**Atividade 2 - DevOps**

O DevOps permite que funções anteriormente isoladas – desenvolvimento, operações de TI, engenharia da qualidade e segurança – atuem de forma coordenada e colaborativa para gerar produtos melhores e mais confiáveis. Ao adotar uma cultura de DevOps em conjunto com as práticas e ferramentas de DevOps, as equipes ganham a capacidade de responder melhor às necessidades dos clientes, aumentar a confiança nos aplicativos que constroem e cumprir as metas empresariais mais rapidamente. O DevOps influencia o ciclo de vida do aplicativo em todas as fases do planejamento, do desenvolvimento, da entrega e da operação. Cada fase depende das demais e elas não são específicas da função. Em uma verdadeira cultura de DevOps, cada função está envolvida de alguma forma em cada fase.

O movimento DevOps começou por volta de 2007, quando comunidades de desenvolvimento de software e operações de TI manifestaram preocupações relacionadas ao modelo de desenvolvimento de software tradicional, em que os desenvolvedores que escreviam os códigos trabalhavam separados das operações que implementavam e prestavam suporte a esses códigos. O termo DevOps, uma combinação das palavras desenvolvimento e operações, reflete a integração dessas disciplinas em um processo contínuo.

O software e a Internet transformaram o mundo e seus mercados, do comércio ao entretenimento e ao sistema bancário. O software já não apenas sustenta uma atividade empresarial, na verdade ele tornou-se um componente integral de cada parte de uma empresa. As empresas interagem com seus clientes por meio de softwares disponibilizados como serviços ou aplicativos online, e em todos os tipos de dispositivos. Elas também usam o software para aumentar a eficiência operacional ao transformar cada parte da cadeia de valor, como logística, comunicação e operações. Similarmente a como essas empresas de bens materiais transformaram como elas projetavam, criavam e disponibilizavam produtos usando a automação industrial durante o século 20, as empresas de hoje devem transformar como elas criam e disponibilizam software.

A transição para o DevOps exige uma mudança de cultura e mentalidade. Em seu nível mais simples, o objetivo do DevOps é remover as barreiras entre duas equipes tradicionalmente separadas em silos: desenvolvimento e operações. Em algumas empresas, podem até não existir equipes de desenvolvimento e operações separadas, os engenheiros se encarregam de ambas. Com o DevOps, as duas equipes trabalham juntas para otimizar a produtividade dos desenvolvedores e a confiabilidade das operações. Elas se esforçam para manter a comunicação frequente, aumentar a eficiência e melhorar a qualidade dos serviços disponibilizados aos clientes. As equipes assumem a responsabilidade total sobre os serviços, geralmente indo além do escopo tradicional de seus cargos e títulos definidos, pensando sobre as necessidades do cliente final e como eles podem contribuir para resolver essas necessidades. As equipes de controle de qualidade e segurança também podem tornar-se altamente integradas a essas equipes. As empresas usando um modelo de DevOps, independentemente de sua estrutura organizacional, têm equipes que supervisionam o ciclo de vida completo de desenvolvimento e infraestrutura como parte das suas responsabilidades.

É difícil mudar um hábito. As equipes acostumadas com formas de trabalho fragmentadas podem ter dificuldade ou até resistir à reformulação das estruturas da equipe para a adoção de práticas de DevOps. Algumas equipes podem acreditar na ideia equivocada de que novas ferramentas são suficientes para a adoção do DevOps. No entanto, o DevOps é uma combinação de pessoas, ferramentas e cultura. Todos os membros de uma equipe de DevOps precisam entender o fluxo de valor como um todo, da concepção ao desenvolvimento e à experiência do usuário final. É necessário acabar com a fragmentação para colaborar durante todo o ciclo de vida do produto.

O ciclo de vida do DevOps foi projetado para otimizar a entrega rápida de software de alta qualidade. Ele inclui uma série de fluxos de trabalho iterativos e automatizados, que fazem parte de um ciclo de desenvolvimento também automatizado e iterativo.Os nomes e a ordem dos fluxos de trabalho podem variar entre organizações, mas o ciclo de vida do DevOps geralmente segue oito etapas principais.

Primeiro, as equipes definem as novas funcionalidades e funções da próxima versão. Durante esse fluxo de trabalho, as equipes utilizam feedbacks de usuários, estudos de caso e inputs de stakeholders internos como engenheiros de plataforma e infraestrutura, segurança, conformidade, governança, gerenciamento de riscos e equipes de negócios.O objetivo do estágio de planejamento é criar um documento de lista de pendências. O backlog é uma lista priorizada de novas funcionalidades, melhorias e correções de erros que serão adicionadas ao produto ao longo do tempo.

A equipe de DevOps codifica as funcionalidades novas e aprimoradas identificadas no backlog. Práticas comuns de codificação em DevOps incluem:

* **Desenvolvimento orientado por testes (TDD):** os desenvolvedores criam primeiro os testes que o código precisa passar antes de escrevê-lo.
* **Programação em par:** dois programadores colaboram na mesma estação de trabalho; um escreve o código enquanto o outro o revisa.
* **Avaliações de código por pares**: os membros da equipe de desenvolvimento analisam o código uns dos outros em busca de bugs, erros ou áreas de melhoria.

Os desenvolvedores geralmente usam suas estações de trabalho locais para escrever e testar o código antes de enviá-lo para o próximo estágio do delivery pipeline.As equipes de DevOps usam testes, geralmente automatizados, para garantir que a aplicação atualizada atenda aos padrões e requisitos adequados.

O modelo clássico de DevOps inclui uma fase de testes específica que acontece entre a construção e a liberação. No entanto, o DevOps evoluiu a ponto de permitir que certos elementos de teste aconteçam ao longo de todo o processo. Os testes unitários, que verificam pequenos trechos de código de forma isolada, podem ser executados durante a fase de codificação. Depois de integrar um novo código, os programas de linting podem analisá-lo em busca de erros.  
  
Os testes contínuos ajudam a aplicar o princípio de shift-left testing, uma abordagem de desenvolvimento de software que prioriza a antecipação das atividades de teste no processo de desenvolvimento. Essa abordagem ajuda as organizações a identificar os problemas mais cedo e corrigi-los com mais eficiência.

**Colaboração**

A principal premissa por trás do DevOps é a colaboração. As equipes de desenvolvimento e operações se unem em uma equipe funcional que se comunica, compartilha feedback e colabora durante todo o ciclo de desenvolvimento e implementação. Muitas vezes, quer dizer que as equipes de desenvolvimento e operações se fundem em uma única equipe que trabalha em todo o ciclo de vida do aplicativo.

Os membros de uma equipe de DevOps são responsáveis por garantir entregas de qualidade em cada faceta do produto, levando a um desenvolvimento mais "full stack", em que as equipes têm as responsabilidades completas de back-end e front-end de uma função ou produto. As equipes vão ser as donas de uma função ou projeto durante todo o ciclo de vida, da ideia até a entrega. O nível aprimorado de investimento e atenção da equipe leva a uma produção de maior qualidade.

**Automação**

Uma prática essencial do DevOps é automatizar o máximo possível o ciclo de vida do desenvolvimento do software. dando aos desenvolvedores mais tempo para escrever código e desenvolver novas funções. A automação é um elemento crucial de um pipeline de integração/implementação contínuas e ajuda a reduzir erros humanos e aumentar a produtividade da equipe. Com processos automatizados, as equipes obtêm melhoria contínua com tempos de iteração curtos, o que permite responder com mais rapidez aos comentários dos clientes.

**Implementação continua**

A melhoria contínua foi estabelecida como um elemento básico das práticas ágeis, bem como da fabricação lean e Kata de melhoria. É a prática de se concentrar na experimentação, minimizar o desperdício e otimizar a velocidade, o custo e a facilidade de entrega. A melhoria contínua também está ligada à entrega contínua, permitindo que as equipes de DevOps enviem atualizações constantes que melhoram a eficiência dos sistemas de software. O pipeline constante de novas versões significa que as equipes promovem com consistência alterações de código que eliminam o desperdício, melhoram a eficiência do desenvolvimento e trazem mais valor ao cliente.

**Ação voltada o cliente**

As equipes de DevOps usam ciclos curtos de feedback com clientes e usuários finais para desenvolver produtos e serviços centrados nas necessidades do usuário. As práticas de DevOps permitem coleta e resposta rápidas ao feedback do usuário por meio do uso de monitoramento em tempo real e implementação rápida. As equipes obtêm visibilidade imediata de como os usuários ativos interagem com um sistema de software e usam os dados para desenvolver melhorias adicionais.

**Criar com o final em mente**

Esse princípio envolve entender as necessidades dos clientes e criar produtos ou serviços que resolvam problemas reais. As equipes não devem "criar uma bolha" ou um software com base em suposições sobre como os consumidores vão usar o software. Como alternativa, as equipes de DevOps devem ter uma compreensão holística do produto, da criação à implementação.

**Referencias:**

<https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-devops>

<https://www.atlassian.com/br/devops>

<https://aws.amazon.com/pt/devops/what-is-devops/>

<https://www.ibm.com/br-pt/think/topics/devops>