

Adriano Martins Antonio

| DEVOPS

Eficiência contínua através da cultura de colaboração



Um guia sobre DevOps e que também prepara para a certificação EXIN DevOps Foundation (DEVOPSF).

DEVOPSF – DevOps: Eficiência contínua através da cultura de colaboração

@PMG Academy

"Este conteúdo foi preparado pela PMG Academy (parceiro oficial) e segue as guidelines do EXIN, auxiliando o profissional na preparação para o exame EXIN Devops Foundation.

O EXIN não se responsabiliza pelo conteúdo e/ou qualquer ideia que venha a ser expressada pelo autor, sendo o mesmo responsável pela obra." (EXIN)

ISBN: 978-65-995022-4-8

Este livro é um guia preparatório para a certificação EXIN DevOps Foundation e também expõe os principais fundamentos da cultura DevOps.

Copyright © 2021 PMG Academy
Todos os direitos reservados.

DEVOPSF é uma fonte de conhecimento para o profissional que deseja conhecer o DevOps, uma cultura que promove colaboração entre as equipes de desenvolvimento e operação.

DevOps: Eficiência contínua através da cultura de colaboração.

@PMG Academy.

Brasil/SP/2021_ Edição 01

Sumário

Introdução

Sobre a Certificação EXIN

Sobre a EXIN

Apresentação

Capítulo 1 – Introdução ao DevOps

Objetivos do Capítulo

As origens do DevOps

Modelo em Cascata

O Manifesto Ágil

Obstáculos a serem superados pela TI

Gerenciamento de infraestrutura por código

Cloud Computing

O que significa gerenciar a infraestrutura remotamente?

Como surgiu o termo DevOps?

Mas afinal, o que é DevOps?

Lean

Desperdícios em TI

Outros Tipos de Desperdícios

Algumas observações

Resumindo... Elementos essenciais do DevOps

Por que usar DevOps?

Acelerar o tempo de lançamento no mercado

Reduzir as “dívidas técnicas”

Eliminar fragilidades

Sistemas frágeis, sistemas resilientes e sistemas antifrágéis

Equívocos mais comuns

DevOps como parte do Agile

DevOps é só uma ferramenta de automação

DevOps é uma nova profissão

Encerramento do Capítulo 1

Pratique o que Aprendeu

Gabarito de Respostas

Capítulo 2 – Princípios do DevOps

Objetivos do Capítulo 2

Fluxo de Valor

Pipeline de Implantação

Funções do Pipeline de Implantação

Desafios do Pipeline de Implantação

Integração, entrega e implantação contínuas

Controle de Versão

Gerenciamento de Configuração

Definição de Pronto

Encerramento do Capítulo 2

Pratique o que Aprendeu

Gabarito de Respostas

Capítulo 3, Principais Práticas de DevOps

Objetivos do Capítulo 3

Diferenças entre DevOps e Práticas Tradicionais

Lançamentos são rotina

Lançamentos de acordo com o plano de negócio

Tudo é automatizado

Incidentes são resolvidos imediatamente

Defeitos são corrigidos imediatamente

Os processos são aprimorados continuamente

Gerenciamento de Construção/Gestão de Builds

Agir como uma Startup

Práticas do DevOps

Times incomuns

Visualização do trabalho

Limite do Trabalho em Andamento (WiP)

Reduzir o tamanho do lote

Cuidado com os requisitos operacionais

Deteção e correção de defeitos mais cedo

Melhorias e inovações gerenciadas, ao invés de controladas

Investimentos que permitem inovações

Priorização de Tarefas

Identificação contínua, explorando e elevando restrições

Encerramento do Capítulo 3

Pratique o que Aprendeu

Gabarito de Respostas

Capítulo 4, Aplicações de DevOps

Apresentação do Capítulo

Aplicação prática

Limitações

Fatores vistos incorretamente como inviabilizadores do DevOps

Uso de softwares de prateleira

Arquitetura em evolução

DevOps e ITSM

Comece onde você está, o progresso será iterativo

Encerramento do Capítulo

Pratique o que Aprendeu

Gabarito de Respostas

Conclusão

Glossário

Simulado Oficial EXIN

Questões

Gabarito de Respostas

Literatura

Introdução

Sobre a Certificação EXIN

A certificação EXIN Devops Foundation atesta os seus conhecimentos sobre os princípios e conceitos que formam o DevOps.

Por isso, o conteúdo deste livro tem como objetivo:

- Descrever os principais conceitos de DevOps, partindo de sua origem, definição do termo, razões para utilizar e equívocos mais comuns;
- Identificar os princípios essenciais para a prática DevOps;
- Descrever as diferenças em relação às práticas tradicionais ;
- Explicar as principais práticas de DevOps;
- Definir sua aplicabilidade e limitações;
- Discutir sobre o uso de software de prateleira, evolução da arquitetura, Modelos Organizacionais, e progressão iterativa.

Sobre a EXIN

A EXIN é o Instituto de Pesquisa para a Ciência da Informação, um provedor global e independente de TI, com mais de 40 anos de experiência e sem fins lucrativos. Sua especialidade são os programas de qualificação, que estabelecem os requisitos educacionais e o desenvolvimento de exames para todas as principais áreas de TI. Então, seus certificados são uma forma de provar suas competências e conhecimentos.

A missão da empresa é melhorar a qualidade do setor, dos profissionais e dos usuários de TI. Consequentemente, é possível ter uma ideia do quão importante é ter um certificado oficial do Instituto.

Público-alvo

O conteúdo abordado neste livro tem um público-alvo bastante amplo, destinando-se a todos que desempenham um papel ou possuam interesse na área de TI e de negócios, que querem entender melhor DevOps e como suas organizações podem se beneficiar com esses princípios. Isto inclui aqueles que participam em times DevOps e qualquer pessoa envolvida no Gerenciamento de TI. Ele é indicado para:

- Gerentes;
- Equipes de negócio e supervisão;
- Líderes de equipe;
- Arquitetos de serviço;
- Arquitetos de TI e Planejadores;
- Consultores de TI;
- Gerentes de Auditoria de TI e auditores;
- Gerentes e colaboradores de segurança de TI;
- Gerentes de programas ou projetos;
- Fornecedores, fornecedores líderes e seus fornecedores subcontratados;
- Clientes de provedores de serviço.

Apresentação

Serei o seu guia DevOps.

Os métodos de gerenciamento de TI não param de evoluir, e isso você provavelmente já percebeu. Eles evoluem dia após dia através da sistematização e aprimoramento dos modelos que já existem.

Ao longo dos capítulos a seguir, nos aprofundaremos no universo do DevOps, método popular de colaboração e comunicação entre executivos e profissionais de TI durante o ciclo de vida de uma aplicação ou serviço.

Lendo este livro, você vai aprender desde suas origens até sua aplicação e benefícios de sua utilização.

Capítulo 1 – Introdução ao DevOps

Objetivos do Capítulo

Ao longo desse capítulo, você vai aprender:

- Como surgiu a prática DevOps, passando por Manifesto Agile, gerenciamento de infraestrutura, Cloud Computing e a origem do termo “DevOps”;
- O que é DevOps e quais são seus elementos essenciais;
- As razões para utilizar as práticas DevOps;
- Os três equívocos mais comuns se tratando de DevOps.

As origens do DevOps

Na década de 80, algumas empresas europeias se tornaram pioneiras, desenvolvendo novas práticas de organização do trabalho e novas abordagens para a resolução de problemas de gestão. Publicações importantes como ITIL®, que assumia o papel de biblioteca de infraestrutura de TI naquela época, precederam algumas das práticas que foram formuladas em 2000-2001, como por exemplo:

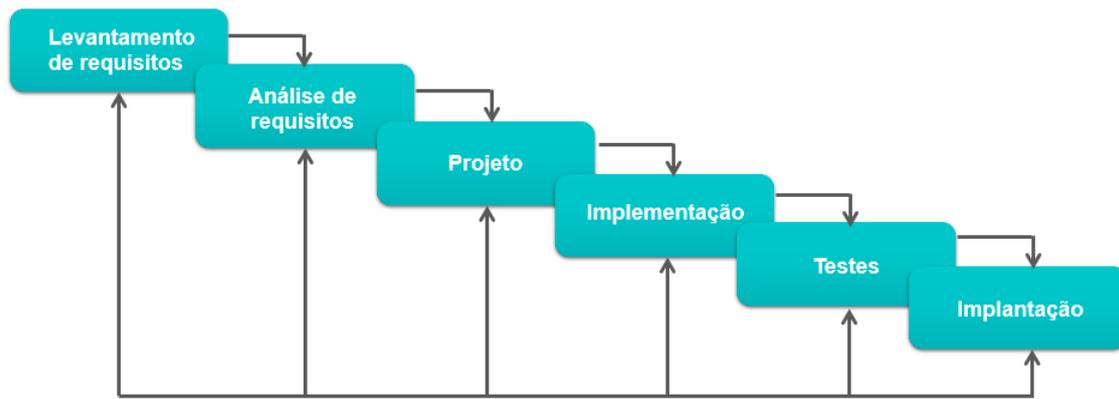
- A introdução de um Service Desk;
- A distinção entre incidentes e problemas;
- O processamento gerenciado e controlado de mudanças na infraestrutura de TI.

Não apenas as organizações líderes optaram por adotá-las, as organizações menores seguiram a mesma tendência, e rapidamente estas passaram a ser consideradas as melhores práticas do mercado.

No ano de 2002, o primeiro padrão para gerenciamento de serviços de TI foi publicado, estabelecendo então uma certa norma a ser seguida por aqueles que buscavam construir um sistema coerente de gerenciamento de serviços de TI.

Modelo em Cascata

No final do século 20, a metodologia que predominava no cenário de desenvolvimento de softwares era o que conhecemos por Modelo em Cascata. Modelo cascata, também conhecido como ciclo de vida clássico ou tradicional, sugere a execução sequencial de estágios pré determinados, onde cada um deles leva um tempo considerável e termina com a conquista dos resultados combinados entre as partes envolvidas.



Fonte: *DevOps – a Business Perspective*

Dentro desse modelo,

- A transição para o próximo estágio, em muitos casos, acontece só quando o estágio anterior está completo;
- Existem funções específicas para pessoas envolvidas em cada estágio, como por exemplo, analistas, arquitetos ou desenvolvedores.

Esse modelo realmente permitia a criação de produtos de alta qualidade, aliados a um controle de custos efetivo e detalhado. Mas, no final da década de 90, com o rápido crescimento das tecnologias da Internet e da programação web, o modelo em cascata já não favorecia a interação entre clientes de sistemas de informação e aqueles que desenvolviam softwares.

Na verdade, para manter a competitividade, era preciso que os produtos chegassem cada vez mais rápido no mercado. E é para suprir essa necessidade que, no começo dos anos 2000, surge o Manifesto Ágil, uma importante declaração de valores e princípios para o desenvolvimento de software.

O Manifesto Ágil

1 - A maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software de valor.

2 - Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que os clientes possam tirar vantagens competitivas.

3 - Entregar o software em funcionamento com frequência, seja na escala de semanas ou meses, dando preferência a períodos mais curtos.

4 - Tanto pessoas relacionadas a negócios como desenvolvedores devem trabalhar em conjunto, diariamente, durante todo o curso do projeto.

5 - Para construir projetos ao redor de indivíduos motivados, é preciso dar a eles o ambiente e o suporte necessários, confiando que farão seu trabalho.

6 - O método mais eficiente de transmitir informações tanto externas como internas para um time de desenvolvimento é por meio de uma conversa cara a cara.

7 - Software funcionando é a medida primária de progresso.

8 - Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante.

9 - Contínua atenção a excelência técnica e bom design aumenta a agilidade.

10 - Simplicidade é a arte de maximizar a quantidade de trabalho que não precisou ser feito.

11 - As melhores arquiteturas, requisitos e designs emergem de times auto-organizáveis.

12 - Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.

No Manifesto Agile, também conhecido como Ágil, temos as seguintes máximas:

- Indivíduos e a interação entre eles, acima de processos e ferramentas;
- Software em funcionamento, acima da documentação abrangente;
- Colaboração com o cliente, acima da negociação de contratos;
- Responder a mudanças, acima de seguir um plano.

Na prática, essas máximas se tornam quatro elementos essenciais de qualquer prática com abordagem ágil, incluindo o Devops. São eles:

- Interação mais próxima entre o cliente e o desenvolvedor;
- Redução do tamanho do lote;
- Produtos entregues em intervalos curtos (ciclos);
- Tamanho limitado das equipes.

Quer dizer, em uma abordagem ágil, o desenvolvimento do código é apenas um dos elos de uma longa cadeia de valores.

Na verdade, antes do desenvolvimento, é necessário passar por outras etapas, destinadas a identificar as necessidades de um negócio, sua elaboração, análise, priorização, e assim por diante.

Isso tudo porque depois do desenvolvimento, os aplicativos precisam ser implantados rapidamente no ambiente de produção, para que os clientes recebam todos os benefícios prometidos e possam fornecer feedback aos desenvolvedores.

Obstáculos a serem superados pela TI

A infraestrutura de TI de quase todas as organizações estabelecidas antes de 2010 fazia com que todas as possíveis alterações fossem muito demoradas e caras.

Existem alguns fatores que podem alterar a produtividade do departamento de TI:

- As soluções de TI utilizadas são extremamente complexas, pois existem muitos milhares de itens interconectados pela infraestrutura;
- A falta de documentação dos sistemas de TI;
- A rapidez com que é ultrapassada a documentação existente.

Mudanças, em geral, não costumam ser bem vindas no departamento de operações de TI. Em muitas organizações, não seria seguro mexer na infraestrutura. Um fluxo constante de mudanças pode comprometer sua segurança e integridade, causando danos irreparáveis.

Isso quer dizer que os métodos avançados de desenvolvimento de software geralmente são impedidos por obstáculos colocados pelas operações de TI.

Esses obstáculos diminuem a chance das abordagens ágeis darem certo.

Gerenciamento de infraestrutura por código

A história da virtualização de softwares e hardwares começou há décadas atrás, em 1964, com o desenvolvimento do sistema operacional IBM CP-40. Desde então, essa tecnologia evoluiu e continua evoluindo consideravelmente.

Os primeiros sistemas para *mainframes* apareceram nos anos 70, e já nos anos 80, outras máquinas mais comuns baseadas na arquitetura Intel x86 foram surgindo.

A virtualização tornou possível não apenas usar hardware caro e poderoso com mais eficiência, mas também introduzir um nível adicional de abstração entre o código executável que fornece algo útil para o cliente e o software do sistema.

Um passo importante foi dado no sentido de separar as competências e responsabilidades de "engenheiros de aplicação" e "engenheiros de sistema", no sentido amplo desses conceitos.

Cloud Computing

A tecnologia da computação em nuvem, ou Cloud Computing, foi desenvolvida num curtíssimo espaço de tempo. As empresas de telecomunicações, até então, ofereciam a seus clientes o serviço Wide Area Network (WAN) até 1990, conectando os terminais com cabeamento direto para cada cliente.

Mas, o surgimento da tecnologia de Virtual Private Network (VPN), as redes virtuais privadas, tornou possível o envio de pacotes de dados de diferentes clientes pelos mesmos canais de transmissão, com o nível necessário de privacidade, segurança e qualidade de serviço.

Você sabia? Naquela época, os provedores passaram a usar o símbolo da nuvem para mostrar a fronteira entre a rede privada do cliente e a rede compartilhada. Assim surgiu o nome Computação em Nuvem (Cloud Computing).

As tecnologias de virtualização e nuvem mudaram significativamente o panorama da computação.

A transferência de dados por longas distâncias já não era apenas utilizada pelos seus clientes para a troca de informações entre sistemas remotos, mas também, para distribuir a carga computacional entre os diferentes nós de suas redes. Assim, a interação foi simplificada e barateada.

O Instituto Nacional de Padrões e Tecnologia dos Estados Unidos (US National Institute of Standards and Technology) identificou cinco características essenciais da computação em nuvem:

1. Autoatendimento sob demanda;
2. Amplo acesso;
3. Combinação de recursos, ou pooling de recursos, que são agrupados para atender a vários consumidores usando um modelo multilocatário, com diferentes recursos físicos e virtuais atribuídos dinamicamente de acordo com a demanda do consumidor;
4. Elasticidade rápida. A elasticidade na cloud computing é a capacidade de expandir e contrair a infraestrutura de processamento conforme a demanda. Para o consumidor, os recursos disponíveis para provisionamento muitas vezes parecem ser ilimitados e podem ser utilizados em qualquer quantidade a qualquer momento;
5. Serviço medido. Os sistemas em nuvem controlam e otimizam automaticamente o uso de recursos, aproveitando a capacidade de medição em algum nível de abstração apropriado para o tipo de serviço.

O que significa gerenciar a infraestrutura remotamente?

O grau de automação cresceu muito, e a velocidade de implementação das mudanças, também. Como resultado da combinação da computação em nuvem, os profissionais de TI desenvolvem nas linhas de comando as partes da infraestrutura que precisam, ou seja:

- Servidores;
- Componentes de rede;
- Sistemas de armazenamento;
- Interfaces;
- Definições e configurações.

E tudo isso é criado por meio de códigos. Não apenas criado, como também gerenciado como um código de programa: há controle de versão e de alterações, o que pode facilitar a reutilização de versões anteriores, por exemplo.

Antes, para implantar uma infraestrutura de TI baseada em hardware interno, chegar ao acordo no orçamento levava pelo menos semanas; esperar o próximo ciclo de compra, meses; solicitar equipamento do fornecedor e pagar por ele, levava dias, isso sem contar a espera pela entrega, receber, instalar, configurar, preparar para o uso...

Hoje é possível criar uma infraestrutura de TI semelhante executando um script, o que dura minutos, e raramente, algumas horas.

Como surgiu o termo DevOps?

Como discutido no tópico anterior, na segunda metade dos anos 2000, várias técnicas como Agile, Scrum, XP, entre várias outras semelhantes, foram amplamente adotadas no campo de desenvolvimento de software. Expandir para as operações era apenas uma questão de tempo.

Uma das primeiras tentativas documentadas foi feita em 2006, quando Marcel Wegemann publicou um artigo sobre a aplicação dos princípios do desenvolvimento ágil ao trabalho de administradores de sistemas.

Nesse artigo, ele sugeriu as seguintes alterações:

- Incluir catálogos de sistema separados no sistema de controle de versão;
- Administradores de sistema deveriam trabalhar em pares;
- Conduzir retrospectivas de operações seria importante daqui em diante.

Em 2008, na conferência regular sobre Agile realizada em Toronto, dois eventos significativos ocorreram ao mesmo tempo.

Andrew Shafer, que acabava de sair do time de desenvolvimento para o time de operação, sugeriu incluir a nova categoria “Infraestrutura Agile” na programação dos eventos, e Patrick Debois, que trabalhava lado a lado com desenvolvedores, apresentou o projeto “Infraestrutura e operações Agile: quão infra-gile você é?”. Aqueles que participaram do evento ficaram muito entusiasmados com as novas ideias para a área, e dessa forma elas cresciam e se popularizavam cada vez mais.

No mesmo ano, Luke Kanies, fundador do Puppet Labs, falou em uma conferência de softwares de código aberto, que o gerenciamento de configuração precisava ser revisado. Seu relatório chamou a atenção de John Willis, que mais tarde teve uma forte influência no desenvolvimento de ideias de DevOps. É importante notar que o próprio termo DevOps ainda não existia.

O termo DevOps foi escolhido após a apresentação feita por John Allspaw e Paul Hammond na conferência Velocity em 2009. A apresentação se chamava "Mais de 10 implantações por dia: Cooperação de desenvolvimento e operações no Flickr", e causou uma impressão marcante em muitas pessoas que já voltavam sua atenção para esse tópico de uma forma ou de outra.

Patrick Debois decidiu organizar a primeira conferência especializada: a DevOpsDays, que foi realizada em Ghent, Bélgica, no ano de 2009.

Já em 2013, Gene Kim que participou da apresentação, publicou The Phoenix Project e fundou a empresa IT Revolution, que promove fortemente o assunto e realiza a conferência DevOps Enterprise Summit duas vezes por ano.

E foi assim que surgiu o termo DevOps e com ele uma enorme comunidade de fãs e entusiastas.

Agora que você já conhece sua história, podemos juntos notar quatro das suas características:

- As ideias-chave do DevOps surgiram como resultado do trabalho intelectual em busca de soluções para problemas reais de gerenciamento.
- DevOps não tem um pai ou um grupo de companheiros fundadores; as pessoas chave desta equação nunca se conheceram antes, mesmo que tenham pensado em direções parecidas.
- O DevOps não tem e não pode ter um proprietário de copyright que dite seu desenvolvimento, ou introduza restrições de uso.
- O assunto DevOps é muito novo. Tão novo, que chega a ser cedo demais para esperar que existam receitas comprovadas ou métodos universais.

Mas afinal, o que é DevOps?

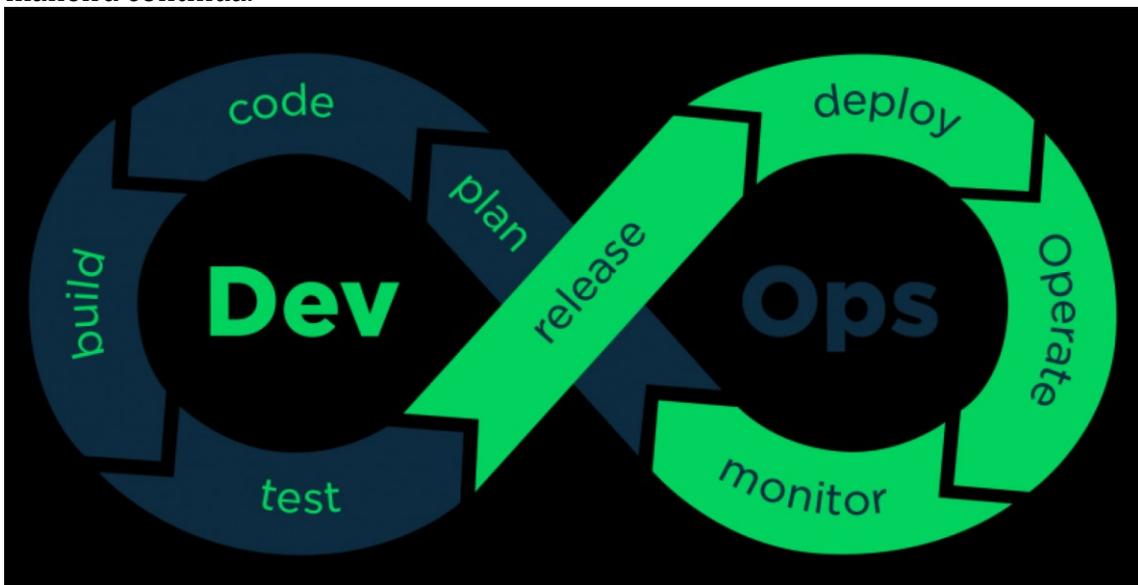
Acabamos de chegar na melhor parte.

O DevOps é uma evolução das ideias de desenvolvimento Ágil e do movimento Lean, que aplicada à cadeia de valor ponta a ponta, alia a modernização do TI com mudanças culturais, organizacionais e técnicas. Dessa forma, permite às empresas alcançarem mais resultados. Em suma, o DevOps promove colaboração e afinidade entre o Time de Desenvolvimento e o Time de Operações.

O DevOps também promove aprendizagem organizacional, ou seja, permite ajudar nossos clientes, garante qualidade e cria vantagem competitiva.

É muito importante ressaltar que o DevOps não substitui as práticas Agile e Lean, mas as absorve e as expande, podendo ser combinado com métodos como Scrum e Kanban, por exemplo.

No DevOps, o departamento de TI da empresa passa a pensar não apenas no desenvolvimento de softwares, mas também em toda a cadeia de valor de um produto, desde a geração de novas ideias junto com as partes interessadas do negócio, passando pelo desenvolvimento, teste, implantação e operação, de maneira contínua.



Fonte: *DevOps – a Business Perspective*

Essa abordagem torna possível que sejam feitas análises e configura loops de feedback não apenas do final do ciclo de vida do produto, mas também entre as etapas. O DevOps, então, presta atenção máxima à abordagem do sistema, ao trabalho com restrições e a gerar feedbacks.

Em uma pequena escala, as equipes que trabalham com o DevOps podem identificar gargalos e superá-los enquanto os problemas ainda são compactos o suficiente para serem tratados de forma mais fácil.

Lean

Conforme dito anteriormente, o DevOps importa muitos princípios do Lean. Alguns até acreditam que o DevOps é a mesma coisa do Lean, o que não é a realidade. Porém, é importante compreender os fundamentos do Lean para conhecermos a base do DevOps.