 17% em 2021 para 19% em 2022.

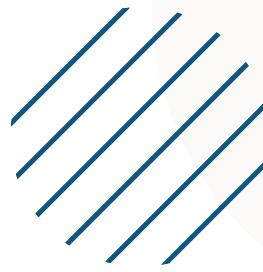


Anos de experiência

Neste ano, mais participantes têm cinco anos ou menos de experiência (35%) do que em 2021 (14%). Portanto, não é surpresa que a proporção de participantes com mais de 16 anos de experiência (13%) seja muito menor que em 2021 (41%).

A mudança pode explicar alguns padrões nos dados, e achamos importante ter isso em mente ao interpretar os resultados, especialmente em comparação ao ano passado.

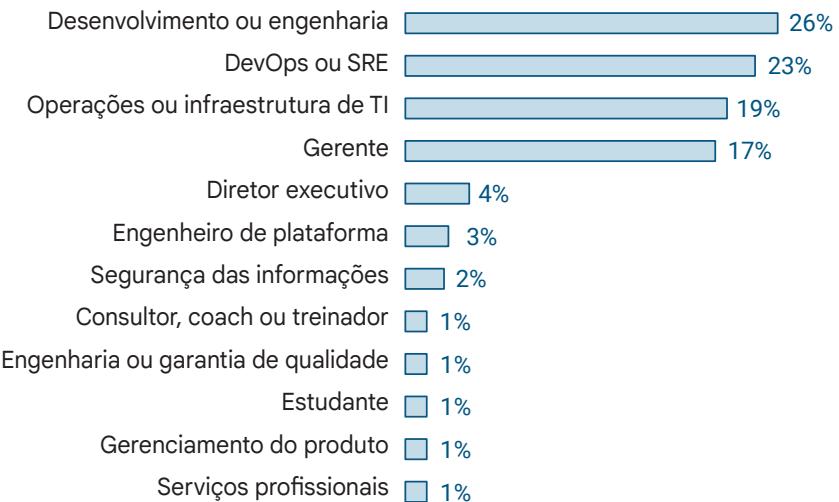




Firmografia

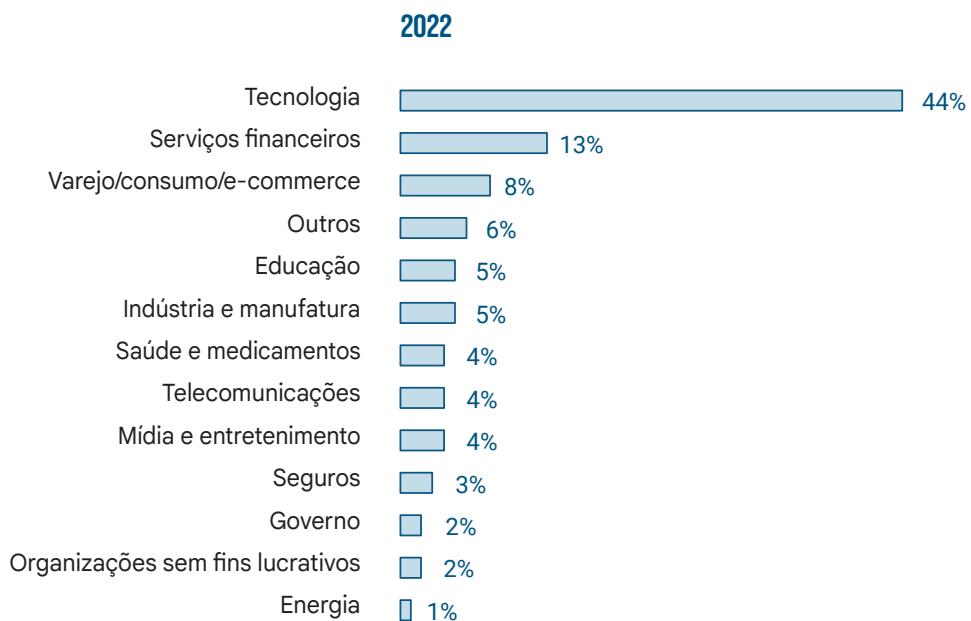
Papel

Dos participantes, 85% são pessoas que trabalham em equipes de desenvolvimento ou engenharia (26%), em equipes de DevOps ou de SRE (23%), em equipes de operações ou infraestrutura de TI (19%) ou são gerentes (17%). A proporção de participantes que trabalham em equipes de operações ou infraestrutura de TI (19%) mais que dobrou desde o ano passado (9%). Os diretores executivos (9% em 2021 para 4%) e os serviços profissionais (4% em 2021 para 1%) tiveram as maiores reduções em relação ao ano passado.



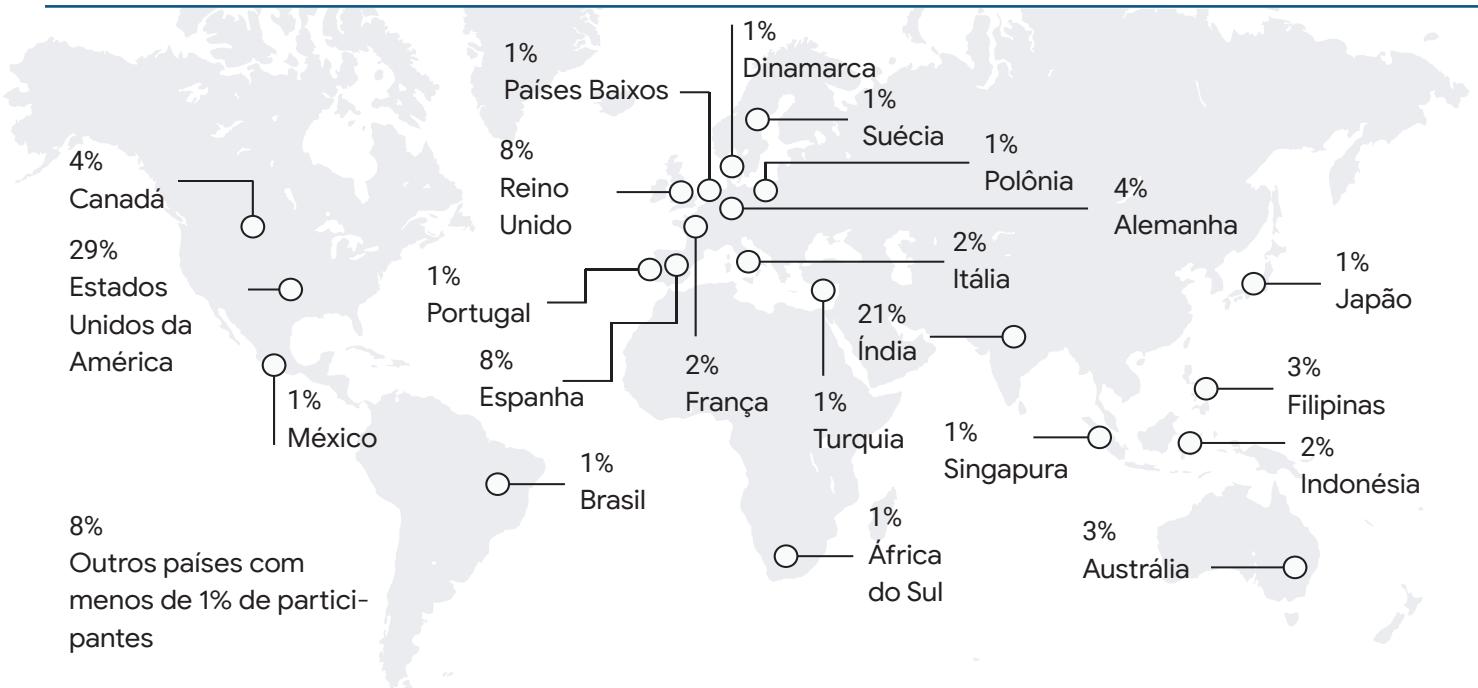
Setor

Como nos relatórios anteriores de State of DevOps, a maioria dos participantes trabalha no setor de tecnologia, seguido por serviços financeiros, "outros" e varejo.



Região

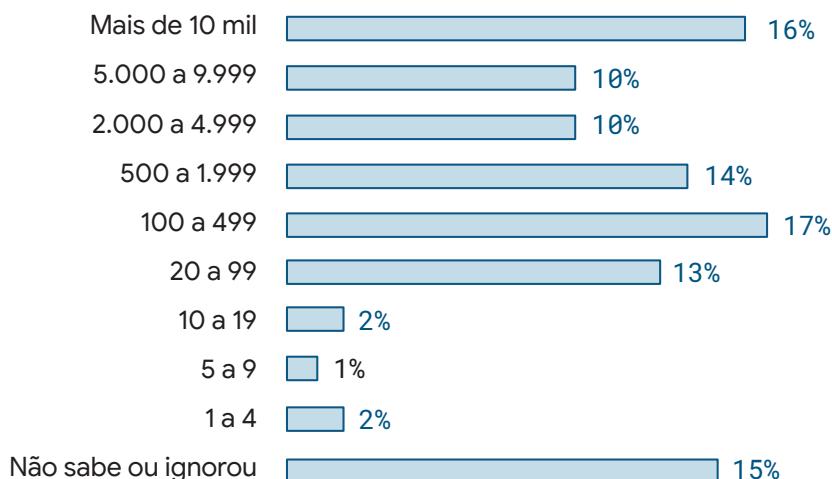
Neste ano, pedimos que os participantes selecionem o país de onde são em vez da região. A região, geralmente representada por um continente, pareceu algo muito grosseiro para entender a composição dos participantes. Recebemos respostas de pessoas em mais de 70 países, e 89% dos participantes vêm de 22 países.

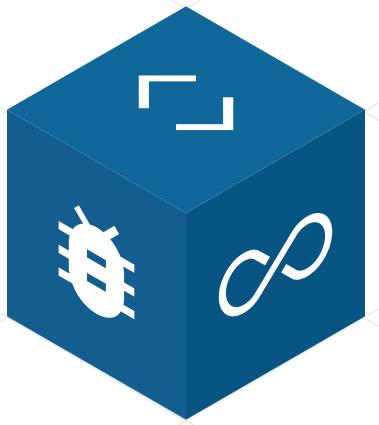


Número de funcionários

Assim como nas pesquisas State of DevOps anteriores, os participantes vêm de organizações de vários tamanhos. 22% dos entrevistados são de empresas com mais de 10.000 funcionários, e 7% são de empresas com 5.000 a 9.999 funcionários. Outros 15 dos participantes

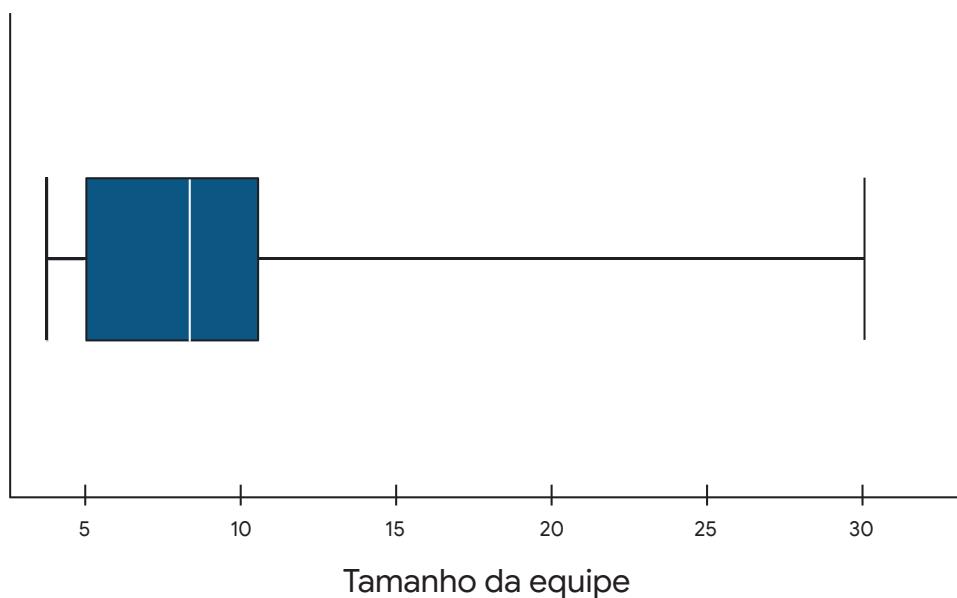
estão com organizações com 2.000 a 4.999 funcionários. Também vimos uma boa representação (13%) de pessoas em organizações com 500 a 1.999 funcionários, 15% com 100 a 499 funcionários e 15% com 20 a 99 funcionários. Neste ano, os participantes tiveram a opção de selecionar "Não sei" sobre o tamanho da organização, e 15% selecionaram isso ou ignoraram a pergunta.





Tamanho da equipe

Neste pedimos, perguntamos aos participantes o número aproximado de pessoas nas equipes deles. 25% dos participantes trabalham em equipes com cinco pessoas ou menos. 50% trabalham em equipes com oito pessoas ou menos. 75% trabalham em equipes com 12 pessoas ou menos.

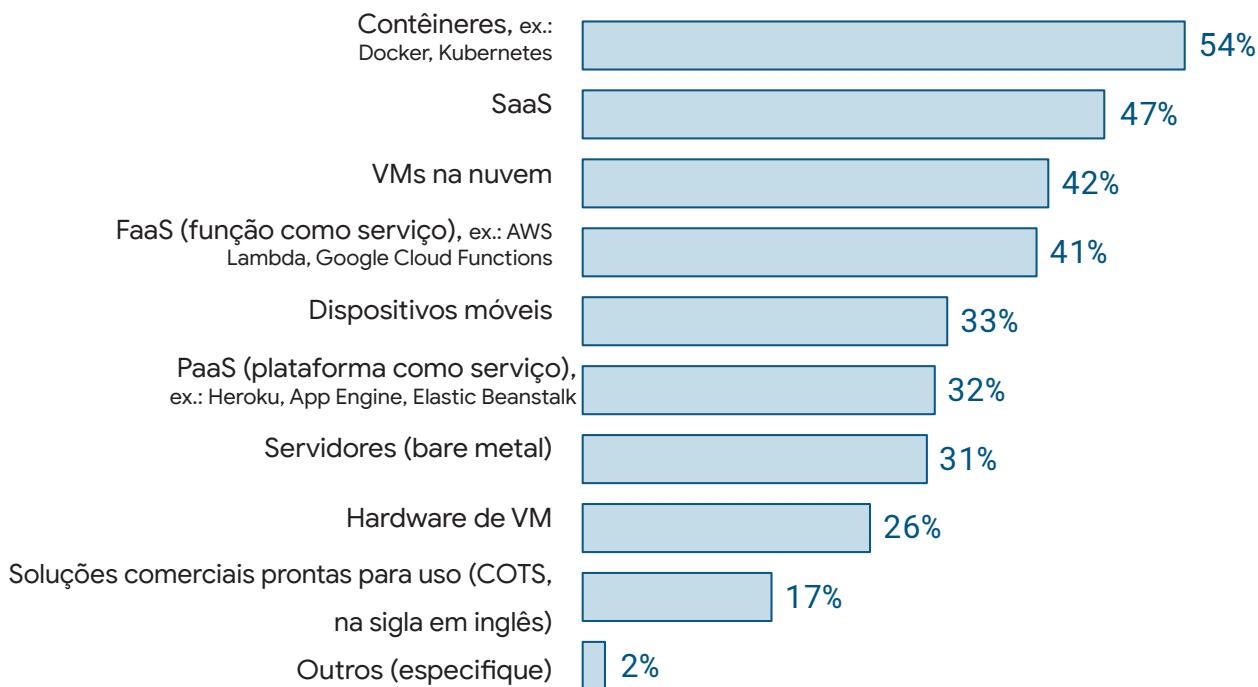


Destino da implantação

O ano de 2021 foi o primeiro em que decidimos perguntar onde os participantes implantam o serviço ou o aplicativo principal em que eles trabalham. Para nossa surpresa, o principal destino de implantação

foram os contêineres. Este ano não foi diferente, mas a uma proporção menor (54%) que no ano anterior (64%).

Também adicionamos mais opções para que os participantes indicassem onde eles implantam.



07

Considerações finais

Sempre que elaboramos este relatório, procuramos fazer uma apresentação rigorosa que mostre como práticas e capacidades levam a resultados essenciais para os negócios, como o desempenho organizacional. Avaliamos se é possível replicar vários efeitos dos relatórios anteriores e estendemos o escopo do trabalho para as prioridades emergentes na área de DevOps. Neste ano, elaboramos a pesquisa e a análise para nos aprofundar nas práticas de segurança. Também alteramos a modelagem estatística para analisar a condicionalidade ou dependência de alguns efeitos. Também exploramos novas maneiras de descrever o cenário do desempenho operacional e da entrega de software.

De muitas maneiras, a narrativa que surgiu neste ano é um eco dos anos anteriores: capacidades técnicas ampliam umas às outras e melhoram o desempenho; o uso da nuvem traz vários benefícios; a cultura e a flexibilidade do local de trabalho melhoram o desempenho organizacional; e o esgotamento dos funcionários impede que as organizações alcancem metas. A análise de interações que adicionamos ao modelo ajudou a entender as condições que levam



Derek DeBellis

ao surgimento de alguns efeitos. Por exemplo, o desempenho da entrega de software só tem impacto positivo sobre o desempenho organizacional quando há alto desempenho operacional (confiabilidade), o que leva à conclusão de que as organizações precisam de ambos para ter sucesso. Há também algumas surpresas, que destacamos em uma seção específica.

Agradecemos a todos que contribuíram com a pesquisa deste ano. Esperamos que nossa pesquisa ajude você e sua organização a criar equipes e softwares melhores, sem abrir mão do equilíbrio entre trabalho e vida pessoal.

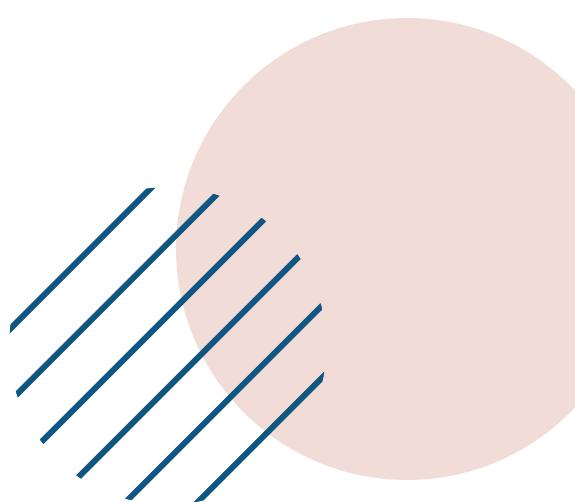


08

Agradecimentos

O relatório deste ano foi possível graças a muitos colaboradores dedicados. Elaboração de perguntas, análises, textos, edições e elaboração de relatórios da pesquisa são apenas algumas das formas como nossos colegas ajudaram a realizar este grande esforço. Os autores gostariam de agradecer a todas essas pessoas pelas contribuições e orientações no relatório deste ano. Os agradecimentos estão em ordem alfabética pelo sobrenome.

Scott Aucoin	Eric Maxwell
Alex Barrett	John Speed Meyers
James Brookbank	Steve McGhee
Kim Castillo	Jacinda Mein
Lolly Chessie	Alison Milligan
Jenna Dailey	Pablo Pérez Villanueva
Derek DeBellis	Claire Peters
Rob Edwards	ConnorPoske
Dave Farley	Dave Stanke
Christopher Grant	Dustin Smith
Mahshad Haeri	Seth Vargo
Nathen Harvey	Daniella Villalba
Damith Karunaratne	Brenna Washington
Todd Kulesza	Kaiyuan "Frank" Xu
Amanda Lewis	Nicola Yap
Ian Lewis	



09

Autores



Claire Peters

Claire Peters é pesquisadora de experiência do usuário no Google, estudando vários aspectos e expressões de DORA em contextos aplicados. Ela estuda o Programa de Modernização de Aplicativos do Google Cloud (CAMP, na sigla em inglês), engajamentos do cliente orientados por DORA e as ferramentas relacionadas a Quatro Chaves, para ajudar equipes e pessoas a aplicar os princípios de DORA no trabalho diário. Claire também faz parte da equipe de pesquisa principal de DORA e elabora a pesquisa anual de DORA e o relatório State of DevOps. Ela tem mestrado em análise cultural aplicada pela Universidade de Copenhague.



Dave Farley

Dave Farley é diretor-gerente e fundador da Continuous Delivery Ltd e criador do canal Continuous Delivery no YouTube. Dave é coautor da série de livros de sucesso Continuous Delivery. Também é autor do best-seller Modern Software Engineering: Doing What Works to Build Better Software Faster. É coautor do Manifesto Reativo e vencedor do Prêmio Duke pelo projeto de código aberto LMAX Disruptor. Dave é pioneiro em entrega contínua, formador de opinião e especialista em CD, DevOps, TDD e design de software. Também tem longo histórico de criação de equipes com alto desempenho, preparação de organizações para o sucesso e criação de softwares excepcionais. Confira mais conteúdo de Dave no [Twitter](#), [YouTube](#), [blog](#) e [site](#).



Daniella Villalba

Daniella Villalba é pesquisadora de experiência do usuário dedicada ao projeto DORA. O foco dela é entender os fatores para deixar os desenvolvedores felizes e produtivos. Antes do Google, Daniella estudou os benefícios do treinamento de meditação, os fatores psicossociais que afetam as experiências de estudantes universitários, a memória de testemunhas e confissões falsas. Ela tem PhD em psicologia experimental pela Universidade Internacional da Flórida.



Dave Stanke

Dave Stanke é engenheiro de relações com o desenvolvedor no Google. Ele dá consultoria sobre práticas recomendadas para adotar DevOps e SRE. Dave já ocupou várias posições ao longo da carreira, incluindo CTO de startup, gerente de produtos, suporte ao cliente, desenvolvedor de software, sysadmin e designer gráfico. Ele tem mestrado em administração de tecnologia pela Universidade Columbia.



Derek DeBellis

Derek DeBellis é pesquisador em experiência quantitativa do usuário no Google. As atividades de Derek no Google são fazer pesquisas, analisar logs e descobrir maneiras de medir conceitos essenciais ao desenvolvimento de produtos. Derek publicou recentemente sobre a interação entre humanos e IA, o impacto do surgimento da COVID-19 sobre a interrupção do tabagismo, design para erros de PLN e o papel da UX nas discussões sobre privacidade.



Eric Maxwell

Eric Maxwell é chefe da área de transformação digital de DevOps do Google, em que orienta as melhores empresas do mundo a serem ainda melhores pouco a pouco, continuamente. Eric passou a primeira metade da carreira como engenheiro, nas trincheiras, automatizando tudo e adquirindo empatia por outros profissionais. Ele é cocriador do Programa de Modernização de Aplicativos Google Cloud (CAMP, na sigla em inglês), membro da equipe principal de DORA e autor de DevOps Enterprise Guidebook. Antes do Google, Eric trabalhava com outros fãs de trocadilhos na Chef Software.



John Speed Meyers

John Speed Meyers é cientista de dados da Chainguard, startup de segurança da cadeia de suprimentos de software. Os projetos de pesquisa de John incluem tópicos como segurança da cadeia de suprimentos de software, segurança de software de código aberto e a resposta do mundo ao crescente poderio militar da China. John trabalhou na In-Q-Tel, na RAND Corporation e no Centro de Avaliações Estratégicas e Orçamentárias. John tem PhD em análise de políticas pela Faculdade de Pós-graduação Pardee RAND, mestrado em assuntos públicos (MPA, na sigla em inglês) pela Faculdade de Assuntos Públicos e Internacionais da Universidade Princeton e bacharelado em relações internacionais pela Universidade Tufts.



Kaiyuan "Frank" Xu

Kaiyuan "Frank" Xu é pesquisador em experiência quantitativa do usuário do Google. Ele analisa logs e dados de pesquisa para entender os padrões de uso e o feedback dos usuários sobre os produtos Google Cloud, melhorando a excelência do produto para os desenvolvedores. Antes do Google, Kaiyuan realizou por anos pesquisas qualitativas e quantitativas de usuário na Microsoft para os produtos Azure Power Platform. Ele tem mestrado em design e engenharia centrados em humanos pela Universidade de Washington.



Nathen Harvey

Nathen Harvey é engenheiro de relações com desenvolvedores do Google. Ao longo da carreira, ele ajudou equipes a realizar o próprio potencial ao mesmo tempo que alinhou a tecnologia aos resultados dos negócios. Nathen teve o privilégio de trabalhar com algumas das melhores equipes e comunidades de código aberto, ajudando a aplicar os princípios e as práticas de DevOps e SRE. Nathen coeditou e contribuiu para *97 Things Every Cloud Engineer Should Know*, O'Reilly 2020.



Todd Kulesza

Todd Kulesza é pesquisador em experiência do usuário no Google, estudando como os engenheiros de software trabalham hoje e como eles vão trabalhar melhor no futuro. Ele tem PhD em ciência da computação pela Universidade Estadual do Oregon.

10

Metodologia

Projeto de pesquisa

Neste estudo, usamos um projeto interseccional e baseado em teoria, chamado **previsão inferencial**, um dos tipos mais comuns usados em pesquisas de negócios e tecnologia hoje. O projeto preditivo inferencial é usado quando o projeto puramente experimental não é viável ou possível.

População e amostragem

A população segmentada para esta pesquisa foi composta por profissionais e líderes que usam (ou trabalham com pessoas que usam) tecnologias e transformações, principalmente os que estão familiarizados com DevOps. Para promover a pesquisa, usamos listas de e-mails, promoções on-line, um painel on-line e mídias sociais. Além disso, pedimos às pessoas que compartilhassem a pesquisa nas redes delas (ou seja, com amostragem de bolas de neve).

Como criar construções latentes

Formulamos nossas hipóteses e construções usando construções validadas anteriormente sempre que possível. Desenvolvemos novas construções com base em teoria, definições e informações de especialistas. Depois procuramos esclarecer a intenção para aumentar a probabilidade de que os dados coletados na pesquisa sejam confiáveis e válidos.¹

Cálculo das diferenças entre baixo e alto desempenho

Na seção "Sua equipe está à altura?", compararmos equipes com baixo e alto desempenho nas quatro métricas do desempenho da entrega. O método é simples. Vamos usar a frequência de implantação como exemplo. O grupo alto implanta sob demanda (ou seja, várias vezes ao dia). Se eles fazem quatro implantações por dia em média, o total é de 1.460 implantações ao ano ($4 * 365$). Já as equipes com baixo desempenho implantam de uma vez ao mês a uma vez por semestre. A frequência média delas é de uma vez a cada 3,5 meses ou cerca de 3,4 implantações ao ano ($12/3,5$). Dividimos 1.460 implantações do grupo alto por 3,4 implantações do grupo baixo para obter o resultado de 417. Essa abordagem foi generalizada para as demais métricas de desempenho de desenvolvimento.

¹ Churchill Jr, G. A. "A paradigm for developing better measures of marketing constructs", Journal of Marketing Research 16:1, (1979), 64–73.

Métodos de análise estatística

Análise de grupo

Para ambas as soluções de agrupamento mostradas na seção "Sua equipe está à altura?", usamos agrupamento hierárquico com uma versão do método de aglomeração de Ward² para avaliar se as soluções eram adequadas aos dados.

Para os primeiros resultados, procuramos por grupos de respostas em termos de frequência de implantação, tempo de lead, tempo para restauração do serviço e taxa de falhas nas alterações. Neste ano, encontramos três grupos depois de avaliar 14 soluções de agrupamento hierárquico, usando 30 índices para determinar o número de grupos.³

A segunda análise que apresentamos tinha o mesmo método da primeira, mas foi implantada sobre dimensões diferentes nos dados. Queríamos procurar

padrões comuns de resposta (ou seja, grupos) em termos de capacidade de processamento (composto por frequência de implantação e tempo de lead), desempenho operacional (confiabilidade) e estabilidade (composto por tempo para restauração do serviço e taxa de falhas nas alterações). Também exploramos vários algoritmos de agrupamento para verificar a sensibilidade dos resultados à nossa abordagem. Não conhecemos uma forma estabelecida de quantificar essa sensibilidade, mas os grupos que surgiram tinham características semelhantes.

Modelo de medição

Antes de realizar a análise, identificamos construções usando análise exploratória de fator com análise de componente principal usando rotação varimax.⁴ Confirmamos a validade divergente e a confiabilidade dos testes estatísticos usando a extração de variação média (AVE, na sigla em inglês), correção, alfa de Cronbach⁵, rhoA⁶, índice heterotrait-monotrait^{5,7} e confiabilidade composta.

² Murtagh, Fionn e Pierre Legendre. "Ward's hierarchical agglomerative clustering method: which algorithms implement Ward's criterion?". *Journal of classification* 31.3 (2014): 274-295.

³ Charrad M., Ghazzali N., Boiteau V., Niknafs A. (2014). "NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set". *Journal of Statistical Software*, 61(6), 1-36. URL: <http://www.jstatsoft.org/v61/i06/>

⁴ Straub, D., Boudreau, M. C., & Gefen, D. (2004). Validation guidelines for IS positivist research. *Communications of the Association for Information systems*, 13(1), 24.

⁵ Nunnally, J.C. *Psychometric Theory*. Nova York: McGraw-Hill, 1978

⁶ Hair Jr, Joseph F., et al. "Partial least squares structural equation modeling (PLS-SEM) using R: A workbook". (2021): 197.

⁷ Brown, Timothy A. e Michael T. Moore. "Confirmatory factor analysis". *Handbook of structural equation modeling* 361 (2012): 379.

Modelagem de equação estrutural

Testamos os modelos de equação estrutural (SEM, na sigla em inglês) usando a análise de mínimos quadrados parciais (PLS, na sigla em inglês)⁸, que é uma modelagem de equação estrutural baseada em correlação.

Análise do segundo modelo de grupo

Para entender o que determina a associação aos grupos, usamos a regressão logística multinomial.⁹

Usamos essa abordagem porque o objetivo era prever a associação aos grupos, que, neste caso, são dados categóricos com mais de dois níveis.

Para entender os resultados previstos pelas associações aos grupos, usamos regressão linear para cada resultado (esgotamento, trabalho não planejado e desempenho organizacional).



⁸ Hair Jr, J. F., Hult, G. T. M., Ringle, C. M., & Sarstedt, M. (2021). "A primer on partial least squares structural equation modeling (PLS-SEM)". Sage publications.

⁹ Ripley, Brian, William Venables e Maintainer Brian Ripley. "Package 'nnet'". Pacote R versão 7.3-12 (2016): 700.

11

Leitura adicional (alguns links estão em inglês)

Participe da Comunidade DORA para discutir, aprender e colaborar para melhorar o desempenho operacional e da entrega de software.

<http://dora.community>

Saiba mais sobre a métrica Quatro Chaves

<https://goo.gle/four-keys>

Confira mais informações sobre capacidades de DevOps.

<https://goo.gle/devops-capabilities>

Confira como implementar as práticas de DORA na sua organização no Enterprise Guidebook

<https://goo.gle/enterprise-guidebook>

Encontre recursos sobre engenharia de confiabilidade do site (SRE)

<https://sre.google>

<https://goo.gle/enterprise-roadmap-sre>

Faça a verificação rápida do DevOps

<https://goo.gle/devops-quickcheck>

Explore o programa de pesquisa de DevOps

<https://goo.gle/devops-research>

Leia o e-book Prêmios DevOps do Google Cloud para aprender com outras empresas que implementaram práticas de DORA

<https://goo.gle/devops-awards>

Conheça o Programa de Modernização de Aplicativos do Google Cloud (CAMP, na sigla em inglês).

<https://goo.gle/3daLa9s>

Como usar dados e métricas de DORA para transformar processos tecnológicos

<https://goo.gle/3Doh8Km>

Leia o artigo: "The ROI of DevOps Transformation: How to quantify the impact of your modernization initiatives" de Forsgren, N., Humble, J., & Kim, G. (2018).

<https://goo.gle/3qECIIh>

Leia o livro: Accelerate: *The science behind devops: Building and scaling high performing technology organizations.* IT Revolution.
<https://itrevolution.com/book/accelerate>

Saiba mais sobre o framework Supply chain Levels for Secure Artifacts (SLSA)
<https://slsa.dev>

Saiba mais sobre Secure Software Development Framework (NIST SSDF)
<https://goo.gle/3qBXLWk>

Saiba mais sobre a cultura de DevOps: Cultura organizacional de Westrum
<https://goo.gle/3xq7KBV>

Leia mais sobre a Open Source Security Foundation
<https://openssf.org/>

Saiba mais sobre in-toto
<https://in-toto.io/>

Saiba mais sobre o Software Bill of Materials de NTIA.gov
<https://www.ntia.gov/SBOM>

Segurança cibernética: reação do governo dos Estados Unidos aos incidentes na SolarWinds e do Microsoft Exchange
<https://www.gao.gov/products/gao-22-104746>

Saiba mais sobre a verificação de segurança no nível do aplicativo como parte da CI/CD
<https://go.dev/blog/survey2022-q2-results#security>

Por último, mas não menos importante, confira os relatórios anteriores de State of DevOps. Todos estão disponíveis em <https://goo.gle/dora-sodrs>:
[Relatório Accelerate State of DevOps de 2014](#)
[Relatório Accelerate State of DevOps de 2015](#)
[Relatório Accelerate State of DevOps de 2016](#)
[Relatório Accelerate State of DevOps de 2017](#)
[Relatório Accelerate State of DevOps de 2018](#)
[Relatório Accelerate State of DevOps de 2019](#)
[Relatório Accelerate State of DevOps de 2021](#)