DevOps: An Architect point of view

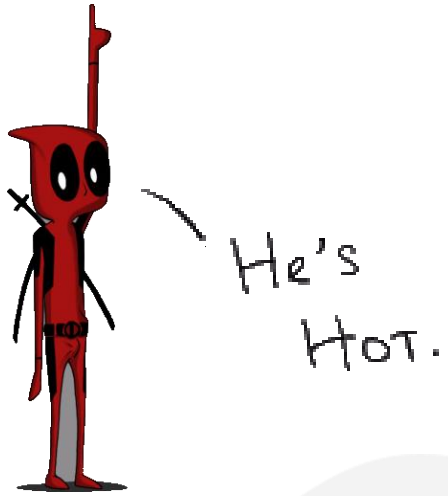
TDC SP Julho 2019

Alexandre B. Daccas Mendonça

avanade

Sobre mim

- Alexandre Bogiani Daccas de Mendonça (Aka Daccas)
- 20 anos de experiência
- Pós graduado em Engenharia de software
- Especialista em arquiteturas web, Azure, Kubernetes, DevOps
- Atuação em várias áreas de negócio: finanças até aviação
- Apaixonado por tecnologia, ferramentas e hobby modelismo



avanade

DevOps as part of modern ALM

TDC SP Julho 2019

Alexandre B. Daccas Mendonça

avanade


Não há nenhum problema no código, ou você está implantando de forma errada ou é algum problema de ambiente.
Infra funciona, seu código é que está ruim.
DEV
Quem nunca...
OPS

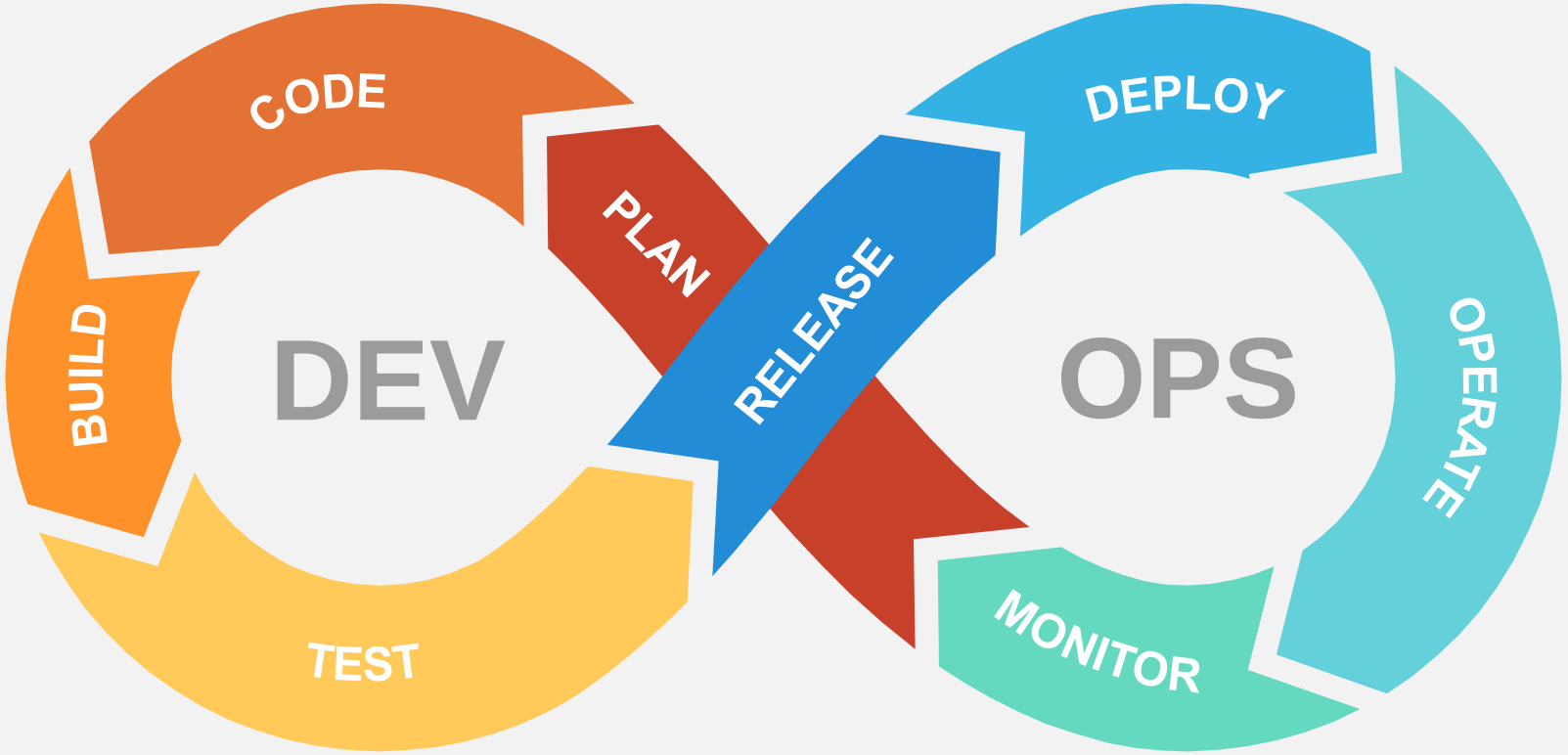
Desenvolvimento ágil necessita de entrega rápida

Agile Development

Agile Delivery

avanade

Entrega e Monitoramento como parte de um processo infinito.

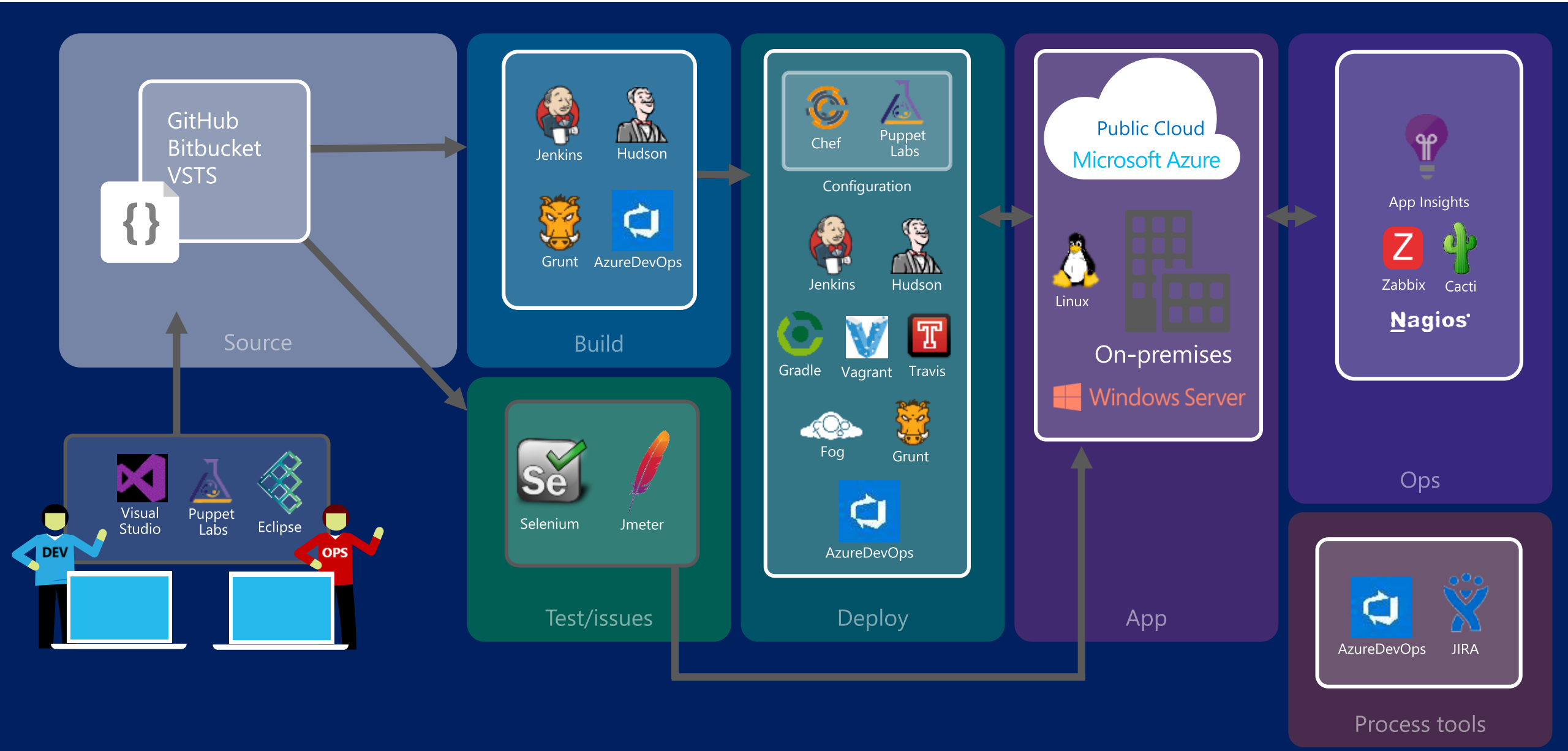


avanade

Cadeia de valor (Value Stream)

- Crie foco aonde realmente importa.

Framework ALM para DevOps



The three ways....

TDC SP Julho 2019

Alexandre B. Daccas Mendonça

avanade

O Conceito de DevOps

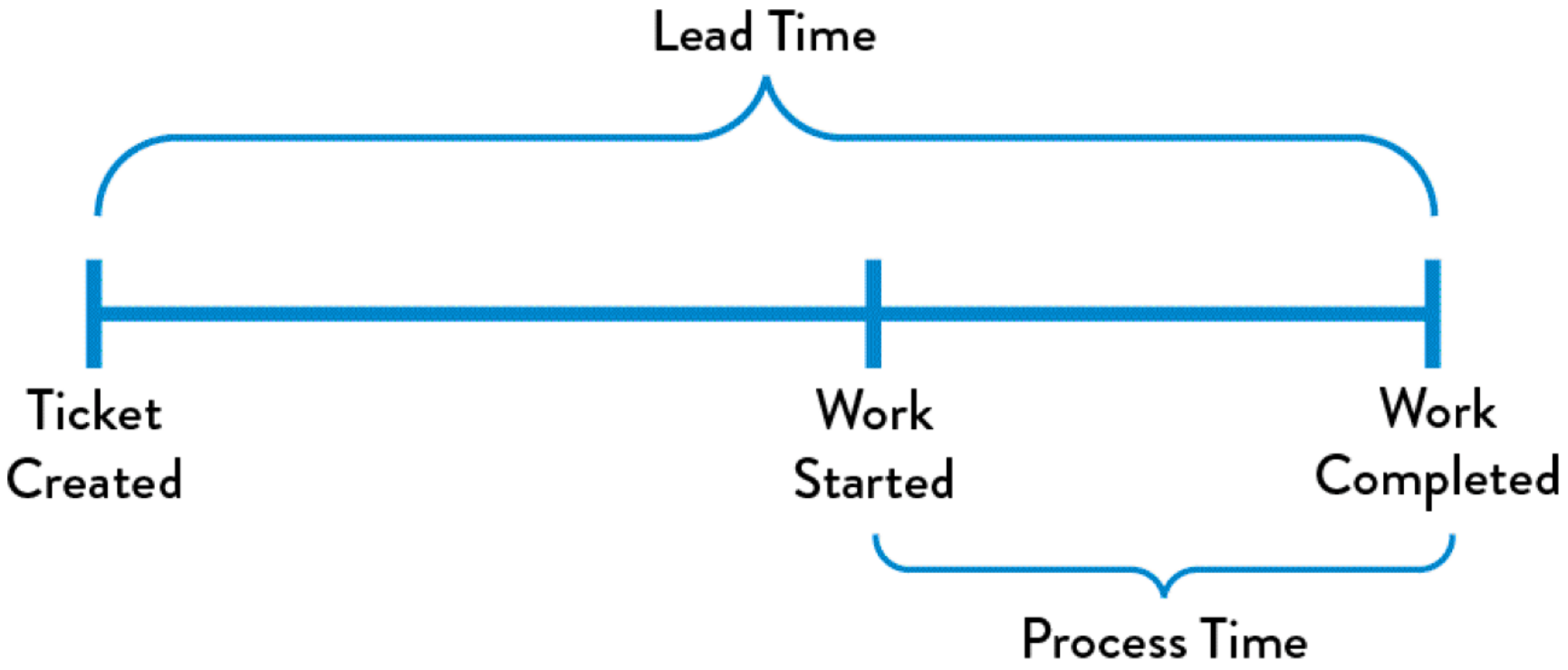
First way: Flow (Entrega)

TDC SP Julho 2019

Alexandre B. Daccas Mendonça

avanade

Foco no tempo de execução (Lead Time)



Elementos chaves da prática do Flow

- Defina tamanho máximo do seu trabalho (WIP – Work in progress)
- Reduza o tamanho das suas tarefas.
- Reduza a quantidade de passagens de bastão.
- Crie as fundações de um pipeline de implantação (deployment pipeline)
- Crie e habilite testes automáticos rápidos e confiáveis
- Crie a prática de integração contínua
- Planeje e automatize entregas de baixo risco

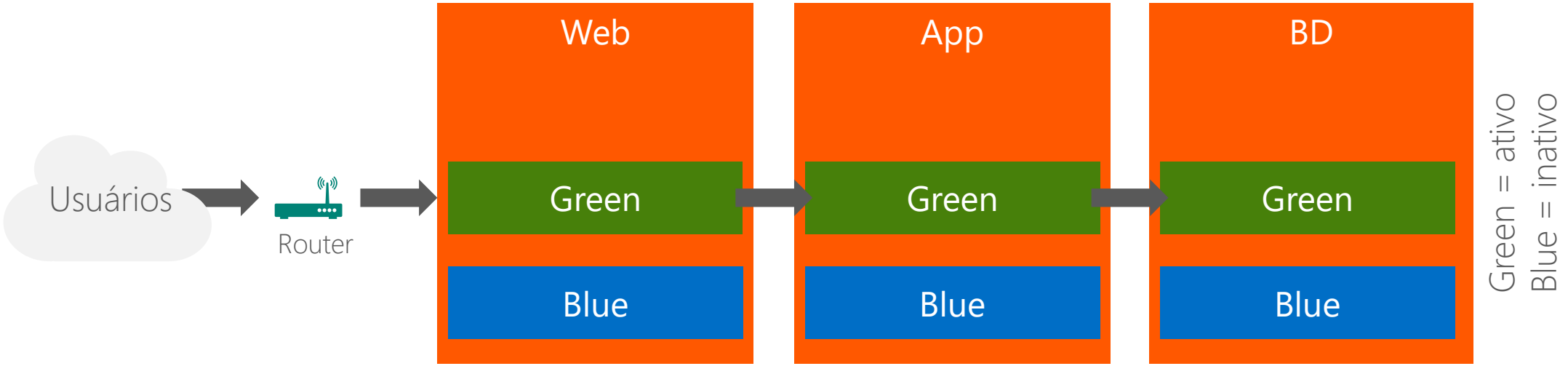
CI/CD pipelines (Continuous Integration/Deployment)

- CI - Integração contínua é uma prática de desenvolvimento software em que os membros do time integram o seu código **frequentemente**. Normalmente cada pessoa integra um pedaço de código/funcionalidade no mínimo 1x ao dia, gerando **múltiplas integrações por dia**.
  - Cada integração é verificada por um processo de build automatizado (que incluem testes), que detectam erros de integração rapidamente.
- CD - Implantação continua é a prática em adição com a integração contínua, em que você implanta funcionalidades de sua aplicação em produção frequentemente através de uma plataforma self-service.

Integração contínua + implantação contínua = Entrega contínua

Estratégias de implantação contínua – blue/green

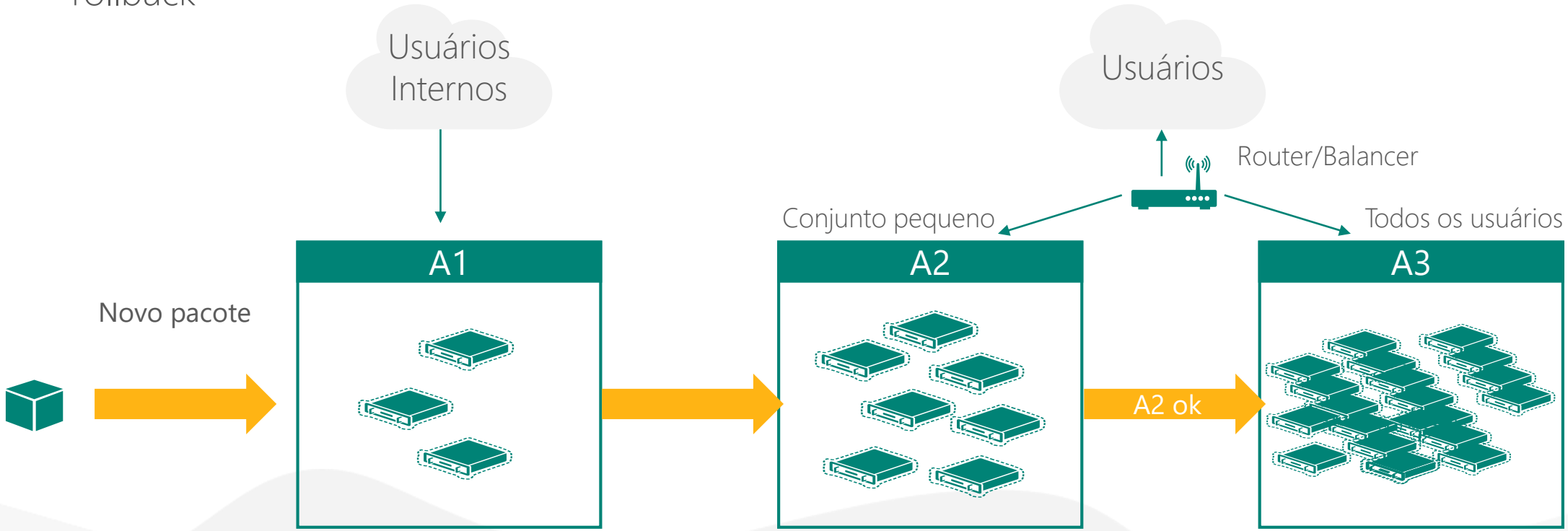
- 2 ambientes de produção – “Blue” e “Green”
  - Somente 1 atendendo as requisições dos usuários, outro serve como distribuição antecipada de novos releases.



- Implantação para o ambiente inativo (Blue) e realização de testes de prontidão
  - Liberação = redirecione o tráfego do ambiente Green para o ambiente Blue.
  - Rollback = volte o tráfego para o ambiente Green

Estratégias de implantação contínua – canary

- Implantação = promova aos poucos para um número menor de usuários e vá aumentando aos poucos.
- Monitore o andamento e o comportamento da aplicação e realize ações para continuar ou rollback



Second way: Feedback (Monitoramento)

TDC SP Julho 2019

Alexandre B. Daccas Mendonça

avanade

A importância do monitoramento...

Um cenário muito comum...

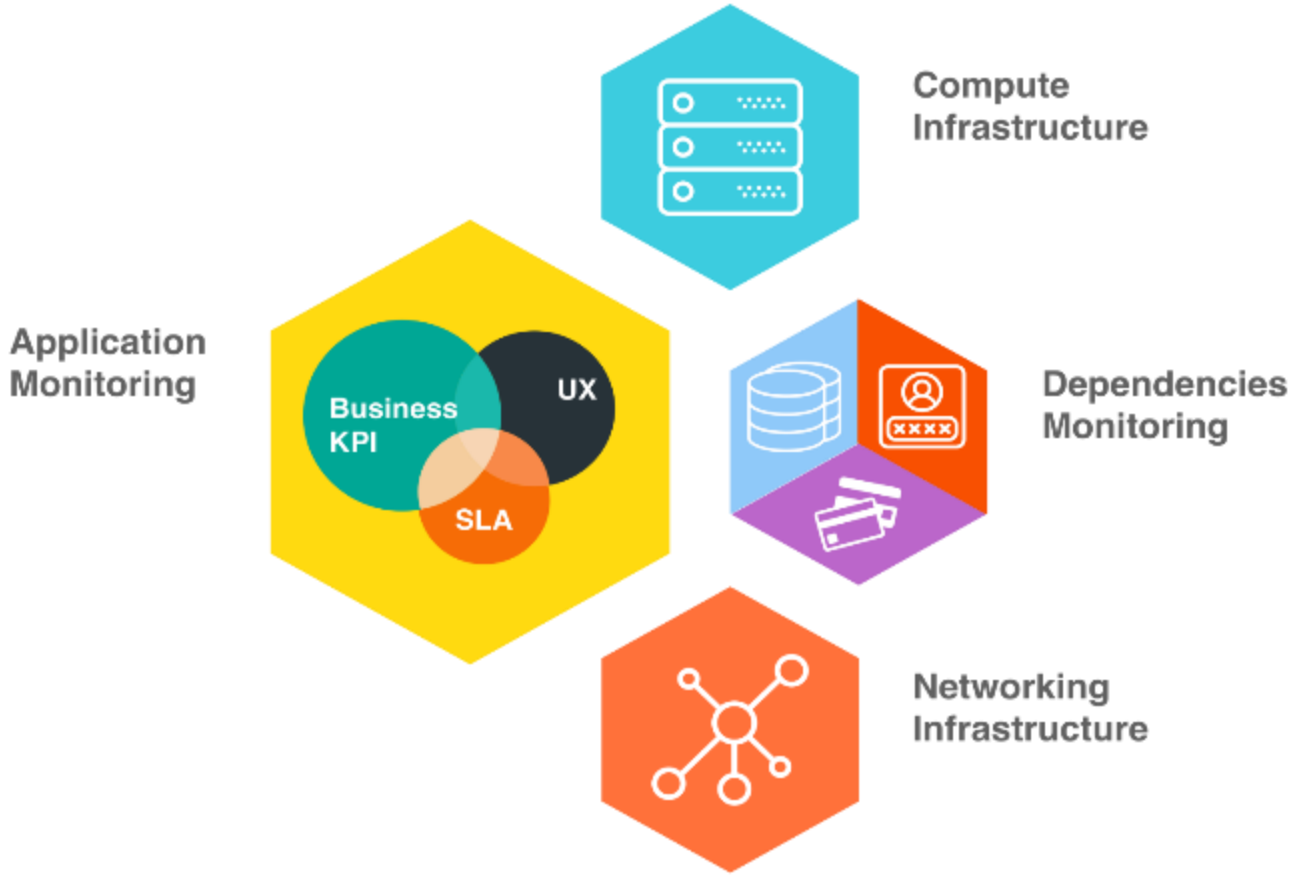
Empresas desorganizadas usam táticas não disciplinadas em servidores

avanade

Elementos chaves da prática do Feedback

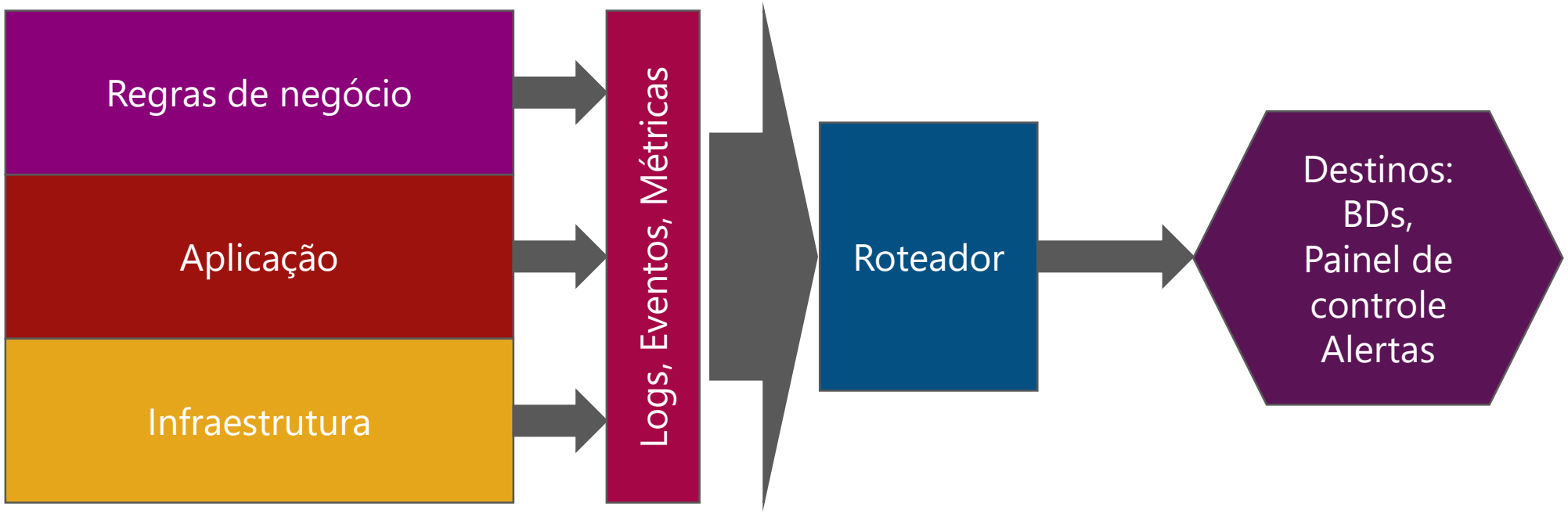
- Crie telemetria do seu produto
  - Tenha dados imediatamente que te informem o que está ocorrendo.
- Analise a telemetria
  - Antecipe problemas e otimize para atingir seus objetivos
- Habilite a prática do retorno ao time
  - Para que cada vez mais o time (dev e ops) trabalhem juntos para entregar códigos com maior qualidade
- Desenvolvimento baseado em hipótese ou teste A/B
  - Através do feedback descubra se seus clientes gostam de uma nova funcionalidade.

O que devemos monitorar?



avanade

Framework de monitoramento



Exemplos de dados para telemetria

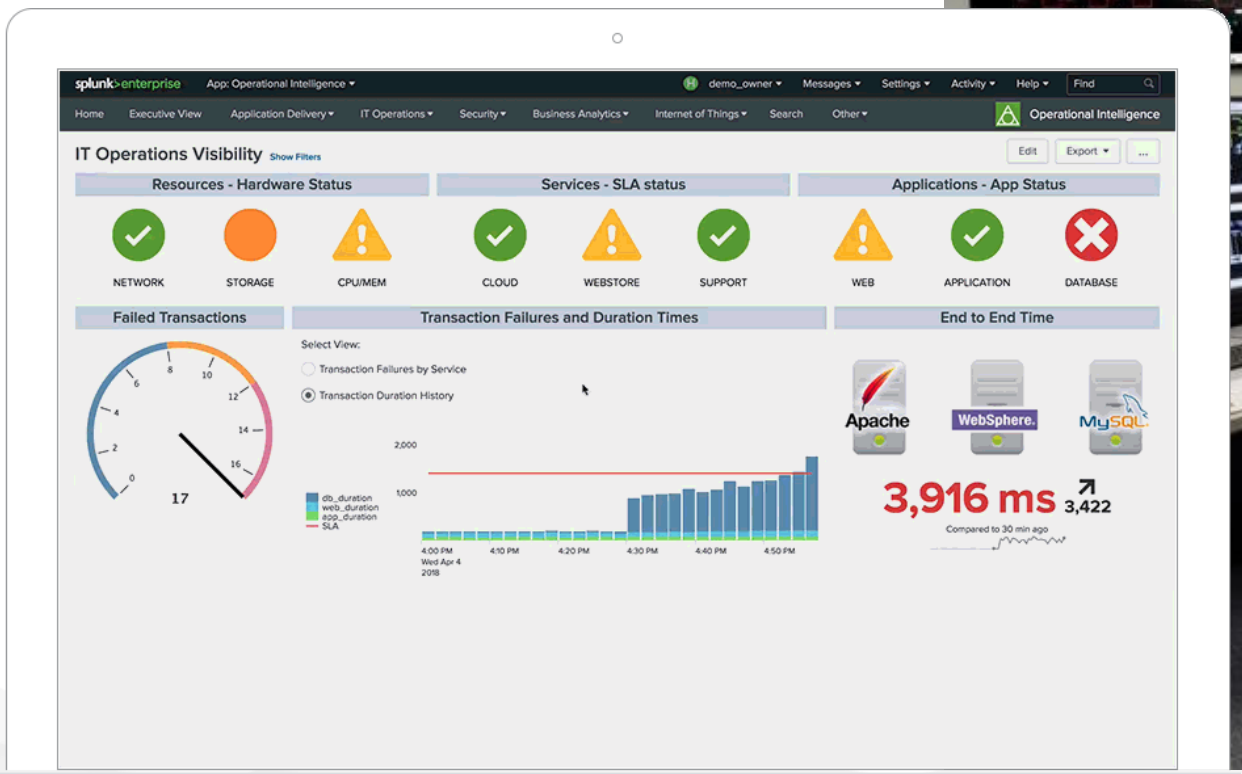
Negócios	Aplicação	Infraestrutura	Software no cliente	Pipelines CI/CD
número de vendas, receitas, qtde novos usuários, taxa de conversão, resultados de teste A/B.	Tempo de resposta, erros do sistema, tempos de transações...	BD, Sistema operacional, rede, discos, Trafego web, CPU, memória, ...	Erros, Falhas, dados estatísticos de uso,...	Resultados dos pipelines, taxa de entrega, lead time, resultado dos testes, ...

Use dados ao invés de rumores, culpa, etc...

avanade

Painéis de controle

- Central de controle de telemetria
- Aplicação de limites para identificar a saúde das aplicações
- Crie alertas para antecipar problemas.
- Analise tendências para prever falhas ou
- maior demanda.



Third way: Experimentation

TDC SP Julho 2019

Alexandre B. Daccas Mendonça

avanade

Práticas chaves da experimentação

- Estabeleça uma cultura de aprendizado...
  - ...seja transparente e aberto quando as coisas saem erradas.
- Injete falhas controladas nas aplicações...
  - ...para criar resiliência no seu software
- De melhorias pequenas para grandes melhorias...
  - ...faça pequenas melhorias em avanços globais
- Separe um tempo...
  - ...para criar melhorias organizacionais e aprendizado
- Crie processos de revisão para melhoria da qualidade do código
  - Foco em peer reviews e pair programming

Reuniões de post mortem

> Erro humano não é a causa de problemas; em vez disso, o erro humano é a consequência das falha de desenho, processos e/ou instrução que recebemos.

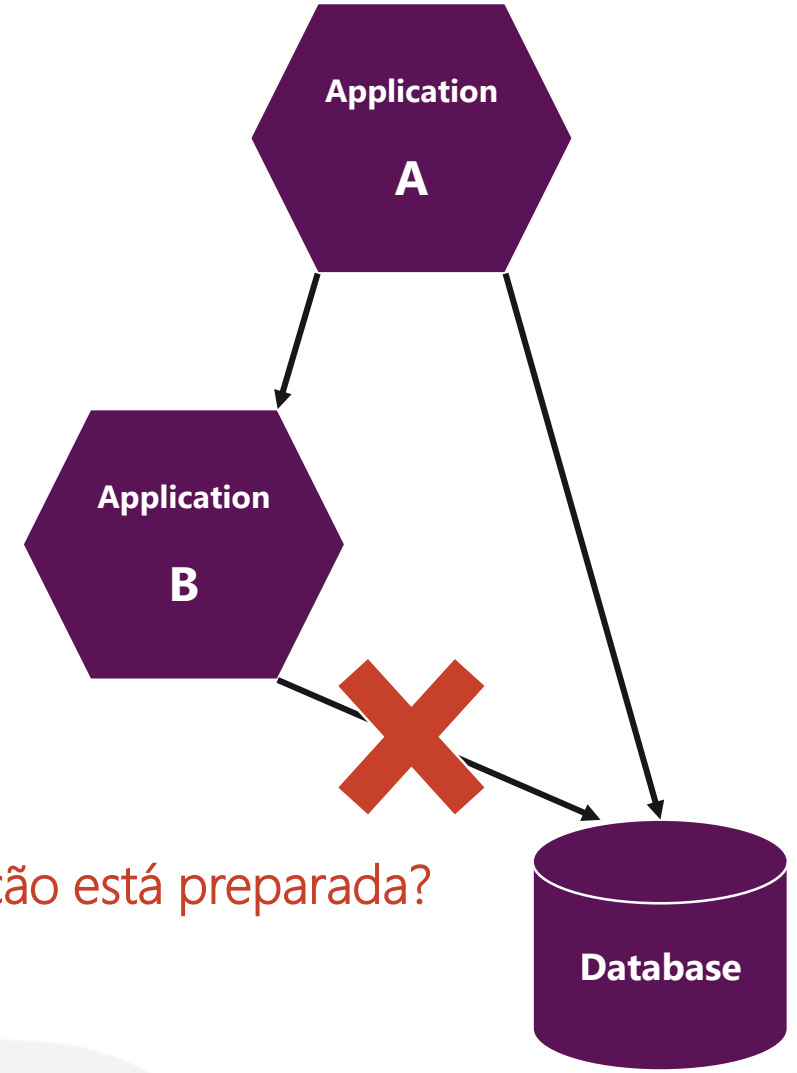
avanade

Reuniões de post mortem

- Construa uma linha do tempo e obtenha todos os detalhes de várias perspectivas sobre a falha, sem punições as pessoas envolvidas.
- Empodere todos os desenvolvedores permitindo que eles contribuam com todas as informações possíveis para evitar novas falhas que já ocorreram.
- Encoraje as pessoas que causaram uma falha que seja porta vozes para educar o restante da organização a como não causá-los no futuro, eles são as melhores pessoas para faze-lo.
- Aceite que sempre há um espaço ilimitado onde as pessoas podem tomar ação ou não, e que houveram um argumento para fundamentação e julgamento dessas decisões.
- Proponha contramedidas para prevenção de acidentes similares no future. Garanta que essas contramedidas sejam documentadas, que todos saibam e prazos para resolução.

Injeção de falhas controladas.

- A construção de aplicações resilientes necessitam de experiência com falhas.
- Não espere que as coisas parem em produção
- Insira falhas de modo controlado e planejado
- Use as ferramentas de monitoramento para detecção de falhas/comportamento
- Comece pequeno e de forma inteligente
- Tipos comum de falhas
  - Rede
  - Hardware
  - Banco de dados
  - Sistema operacional
  - ...



Sua aplicação está preparada?

Crie um product backlog para Operações

- Automatize o máximo possível...
- ... tarefas não automatizáveis devem ser representadas como user stories no product backlog
- Coloque as tarefas de operações junto com as de desenvolvimento
  - Um único product backlog
  - Melhor planejamento e entregas mais confiáveis

avanade

Reserve tempo para criar uma cultura de aprendizado e melhorias (Aka improvement blitzes)

- Todos do time devem focar em resolução de problemas ocorridos
  - Nenhuma nova funcionalidade é permitida
  - Foco em resolver problemas corrigidos através de soluções alternativas
  - Pode ser relacionado a código, ambiente, arquitetura, etc...
  - Deve abranger pessoas de todo a cadeia de valor
  - Apresente os resultados para todos os envolvidos.

avanade

Considerações finais

TDC SP Julho 2019

Alexandre B. Daccas Mendonça

avanade

Não esqueça dos ´ilities´

- Security
  - Faça com que Infosec seja parte da sprint review
  - Inclua Infosec na analise de defeitos e nas reuniões de post mortem
  - Configure segurança no seus pipelines (configure acessos, esconda dados protegidos como senhas etc...)
  - Inclua testes de segurança na sua aplicação
  - Garanta a segurança dos ambientes.
  - Integre segurança na telemetria
- Repeatability
  - Reuse tarefas comuns
  - Crie templates de pipelines para aplicações similares
  - Crie templates para workflows de aprovação
- Performance
  - Crie cache de dependências nos pipelines.
  - Paralelize tarefas quando possível
  - Use o mínimo necessário para montagem de ambientes/pipelines
- Reliability
  - Use repositórios de dependências locais ao invés de externos.
  - Reduza a quantidade de tecnologias/arquiteturas para não aumentar complexidade.
  - Use a mesma abordagem/tecnologias para todas as aplicações quando possível

avanade

Don´t forget the ´ilities´

- Recoverability
  - Adicione telemetria e monitore seus pipelines
  - Crie tarefas de rollback
- Interoperability
  - Use ferramentas que tenham fácil integração com outras
  - Desenhe e orquestre seu workflow para diminuir quantidade de integrações
- Testability
  - Teste seus scripts de tempos e tempos
  - Recrie ambientes e check se seus scripts estão atualizados.
- Modifiability
  - Garanta que se uma mudança ocorrer, seu pipeline não será impactado.
  - Teste seu sistema para novas variáveis de ambiente

Referências

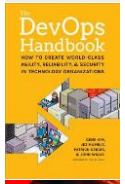
TDC SP Julho 2019

Alexandre B. Daccas Mendonça

References

The Phoenix Project

Gene Kim, Kevin Behr and George Spafford



DevOps Handbook

Gene Kim, Jez Humble, Patrick Debois and John Willis

DevOps: A Software Architect Perspective

Len Bass, Ingo Weber and Liming Zhu

- And some Avanade´s materials…

avanade