

DevOps

DevOps é um conceito comum no desenvolvimento de software. Mas se você é novo no setor ou está nele há anos, deve saber o que é DevOps e como pode afetar sua forma de trabalhar. Nesta leitura, você vai conhecer o DevOps e examinará sua cultura e seus componentes comuns.

O que é DevOps?

A definição de DevOps depende de quem responde sua pergunta! Mas, para a maioria dos especialistas, DevOps resume-se ao seguinte:

DevOps é uma abordagem para o desenvolvimento de software na qual as equipes de desenvolvimento (dev) e operações (Ops) colaboram para oferecer atualizações de software rápidas e de alta qualidade.

Vamos detalhar os principais componentes dessa definição.

Desenvolvimento

O "desenvolvimento" no DevOps refere-se à equipe de **desenvolvimento**. Essa equipe escreve, edita e documenta o código do software e o mantém em uma base de código ou repositório.

Operações de TI

O "Ops" no DevOps refere-se à equipe de **operações de tecnologia da informação (TI)**. Essa equipe configura, gerencia e otimiza a infraestrutura em que o desenvolvimento depende, incluindo hardware, software, rede e armazenamento em nuvem. A equipe também implementa atualizações de software para os usuários e, de outra forma, oferece suporte ao código.

Velocidade

O objetivo do DevOps é liberar resultados, que são produtos ou serviços que fazem parte de um projeto, de forma rápida e frequente por meio de implantação rápida. Com o DevOps, as organizações lançam atualizações pequenas, mas contínuas, em vez de alterações grandes e pouco frequentes.

Qualidade

Os entregáveis devem ser de alta qualidade, o que significa que atendem às expectativas das partes interessadas quanto ao desempenho, funcionalidade, estabilidade, confiabilidade, disponibilidade e segurança. As partes interessadas incluem equipes de desenvolvimento e operações, mas também equipes de segurança, equipes de linha de negócios e usuários, entre outras partes.

Cultura

O DevOps só funciona se a organização adotar uma cultura de DevOps e a base dessa cultura é a **colaboração**.

Tradicionalmente, as equipes de desenvolvimento e operações trabalham isoladas em seus próprios silos. Mas com o DevOps, elas trabalham juntas e mantêm uma comunicação aberta durante todo o ciclo de vida de **desenvolvimento de software (SDLC)**.

As equipes compartilham informações de forma livre e transparente e trabalham juntas para garantir que os desenvolvedores recebam **feedback rápido** sobre quaisquer problemas de software.

As equipes também **compartilham a responsabilidade** pela qualidade do produto.

- Os desenvolvedores não apenas constroem o código, mas o testam e ajudam a mantê-lo ao longo do tempo; eles não deixam essas tarefas para a equipe de operações.
- O pessoal de operações participa de discussões de planejamento para atualizações de software. Fazendo isso, podem garantir que os desenvolvedores incluam monitoramento, registro, testes de segurança e outros recursos que o sistema exige.

Ao colaborar em todo o SDLC, as equipes entendem melhor as necessidades umas das outras. Por sua vez, podem ajustar suas práticas de trabalho para atender a essas necessidades, melhorando o relacionamento entre os membros e eliminando obstáculos que atrasam as liberações.

Componentes

Vamos discutir outros componentes comumente reconhecidos do DevOps.

Agilidade

O DevOps normalmente envolve uma abordagem ágil para o desenvolvimento de software.

Para explicar a abordagem ágil, vamos primeiro discutir sua antecessora, a abordagem em cascata.

A **abordagem em cascata** é linear. A equipe segue um plano longo e detalhado com fases sequenciais distintas, como planejamento, design, desenvolvimento, teste e suporte. Esse plano especifica os prazos para cada fase e responsabilidades de cada membro da equipe e o objetivo é executar o plano da forma mais precisa possível.

Mas essa abordagem apresenta vários problemas para o desenvolvimento de software.

- À medida que o projeto avança, os desenvolvedores podem encontrar obstáculos ou pensar em ideias novas e melhores, e os clientes podem mudar suas necessidades. No entanto, o plano rígido torna a adaptação difícil e às vezes cara.
- O pior é que clientes e usuários normalmente não veem o produto até a fase de teste, então os desenvolvedores não recebem feedback até lá. Resolver erros e preocupações do cliente no fim do processo pode ser mais caro e demorado do que no início do desenvolvimento.

Por outro lado, o **desenvolvimento ágil** é iterativo. Em vez de concluir o projeto em fases mais longas e distintas, as equipes trabalham em iterações curtas, conhecidas como *sprints*, geralmente com várias semanas de duração.

O objetivo de cada sprint é produzir um pequeno entregável no qual o cliente pode apresentar comentários rapidamente. Dessa forma, os desenvolvedores podem ter conhecimento sobre as preocupações do cliente logo no início e levá-las em consideração daqui em diante.

Além disso, os testes ocorrem em cada sprint e não no fim da linha do tempo do projeto, de modo que as equipes identificam e corrigem erros mais cedo.

Como um projeto em cascata, um projeto ágil ainda tem uma meta geral. No entanto, o plano é menos detalhado e mais flexível para acomodar novas ideias e requisitos.

Resumindo, a abordagem ágil é ideal para DevOps.

- Concentrar-se em breves sprints, pequenos resultados e testes contínuos combina bem com a meta do DevOps de atualizações de software frequentes e de alta qualidade.

- Os desenvolvedores e o pessoal de operações podem se adaptar rapidamente às mudanças nas demandas do projeto e nos padrões de qualidade.

Microserviços

Para acelerar a implementação, o DevOps exige um projeto de aplicativo concentrado em microsserviços. Um **microserviço** é um pequeno componente funcional independente de um aplicativo capaz de funcionar com outros componentes para constituir todo o aplicativo.

No desenvolvimento tradicional, os desenvolvedores constroem todas as partes de um aplicativo em uma grande peça monolítica. Para adicionar um recurso ou corrigir um bug, os desenvolvedores devem reimplementar todo o aplicativo.

Ao utilizar microsserviços, os desenvolvedores só precisam editar o componente correspondente, não o aplicativo inteiro. Essas alterações em menor escala tornam as atualizações mais rápidas e reduzem o risco de interromper o aplicativo.

Automação

O DevOps envolve o uso de ferramentas para **automatizar** o movimento do código por meio do SDLC.

Considere os testes, por exemplo. O código deve passar por diferentes níveis de testes durante todo seu ciclo de vida, e várias ferramentas podem executar automaticamente os testes apropriados em cada estágio.

As ferramentas também podem preparar e configurar componentes de infraestrutura, monitorar problemas de desempenho e segurança, identificar componentes ou ativos incompatíveis e implementar código na produção.

A automação reduz o tempo de implementação e a chance de erro humano. Isso também libera os desenvolvedores e a equipe de operações para se concentrarem na codificação, na solução de problemas e em outras tarefas cruciais para aprimorar o produto.

Integração e entrega contínuas

Integração e entrega contínuas (IC/EC) são dois processos automatizados essenciais para DevOps.

- **Integração contínua (IC)** é um processo no qual os desenvolvedores integram regularmente pequenas partes do código recém-escrito em sua base de código principal. O código passa por testes automáticos, e o gerente de base de código analisa e aprova antes da integração.

- **A entrega contínua (EC)** vem em seguida. É um processo que envia esse novo código para um ambiente de pré-produção para verificar se está pronto para implantação rápida e segura na produção. Aqui, o código passa por testes e análises adicionais e, em seguida, segue para a produção.

IC/EC são processos valiosos para DevOps.

- Eles automatizam os testes, economizando tempo gasto testando manualmente o código e apresentando comentários instantâneos aos desenvolvedores sobre bugs.
- Eles automatizam verificações de cibersegurança, como verificação de vulnerabilidade e aplicativos, que são partes padrão do **DevSecOps** (DevOps com segurança integrada).
- Testes, verificações e revisões frequentes também significam um código de maior qualidade.
- IC/EC automatizam a implementação no ambiente de pré-produção.

Com IC/EC, as atualizações chegam à produção mais rápido e com menos esforço.