DevOps

Origem: Wikipédia, a enciclopédia livre.

A engenharia de software é a área responsável pelo estabelecimento de técnicas e práticas para o desenvolvimento de software cobrindo uma ampla área de aplicações e diferentes tipos de dispositivos.[1]

Na Ciência da Computação o DevOps (contração de development e operations), é uma cultura na engenharia de software que aproxima os desenvolvedores de software (Dev) e os operadores do software / administradores do sistema (Ops),[2] com característica principal de melhorar a comunicação dos dois papéis dentro de um projeto e defender a automação e monitoramento em todas as fases da construção de um software (desde a integração, teste, liberação para implantação, ao gerenciamento de infraestrutura), auxiliam empresas no gerenciamento de lançamento de novas versões, padronizando ambientes em ciclos de desenvolvimento menores, frequência de implantação aumentada, liberações mais seguras, em alinhamento próximo com os objetivos de negócio.[3][4][5][6][7]

Empresas que liberam novas versões de software frequentemente podem precisar das considerações ou orientações de um SysAdmin (Administrador do sistema). O Flickr desenvolveu a cultura de DevOps para suprir uma necessidade do negócio de realizar dez implantações por dia, este ciclo diário de implantações será muito maior em organizações que produzem aplicações multi-foco ou multi-funções. É conhecido como implantação contínua[8] ou entrega contínua[9]. Grupos de trabalho, associações de profissionais e blogs estão tratando do tema desde 2009.[6][10][11]

A cultura DevOps auxilia empresas no gerenciamento de lançamento de novas versões, estimular a comunicação entre os dois papéis. Eventos podem ser acompanhados com maior facilidade, assim como o controle de processos documentados e emissão de relatórios granulares. Empresas com problemas no processo de liberação/implantação de novas versões, normalmente possuem automação, mas querem maior flexibilidade para gerenciar e conduzir esse processo - sem precisar editar tudo na linha de comando. Idealmente, essa automação deve ser disparada por recursos não operacionais, em ambientes específicos que não estejam "em produção". O desenvolvedor ganha maior controle sobre o ambiente, e o administrador do sistema maior entendimento sobre os aplicativos.

Processos simples se tornam claramente articuláveis, através do DevOps. O objetivo é automatizar a maior quantidade possível de processos operacionais.

Integrações DevOps visam a entrega de produtos, testes de qualidade, desenvolvimento de características e releases de manutenção, de modo a incrementar a confiança e segurança, desenvolvimento rápido e ciclos de desenvolvimento. Muitas das ideias (e pessoas) envolvidas com a cultura DevOps vieram dos movimentos de Gerenciamento de sistemas empresariais e Desenvolvimento ágil de software.[12]

Índice

1 Etimologia

2 História

2.1 Impacto nos lançamentos de aplicações

3 Cadeia de ferramentas DevOps

4 Coordenador de lançamento

5 Referências

6 Ver também

7 Ligações externas

Etimologia

O termo DevOps deriva da junção das palavras development e operations, onde em português une o desenvolvimento de software com as operações das tecnologias.

História

O termo "surgiu" inicialmente no evento Velocity em 2009, onde John Allspaw e Paul Hammond apresentaram a palestra de título: "10+ Deploys Per Day: Dev and Ops Cooperation at Flickr" que contava sobre os resultados e desafios da maior aproximação entre a equipe de desenvolvimento e de operações no Flickr. Patrick Debois, que assistiu a palestra online e viria ser o criador do termo, teve a ideia de criar o evento "DevOps Days".[13]

O termo "devops" foi popularizado através de uma série de eventos intitulados "DevOps Days", começando em 2009 na Bélgica.[14] Desde então, ocorreram conferências "DevOps Days" na Índia, EUA, Brasil, Austrália, Alemanha e Suécia.[15]

Ilustração mostrando como são as intersecções entre Desenvolvimento (Engenharia de Software), Operações e Controle de Qualidade.

Metodologias como Desenvolvimento Ágil, que são adotadas por organizações tradicionais, separam os departamentos de desenvolvimento, operações de TI, e controle de qualidade, não integrando as atividades de desenvolvimento e implantação entre os departamentos de TI e controle de qualidade. DevOps promove um conjunto de processos e métodos para pensar sobre comunicação e colaboração entre departamentos.[16]

A adoção do DevOps é conduzida por fatores, tais quais:

Uso de processos e metodologias de desenvolvimento ágil

Demanda crescente na taxa de produção de aplicativos e unidades de negócios

Ampla disponibilidade de infraestrutura na Cloud e virtualizada[17] por provedores internos e externos

Uso crescente de automação de data center[18] e ferramentas de gerenciamento de configuração

DevOps é frequentemente descrito como uma relação mais colaborativa e produtiva entre equipes de desenvolvimento e operações. Essa melhora na relação e colaboração aumenta a eficiência e reduz os riscos da produção associados com mudanças frequentes. Estão surgindo medidas de ROI[19] e métricas potenciais para DevOps.

A função de um profissional DevOps possui semelhanças com as de um Engenheiro Chefe dentro do Sistema de Produção da Toyota.[20] Tais pessoas têm responsabilidades para o sucesso do projeto, mas não possuem autoridade formal sobre as diferentes equipes envolvidas. Isto exige conhecimento técnico, de modo a convencer os gerentes das necessidades. O aval dos executivos da empresa para o Engenheiro chefe pode auxiliar.

Muitas organizações dividem Desenvolvimento e Administração de sistemas em departamentos diferentes. Enquanto os departamentos de desenvolvimento são guiados pelas necessidades do usuário para entregar frequentemente novos recursos, os departamentos de operações focam mais na disponibilidade e estabilidade dos serviços de TI e eficiência de custos em TI. Esses dois objetivos contraditórios criam um abismo entre Desenvolvimento e Operações, o que desacelera a entrega dos valores nos negócios de TI.

Há alguns anos as técnicas de DevOps são utilizadas por programadores no desenvolvimento de softwares, em conjunto com as técnicas do framework Agile + Scrum. [21] [22] [23]

Impacto nos lançamentos de aplicações

Em muitas empresas, eventos de lançamento de aplicações são atividades muito estressantes e de alto risco, que envolvem várias equipes. Em uma organização DevOps, o lançamento de aplicações é de baixo risco pelas seguintes razões:

Este diagrama mostra as diferenças, em termos de frequência de lançamentos e escala de impacto, entre as metodologias ágeis e as iterativas tradicionais.

Escopo de mudança reduzido

Adoção do desenvolvimento iterativo ou ágil, em contraste com o tradicional modelo em cascata. Isto significa que cada lançamento possui menos mudanças, mas ocorrem com maior frequência. O que ocorre, então, é a criação de uma suave taxa de mudança progressiva na aplicação vs. o grande impacto causado por implantações raras - cada uma contendo um grande número de mudanças.

Coordenação de lançamentos crescente

Uso de um coordenador de lançamentos forte para criar a ponte sobre o abismo comunicacional e de conhecimento entre desenvolvedores e operadores; emprego de ferramentas de colaboração tais como planilhas, teleconferências, mensagens instantâneas, portais corporativos e wikis, para garantir um entendimento minucioso da mudança e cooperação total dos envolvidos.

Automação

Automação completa de implantação garante replicabilidade das tarefas e reduz os erros.

Cadeia de ferramentas DevOps

Ilustração mostrando etapas em uma cadeia de ferramentas DevOps

Ilustração mostrando etapas em uma cadeia de ferramentas DevOps

Como o DevOps é destinado a ser um modo de trabalho inter-funcional, em vez de uma única ferramenta DevOps, existem conjuntos (ou "cadeia de ferramentas") de várias ferramentas. Espera-se que essas ferramentas DevOps se encaixem em uma ou mais destas categorias citadas abaixo, refletidas de aspectos-chave do processo de desenvolvimento e entrega de sistemas:

Codificação - desenvolvimento e revisão de código, ferramentas de gerenciamento de código-fonte, fusão (merge) de código

Compilação - ferramentas de integração contínua, estado de compilação

Teste - ferramentas de teste contínuo que fornecem feedback sobre riscos do negócio

Pacote - repositório de artefato, etapa de pré-implantação de aplicação

Liberação - gerenciamento de mudança, aprovações de liberação, automação de liberação

Configuração - configuração e gerenciamento de infraestrutura, ferramentas de Infraestrutura como Código

Monitoramento - monitoramento de desempenho de aplicações, experiência do usuário final

Algumas categorias são mais essenciais em uma cadeia de ferramentas DevOps que outras, especialmente a integração contínua (como, por exemplo, o Jenkins) e infraestrutura como código (como, por exemplo, o Puppet).

Coordenador de lançamento

Uma função relativamente nova na TI corporativa, está incumbida da coordenação da implantação dos softwares da empresa até a pré-produção dos ambientes. A necessidade do coordenador de lançamento foi impulsionada por:

Necessidades para preencher o "abismo" DevOps;

Complexidade crescente da infraestrutura - camadas e plataformas múltiplas, que formam aplicações web;

Crescimento na taxa de lançamentos - devido ao desenvolvimento ágil e iterativo;

Equipes distribuídas - implantação globalizada, desenvolvimentos terceirizados e híbridos.

A função do coordenador (também conhecido como coordenador de implantação e integração) emergiu das equipes de gerenciamento de lançamento ou engenheiros de lançamento. Esta função é similar a do operador de tráfego aéreo - coordenando em tempo real atividades em diversas equipes, para atingir o objetivo de todo o grupo (decolagem e aterrissagem seguras) usando recursos compartilhados (espaço aéreo, caminhos de voo, pistas de aeroportos e terminais de embarque).

O coordenador de lançamento contrasta com o gerente de lançamento, que está sempre focado no planejamento e aviso de mudanças no software, de modo a controlar mudanças específicas na aplicação em produção. O engenheiro de lançamento está preocupado com o trabalho sistêmico e técnico relacionado a construção e implantação do código nos ambientes.

Gerenciamento de mudanças é a disciplina de infraestrutura para acompanhar todos os tipos de mudanças no ambiente de TI da empresa - incluindo tanto mudanças na aplicação quanto na infraestrutura. Gerenciamento de mudanças é parte integrante do ITIL v3.