

Aula 9: DevOps

Apresentação

Nesta aula focaremos no tópico de DevOps, um conceito avançado no cerne da Gerência de Configuração moderna.

Objetivo

- Definir DevOps e seus fundamentos gerais;
- Identificar a cultura necessária para a sustentação do DevOps;
- Identificar os papéis envolvidos e vantagens e desvantagens do DevOps.

O DevOps e seus fundamentos

Ao discutirmos previamente o conceito de Integração Contínua, mencionamos brevemente o contexto ágil em que tal conceito se encontrava inserido. Basicamente, usamos a tendência cada vez maior do uso de frameworks ágeis, tais como o *eXtreme Programming* (XP), para justificar, até certo ponto, a adoção da Integração Contínua.

Comentário

É importante reforçar o caráter parcial que demos a esta justificativa, pois a agilidade não era e não é o único fator que justifica a existência da Integração Contínua como prática aliada à Gerência de Configuração.

Ao falarmos de DevOps, um conceito mais avançado do que Integração Contínua, e que também pode ser associado à Gerência de Configuração, precisamos novamente voltar à tendência da agilidade, pois diferentemente da Integração Contínua, no caso do DevOps, esta tendência se encontra bem no cerne de sua adoção.

O termo em si surgiu pela primeira vez em 2008, após o manifesto ágil e dentro do período em que *frameworks* ágeis, tais como XP, Scrum, Lean e outros, se expandiram e popularizaram, influenciando profissionais e organizações como um todo. Não surpreende que o DevOps já tenha sido conhecido por “infraestrutura ágil” ou “operações ágeis”.

O DevOps é a junção entre Desenvolvimento (Development) e Operações (Ops). É uma mudança de paradigma fundamental em relação às abordagens tradicionais, em que a fase do Desenvolvimento, ou seja, a fase de projetos, sempre foi encarada como sendo completamente segregada da fase de Operações.

Esta conceituação do DevOps deixa clara a presença de uma mentalidade ágil, pois com a junção de ambas as fases busca-se obter um relacionamento agilizado entre elas, melhorando a comunicação e encurtando-se o tempo de desenvolvimento, ao mesmo tempo em que produz software de melhor qualidade.

O DevOps facilita a comunicação e a colaboração através das organizações, objetivando a produção e melhoria de produtos de software em um ritmo maior do que seria possível caso abordagens tradicionais fossem utilizadas.

Além disso, de maneira similar (e por que não complementar?) à Integração Contínua, dedica forte atenção em automação de tarefas, de forma a liberar os desenvolvedores de software para que foquem no que realmente entrega valor:


Desenvolvimento de código e outras tarefas que, em larga medida, não podem ser automatizadas; ao menos, não em sua totalidade.

Saiba mais

Esta automação entre desenvolvimento e operação entrega construção, testes e liberação, além de contar com o uso extensivo de filosofia de trabalho, práticas e habilidades próprias. Entre as várias tecnologias usadas para se atingir este fim, podemos citar por exemplo máquinas virtuais e de containerização (*containers*).

Entre as práticas, podemos citar, por exemplo, a participação conjunta entre engenheiros de desenvolvimento e de operação no ciclo de vida inteiro do produto, dos requisitos ao suporte continuado.



 O ciclo integrado do Desenvolvimento e Operação. Fonte: Adaptado de DevOps Whitepaper, HGS Digital.

O DevOps se organiza ao redor de três princípios:

O Princípio do Fluxo

- A ideia aqui é reduzir o tempo da confirmação (*commit*) ao código rodando em produção.

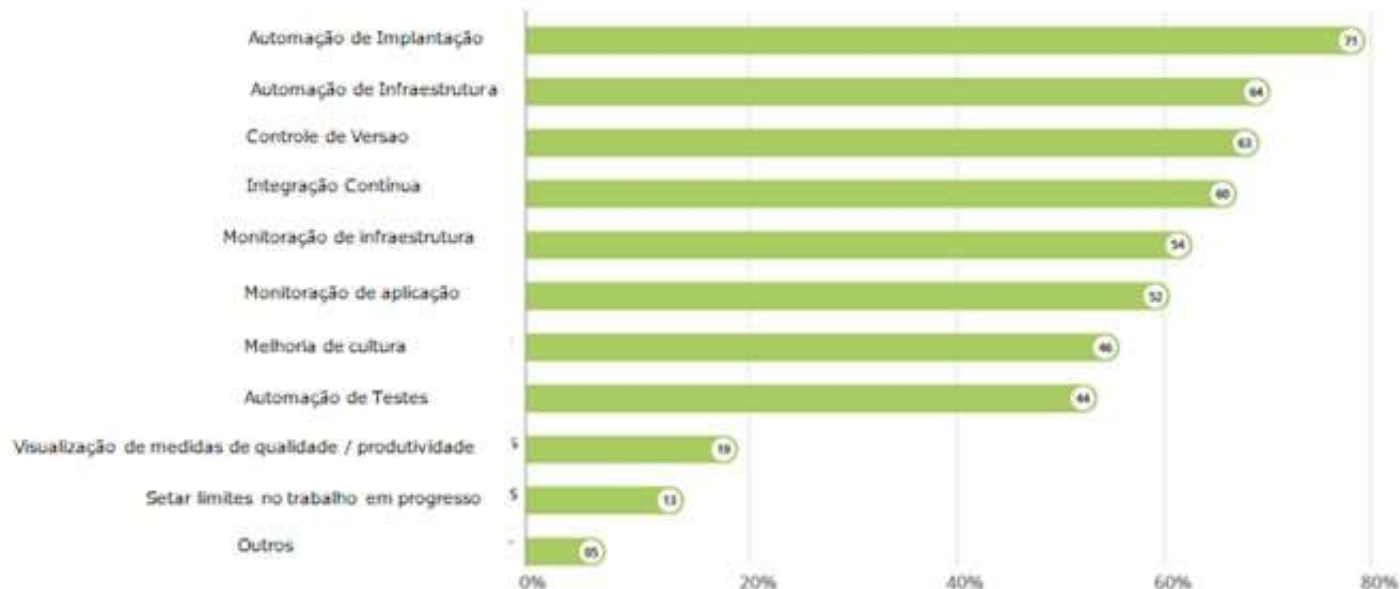
O Princípio do *Feedback*

- Deve-se aumentar o *feedback* do desenvolvimento para produção e da produção para o desenvolvimento.

O Princípio do Aprendizado Contínuo e Experimentação

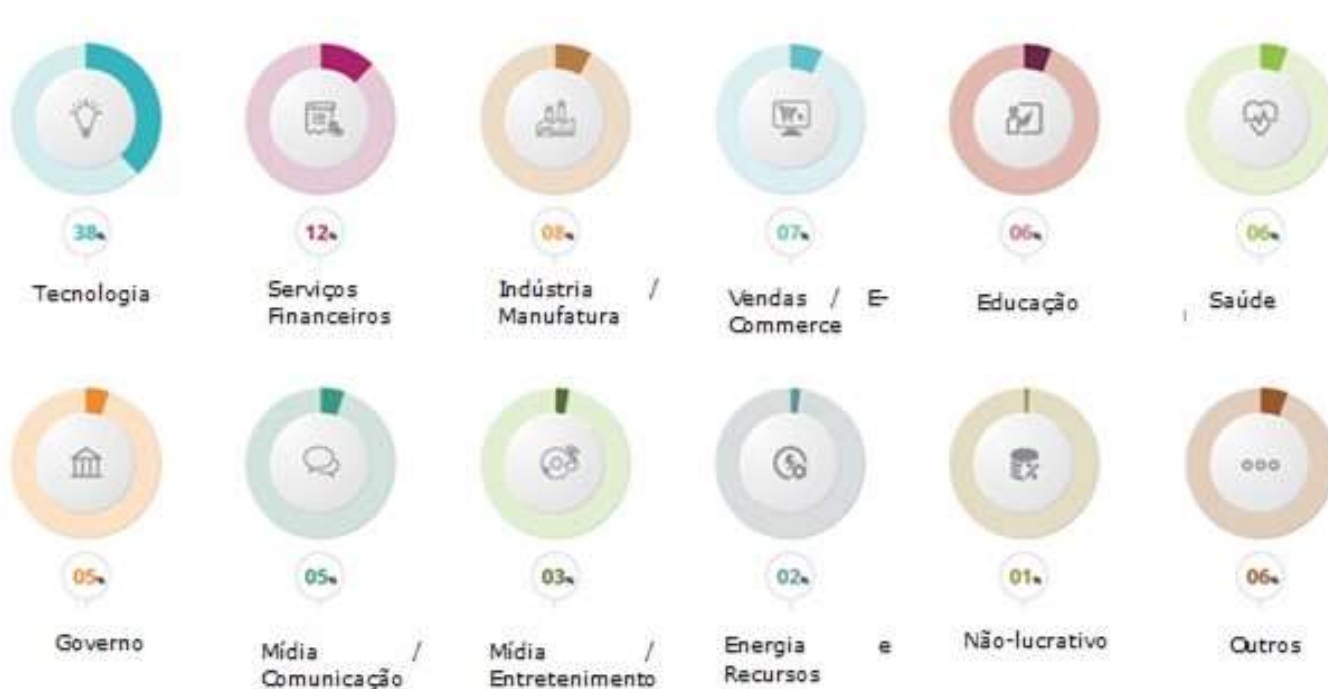
- Aqui se reconhece a importância de se aprender continuamente e usar este aprendizado para melhorar os processos envolvidos.

O DevOps se estrutura ainda ao redor de várias práticas fundamentais, que podem ser adotadas de forma isolada ou em conjunto. De fato, é comum que as organizações se encontrem em níveis diferentes de maturidade na adoção do DevOps, conforme indicado pelo gráfico extraído do Survey Global em DevOps, conduzido no ano de 2017. Note que algumas práticas ainda são muito pouco adotadas.



Adoção de práticas do DevOps. Fonte: Adaptado de 3 Steps to Ensure Compliance and Audit Success with DevOps.

Outro dado que demonstra claramente que a adoção do DevOps como um todo ainda se encontra em andamento, devendo evoluir muito nos próximos anos, é o gráfico de adoção do DevOps por segmento. Não é surpreendente que a indústria da tecnologia, mais especificamente a Indústria de Tecnologia da Informação, esteja na vanguarda da adoção, com ampla vantagem sobre o segundo colocado, no caso a Indústria de Serviços Financeiros.



Adoção do DevOps por Segmento. Fonte: Adaptado de DevOps Whitepaper, HGS Digital.

Atenção! Aqui existe uma videoaula, acesso pelo conteúdo online

A Cultura que Suporta a Adoção e Manutenção do DevOps

O principal ponto cultural que precisa mudar ao se decidir pela adoção do DevOps é a eliminação dos chamados silos funcionais, ou seja, a concepção de que equipes totalmente diferentes devem existir para desenvolvimento e operação de software.

É possível que duas equipes formais separadas ainda existam para cumprir estes propósitos; contudo, no DevOps a linha que as separa se torna cada vez mais difusa, até um ponto em que se percebe que é melhor fundir as duas equipes em uma única entidade.

Comentário

Independentemente da abordagem seguida (se separar ou fundir), o DevOps extingue a cultura da “passagem do bastão”, ou seja, a cultura que se centrava em um ponto de corte, em que o software, recém entregue pela equipe de projetos, era repassado à equipe de operações, que passaria então a ser a responsável por suportar o software adiante.

Esta cultura anterior era baseada mais em performances individuais, tanto das equipes em si quanto dos profissionais que as integravam.

No DevOps, ao contrário, se valoriza o compartilhamento de responsabilidades e a flexibilidade em se ter profissionais trabalhando em pontos ou atividades distintas, sendo que o valor é gerado ao se atender a necessidade imediata do esforço de desenvolvimento ou operação corrente.

Exemplo

Se um desenvolvedor está disponível e existe a necessidade em se corrigir defeitos vindos da operação de um produto de software, então não deve ser problema que tal desenvolvedor possa realizar a correção ou mesmo testar a correção.

Outro ponto importante derivado dessa mudança de mentalidade é o endereçamento de um problema que afeta organizações de TI ao redor do mundo: A taxa de rotatividade (turnover). Não é incomum encontrar organizações enfrentando problemas com membros-chave, deixando as equipes, seja de projetos, seja de operações, também em momentos chave ou críticos.

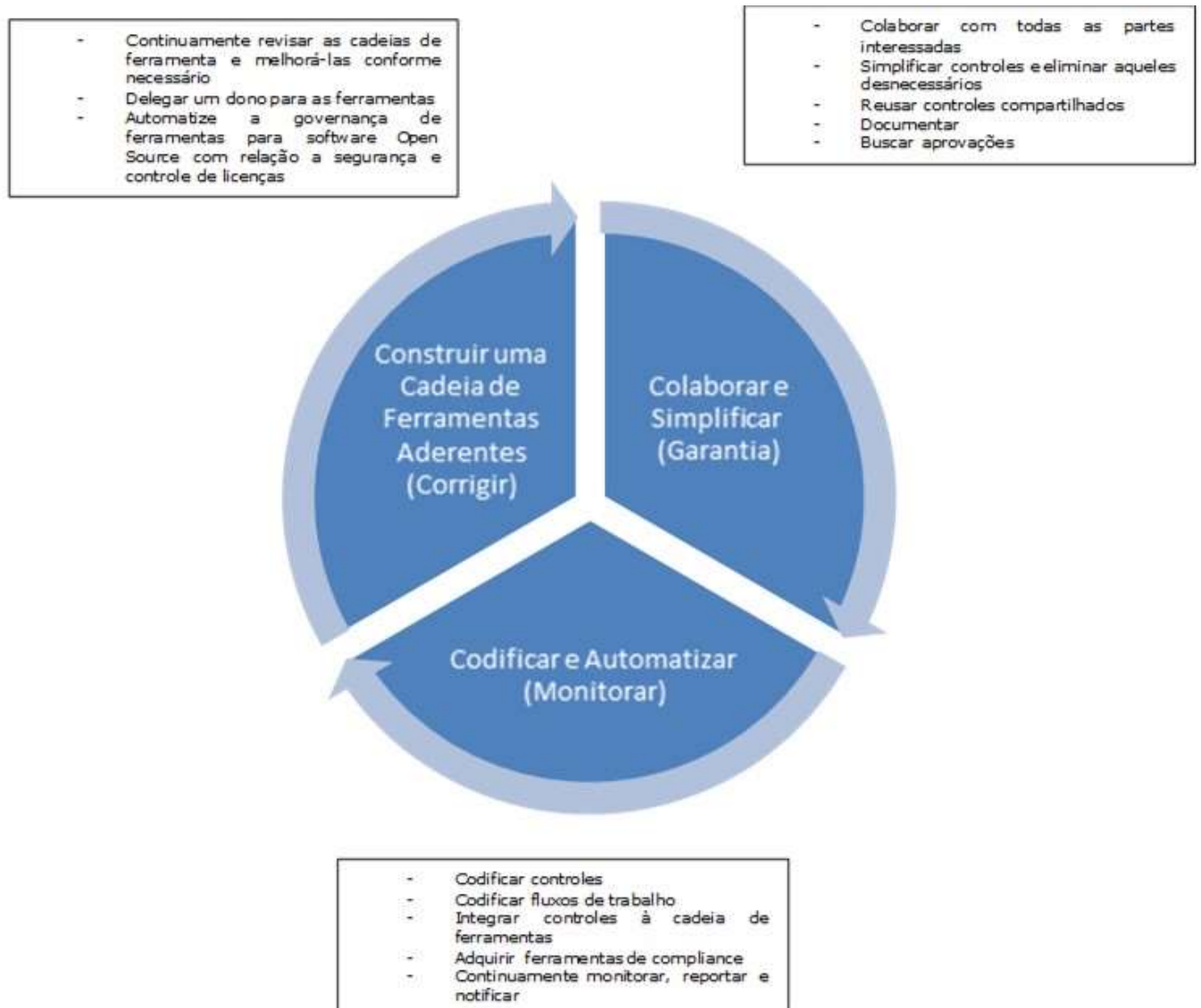
Com a nova cultura de compartilhamento de responsabilidades, é possível expor membros do time a vários tipos de trabalho.

Isso leva ao aumento do conhecimento em relação ao todo, fazendo com que não só contribuam em algum momento com todo o ciclo de desenvolvimento, implantação e sustentação, mas também reduzindo a dependência em relação a profissionais específicos, tornando toda a organização mais segura, e também protegendo a relação com o cliente, que normalmente é quem vê diretamente o impacto negativo da rotatividade.

Dados estes fatores, fica claro que a adoção do DevOps não ocorre com sucesso sem uma estratégia clara para se quebrar os silos funcionais.

A Integração Contínua e, mais especificamente, sua evolução encontrada na Entrega Contínua também são pilares do DevOps, e, portanto, devem ter sido internalizadas culturalmente pelas organizações antes que o DevOps possa vir a ser adotado.

Tais práticas automatizam partes importantes do desenvolvimento e implantação, liberando as equipes para focarem em pontos mais críticos, como, por exemplo, em reduzir os tempos de desenvolvimento e a apresentarem liberações mais rápidas, frequentes e com menos erros. Um panorama da automação requerida pelo DevOps pode ser vista a seguir.



A existência de automação no estado da arte ajuda ainda as equipes técnicas a focar nos clientes e suas necessidades, uma clara distinção que separa o DevOps de abordagens mais tradicionais.

Comentário

Aliás, o DevOps estimula a comunicação próxima entre desenvolvedores e clientes de forma a assegurar que o software desenvolvido atinja as expectativas iniciais, além é claro de trazer a vantagem secundária de que problemas sejam endereçados mais rapidamente do que era possível anteriormente.

Dada a natureza das mudanças organizacionais necessárias, nem toda organização está preparada para adotar o DevOps, seja por falta de apoio, seja por considerar outras prioridades. Outro ponto que pode ser um inibidor inicial é o receio de que a agilidade não combine com regulações, por exemplo. Apesar da preocupação em si ser válida, é possível aliar as duas perspectivas.

A conhecida empresa de consultoria Gartner propõe três passos para a adoção inicial, bem como adoção continua do DevOps, sempre em alinhamento com requisitos regulatórios. Tais passos podem ser vistos na figura a seguir.



 Possíveis Aspectos a Serem Cobertos por Ferramentas em um ambiente DevOps. Fonte: Autor.

Cases de sucesso

 Clique no botão acima.

Vejamos agora alguns casos de sucesso na adoção do DevOps.

Enquanto o [Walmart](#) é o rei das empresas de venda baseadas em lojas físicas nos Estados Unidos, no mundo online a empresa sempre esteve às sombras da Amazon. Visando ganhar espaço, o Walmart montou um time avançado por meio de várias aquisições, fundando o [WalmartLabs](#), que nada mais é do que o braço tecnológico, de inovação e desenvolvimento da empresa.

Uma das abordagens chave da missão do WalmartLabs é o uso do DevOps. Ele incorporou a tecnologia OneOps baseada na nuvem, que automatiza e acelera a implantação de aplicações. O WalmartLabs também criou várias ferramentas abertas, como o [Hapi](#), um *framework* baseado em Node.js para construção de aplicações e serviços que permitem aos desenvolvedores focarem em escrever lógica reutilizável de aplicações ao invés de gastarem tempo com infraestrutura. Mais recentemente, implantou mais de 100,000 cores OpenStack de forma a construir e dispor de sua própria nuvem privada, continuando a evoluir sua abordagem ágil.

Quando a Netflix evoluiu seu modelo de negócio do envio de DVDs para streaming de vídeo, havia muitas dúvidas. Não existiam ferramentas comerciais disponíveis para ajudar a massiva infraestrutura em rede da empresa a rodar de forma suave. Assim, a Netflix decidiu se voltar para ferramentas open source. A empresa tomou a iniciativa de alistar a ajuda voluntária de centenas de desenvolvedores, criando com essa ajuda o “Exército Sírio”, ou uma suíte de ferramentas automatizadas que testava a infraestrutura disponível, permitindo que a empresa fosse pró-ativa ao identificar e resolver vulnerabilidades antes que os clientes fossem impactados.

Desde então a Netflix continua a investir em automação e open source, e hoje, com o avanço de sua cultura DevOps, seus desenvolvedores integram e implantam código milhares de vezes ao dia.

A Fidelity Investments possuía várias unidades de negócio desenvolvendo software, além de muitas aplicações e processos legados que demandavam muito das equipes. As implantações eram manuais em centenas de servidores, gerando erros frequentes. Quando foi necessário desenvolver uma aplicação crítica de suporte a negociações sob um cronograma fixo, a organização aproveitou para adotar o DevOps e implementar um framework automatizado de liberação de software.

Tal solução resultou em mais de US\$2.3 milhões por ano em economia de custos somente para o aplicativo que precisava ser desenvolvido. Desde então, dezenas de outros aplicativos tiveram suas liberações automatizadas, reduzindo o tempo necessário de três dias para uma ou duas horas, além de reduzir o downtime de teste. O processo também ajuda com as questões regulatórias, além de dar maior visibilidade às partes interessadas sobre os cronogramas de liberação.

Atenção! Aqui existe uma videoaula, acesso pelo conteúdo online

Vantagens e Desvantagens do DevOps

A adoção do DevOps pode por si só ser intimidadora. Com tanta coisa envolvida, incluindo tecnologias e perfis profissionais diversos, decisões, itens a serem priorizados, apoio institucional, apenas para citar alguns fatores, é compreensível que muitas organizações ainda não tenham dado os passos necessários para que esta adoção se concretize.

Em alguns dos casos, o incentivo para tomada de decisão pode vir de uma análise cuidadosa das vantagens e desvantagens do DevOps para o contexto da organização.

A seguir, apresentamos uma lista não-exaustiva das vantagens e desvantagens conhecidas trazidas pelo DevOps.

Vantagens

- Redução no tempo necessário para o desenvolvimento: Com a aproximação entre desenvolvimento e operações, podemos esperar menores tempos necessários para aceitação e início do uso efetivo do software, tornando todo o processo mais econômico em termos de custos de produção.
 - O tempo menor, bem como os custos menores, dependendo da situação (se a criação do software se destina ao consumo externo à organização), pode acabar por levar a uma fatia maior de participação no mercado.

- Redução no número de profissionais e custos associados: Algumas estatísticas demonstram que é possível atingir uma redução de até 35% no número de profissionais de TI necessários para suportar o desenvolvimento e operações em organizações que usam DevOps.
 - Isto é possível pelo forte uso da automação, bem como pela integração maior entre as equipes, reduzindo ruídos de comunicação e tornando o ciclo menos sujeito a erros. A redução geral em custos fica em torno de 30%.
- Possível endereçamento de questões relacionadas à segurança: Dado que considerações acerca de segurança sejam trazidas para dentro do ciclo de desenvolvimento do produto, por meio da inclusão da integração e automação de práticas de segurança e procedimentos relacionados, organizações mais maduras fazem uso do chamado “DevSecOps”, ou seja, há integração e diálogo constantes entre desenvolvimento, segurança e operações.
- Auto-responsabilização dos membros da equipe: Já que estes membros são incentivados a “empoderarem-se”, tomando para si mais responsabilidade sobre o produto, aumenta a capacidade da equipe como um todo a abraçar novas tecnologias, pois, ao interagirem mais com outros profissionais de outras equipes, tendem a absorver novos conhecimentos e se sentirem propelidos a inovar.

Desvantagens

- Problemas relacionados à segurança: Este risco advém principalmente das situações em que as equipes de segurança não estão integradas ao grupo de DevOps, carecendo portanto de visibilidade sobre as abordagens usadas para acelerar o processo de desenvolvimento.
- Mudança cultural: O DevOps, no momento em que passa a ser adotado, requer grandes mudanças e adaptações na mentalidade e cultura da organização e dos funcionários. São mudanças que mexem com o status quo, muitas vezes construído ao longo da existência da organização e/ou da carreira destes indivíduos.
 - Logo, não são mudanças fáceis de implementar. Resistências são esperadas, é preciso haver todo um trabalho de convencimento, se possível, baseado em fatos e números, para que a mudança de paradigma seja facilitada.
- Necessidade em lidar com experiências individuais históricas: A ligação entre DevOps e frameworks ágeis é inegável, e ambas privilegiam o emprego de profissionais com perfil autônomo e que tomam a iniciativa.
 - Infelizmente, muitos profissionais vêm de situações anteriores que não privilegiavam este comportamento, logo eles têm dificuldade em entender que não existe mais separação entre desenvolvimento e operações.
 - Ainda tendem a ver certas atividades como sendo Dev e outras atividades como sendo Ops.
 - Esta linha de raciocínio não funciona no DevOps pleno, pois, ao lidar com um problema, o profissional precisa fazer sua própria investigação, já que não existe uma equipe de suporte separada.
 - No máximo, o ticket existirá para dar suporte ao DevOps ao indicar e descrever toda a investigação que foi feita pelo próprio profissional, devendo o problema ser escalado horizontalmente (ou até mesmo verticalmente) apenas quando o profissional em questão realmente não possuir condições técnicas de endereçá-lo.


Os papéis relacionados ao DevOps

É importante diferenciar funções de papéis. Enquanto funções normalmente são formais, sendo o que define a relação da organização com o funcionário, papéis podem ser formais ou informais, sendo que um profissional com uma função específica pode ser revestido de um dado número de papéis.

A seguir, apresentaremos papéis que, no geral, fazem sentido à organizações que abraçam DevOps.

É possível, portanto, que um profissional vista o chapéu de dois ou mais destes papéis, a depender de vários fatores, tais como a forma como a organização está estruturada, ou mesmo o número de profissionais envolvidos. Estes papéis também não podem ser considerados “canônicos”, ou seja, universalmente aceitos.

Como o DevOps em si é um *framework*, cada organização irá adotar os papéis individuais, as combinações de papéis e a denominação de papéis que fizerem sentido.

 Clique nos botões para ver as informações.

[Evangelista](#)



A adoção do DevOps requer planejamento, supervisão e promoção. Mudanças de hábito são necessárias, o que não irá se realizar a não ser que haja visibilidade clara dos benefícios oriundos dessas mudanças.

É aqui que o evangelista entra. Ele pode trabalhar como agente dessas mudanças para obter apoio das equipes de desenvolvimento e operação.

A pessoa que ocupar este papel pode identificar novos papéis que irão suportar a entrega e assegurar que os membros do time tenham acesso aos conhecimentos e recursos que eles precisam para completar a transição para uma cultura DevOps bem sucedida.

[Gerente de Liberação](#)



Trabalham para coordenar e gerenciar o produto à medida que ele se move de desenvolvimento até a produção. Preocupados com a criação do produto, estão ainda mais focados em sua movimentação por meio do desenvolvimento, teste e implantação, de maneira que possam suportar a entrega contínua.

Isto significa que, por vezes, irão trabalhar em detalhes técnicos com os quais gerentes de projeto não se envolvem.

[Arquiteto de Automação](#)



Também chamados de especialistas em integração, focam principalmente no desenho e implementação dos sistemas automatizados que são centrais ao componente da entrega contínua do DevOps.

Estes arquitetos desenham e implementam os sistemas automatizados que todos os membros da equipe DevOps irão usar durante o desenvolvimento e a produção.

[Desenvolvedor/Testador](#)



É um papel expandido, já que não só existe codificação envolvida; no DevOps devem participar de testes unitários, implantação e monitoração contínua do produto. Esta mudança de mentalidade é construída sobre automações que fazem boa parte do papel pesado, mantendo ainda a agilidade esperada.



Uma evolução da função da Garantia da Qualidade. Os Profissionais da Garantia da Experiência procuram focar nas características e funcionalidades de novos produtos sempre sob a perspectiva da experiência do usuário.

Ao invés de simplesmente garantirem que algo funciona, procuram encontrar maneiras de otimizar e melhorar a experiência do usuário.

Engenheiro de Segurança



Ao invés de ser uma preocupação secundária, no DevOps procura-se tratá-la como preocupação primária, incorporando seus elementos desde a concepção do produto.

O DevSecOps, como alguns preferem chamar esta prática, assegura que engenheiros de segurança trabalhem em conjunto com todo o time DevOps, integrando melhor estes princípios.

Atenção! Aqui existe uma videoaula, acesso pelo conteúdo online

Atividade

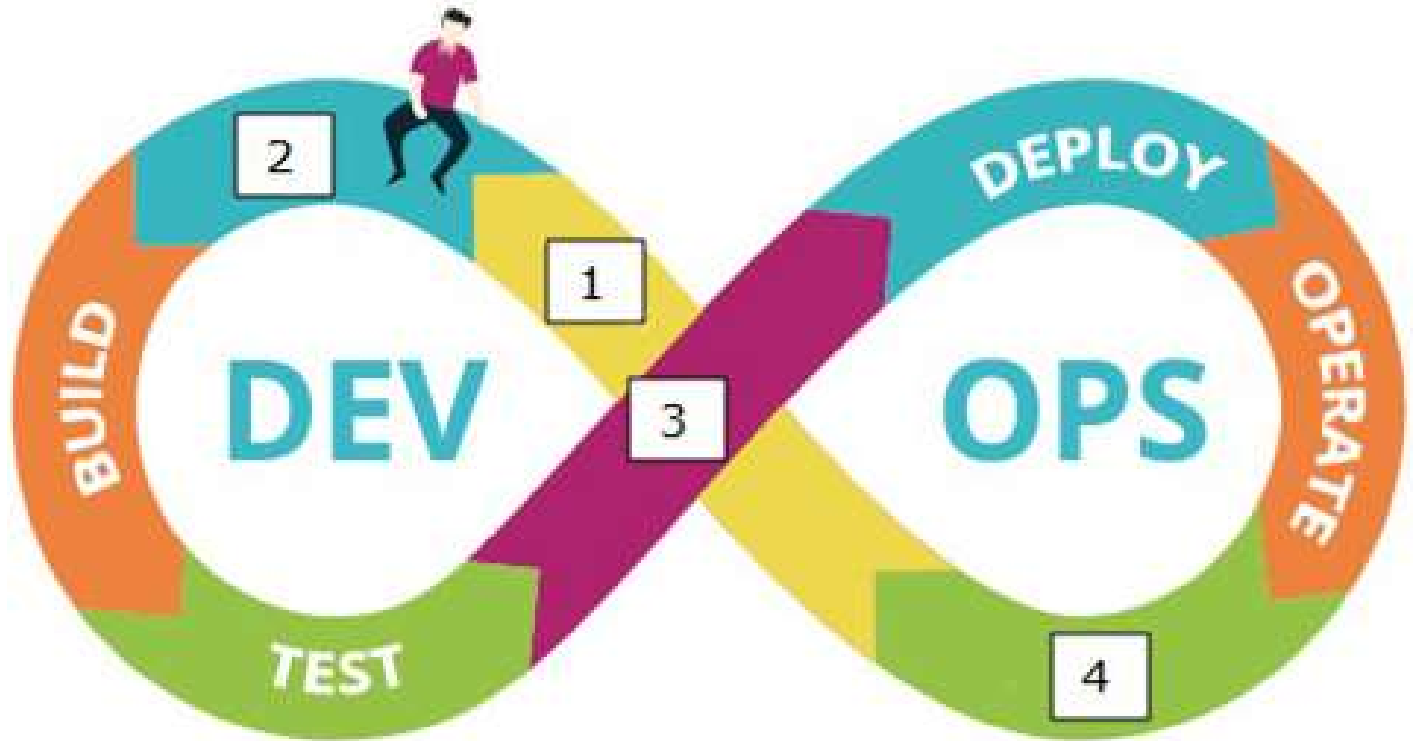
1. Qual a ligação entre DevOps e *Frameworks* Ágeis?

- a) Os *frameworks* ágeis se originaram no DevOps.
- b) Os *frameworks* ágeis são o DevOps.
- c) O DevOps se originou nos *frameworks* ágeis.
- d) O DevOps é um *framework* ágil.
- e) Não há relação.

2. Qual indústria se encontra mais avançada em termos de adoção do DevOps?

- a) Manufatura.
- b) E-commerce.
- c) Saúde.
- d) Tecnologia.
- e) Finanças.

3. Os itens que completam o ciclo de DevOps são, respectivamente:



- a) Prepare, Code, Quality Assurance, Monitor.
- b) Plan, Code, Release, Monitor.
- c) Plan, Develop, Release, Monitor.
- d) Proceed, Develop, Quality Assurance, Support.
- e) Proceed, Develop, Release, Support.

4. Em geral, _____ e _____ são _____ pela adoção do DevOps:

Objetivo: Identificar os papéis envolvidos e vantagens e desvantagens do DevOps

- a) Problemas de segurança, liberações, aumentados.
- b) Custos, problemas de segurança, reduzidos.
- c) Número de profissionais, mudanças, aumentados.
- d) Mudanças, liberações, reduzidos.
- e) Número de profissionais, custos, reduzidos.

5. Qual dos papéis abaixo recebeu uma evolução direta com o DevOps?

- a) Auditor de Configuração.
- b) Dono do Processo de Gerência de Configuração.
- c) Gerente de Projeto.
- d) Analista de Qualidade.
- e) Analista de Item de Configuração.

Notas

Mudança emergencial¹

As mudanças emergenciais são basicamente o oposto das mudanças padrão: Geralmente, mais do que se caracterizarem por trazerem riscos médios a altos, exigem implementação o mais rapidamente possível, de forma que também possam ser operacionalizadas com grande rapidez. São, portanto, mudanças para “situações de crise”, e que requerem procedimentos à altura.

Referências

AIELLO, B. **Configuration Management Best Practices**. 1.ed. Pearson, 2013.

BOURQUE, P.; FAIRLEY, R. **Software Engineering Body of Knowledge (SWEBOK)**. 3.ed. IEEE Computer Society, 2017.

HAAS, J. **Configuration Management Principles and Practice**. 1.ed. Addison Wesley, 2003.

REED, M. **DevOps**: The Ultimate Beginners Guide to Learn DevOps Step-by Step. 1.ed., Produção Independente, 2019.

Próxima aula

- Visão geral de Inteligência Artificial;
- O emprego da AI na Gerência de Configuração;
- Benefícios da AI na Gerência de Configuração.

Explore mais

Assista o vídeo sobre DevOps do canal Dicionário do Programador.