



Folheto do GitHub

6 dicas do DevOps para ajudar líderes da engenharia a entregar software em escala

O foco do DevOps é a velocidade: desenvolvimento de software mais rápido, atualizações mais rápidas e remessas mais rápidas, e tudo isso gera ciclos de vida de desenvolvimento de sistemas mais curtos. [De acordo com o Relatório do DevOps corporativo da Microsoft](#), as organizações que fazem a transição com sucesso para um modelo de DevOps enviam código com mais rapidez e superam outras empresas em 4 a 5 vezes.

Não é de admirar que organizações de praticamente todos os portes e de vários mercados verticais e geografias globais tenham adotado o DevOps. Outra [pesquisa constatou](#) que mais de 80% dos tomadores de decisão de TI dizem que suas organizações investiram no DevOps.

Mas há desafios reais para qualquer organização que busca o sucesso no DevOps. Quais desafios estão entre os principais? Simplificar um stack de tecnologia complexo, confrontar a resistência organizacional à mudança, lidar com arquiteturas ultrapassadas, reduzir todos riscos de segurança e aplicar a automação a todo o ciclo de vida de desenvolvimento de software (SDLC), entre outras coisas.

Abaixo, exploraremos seis dicas práticas de DevOps que as organizações de alta performance seguem para maximizar seu investimento e benefícios com o DevOps.

1. Criar uma cultura forte de DevOps é importante.

[De acordo com o Relatório anual do Estado do DevOps](#), as empresas frequentemente "se concentram demais nos aspectos de automação do DevOps em detrimento das interações em equipe, fluxo rápido, colaboração e otimização de todo o sistema". Esse foco na automação frequentemente deixa de fora um elemento mais fundamental em qualquer prática bem-sucedida de DevOps: a criação de uma cultura de DevOps que favoreça a colaboração e o compartilhamento de conhecimentos.

Esse compartilhamento é fundamental para criar uma forte cultura de colaboração multifuncional, o que ajuda a proporcionar ao desenvolvedor uma experiência positiva. No GitHub, sabemos o valor de investir na experiência do desenvolvedor em todo o SDLC, porque quando os desenvolvedores estão investidos em seu trabalho, o resultado e a entrega de software são muito melhores.

"As organizações de alta evolução fizeram trabalhos de cima para baixo e de baixo para cima para criar impulso por trás de suas práticas de DevOps, criaram práticas de compartilhamento de conhecimentos e padrões que permitem a otimização rápida do fluxo e estabeleceram processos produtivos de aprovação de mudanças", segundo os autores do Relatório sobre o Estado do DevOps. Os desenvolvedores precisam de uma



plataforma leve para compartilhar com segurança os sucessos iniciais e as práticas comprovadas em toda a organização. Esse compartilhamento é fundamental para criar uma forte cultura de colaboração multifuncional e gerar logs de tomada de decisão e documentação.

O resultado: não exagere no início da sua jornada do DevOps. Em vez disso, concentre-se na criação de uma cultura em que a colaboração, a automação e a comunicação multifuncional sejam priorizadas desde o início.

2. Priorize a segurança desde o início.

DevSecOps é uma prática crescente que se baseia no DevOps para tornar a segurança uma parte central de cada fase do desenvolvimento de software. Há a automatização das configurações de segurança na maior medida possível. Há também a adoção de ferramentas de segurança de DevSecOps e DevOps, como ferramentas de teste estático de segurança de aplicação (SAST) e teste dinâmico de segurança de aplicação (DAST), entre outras coisas.

O GitHub Enterprise fornece uma solução de DevSecOps totalmente integrada compatível com o GDPR, criptografa todos os dados em trânsito e oferece suporte a considerações de controle líderes da indústria com a avaliação CSA-CAIQ da Cloud Security Alliance. As organizações também têm a opção de usar uma solução auto-hospedada, que oferece suporte a requisitos rigorosos de segurança, auditoria e conformidade. O GitHub Enterprise oferece uma experiência de desenvolvimento coesa, projetada para reduzir a troca de contextos, o que auxilia as equipes na implementação de práticas de segurança desde o início do processo de desenvolvimento. Isso é feito sem adicionar atrito desnecessário para as equipes de desenvolvedores e as organizações de DevOps.

O GitHub também oferece o pacote de segurança complementar [GitHub Advanced Security](#), que oferece várias soluções de segurança líderes da indústria para automatizar a detecção de problemas, vulnerabilidades e falhas de codificação em todo o SDLC. Isso facilita o foco em manter os dados organizacionais e de clientes seguros em todas as etapas. E o mais importante: as equipes podem criar códigos mais seguros mais rapidamente com ferramentas abrangentes que oferecem inteligência pré-commit em código vulnerável. Isso capacita os desenvolvedores a corrigir as vulnerabilidades mais importantes para sua organização em poucos minutos.

O resultado: as organizações de DevOps precisam mudar o foco dos detalhes das auditorias e da conformidade para o desenvolvimento e o fornecimento de código seguro em todas as fases do SDLC.



3. Aplique a automação estrategicamente o máximo que puder.

Automação e DevOps não são sinônimos. Há organizações praticando o DevOps que aplicam alguma ou nenhuma automação aos seus processos principais. Entretanto, a automação permite que as organizações de DevOps alcancem graus mais altos de performance, removendo tarefas repetitivas e limitando a chance de erro humano, entre outras coisas.

Quando implementada corretamente, a automação também pode melhorar a experiência do desenvolvedor, agilizando as implantações de software. Por quê? Porque quando as organizações usam estrategicamente a automação no SDLC, elas podem ajudar ativamente seus desenvolvedores a resolver problemas maiores sem ficarem atolados em tarefas repetitivas que não estão diretamente relacionadas à criação de novos códigos.

Algumas áreas estratégicas para aplicar a automação em uma prática de DevOps incluem configurações de sistemas, fluxos de trabalho e provisionamento de sistemas. Essa automação de infraestrutura enfrenta o desafio comum de manter o ritmo do trabalho dos desenvolvedores e não se antecipar aos recursos operacionais, e isso está diretamente relacionado à capacidade de implantar software.

Mas a automação também oferece outro benefício importante. Ela permite que as organizações obtenham maior eficiência, trazendo um maior grau de consistência e velocidade para o SDLC. O GitHub oferece diversos modelos de CI/CD e automação que permitem aos usuários configurar automaticamente fluxos de trabalho e manter o status dos cards de quadros de projeto em sincronia com problemas associados e pull cards. Os usuários podem automatizar ações com base no acionamento de eventos para eliminar tarefas manuais demoradas envolvidas no gerenciamento de um quadro de projeto, ou o SDLC mais amplo.

O resultado: procure áreas em todo o SDLC que possam ser automatizadas, como nas fases de compilação, teste e implantação. Para algumas organizações, isso pode começar com explorações para criar um pipeline de CI/CD ou melhorar um pipeline existente. Seja qual for o caminho que você seguir, é importante identificar tarefas manuais repetitivas como candidatas à automação para melhorar a experiência do desenvolvedor e ajudar suas equipes a enviar software mais rapidamente.



4. Simplifique seu stack de tecnologia.

Alguns dos principais desafios para alcançar o sucesso com o DevOps são orientados às pessoas e frequentemente focados em eliminar gargalos de processos e otimizar a colaboração de desenvolvedores.

Entretanto, outro desafio comum é selecionar, implementar e gerenciar [um ecossistema de ferramentas específicas de DevOps](#) para funções importantes, como CI/CD, monitoramento de aplicações, gerenciamento de infraestrutura e automação. Uma solução para esse desafio é escolher uma plataforma de desenvolvimento extensível que se integre à mais ampla variedade possível de ferramentas. No entanto, muitas organizações estão buscando evitar completamente esse problema, utilizando uma plataforma de desenvolvimento que oferece vários recursos, desde CI/CD até gerenciamento de projetos e segurança.

O objetivo delas? Simplificar seu stack de tecnologia para priorizar uma plataforma que alcança vários objetivos de negócios.

No GitHub, criamos o GitHub Enterprise como uma plataforma singular com gerenciamento de código interno, gerenciamento de projetos, CI/CD, segurança, IDEs, ferramentas de codificação alimentadas por IA e muito mais. Fizemos isso para ajudar a simplificar os stacks de tecnologia de que as equipes de software precisam para navegar e melhorar a experiência geral do desenvolvedor, reduzindo a quantidade de alternância de contexto que as equipes precisam fazer enquanto trabalham.

O resultado: você pode gerenciar todas as suas práticas de desenvolvimento de software e DevOps em um só lugar.

O resultado: priorize as plataformas de desenvolvimento que reduzem o número de ferramentas adicionais necessárias em sua prática de DevOps e oferecem um ecossistema de integrações para encontrar você onde estiver com o stack de tecnologia que você tem.

5. Maximize a eficiência do código aberto com o innersource.

Um número cada vez maior de organizações de SA está adotando e usando o [innersource](#), uma metodologia de desenvolvimento que busca combinar as práticas recomendadas de projetos de código aberto em grande escala com segurança. Esses tipos de esforços de equipe multidesenvolvimento altamente complexos exigem coordenação entre centenas (ou milhares) de desenvolvedores e equipes.



O innersource explica como criar grandes projetos de software de forma distribuída e assíncrona e está provando ser uma arma formidável para combater os muitos desafios criados por projetos de DevOps em grande escala. Ele também provou ser uma ótima maneira de melhorar a eficiência por meio da reutilização de código. As empresas podem acessar e utilizar as práticas recomendadas de código aberto por trás do firewall do innersource.

É no GitHub que está a maior comunidade de código aberto do mundo, e a plataforma em si foi criada, projetada e otimizada para facilitar esforços de desenvolvimento em grande escala em equipes distribuídas. Assim, as organizações de DevOps podem adotar metodologias do innersource com mais simplicidade. Os desenvolvedores e as equipes na plataforma do GitHub podem exibir, modificar e distribuir um projeto para qualquer finalidade que escolherem, conforme determinado pelas licenças de código aberto. Além disso, o GitHub oferece uma ampla série de guias de autoajuda para encontrar usuários para o seu projeto, criar comunidades acolhedoras, investir em manutenção e governança e analisar métricas de código aberto, para citar alguns.

O resultado: o innersource permite que as empresas incorporem as práticas recomendadas de código aberto aos seus próprios esforços de desenvolvimento. Invista na adoção e na implementação de metodologias de innersource entre suas equipes de software para proporcionar maior eficiência, mais transparência e comunicação mais clara.

6. Promova comentários contínuos, integração contínua, gerenciamento de alterações e entrega.

Descobrir práticas comuns e padrões da indústria no DevOps é uma coisa. Outra coisa é integrá-los e incorporá-los profundamente em um fluxo de processo que cruza os pontos de isolamento de dados e informações do desenvolvimento por meio da implantação e das operações.

Para isso, a [integração contínua](#) se baseia na automação de testes para garantir que o novo código seja adicionado a uma aplicação. Como prática, ela se concentra na qualidade no nível da aplicação à medida que códigos novos ou modificados são integrados. O gerenciamento de alterações coloca as operações em cena para mostrar quais outros sistemas podem ser afetados e quais consequências ou oportunidades uma mudança pode gerar em um nível mais abrangente. As estratégias de implantação incremental frequentemente são vítimas de restrições operacionais que podem limitar ou interromper a implantação contínua em seu caminho. Esse é um problema de pessoas, não de tecnologia. E o remédio é a inclusão de todos os stakeholders no desenvolvimento, nas operações e no suporte de aplicações.



Esse tipo de entrada contínua entre departamentos aumentará significativamente a probabilidade de que uma aplicação também atenda às necessidades e expectativas do usuário.

O resultado: crie pontos de verificação automatizados estrategicamente em todo o SDLC para garantir testes contínuos, gerenciamento de alterações e entrega de software, tudo com comentários em tempo real para garantir que as pessoas certas saibam quando algo der errado. Esse nível de automação remove o risco de erro humano e permite que as equipes trabalhem mais rapidamente em grande escala para criar software de qualidade.



Envie software mais rapidamente na plataforma completa para desenvolvedores.

O GitHub oferece a plataforma para desenvolvedores completa para fornecer software seguro em grande escala. Com a confiança de mais de 83 milhões de desenvolvedores e 90% das empresas da Fortune 100, o GitHub ajuda equipes de DevOps de todos os tamanhos a colaborar de forma segura e oferecer aos clientes melhores experiências com mais rapidez.

Descubra por que as empresas mais inovadoras do mundo usam o GitHub.

Iniciar um trial gratuito →

Dúvidas sobre o DevOps? Estamos aqui para ajudar.

Fale conosco pelo email sales@github.com. Saiba mais em github.com/enterprise

