

UNIVERSIDADE FEDERAL DO PARANÁ

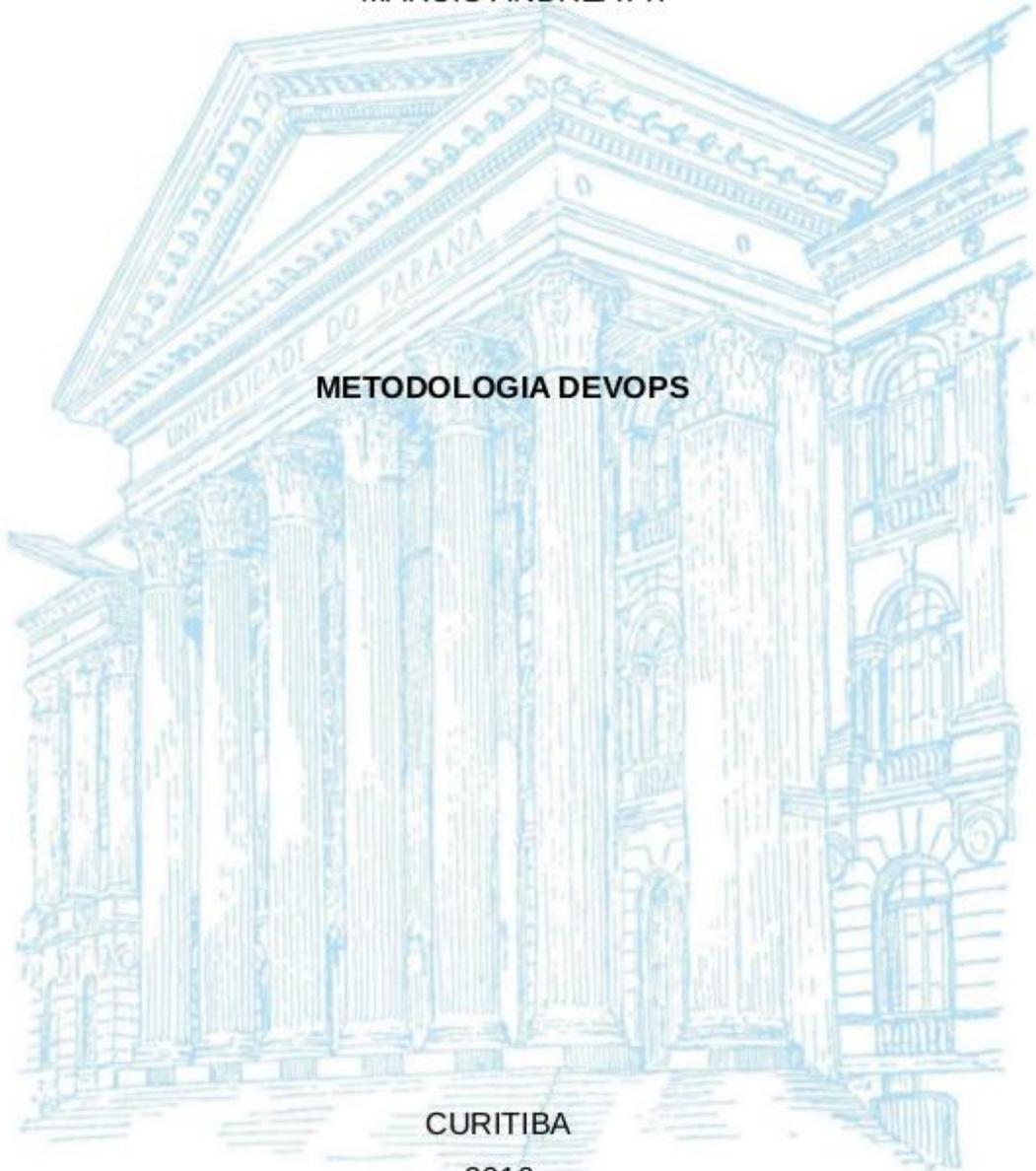
GUILHERME SECONDE BOZELLI

MÁRCIO ANDREATTI

METODOLOGIA DEVOPS

CURITIBA

2016



MÁRCIO ANDREATTI
GUILHERME SECONDE BOZELLI

METODOLOGIA DEVOPS

2016

MÁRCIO ANDREATTI
GUILHERME SECONDE BOZELLI

METODOLOGI DEVOPS

Dissertação apresentada como requisito parcial à obtenção do grau de Cientista da Computação, no Curso de Ciências da Computação, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Orientador: Prof. Setembrino Soares Ferreira Jr.

CURITIBA
2016

SUMÁRIO

- 1.Introdução:
- 2.Entendendo a abordagem DevOps e suas partes interessadas:
 - 2.1.Desenvolvimento:
 - 2.2.Operações:
 - 2.3.Negócios:
- 3.Devops e Agil:
- 4.A necessidade de DevOps
 - 4.1.DevOps: uma perspectiva
 - 4.2.Desenvolvimento contínuo da equipe Ops
 - 4.3.Dono do produto
 - 4.4.Product backlog
 - 4.5.Representação Ops na equipe
 - 4.6.Definição de pronto
 - 4.7.Planejamento do sprint e diariamente stand-up
 - 4.8.Avaliação Sprint
 - 4.9.Scrum de Scrums
 - 4.10.Alinhamento
 - 4.11.Métricas
 - 4.12.Em relação as características técnicas
 - 4.13.Em relação aos valores humanos
 - 4.14.Em relação a forma de trabalho
 - 4.15.Aplicando a cultura
 - 4.16.A realidade no ambiente Startup
 - 4.17.A realidade no ambiente corporativo
- 5.O especialista DevOps no meio corporativo
 - 5.1.Como esse especialista atua?
 - 5.1.1Pé no devel
 - 5.1.2.Pé na infra
- 6.Open Source DevOps
- 7.Conclusão

Resumo. Este texto tem por objetivo tratar de forma abrangente a filosofia de desenvolvimento conhecida como DevOps.

DevOps é um termo para um grupo de conceitos que, embora não seja totalmente novo, catalisaram em um movimento e estão rapidamente se espalhando por toda a comunidade técnica. Como qualquer novo e popular prazo, as pessoas têm impressões

um pouco confuso e por vezes contraditórias de que se trata. Aqui é a nossa opinião sobre como DevOps pode ser útil

Propomos esta definição como uma estrutura padrão para discutir com mais clareza as diversas questões DevOps. Como DevOps "Qualidade" ou "Agile", é um conceito suficientemente grande que requer alguma nuance de compreender plenamente.

DevOps é um termo novo que emerge da colisão de duas grandes tendências relacionadas. O primeiro também foi chamado de "administração ágil sistema" ou "operações ágeis"; que saltou de aplicação de abordagens mais recentes ágil e enxuta para o trabalho operações. A segunda é uma compreensão muito mais expandida do valor da colaboração entre desenvolvimento e equipe de operações em todas as fases do ciclo de vida de desenvolvimento ao criar e operar um serviço, e como operações importantes tornou-se no nosso mundo cada vez mais orientada a serviços.

Abstract. *This text aims to address comprehensively the development of philosophy known as DevOps.*

DevOps is a term for a group of concepts that, although not entirely new, catalysed in a move and are rapidly spreading throughout the technical community. Like any new and popular term, people have impressions a bit confusing and sometimes contradictory that it is. Here is our opinion about DevOps can be useful

We propose this definition as a standard framework to discuss more clearly the various DevOps issues. How DevOps "Quality" or "Agile" is a large enough concept that requires some nuance to understand fully.

DevOps is a new term that emerges from the collision of two major trends related. The first was also called "agile management system" or "agile operations"; which jumped from applying more recent approaches agile and lean to work operations. The second is a much more expanded understanding of the collaboration between development value and operations team in all phases of the development life cycle to create and operate a service, and how important operations has become in our world increasingly oriented services.

DEVOPS

1. Introdução:

Seja em empresas de pequeno, médio ou grande porte, a necessidade de automatizar a operação para aumentar a produtividade é uma constante. E isso independe do segmento de atuação. No objetivo de proporcionar ganhos em escala, reduzindo custos operacionais, tempo gasto em processos, a dependência de setores para com o departamento de tecnologia da informação, a tecnologia DevOps vem revolucionando muitas empresas com processos antes demorados e complicados, com simples apertar de botões. O universo analógico já sucumbiu, e processos digitais enfrentam mudanças permanentemente.

O peso e a importância das aplicações legadas que ainda não rodam em plataforma de cloud computing de forma inteligente é enorme, sem contar no risco e investimento de se alterar tais aplicações para migrar para nuvem. É exatamente nesse contexto, e para suprir essas necessidades, que o DevOps vem ganhando espaço.

No contexto do cloud computing, o DevOps se tornou uma ferramenta que traz economia de tempo e ganho em escala de serviços. A ideia construída com isso é automatizar rotinas de operação para que seja consumidos recursos de cloud computing em escala, tirando a dependência de pessoas no processo.

Devops é um conjunto de principio e valores que visa facilitar a coloboração entre disciplinas, para possibilitar a rápida evolução de implantação de serviços para reduzir riscos e custos aumentando a qualidade em todo portfólio.

Esta tecnica é a chave para implantação em um tripe onde processos, pessoas e ferramentas, trabalham em conjunto.

Seguindo a linha utilizando as praticas ágeis e filosofia Lean (semelhante ao TOYOTISMO).

A diferença entre o scrum conceitual comum e o devops, é que o devops engloba todas as áreas necessárias para a especificação, desenvolvimento, implantação e manutenção. Todas as áreas se tornam uma só, tendo a comunicação direta entre as mesmas.

O ciclo de vida DevOps é regido por 5 princípios básicos que podem ser lembrados pelo acrônimo **CALMS** (em inglês):

- Culture (Cultura)
- Automation (Automação)
- Lean (Enxuto)
- Measurement (Medição)
- Sharing (Compartilhamento)

2. Entendendo a abordagem DevOps e suas partes interessadas:

Geralmente combina com a abordagem ágil de fazer mudanças rápidas e pequenas para garantir o foco no trabalho de maior valor, minimizar o risco de defeitos associados com mudanças grandes e minimizar a mudança de valor frequentemente associada com grandes lacunas entre um caso de negócios de software e a conclusão de um projeto. A visão de mundo de operações, por outro lado, é tratar a mudança como uma causa de instabilidade. A instabilidade pode, obviamente, levar a tempo de inatividade, que é a principal ocorrência negativa nas disciplinas de operações. Por isso, as organizações de operações geralmente resistem a mudanças.

No entanto, DevOps é geralmente visto como um movimento de mudança, pois, na maioria das organizações, desenvolvimento, operações e negócios têm uma série de comportamentos e recompensas em conflito, que levam a incontáveis problemas de comportamento. Os negócios recompensam o desenvolvimento por novos recursos e mudanças, mas penalizam operações pelo tempo de inatividade. Dessa forma, o desenvolvimento empurra operações para mais implementações, e operações empurra desenvolvimento para mais estrutura e rigor nos artefatos entregues. A abordagem DevOps procura eliminar essas barreiras, nivelar o campo de atuação entre os grupos concorrentes e fazer todos concentrarem-se em um objetivo em comum.

2.1. Desenvolvimento:

Desenvolvimento é visto como tendo avançado mais em seus esforços para tornar-se mais responsivo às necessidades de negócios. O Manifesto Ágil já tem mais de 10 anos. O próprio manifesto é, em parte, o resultado de abordagens e experimentos anteriores de programação extrema e programação de pares. Para ser justo, a parte de software do quebra-cabeça era vista como um objetivo fácil de atingir -- facilmente alterado e, teoricamente, isolado dos problemas da infraestrutura e plataforma. E infraestrutura tem sido tratada, tradicionalmente, como um gasto enorme de capital, com longos ciclos de amortização, muito mais difícil de alterar. Infelizmente, software sofisticado depende muito da infraestrutura e exige que ela evolua no mesmo ritmo que o próprio software. A noção de manter software e infraestrutura separados não é sustentável se a tecnologia tiver que manter o ritmo com as necessidades de negócios. Isso está forçando o desenvolvimento a engajar-se mais nas necessidades para sustentar infraestruturas de software complexas em produção.

2.2. Operações:

Felizmente, algumas novas tecnologias e técnicas surgiram em operações para ajudá-la a tornar-se mais responsiva. A principal tecnologia inovadora no reino das operações é

a ampla disponibilidade de virtualização barata em hardware de mercadoria. Isso deu origem a novas abordagens de gerenciamento de sistemas e, é claro, à computação em nuvem. Essa tecnologia obteve popularidade porque proporcionava valor imediato ao permitir que organizações consolidassem, rápida e facilmente, seus recursos de cálculo subutilizados.

A consolidação simples, no entanto, é apenas uma medida de corte de custos, com um índice finito de captura de valor. A tecnologia de virtualização também permite um nível de mutabilidade na infraestrutura que permite que as operações avancem-na sem afetar a estabilidade, em níveis que antes não eram possíveis. As técnicas usadas para explorar esses recursos de virtualização geralmente aparecem dentro das disciplinas relacionadas à computação em nuvem. Essas técnicas emergentes deram às operações uma maneira de ser tão ágeis quanto o desenvolvimento e de atender aos requisitos de responsividade dos negócios.

2.3.Negócios:

Os negócios, por sua parte, descobriram que entender e explorar tecnologia é mais crítico do que nunca para obter resultados. De acordo com a Pesquisa de Opinião de CEOs 2012 da IBM, que se baseou em entrevistas com mais de 1.700 executivos chefes em 64 países, "fatores de tecnologia" são o principal fator que afeta organizações. Isso tem subido desde 2004, quando ocupada apenas a sexta posição dentre os nove fatores listados.

Portanto, os líderes de negócios perceberam que a capacidade de responder às necessidades de seus clientes é uma vantagem competitiva. A consequência, obviamente, é que a execução tecnológica precária representa uma ameaça intrínseca aos negócios. Essa não é uma percepção súbita, mas atingiu um ponto de mudança. A mesma pesquisa discutiu como os respondentes deram um peso igual (sete de dez) para entender o que clientes individuais queriam e ser capaz de responder a eles com prazos de lançamento nos mercados menores. Em última instância, essa é a pressão de negócios colidindo com as realidades processuais de como as disciplinas técnicas de desenvolvimento e operações operaram no passado e estimularam a discussão de como fazer melhor.

3.Devops e Agil:

DevOps e Agil se complementam para implementar a funcionalidade de trabalho em produção mais rápido.

Em uma organização de TI, a equipe de operações (PO) fornece a interface direta entre a TI e os usuários de negócios / fim. Muitas vezes, é o primeiro ponto de contato para a comunidade de usuários finais. Uma organização DevOps normalmente incluem o grupo que assume as "deployables" concluídos da equipe de desenvolvimento (Dev) e coloca-os em produção, bem como a equipe que fornece suporte "Nível-1" para o negócio.

A equipe DevOps tem que ser eficiente e experiente para que eles possam prestar serviço de qualidade para o negócio - eficiente do ponto de vista de liberar os executáveis à produção rapidamente, e experiente em termos de ser capaz de compreender as consultas do usuário e fornecer a resposta certa.

4.A necessidade de DevOps

Muitas vezes, quando a equipe DevOps cria a solução, os requisitos funcionais recebem atenção completa, mas os requisitos de implementação e suporte não são abrangentes. Isto leva a surpresas durante a implantação de aplicações, suporte de produção e recuperação de desastres.

Em um cenário de Agil, a equipe de desenvolvimento produz a funcionalidade de trabalhar no final de cada sprint. No entanto, a funcionalidade concluída teria que esperar até que a data de lançamento chegue. Mesmo na data de lançamento, se a equipe DevOps não estiver preparada para a integração e implantação ou negócio não está pronto para conhecer a nova funcionalidade, não haverá liberação em atrasos. Menor tempo de mercado - um dos principais benefícios do Agil - não é plenamente realizado.

4.1.DevOps: uma perspectiva

DevOps é uma filosofia em que as equipes de negócios, as equipes de desenvolvimento e as equipe de operações trabalham unidas para colaborar em uma base contínua e se certificar de que as soluções de TI estarão disponíveis para as áreas usuárias em tempo, e que eles sejam executados sem interrupção. A filosofia foca em automação, colaboração, mudança cultural, e uma estrutura organizacional que é menos complexo e é fácil de navegar. Ela aborda as pessoas, processos e ferramentas, bem como as dimensões tecnológicas necessárias para garantir essa colaboração e sincronizar às diferentes partes interessadas para mover funcionalidade a produção mais rápida.

DevOps permite a realização dos benefícios de uma entrega mais rápida de funcionalidade alcançado através do Agil. As seções seguintes descrevem as considerações-chave para esta habilitação:

4.2.Desenvolvimento contínuo da equipe Ops

Um requisito fundamental do DevOps é que a equipe Ops está continuamente envolvida com a equipe de desenvolvimento em todo o ciclo de vida de desenvolvimento de soluções. Ops devem participar desde a fase de visão para compreender a visão de negócio, os épicos, e as linhas de tempo de liberação. Eles também devem contribuir para determinar a viabilidade técnica e cronograma da solução.

Desde a fase de visionamento através da fase de desenvolvimento, a equipe de operações deve fornecer os insumos necessários para a equipe de desenvolvimento para que eles possam construir e validar os requisitos relacionados-Ops. Um modelo ilustrativo de envolvimento e alguns exemplos de entradas Ops são mostrados abaixo.

4.3.Dono do produto

O proprietário do produto (PO) é o cara do negócio para a equipe de desenvolvimento. O PO tem uma visão para o produto e tem uma visão completa sobre os requisitos funcionais que devem ser cumpridas. No entanto, se os requisitos não funcionais (NFRs) não são bem compreendidos e articulada pelo PO, a equipe de desenvolvimento não será capaz de levá-los em conta ao criar a arquitetura e construção da solução final.

A liderança da equipe de TI e de negócios devem equipar o PO com a valorização básica de NFRs, juntamente com os requisitos relacionados com aspectos técnicos, tais como o seguinte:

- Plataformas de implementação e suporte
- A sua disponibilidade e limitações
- Dependência de parceiros fornecedores sobre a manutenção da infra-estrutura
- Interfaces de terceiros / aplicações necessários para as soluções finais

O PO não é esperado para ser um especialista nestas áreas, mas deve ser capaz de prever esses requisitos e comunicá-los às partes interessadas relevantes na TI e do negócio.

4.4.Product backlog

Normalmente, um product backlog concentra-se em histórias relacionadas com os requisitos funcionais. O PO e a equipe Dev são bem treinados para brainstorm e documentar os requisitos funcionais. No entanto, muitas vezes os NFRs não são bem especificados no backlog. Além dos requisitos funcionais, o atraso deve descrever itens tais como a seguinte:

- Requisitos de desempenho
- requisitos de tecnologia relacionados à implantação e suporte
- Requisitos para desenvolver as diretrizes para a reversão rápida e rolar para a frente
- Requisitos de segurança / firewall

4.5.Representação Ops na equipe

A equipe de desenvolvimento Agile é multi-funcional e auto-organizada. Então [e possível incluir uma pessoa Ops também? Talvez - isso depende de quanto qualificados os membros da equipe são. Normalmente em uma organização de TI start-up, alguns dos desenvolvedores ou testadores também seria responsável pela implantação e suporte de Nível 3 (correção de bugs). Nesses casos, os requisitos relativos à implantação e suporte seria bem discutido em reuniões de planejamento e revisão.

No entanto, em grandes organizações, as operações que precisam de uma grande equipe dedicada para assumir o código completo de várias equipes Dev e implantá-las. Em tais casos, a rotação de trabalho poderia ser uma proposta útil. Ou seja, alguns dos desenvolvedores poderia desempenhar o papel Ops para determinado período de tempo, e alguns dos membros da equipe Ops com aptidão poderia ser parte da equipe de Dev por um determinado período de tempo. Desta forma, o lado Ops seria bem representados durante o ciclo de desenvolvimento.

4.6.Definição de pronto

Outra alavanca fundamental para envolver Ops no ciclo de desenvolvimento é introduzi-lá em aspectos relacionados-Ops na definição de pronto. Junto com o padrão de codificação, testes e elementos de documentação, validação do código na plataforma de implementação (por exemplo, uma caixa de produção simulada), instruções específicas de apoio como parte da documentação, e uma corrida seca destas instruções

também devem ser incluídos na definição de pronto. Aqui, novamente, as entradas da equipe Ops são cruciais.

4.7.Planejamento do sprint e diariamente stand-up

Planejamento do sprint backlog e stand-ups diárias devem prestar atenção aos Ops priorizando itens do backlog e discutir o progresso. O sprint backlog deve incluir linhas de itens específicos relacionados com a segurança das plataformas de tecnologias necessárias para a implantação de simulação e outras atividades de coordenação. É uma boa idéia incluir a equipe de operações durante o planejamento do sprint e em stand-ups diários selecionados, onde a equipe iria discutir aspectos Ops. Qualquer dependência de fornecedores de infra-estrutura e integradores de sistemas devem ser consideradas, nesta fase, utilizando a entrada da equipe Ops.

4.8.Avaliação Sprint

A equipe de Dev deve demonstrar as características relacionadas-Ops da solução. Claramente, todos os Sprint comentados podem não incluir recursos OPS-relacionados. No entanto, se a equipe Ops faz parte das demos, eles têm a oportunidade de ver o que está vindo acima e fornecer insumos para os sprints subsequentes para melhorar o produto e incluir requisitos Ops também. Aqui, novamente, se um ou mais dos membros da equipe Dev representam Ops, é fácil de obter este alinhamento.

4.9.Scrum de Scrums

Quando várias equipes trabalhar em uma solução, a integração da saída de cada equipa tem de ser cuidadosamente planejada e executada. Cada equipe Scrum deve ter em conta os requisitos Ops e construir recursos em alinhamento com os requisitos. Os proprietários do produto devem ter uma visão do produto final, como ele será desenvolvido através de várias equipes, e onde e como ele será implantado. Eles devem envolver as equipes Ops para fornecer insumos específicos para cada equipe Scrum.

4.10.Alinhamento

DevOps e Agil se complementam bem em ajudar a empresa e as equipes a planejar o calendário de divulgação anual. Com o envolvimento contínuo e colaboração com a equipe de desenvolvimento, a equipe Ops fica sabendo quais funcionalidades estarão saindo. Com essa percepção, e usando o padrão de sprint conclusão, o Dev e equipes Ops deve ser capazes de prever com razoável precisão as potenciais datas de lançamento. Eventualmente, eles devem se esforçar para alinhar o cronograma de lançamento com os planos da Sprint. E quando este alinhamento acontece, a equipe de suporte seria capaz de mover a funcionalidade concluída a produção mais rápida e com intervalos mais curtos - e um dos principais benefícios do Agile é realizada!

4.11.Métricas

Para medir como DevOps ajudaria em lançamentos mais rápidos, um indicador de resultado poderia ser utilizado. Algumas estatísticas da indústria demonstram que as organizações usando DevOps e Agile são capazes de fazer vários lançamentos para a produção em um único dia.

- Lançamento percentual data de adesão
- Aumento percentual no número de lançamentos
- Tempo necessário para a liberação para produção
- Defeitos atribuíveis aos requisitos de plataforma / apoio
- Percentagem de NFRs conhecido

4.12.Em relação as características técnicas

Um ambiente DevOps deve ter/possuir/oferecer/permitir:

- Infraestrutura como código
- Orquestração de servidores
- Gerência de configurações
- Provisionamento dinâmico de ambientes
- Controle de versões compartilhado entre infra e devel
- Ambiente de desenvolvimento, teste e produção (no mínimo)
- O ambiente de devel deve possibilitar TDD
- Infra deve participar dos projetos desde o início
- Infra deve participar das reuniões de devel
- Devel deve participar das reuniões de infra
- Ambiente de entrega contínua
- Os desenvolvedores devem conseguir fazer o deploy sem interferência da infra

- Monitoramento eficaz com processamento adequado dos eventos e métricas
- Capacidade de resposta rápida a incidentes e problemas
- Backup e restore confiáveis

O devel precisa envolver a infra nos projetos desde o início - isso significa participar das reuniões técnicas ou SCRUM, afinal sem a infra não há projeto, e além disto, quanto mais problemas foram resolvidos durante o projeto - com ajuda da infra, menos problemas serão expostos aos clientes.

A infra também precisa observar quais são as metas da empresa a longo prazo, principalmente aquelas ligadas ao devel, pois enxergando onde o devel quer chegar, ela pode se programar melhor para ter certeza que a infraestrutura tecnológica estará preparada para atendê-los quando esse momento chegar.

A infra precisa envolver o devel em suas reuniões técnicas para que o devel entenda e tenha ciência da realidade da infra, assim eles vão conseguir enxergar suas qualidades, atribuições, planos de melhorias, atualizações programadas, agendas de manutenção, eles vão conhecer os recursos disponíveis e também descobrir as limitações da equipe, sejam elas técnicas ou materiais. Além disto, o devel pode ser um grande aliado da infra na solução de problemas, afinal o conhecimento que o devel traz pode ajudá-los a melhorar a forma com que administram seu ambiente, tornando este processo mais eficiente.

O devel precisa adotar alguma metodologia de entrega ou desenvolvimento contínuo e a infra precisa entender esse processo para que juntos criem os ambientes com as ferramentas certas.

A infra precisa ceder um pouco e evoluir, precisa oferecer ao devel um ambiente adequado onde eles sejam o dono do produto, onde o devel consiga fazer todo o ciclo de desenvolvimento de forma direta, o devel precisa conseguir gerar e controlar o código, precisa fazer o commit com segurança, precisa fazer o build, testar o build, validar a aplicação e entregar a nova versão de forma transparente sem que para isso precise passar por um burocrático e engessado processo de mudança.

4.13. Em relação aos valores humanos

Para a adoção da cultura funcionar, a equipe precisa observar e exercitar os seguintes valores:

- Confiança no trabalho de sua equipe
- Respeito pessoal e profissional por todos da equipe
- Sinceridade sobre eventos e incidentes ocorridos
- Honestidade sobre as causas dos incidentes (não esconda nada da sua equipe)
- Entendimento de que o problema é responsabilidade de todos
- Entendimento de a solução é responsabilidade de todos

- Entendimento de que os resultados são o reflexo do trabalho de toda a equipe
- Comunicação efetiva e dinâmica
- Postura construtiva sempre
- Espírito de colaboração

4.14.Em relação a forma de trabalho

É recomendável que a equipe:

- Internalize e adapte métodos ágeis como KANBAN e SCRUM para seu dia-a-dia
- Aprofunde estudos em entrega contínua
- Aprofunde estudos em gerência de configurações e orquestração

4.15.Aplicando a cultura

Após observar as principais características deste movimento, normalmente pensamos em como aplicar isto em nosso meio. Para ajudar na reflexão vamos avaliar o meio startup e o meio corporativo.

4.16.A realidade no ambiente Startup

A cultura DevOps combina muito com startups, nestes locais normalmente já se trabalha desenvolvimento utilizando metodologias ágeis, foi inclusive neste nicho em que começaram a discutir infraestrutura ágil - a precursora do movimento devops, portanto, as pessoas deste meio conseguem absorver os conceitos e a cultura DevOps sem grandes dificuldades, eles conseguem compreender os preceitos de colaboração e feedback pois já fazem isto em seu dia-a-dia.

Quem está em uma startup não tem as amarras e vícios da corporação, este é um grande facilitador e não é necessário nenhum tipo de intervenção para internalizar a cultura, a partir do estímulo de um líder as pessoas começarão a estudar e aplicar DevOps naturalmente.

Na startup normalmente não existe divisões, departamentos, todos trabalham juntos e isso também é um facilitador, afinal não existem barreiras para se comunicar.

4.17.A realidade no ambiente corporativo

A corporação não funciona como a startup, lá existe burocracia e o uso vicioso de métodos ultrapassados, portanto não bastará o estímulo da alta hierarquia para que equipes de infra e desenvolvimento comecem a vivenciar a cultura DevOps, é necessário intervir cirurgicamente para conseguir internalizar a cultura DevOps.

Em resumo, você precisa trazer alguém - de fora - que conhece DevOPs para que esta pessoa passe a contaminar os demais.

Esse processo é lento, mas se o especialista tiver os meios e o apoio do alto escalão, mudanças fantásticas poderão ocorrer.

5.0 especialista DevOps no meio corporativo

Atua como um agente de mudanças, ele precisa contaminar as áreas e mostrar que a cultura DevOps funciona.

5.1. Como esse especialista atua?

Ele será a ponte entre as áreas de infra e devel, ele conhece a infra a fundo e entende de forma ampla processos de desenvolvimento ágil.

5.1.1. Pé no devel

Ele participa dos projetos de desenvolvimento desde o seu nascimento, seu foco é oferecer os recursos para os desenvolvedores trabalharem da forma mais eficiente, além disto, com sua ótica de infra ele toma todas as precauções para que os aspectos de segurança, monitoramento, eficiência e escalabilidade sejam observados desde o início do projeto.

O DevOps vai ainda estudar todo o processo de desenvolvimento e definir - em conjunto com o devel - as ferramentas que irão permitir um processo de desenvolvimento e entrega contínua. Após definir ele vai instalar e manter esse infra.

Alguns DevOps conseguem até avaliar o código do produto e enxergar problemas de performance, esse tipo de visão sistêmica e raciocínio rápido são diferenças importantes para uma entrega com mais qualidade.

5.1.2. Pé na infra

Na infra ele é o principal agente de mudanças, é ele quem vai puxar a fila para iniciar a implantação de uma infraestrutura ágil, ele domina as ferramentas de orquestração, gerência de configuração e provisionamento e vai usar esse conhecimento para que a equipe passe a trabalhar a infraestrutura como código.

Este profissional também vai ajudá-los a mudar seu comportamento e cultura, ele vai orientá-los nos métodos ágeis de execução de atividades, aqueles inspirados no SCRUM e KANBAN.

6. 8 Open Source DevOps

1. Nagios e Icinga

Monitoramento de infra-estrutura é um campo que tem tantas soluções do *Zabbix* para *Nagios* a dezenas de outras ferramentas de código aberto. Apesar do fato de que agora softwares muito mais novos no bloco, *Nagios* é uma solução de monitoramento veterano que é altamente eficaz devido à grande comunidade de colaboradores que criam plug-ins para a ferramenta. O *Nagios* não inclui todas as habilidades em torno da descoberta automática de novas instâncias e serviços, por isso houve a necessidade de contornar esses problemas com plugins da comunidade. Felizmente, não foi muito difícil, e *Nagios* funciona muito bem.

A *Icinga*, que foi originalmente criado como um *fork* do *Nagios*. Seus criadores pretendem elevar *Nagios* para o próximo nível com novos recursos e uma experiência de usuário moderna. Há um debate dentro da comunidade de código aberto sobre os méritos do *Nagios* e seu entendo, mas por enquanto vamos continuar a usar o *Nagios* pois estão satisfeitos com a sua escala e desempenho. A mudança para a nova tecnologia, como *Icinga*, pode ser apropriado no futuro à medida que progredi.

2. Monit

Às vezes, as ferramentas mais simples são as mais úteis, como comprovado pelo cão de guarda simples *Monit*. O seu papel é assegurar que qualquer processo em uma máquina está funcionando de forma adequada. Por exemplo, uma falha ocorre em *Apache*, *Monit* vai ajudar a reiniciar o processo *Apache*. É muito fácil de instalar e configurar e é especialmente útil para a arquitetura multi-serviço com centenas de micro-serviços. Se você estiver usando *Monit*, certifique-se de monitorar as reinicializações que ela executa, de modo a vir à tona problemas e implementar soluções (ao invés de apenas reiniciar e ignorar falhas). Pode se fazer isso por meio do monitoramento de arquivos de log do *Monit* e garantir que você está alerta para cada reinicialização.

3. ELK - ElasticSearch, Logstash, Kibana - via Logz.io

O *ELK Stack* é a solução de análise de log mais comum no mundo de TI moderna. Ele coleta os logs de todos os serviços, aplicações, redes, ferramentas, servidores, e muito mais em um ambiente local único e centralizado para processamento e análise. Utiliza-se para fins analíticos (por exemplo, para solucionar problemas, monitorar serviços, e reduzir o tempo que leva para resolver questões operacionais). Outro uso

para esta ferramenta é para segurança e auditoria (por exemplo, para monitorar as mudanças em grupos de segurança e alterações nas permissões). Depois de receber alertas sobre estas questões, é fácil para agir em usuários e atividades não autorizadas. É também usado *ELK* para inteligência de negócios, tais como monitoramento de nossos usuários e seu comportamento. Você pode configurar o seu próprio *ELK*.

4. Consul.io

Consul é uma gama de serviços e configuração em aplicações modernas, elásticos que são construídos a partir de micros serviços. A ferramenta de código aberto faz uso da mais recente tecnologia no fornecimento de nomes de *DNS* interno dos serviços. Ele atua como uma espécie de corretor para ajudá-lo a assinar e registrar nomes, permitindo-lhe aceder a nomes de serviços em vez de máquinas específicas. Se, por exemplo, existe um conjunto de várias máquinas, pode-se simplesmente registrá-los como uma única entidade de *Consul* e acessar o cluster facilmente. É uma ferramenta de alta eficiência, embora ainda há um sentimento de que existe à ser feito com ele.

5. Jenkins

Não é o mais rápido ou o mais badalado, mas é realmente fácil de começar a usar e tem um grande ecossistema de *plugins* e *add-ons*. Ele também é otimizado para a personalização fácil. Tem-se configurado *Jenkins* para construir código, criar recipientes *Docker* (projeto de código aberto que automatiza a implantação de aplicativos dentro de recipientes de software, fornecendo uma camada adicional de abstração e automação de virtualização de sistema de nível operacional), toneladas de executados de testes e empurrar para preparo / produção. É uma ótima ferramenta, mas há algumas questões a respeito de escala e desempenho (que não é tão incomum).

6. Docker

Usado na produção para a maioria dos serviços. Ele facilita o gerenciamento de configuração, problemas de controle e dimensionamento, permitindo os recipientes a serem movidos de um lugar para outro.

O *Docker* está progredindo e deixando os desenvolvedores ansiosos para receber novas soluções de gerenciamento e orquestração de empresa.

7. Collectd / collectl

Collectd são ferramentas que recolhem e armazenam estatísticas sobre o sistema no qual eles são executados e são muito mais flexíveis do que outras ferramentas. Eles

permitem aos usuários medir os valores de várias métricas do sistema e ao contrário de outras ferramentas de coleta de log que são projetados para medir parâmetros de sistema específicos, *Collectd* pode monitorar diferentes parâmetros em paralelo. Foi usado essas duas ferramentas para medir parâmetros de desempenho do cliente e enviá-las para o *ELK-as-a-Service* plataforma. Nós especificamente envolveu um agente *collectl* em um recipiente de encaixe e empurre-o com Ansible a todos os nossos servidores. Ele coleta informações a cada dois segundos e, em seguida, move para *ELK* para permitir gerar relatórios e enviar alertas.

8. Git (GitHub)

Git foi criado há 10 anos por necessidade da comunidade Linux para o software de SCM (Gestão de controle de origem) que poderia suportar sistemas distribuídos. Git é provavelmente a mais comum ferramenta de gerenciamento de fonte disponível hoje. Além de suas grandes características que se bifurcam, GitHub também tem plugins que podem se conectar com Jenkins para facilitar a integração e implantação.

7. Conclusão

O que torna o DevOps tão necessário?

Vimos que o crescente pela demanda de desenvolvimento com mais agilidade e menos ocorrência de problemas levaram a evolução no modos de desenvolvimento no meio corporativo

Um relatório chamado “*2013 State of DevOps Report*”, publicado pela Puppet Labs feito a partir de uma pesquisa com 4 mil profissionais de TI, tanto de operações como da área de desenvolvimento. Mostra que a adoção desta prática está se acelerando. A maioria dos pesquisados (63%) diz que suas empresas já adotam o conceito e entre os benefícios reporta que conseguem entregar código 30 vezes mais rápido e com 50% menos falhas.

Ora, simplesmente por esses números, todo executivo responsável pelo que chamamos TI nas empresas deve, no mínimo, estudar o assunto. Ignorá-lo pode significar perder excelente oportunidade de virar o jogo, ou seja, de desenvolver e entregar sistemas muito mais rapidamente que hoje.

O tempo é cada vez mais reduzido e a competição no cenário de negócios vem de todos os lados. As áreas de TI que não conseguem acompanhar o ritmo das demandas do negócio que tendem a se tornar secundárias. Perdem a oportunidade de ser contributivas para novos negócios e acabam apenas responsáveis pela “casa das máquinas”, ou seja, necessário, mas simplesmente operacional.

DevOps é uma mudança das práticas convencionais de desenvolvimento e entrega de sistemas. Provoca, claro, reações contrárias, como o famoso “não é novidade, sempre fizemos isso”. Mas, por outro lado, gera benefícios de mudanças comportamentais, como uma maior colaboração entre desenvolvimento e operações. Estes setores sempre atuaram de forma compartimentalizada, com objetivos diferentes. É absolutamente necessário para as empresas criarem coisas novas e rapidamente, para explorar as janelas de oportunidade que cada vez mais se encurtam. O setor de desenvolvimento busca criar estas coisas novas e rápidas (os métodos ágeis são um exemplo). Por outro lado, operações precisa garantir a estabilidade do ambiente operacional de TI. Esta é sua missão e qualquer novidade que perturba seu ambiente não é bem visto. O desafio é como conciliar estes objetivos divergentes.

A questão da integração entre desenvolvimento e operações é fundamental. Apenas olhar um lado da equação não é suficiente.

Por exemplo, nos últimos anos vimos uma verdadeira explosão dos métodos ágeis, com encurtamento significativo dos prazos de desenvolvimento. Mas o *deployment* dos sistemas não avançou na mesma velocidade. A causa é direta: falta de colaboração entre os setores de desenvolvimento e operações. Implementar métodos ágeis sem mudança nas práticas que envolvam o deployment (operações) apenas piora as coisas. O mesmo acontece quando operações buscam criar processos mais padronizados, como adoção de práticas como ITIL e deixa em segundo plano a velocidade que o desenvolvimento demanda. O resultado de tentarmos otimizar os dois lados da equação, desenvolvimento e operações, de forma separada, é a criação de mais barreiras e burocracias, gerando menos eficiência. Tiro n'água.

O conceito de DevOps começou a despertar atenção quando se descobriu que as principais empresas do mundo da Internet como Google, Amazon, Facebook, Twitter e outras o adotam como prática básica.

As grandes empresas da Internet apresentam números impressionantes. Por exemplo, a Amazon faz uma mudança em seus sistemas, em média, uma vez a cada 11,6 segundos! Outros benefícios que estas empresas conseguiram foi se recuperarem de paradas muito rapidamente e os números da pesquisa mostram uma taxa de falhas bem baixa. Voltando ao exemplo da Amazon, apenas 0,001% dos deployments causaram uma pane no sistema.

A média de falhas geradas pelas empresas que adotaram DevOps é cerca de 50 vezes menor em comparação das que não usam esta prática. Imaginem, o que é reduzir em 50 vezes o numero de falhas. Muito menos esforço e recursos para refazer coisas que já deveriam estar prontas e muito mais confiabilidade nos *deployments*. Cada vez

mais fundamental em um ambiente de negócios hiperconectado, com usuários acessando seus sistemas baseados em nuvem, via smartphones e tablets

DevOps é uma evolução nas práticas que chamamos de gestão do ciclo de vida dos sistemas ou *Application Development Life cycle Management* (ADLM). Demanda tecnologia sim, mas principalmente mudanças culturais, funcionais e organizacionais em TI.

As principais barreiras são o desconhecimento do conceito (e muito frequentemente a confusão que apenas implementar método ágeis resolve, vimos que não, pois apenas endereça um lado da equação), cultural (ambiente que não incentiva colaboração), e a falta de capacitação, tanto gerencial como dos profissionais técnicos, no desenvolvimento e em operações.

O diálogo entre desenvolvimento e operações é fundamental. Importante frisar que DevOps não implica que desenvolvedores possam ou devam operar seus sistemas. Não é esta a função do grupo de operações. O que DevOps incentiva é um processo colaborativo entre estes dois setores, de modo que eles atuem em harmonia, criando uma prática de entrega contínua de software.

Não existe mais a barreira do desenvolvimento parar o processo, criar um protocolo de solicitação ao grupo de operações e aguardar que este setor responda. Os dois setores atuando em conjunto fazem com que o processo seja contínuo ou seja, sai do código para a nuvem de produção, sem interrupções.

Em resumo, vale a pena ler a pesquisa e refletir. E aí, indagar, porque ainda não adotamos DevOps?

Referências:

ARTIGO DE PERIÓDICO	Portal Information Management. Acesso em: 16 junho. 2016 < http://docmanagement.com.br/ >
ARTIGO DE PERIÓDICO	IBM, SãoPaulo, em:< http://www.ibm.com/developerworks/br/rational/library/defining-deployment-deliverable-devops/ > . Acesso em: 15 junho 2016.

BLOG	BLOG O blog dos Evangelistas Técnicos do Instituto Eldorado < https://talkitbr.com/2016/03/22/o-ciclo-de-vida-devops/ > . Acesso em: 18 junho. 2016
BLOG	BLOG Guto Carvalho < http://gutocarvalho.net/octopress/2013/03/16/o-que-e-um-devops-afinal/ > . Acesso em: 18 junho. 2016
BLOG	BLOG TOMER LEVY < http://devops.com/2015/08/07/9-open-source-devops-tools-love/ >. Acessado em: 18 de junho. 2016
BLOG	BLOG John Willis - The Convergence of DevOps < http://itrevolution.com/the-convergence-of-devops/ >, Acessado 18 de junho. 2016
Livros	Michael Hutterman, DevOps for Developers: Integrate Development and Operations, the agile way. New York, 2001
Livros	Mike Loukides, What is Devops? , Tokio, 06.04.2012
Relatório Técnico	Puppet Labs, <i>2013 State of DevOps Report.</i> Miami, 2012.