

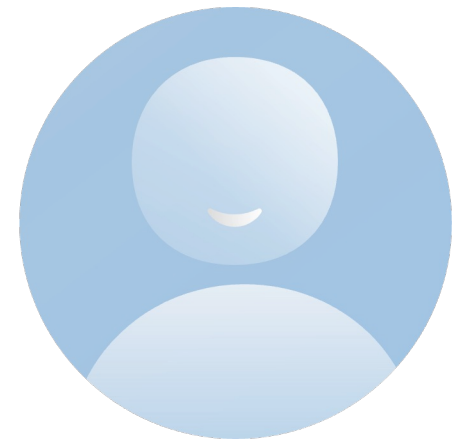


PUC Minas
Virtual

Maturidade DevOps e FinOps

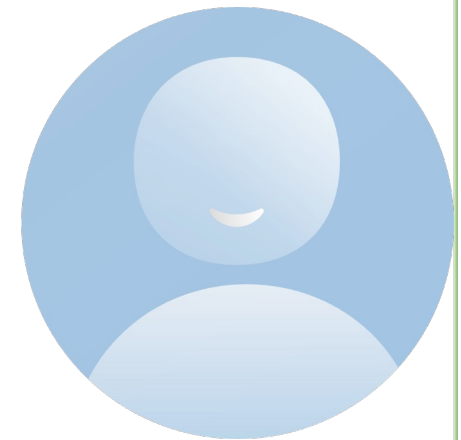
Marco Mendes

Escala de Maturidade DevOps



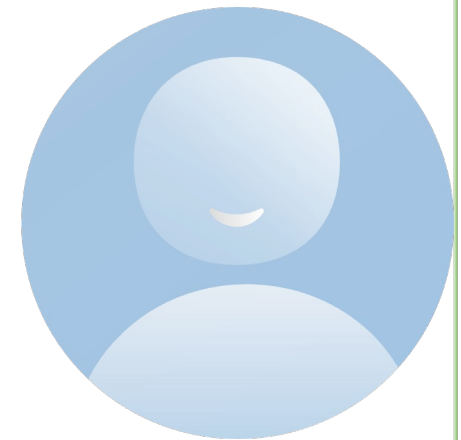
1. Inicial

- Não temos consciência da prática DevOps.
- Não temos um processo claramente definido
- Não temos qualidade consistente.
- Não existe instrumentação nem medições sobre os problemas e ações.
- Não temos resultados efetivos para comunicar para a gestão e diretorias.



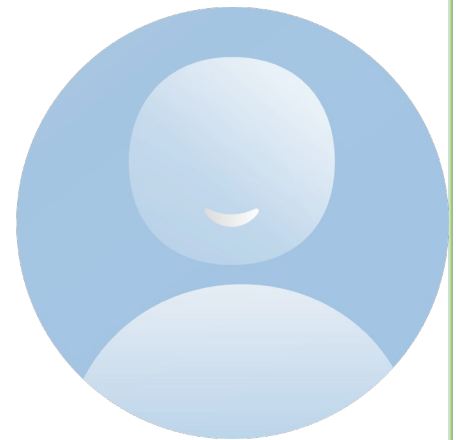
2. Consciente

- Temos consciência da prática DevOps.
- O processo de trabalho está emergindo.
- Realizamos já laboratórios e experimentações.
- Não temos qualidade consistente.
- Não existe instrumentação ou ela é muito precária.
- Não temos resultados efetivos.



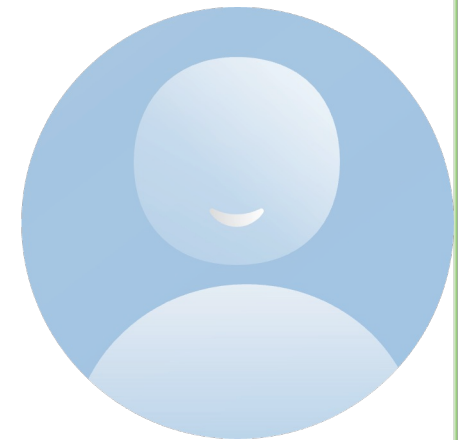
3. Gerenciado

- Temos consciência da prática DevOps.
- O processo de trabalho está padronizado.
- Temos uma suíte de ferramentas definido e que suporta o processo de trabalho.
- Não temos qualidade plenamente consistente.
- A instrumentação está mais avançada e temos plena consciência de nossos desvios.
- Os resultados começam a surgir, mas ainda não são plenamente efetivos.
- Sabemos que ir além desse nível é desafiante, longo e árduo, pois as fontes de variabilidade devem ser atacadas dia após dia.

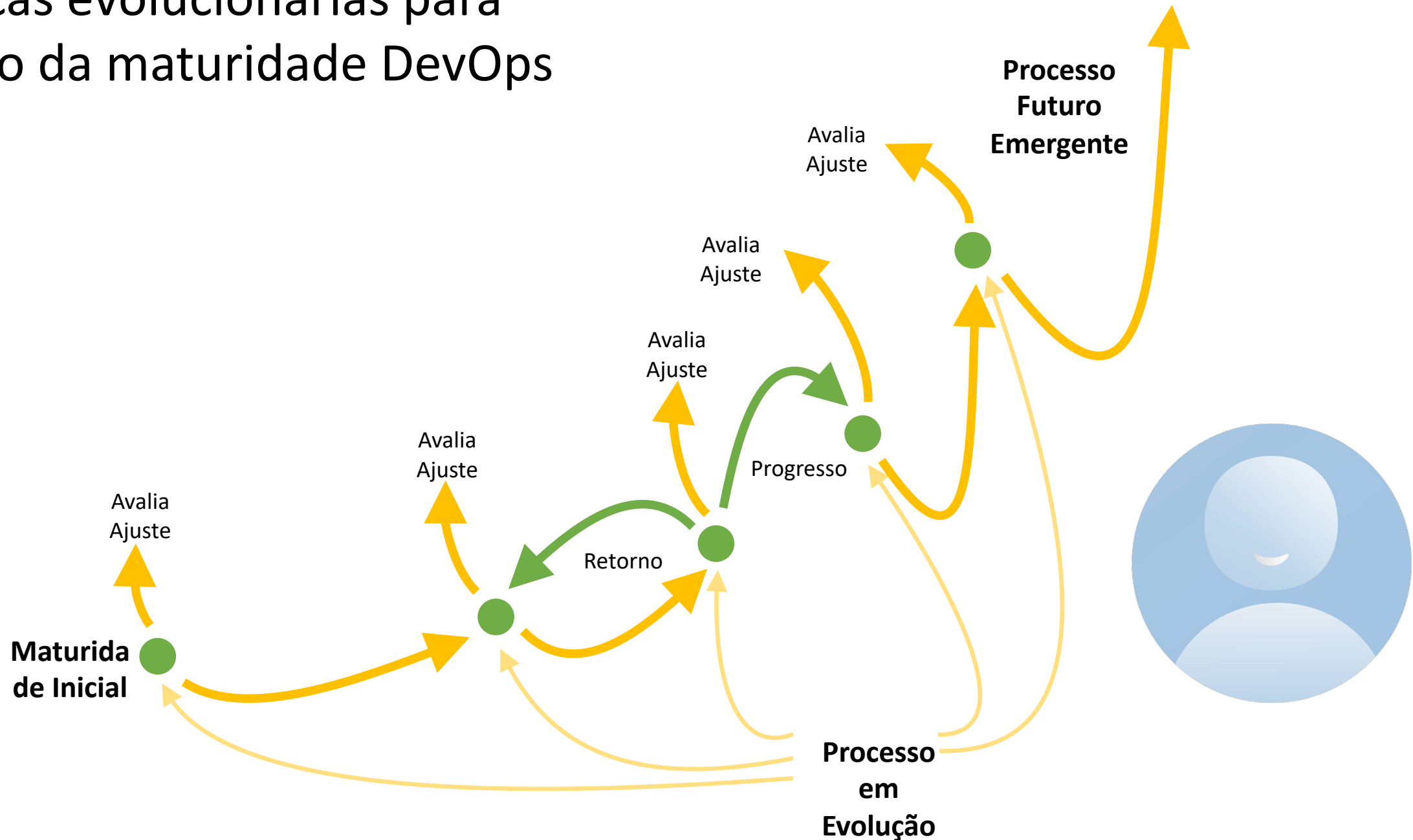


4. Quantitativamente Gerenciado

- Temos consciência da prática DevOps.
- O processo de trabalho está padronizado.
- Temos uma suíte de ferramentas definido e que suporta o processo de trabalho.
- A instrumentação está mais avançada e já endereçamos nossos desvios ao longo do tempo.
- Temos qualidade plenamente consistente.
- Os resultados são claros e conseguimos demonstrar o efeito em indicadores como redução do tempo de entrega, aumento da vazão ou redução do uso de horas humanas.



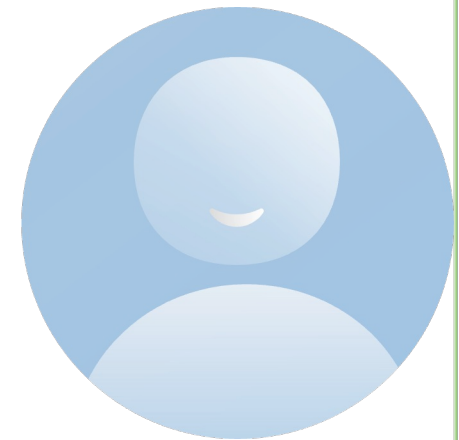
Mudanças evolucionárias para aumento da maturidade DevOps



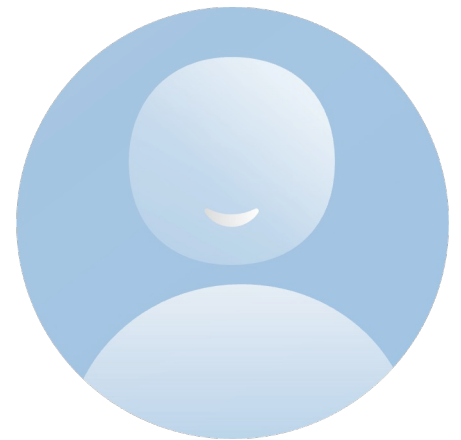
Fonte: Kanban University

Contexto da Maturidade

- Sempre avalie a maturidade dentro de um contexto. Times diferentes podem ter maturidade diferentes.
- Não use maturidade para falar da “empresa” como um todo.
- Qual o limite da “organização” que estamos observando? Um time? Uma tribo? Uma área? Um serviço que envolve múltiplos times?
- Avalie o dialeto dos desenvolvedores e times.
- Exemplo: Se as pessoas acreditam que fazem integração contínua mas não tem automação de testes, observamos a maturidade baixa do time.
- Colete evidências (falar é fácil)

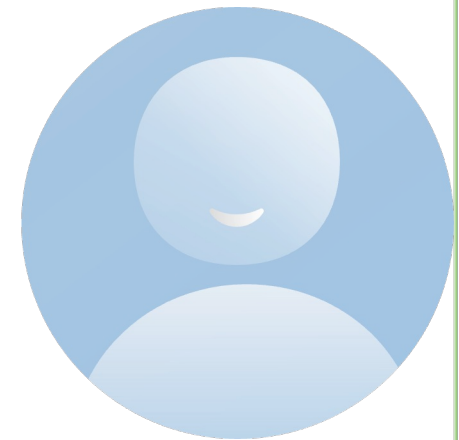


Maturidade de Automação de Builds



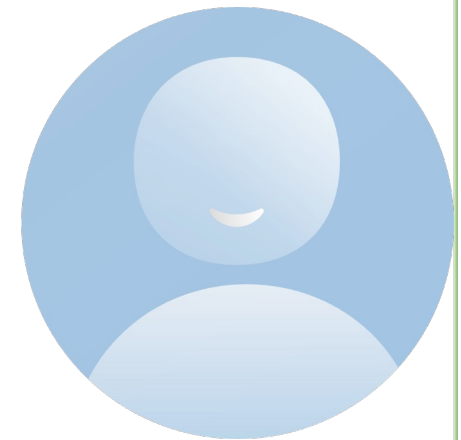
Feedback do cliente

- A **automação de builds** é prática essencial para garantir que os executáveis dos seus produtos sejam gerados de forma consistente, em base diária.
- Esta prática busca evitar o problema comum do código funcionar apenas máquina do desenvolvedor, fornecer executáveis sólido a qualquer momento e reduzir o trabalho manual gasto para gerar executáveis.



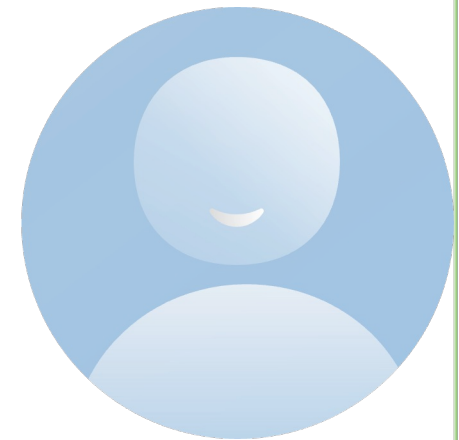
1. Inicial

- Não temos consciência sobre automação de builds.
- Os processos são manuais.
- Não temos nenhum automação de testes.
- Dependemos das IDEs para compilar o código
- A qualidade é completamente dependente do artesanato individual.
- Aqui ouvimos “*Na minha máquina funciona*” com mais frequência que gostaríamos.



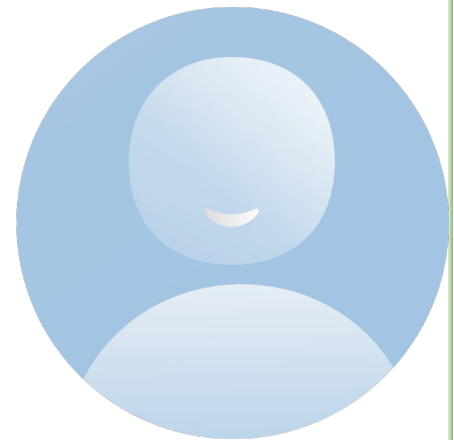
2. Consciente

- Temos consciência sobre o processo de automação de builds.
- O processo de trabalho de construir builds está emergindo.
- Temos já experimentos em curso com ferramentas como maven, gradle, npm, yarn e não dependemos mais das IDEs para compilar o código
- Estamos começando a testar ferramentas de pipelines de automação de builds. (ex. GitLab ou GitHub Actions)
- A automação de testes de unidade começa a ser discutida.
- A instrumentação do processo de builds ainda não existe e não temos métricas desse processo.



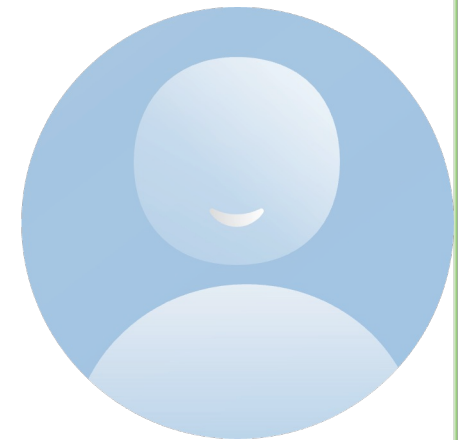
3. Gerenciado

- Temos uma suíte de ferramentas para pipelines de builds definido e que suporta o processo de trabalho.
- Os builds rodam em base diária, mas ainda temos problemas de qualidade devido a baixa automação de testes ou políticas pobres de gestão de configuração.
- Os esforços de aumento da cobertura de testes estão a pleno vapor.
- Introduzimos processos técnicos sólidos nos builds, como verificação de qualidade de código ou segurança.
- Os resultados começam a surgir, mas ainda não são plenamente efetivos.
- Podemos ficar meses ou trimestres aqui até que tenhamos larga automação de testes e mais frequência de builds sólidos.

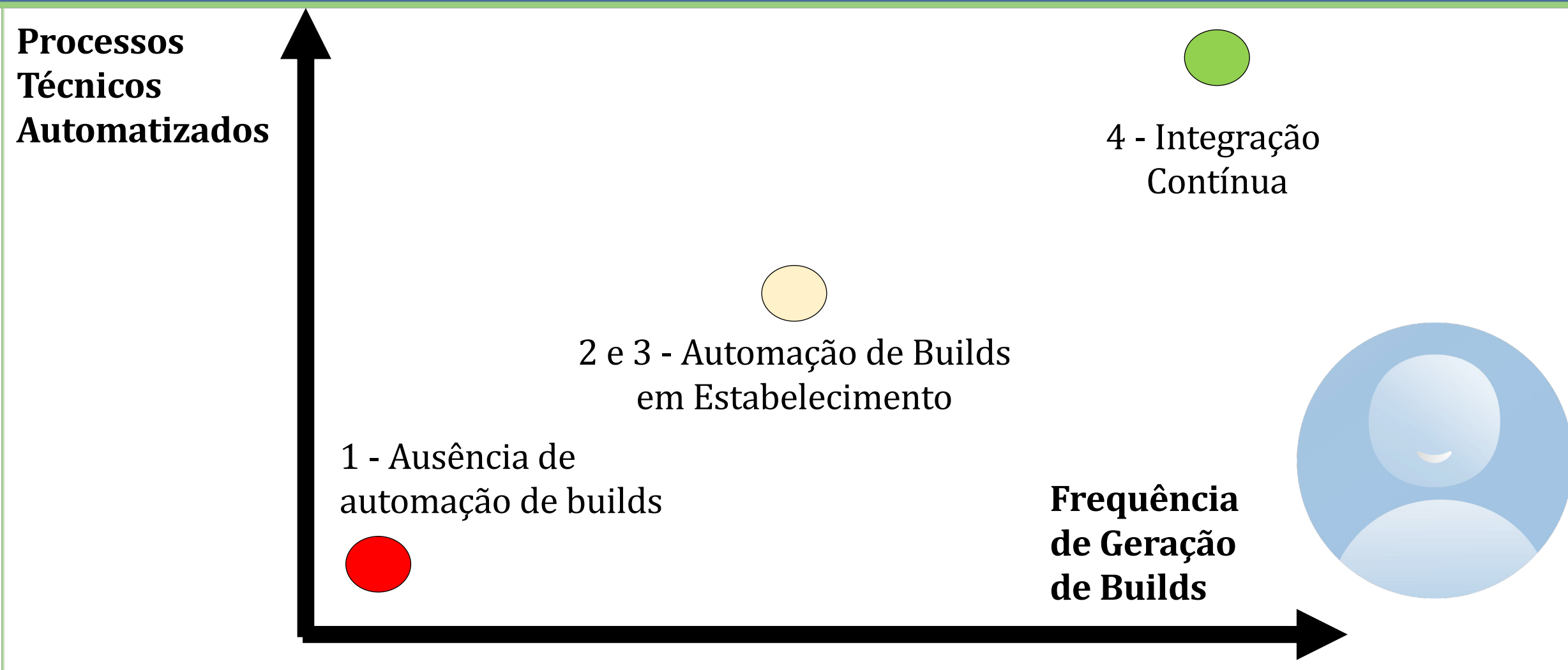


4. Quantitativamente Gerenciado

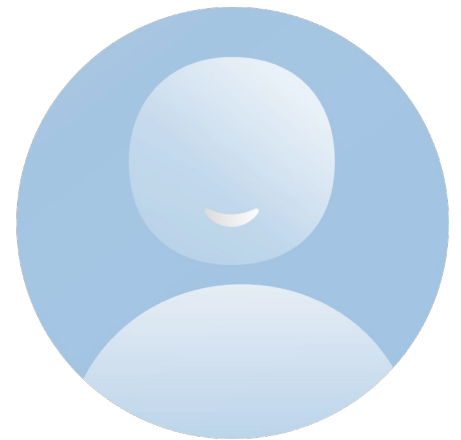
- Entramos no terreno da *Integração Contínua*.
- Grande frequência de builds e os builds tem processos sólidos de verificação de qualidade de código, segurança e automação de testes.
- Abordagens TDD e BDD são comuns para suportar a automação de testes.
- A cobertura de código excede em muitas empresas 80%, fornecendo muita solidez para melhorias e inovações.
- A instrumentação é sólida. Podemos demonstrar a redução de horas humanas, do tempo de entrega e defeitos em produção.



Escala de Maturidade Visual

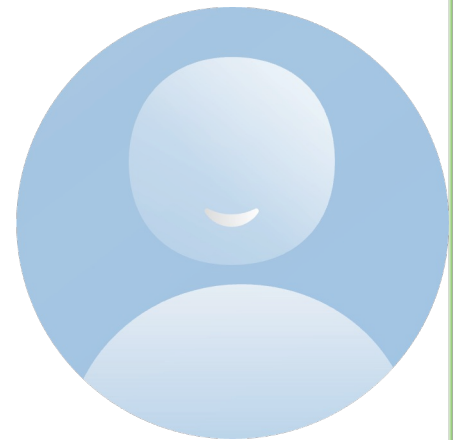


Maturidade de Automação de Entregas



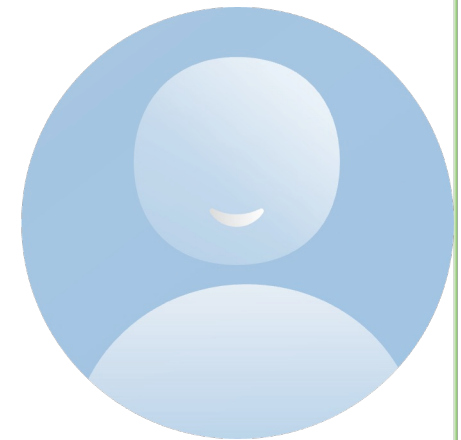
Automação de Entregas

- A automação das entregas é uma prática que busca garantir que o processo de promoção do executável para os ambientes de testes, homologação e produção sejam automatizados e assim tornados consistentes e eficientes.



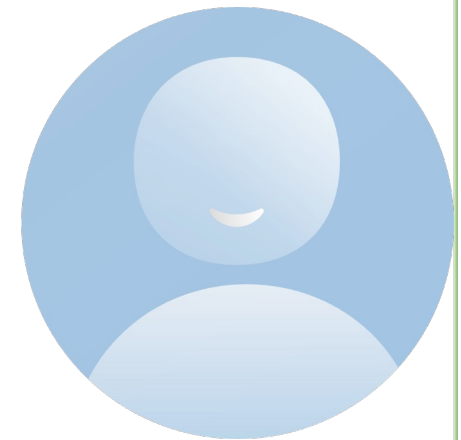
1. Inicial

- Não temos consciência sobre automação de entregas.
- Aqui ouvimos “*Na minha máquina funciona*” com mais frequência que gostaríamos.
- A promoção de builds é manual.
- Os processos de transporte de esquemas e dados nos bancos de dados são morosos.
- Temos muitas instabilidades em produção.
- Não é incomum que o transporte para produção demore horas ou dias.



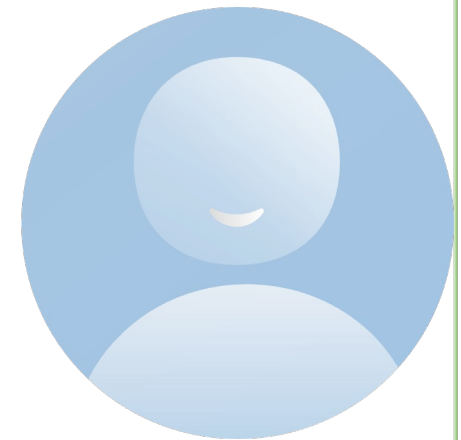
2. Consciente

- Temos consciência sobre o processo de automação de entregas.
- Ferramentas como Chef, Puppet ou Azure Pipelines começam a ser experimentadas.
- O conceito de containerização de ambientes começa a ser discutido e ferramentas como Docker começam a ser usadas.
- Promoções do ambiente de integração para ambiente de testes começam ser realizados de forma automatizadas.



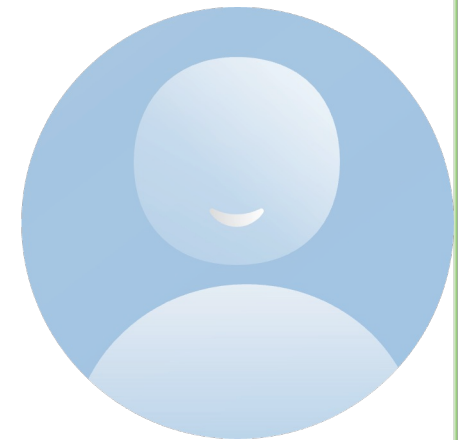
3. Gerenciado

- Temos uma suíte de ferramentas para promoção de builds.
- Os processos de gestão de mudanças ITIL começam a ser incorporados nas ferramentas de automação de entregas.
- A promoção de contêineres é usual já e temos propriedade no uso de contêineres.
- Introduzimos processos técnicos sólidos nos builds, como verificação de qualidade de código ou segurança.
- Os resultados começam a surgir, mas ainda não são plenamente efetivos.
- Ainda podemos lutar com o transporte automatizados de dados de banco de dados e dependências de outras aplicações
- Podemos ficar meses aqui até que tenhamos resolvido a automação do transporte de dados e dependências de outras aplicações.

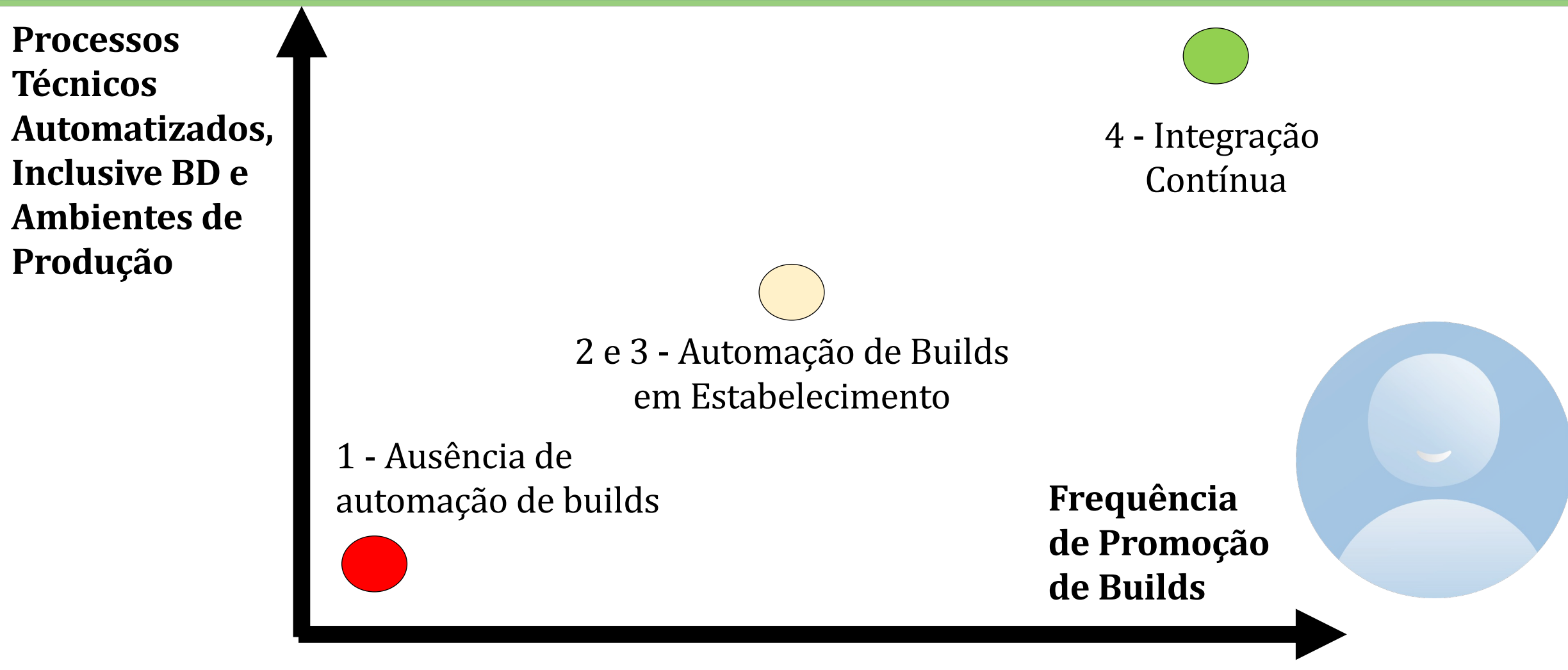


4. Quantitativamente Gerenciado

- Entramos no terreno do CD (*Implantação Contínua e Entrega Contínua*)
- Entregamos na frequência demanda pelo negócio.
- O transporte de builds, esquema e dados são todos automatizados.
- Containeres são usados em largas escala e já temos repositórios privados implementados.
- Os processos de entrada em produção são sólidos, inclusive a reversão de builds.
- Práticas como implantações canários, implantações azul-verde e testes A/B são suportadas sem maiores problemas.
- Medimos o efeito econômico da melhoria de qualidade, tempo e custos associados aos processos de entrega.



Escala de Maturidade Visual

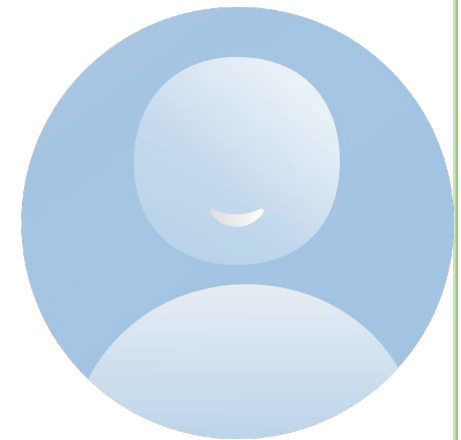


Maturidade de Automação de Ambientes e Infraestrutura como Código



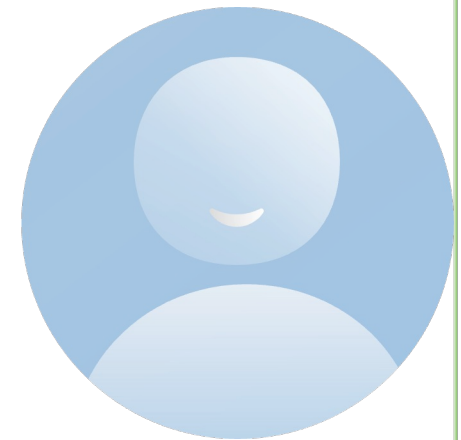
Automação de Ambientes e Infraestrutura como código

- A automação de ambientes e infraestrutura como código busca garantir que o provisionamento de ambientes seja realizado de forma automatizada e consistente para o aumento da produtividade e segurança no uso de ambientes virtualizado e de nuvens.



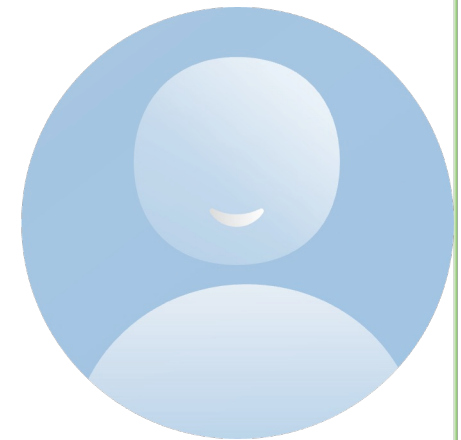
1. Inicial

- Aqui softwares, aplicativos e servidores Web são instalados manualmente.
- A infraestrutura é física ou virtualizada. Não temos consciência sobre contêineres e orquestradores de contêineres.
- O trabalho de infraestrutura e produção é manual e temos, se existirem, scripts locais usados no nível pessoal dos vários SysAdmins.



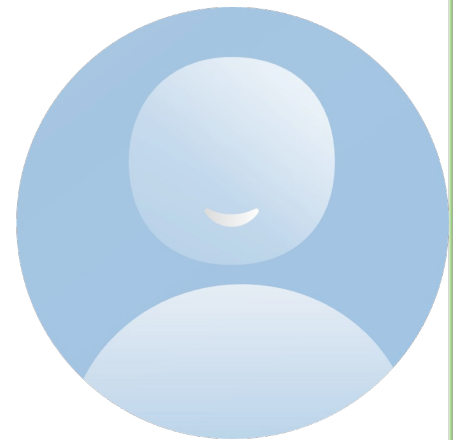
2. Consciente

- Temos consciência sobre o processo de automação de ambientes.
- Ferramentas como Docker, Kubernetes, Chef, Puppet, Ansible, Terraform e outras começam a ser experimentadas.
- Pequenos scripts de automação de ambientes começam a ser experimentados.
- A cultura de infraestrutura como aluguel (nuvens) se torna lugar comum.



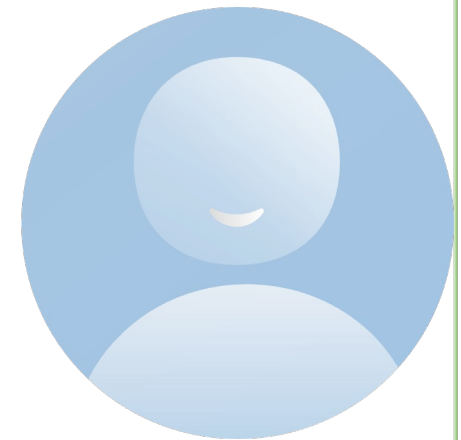
3. Gerenciado

- Temos uma suíte de ferramentas para automação de ambientes e elas são usadas em larga escala pelos times de produção.
- Times de desenvolvimento tem governança para uso de ambientes provisionados dinâmica
- O uso de Docker com repositórios privados e o uso de Kubernetes e soluções como OpenShift se torna pervasivo por toda a TI.
- Ainda não temos instrumentação do processo nem medições dos seus benefícios econômicos.

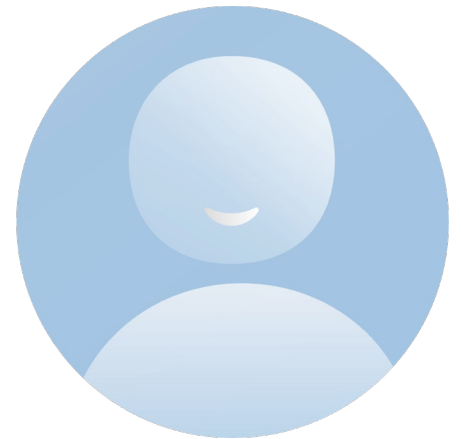


4. Quantitativamente Gerenciado

- Aqui temos automação da infraestrutura em larga escala.
- O time está apto a operar seus ambientes em estruturas virtualizadas, nuvem e até mesmo multi-nuvem, conforme direcionadores de negócio e critérios de aptidão dos seus produtos.
- O uso de plataformas como Ansible, Terraform e Kubernetes está largamente disseminado, até mesmo em ambientes de desenvolvimento.
- Práticas avançadas como Engenharia do Caos são experimentadas.
- Temos indicadores claros sobre os benefícios econômicos, temporais e de qualidade no uso da infraestrutura como código.

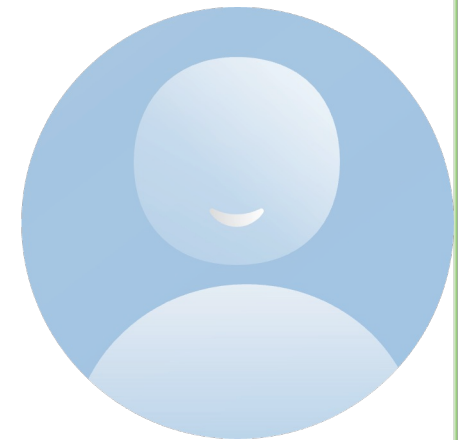


DevOps para Processos de Transformação Digital



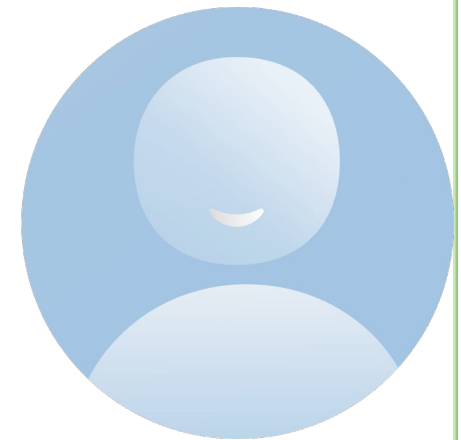
Abordagem de Transformação para Aumento de Maturidade

- A transformação DevOps é um processo contínuo de mudança organizacional.
 - *Não trate a transformação DevOps como um projeto*
- A direção da mudança é crucial para alcançar os resultados desejados.
- O foco deve estar nos resultados do sistema, não apenas em metas individuais.



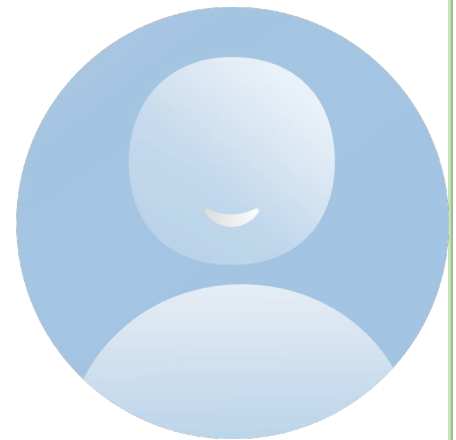
Desafios da Transformação

- Muitos programas de transformação falham devido à falta de clareza e direção.
- A transformação consome recursos e capacidade organizacional, portanto, deve ser bem planejada.
- A execução bem-sucedida da transformação requer uma compreensão clara dos objetivos e do caminho a seguir.



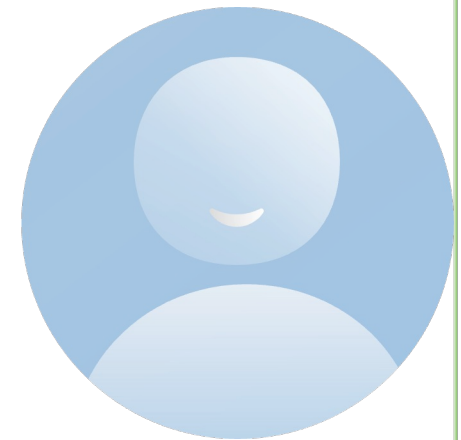
Implementação da Transformação

- A transformação envolve mudar processos organizacionais, definir metas claras e permitir a experimentação da equipe.
- A experimentação da equipe é essencial para descobrir o que funciona melhor.
- As boas práticas descobertas devem ser disseminadas em toda a organização.



Metas e Experimentação da Equipe

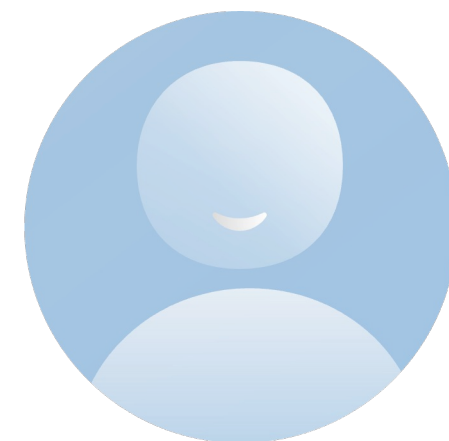
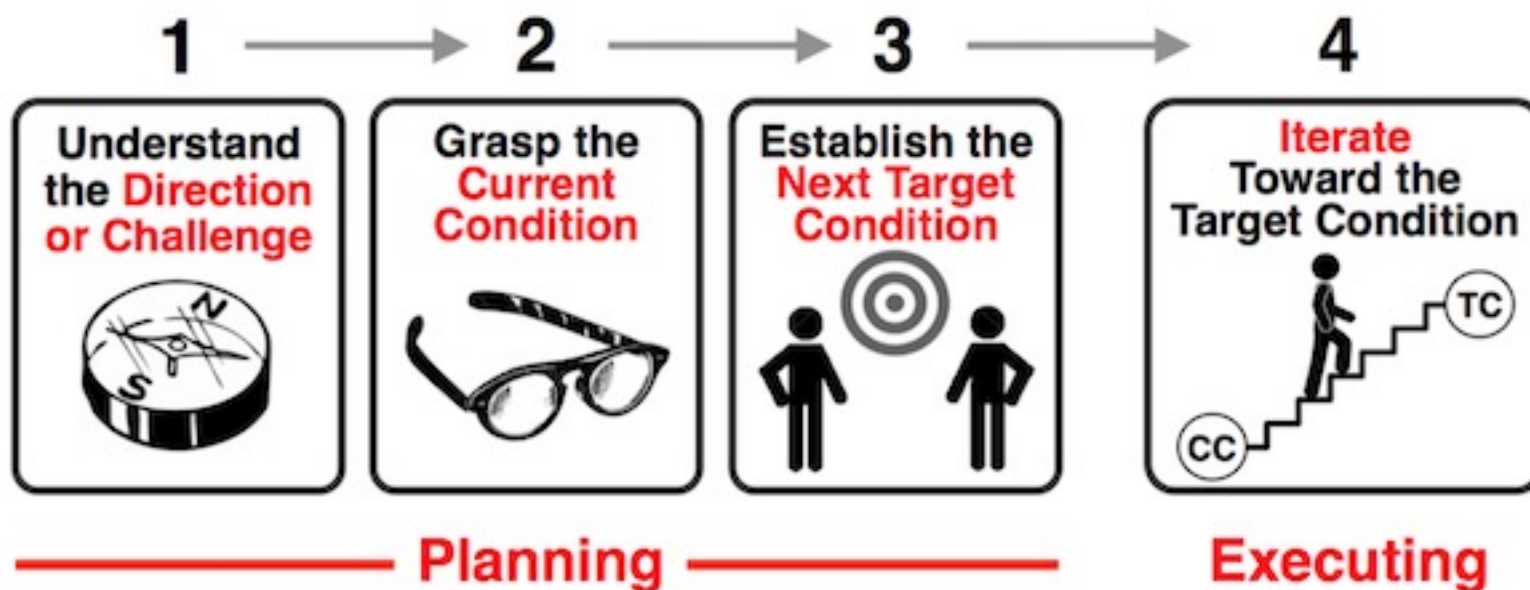
- Frameworks como o OKRs podem ajudar a definir metas claras.
- A melhoria e o coaching kata são técnicas úteis para facilitar a experimentação da equipe.
- A dinâmica básica da transformação é comum a todos os frameworks e envolve a definição de metas, experimentação e disseminação de boas práticas.



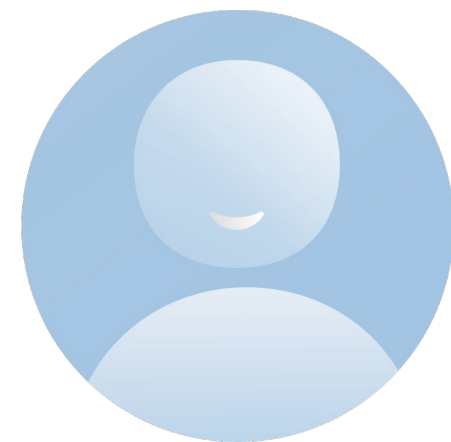
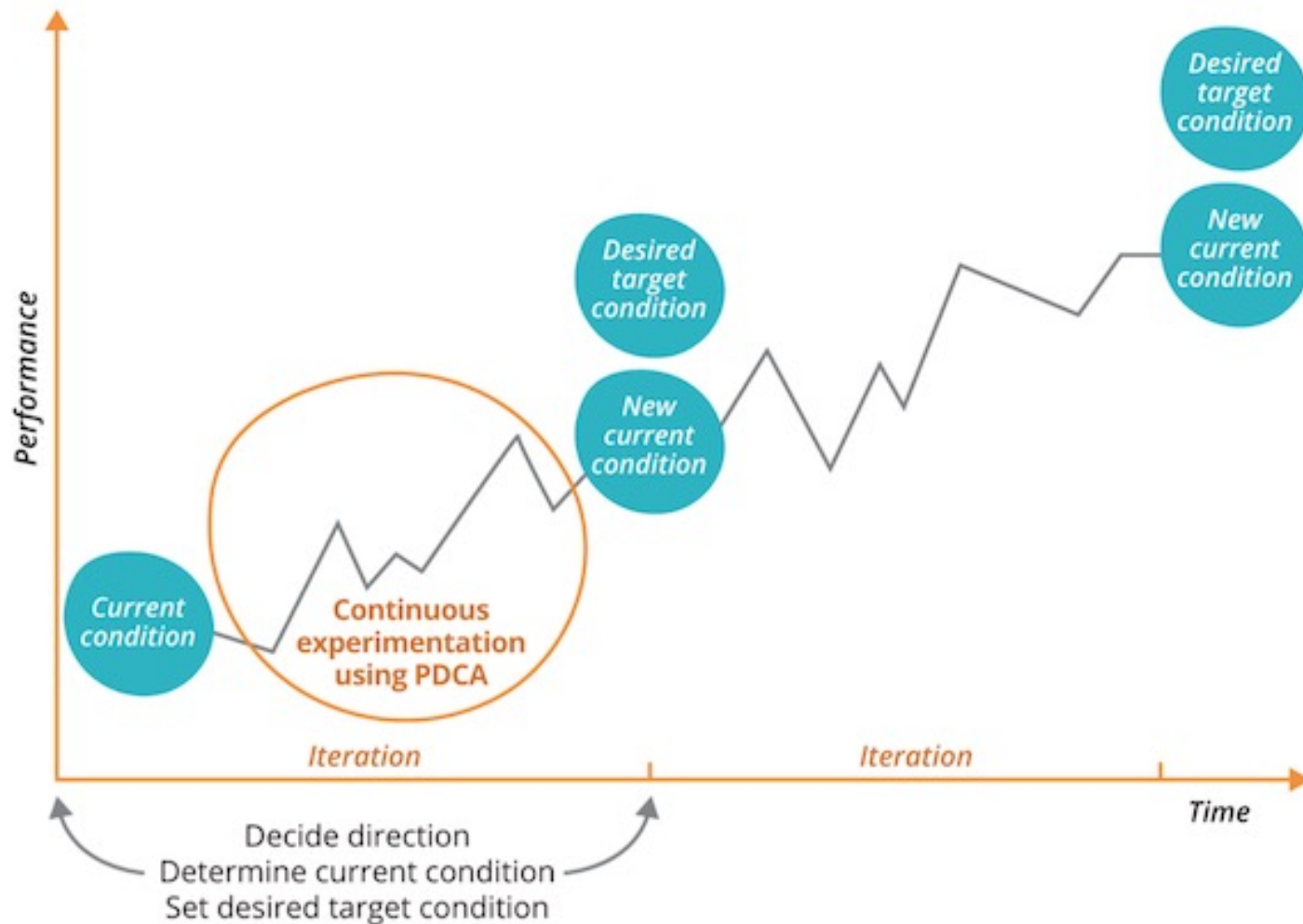
Metas e Experimentação da Equipe

THE FOUR STEPS OF THE IMPROVEMENT KATA MODEL

A systematic, scientific pattern of working



Metas e Experimentação da Equipe



DevOps para Processos de Transformação Digital

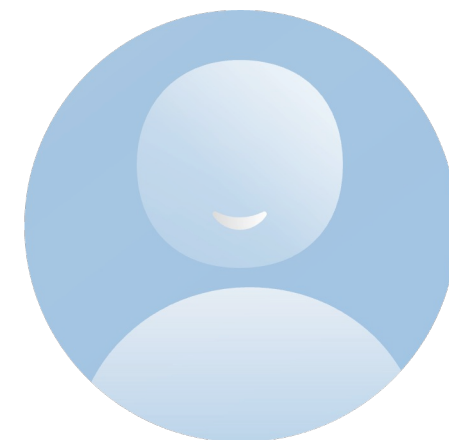
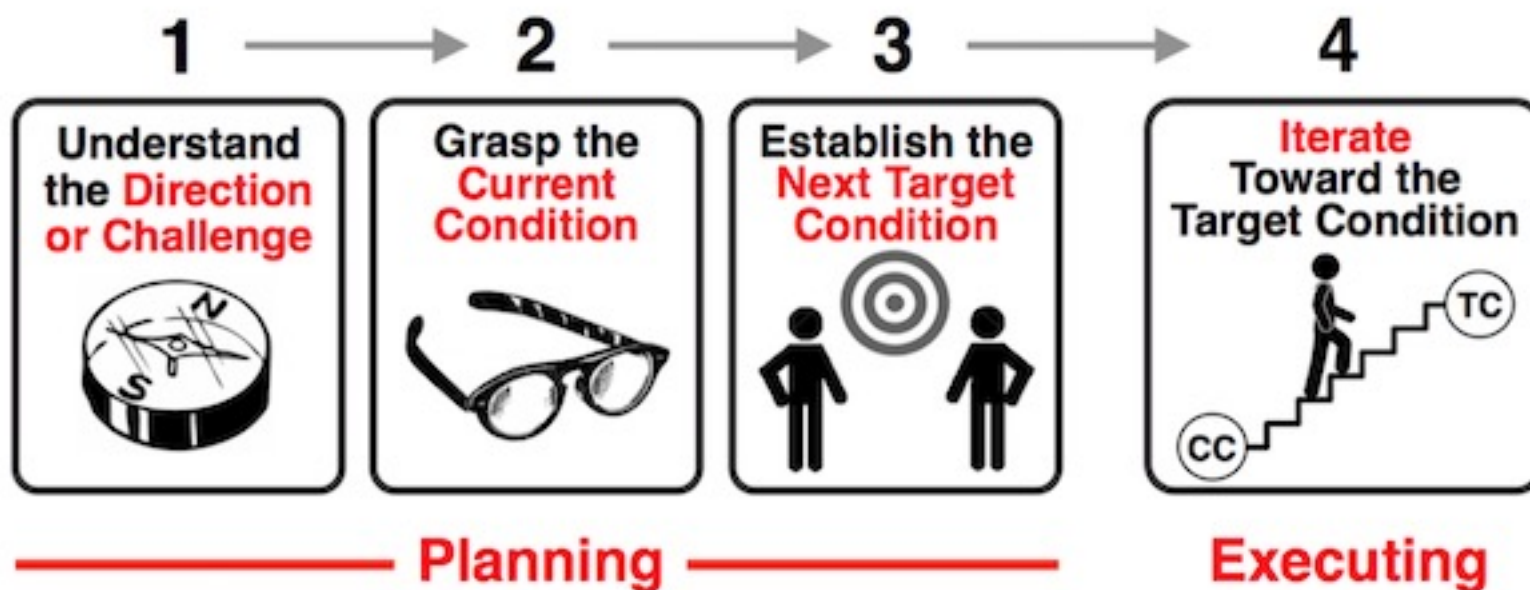
Parte 2



Metas e Experimentação da Equipe

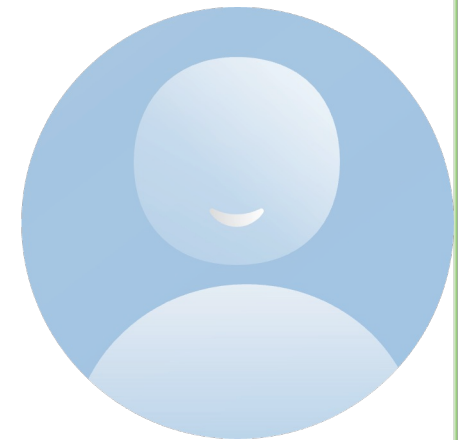
THE FOUR STEPS OF THE IMPROVEMENT KATA MODEL

A systematic, scientific pattern of working



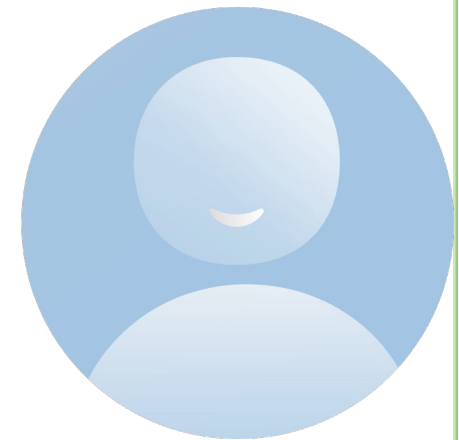
Direção da Transformação

- A direção da transformação, ou "verdadeiro norte", deve ser claramente definida e compreendida por todos na organização.
- A meta de negócios de nível de sistema serve como um guia para a transformação.
- A meta pode ser um ideal inatingível que serve como um farol, ou um objetivo desafiador, mas alcançável.



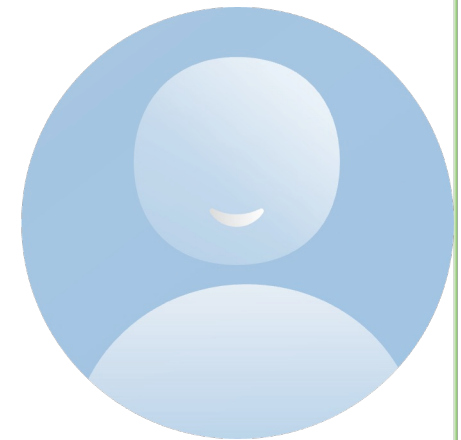
Entendendo a Condição Atual

- O modelo de maturidade apresentado no começo dessa unidade pode te ajudar a entender o nível atual de cada prática.
- *A verificação rápida de métricas DORA* pode ajudar a avaliar a condição atual da organização.
- O mapeamento do fluxo de valor pode ajudar a entender como o trabalho flui através da organização.
- A organização deve ser entendida em termos mensuráveis para que o progresso possa ser rastreado.



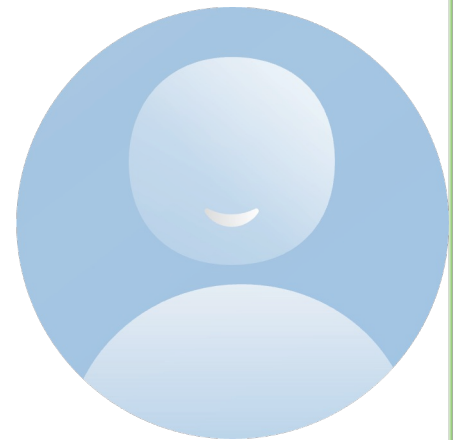
Definição de Metas Mensuráveis

- As metas devem ser definidas para uma data futura específica.
- As OKRs começam com um objetivo qualitativo, seguido por resultados-chave mensuráveis.



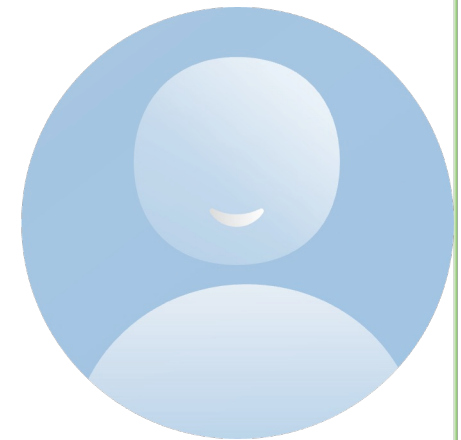
Experimentação da Equipe

- Experimentação da equipe deve seguir uma abordagem científica, testando hipóteses e aprendendo com os resultados.
- O método OODA é uma estrutura útil para a experimentação.
- O ciclo de experimentação é uma parte integral da transformação DevOps.



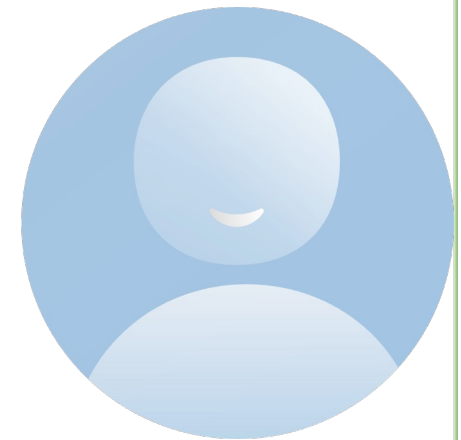
Perguntas Diárias do Equipe

- As perguntas diárias podem ajudar a equipe a entender sua condição alvo e atual, identificar obstáculos e planejar os próximos passos.
- As perguntas podem incluir: "Qual é a nossa condição alvo? O que estamos tentando alcançar? Quais obstáculos estamos enfrentando? Quais são os próximos passos?"
- A equipe deve avaliar os resultados de suas ações e aprender com eles.



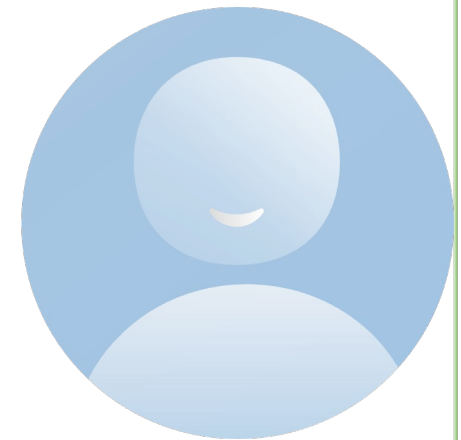
Progresso Não-Linear

- O progresso na transformação DevOps pode ser não linear devido à natureza incerta e complexa da mudança organizacional.
- As condições alvo devem ser revisadas regularmente e novas metas devem ser definidas com base nos resultados.
- A organização deve estar preparada para se adaptar e mudar de direção com base nos resultados.



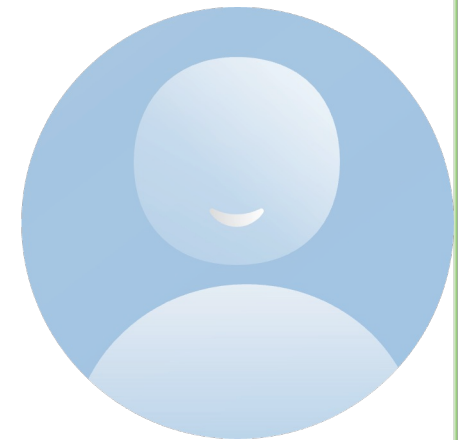
Metas da Equipe

- As metas da equipe devem ser definidas pela própria equipe, não impostas de cima para baixo.
- É aceitável não atingir todas as metas; o importante é aprender com a experiência.
- As equipes precisam de autonomia para experimentar, além de capacidade, recursos e apoio para realizar a transformação.



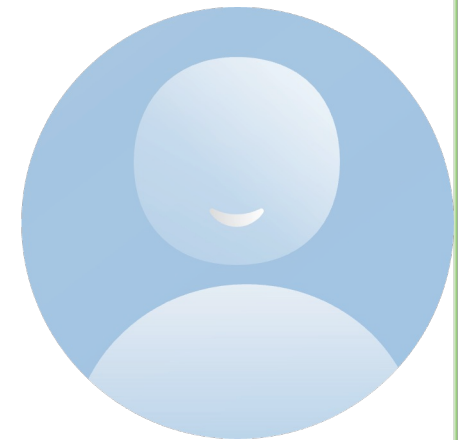
Disseminação do Conhecimento

- Depois que as equipes descobrem melhores formas de trabalhar, as lições aprendidas devem ser disseminadas em toda a organização.
- Existem várias estratégias para disseminar métodos DevOps e Agile, incluindo comunidades de prática e esforços de base.
- A disseminação do conhecimento ajuda a acelerar a transformação e a criar uma cultura de aprendizado contínuo.



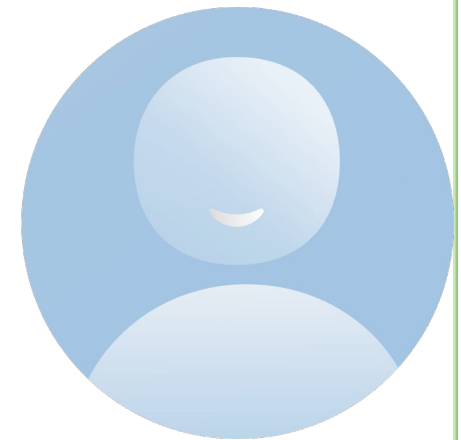
Práticas de Disseminação do Conhecimento

- Centro de treinamento (às vezes referido como um *dojo*)
- Centro de Excelência (CoE)
- Prova de conceito (PoC), mas parada
- Prova de conceito como um modelo
- Prova de conceito como uma semente
- Comunidades de prática
- De baixo para cima ou de base



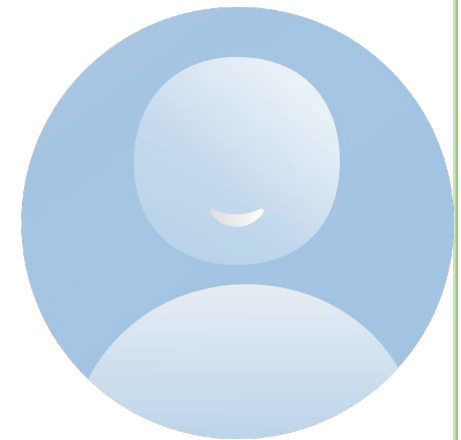
Padrões de Alta Performance

- Organizações de alto desempenho seguem padrões tais como:
 - Construtores de Comunidade,
 - Universidades Corporativa,
 - Emergente,
 - Experimentadores.
- Cada padrão tem suas próprias características e abordagens para a transformação DevOps.
- As organizações podem aprender com esses padrões e adaptá-los às suas próprias necessidades e contexto.

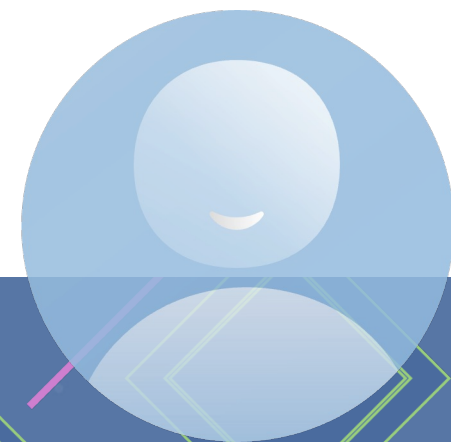


Erros comuns na transformação

- Tratar a transformação como um projeto único, em vez de um processo contínuo.
- Tratar a transformação como um esforço de cima para baixo, em vez de envolver todas as equipes.
- Falhar em concordar e comunicar o resultado pretendido, levando a confusão e falta de direção.



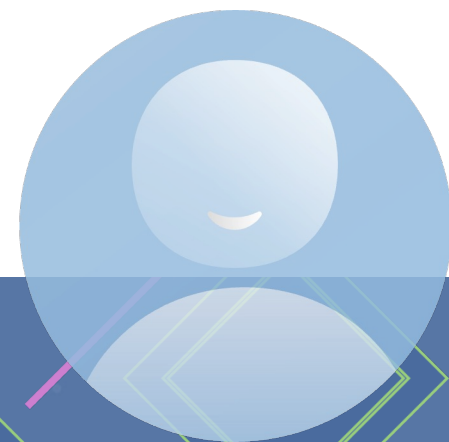
Verificação de Maturidade





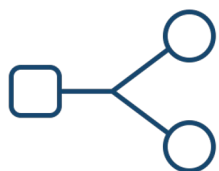
PUC Minas
Virtual

FinOps



Os desafios de uso de nuvem

A nuvem é fundamental para todos os negócios, mas é um modelo de consumo fundamentalmente diferente...



Decentralizado

Engenheiros e times de Finanças (e Compras) são empoderados em compromissos de compra



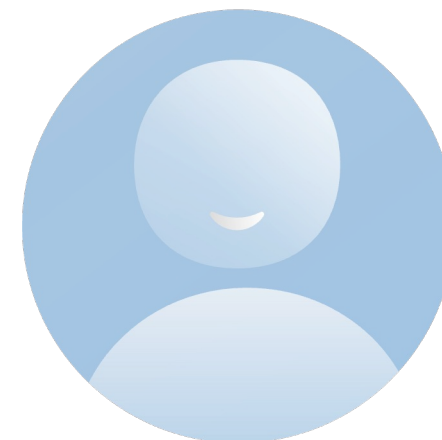
Variável

Gasto variável em nuvem substitui o data center com custo fixo



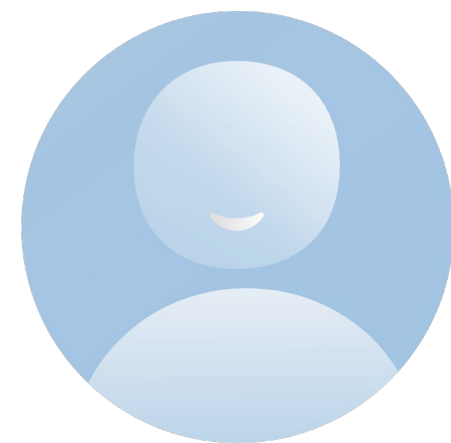
Escalável

O acesso instantâneo aos recursos permite a inovação, mas muitas vezes resulta em excesso de provisão



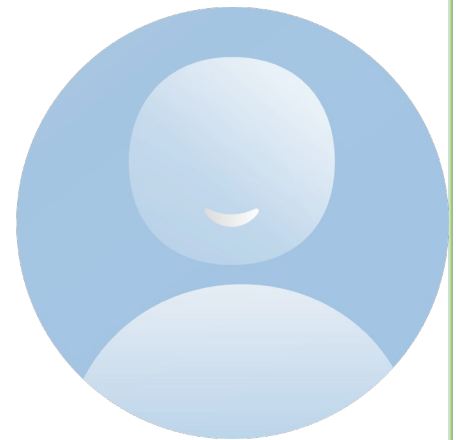
A complexidade dos custos de nuvem

- Tradicionalmente, a aquisição era em grande parte no controle de qualquer gasto de TI material porque eles tinham que aprovar todas as grandes compras de equipamentos.
- À medida que as organizações se movem para a nuvem, no entanto, o modelo pay-as-you-go — também conhecido como modelo de gasto variável — permite que os engenheiros contornem esse processo de aprovação de aquisição.
- Quando os gastos em nuvem atingem níveis materiais, as organizações são deixadas para alocar, explicar e controlar esses custos.
- Com mais de 300.000 SKUs oferecidos por provedores de serviços em nuvem, e milhares de novos recursos introduzidos a cada ano, **a gestão financeira em nuvem é um problema que não vai se resolver sozinho.**

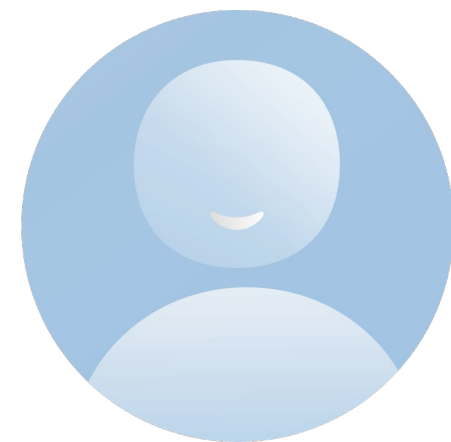


Em cena – O *FinOps*

- FinOps não é abreviação para "Operações Financeiras".
- É um junção de "Finanças" e "(Dev)Ops" enfatizando a comunicação e colaboração entre times de negócios e engenharia.



“Se parece que FinOps é sobre economizar dinheiro, então pense novamente. FinOps é sobre ganhar dinheiro.”



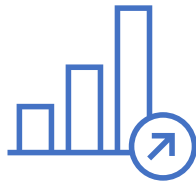
Cloud *FinOps* é uma disciplina em evolução de gestão financeira em nuvem e também uma prática cultural que:



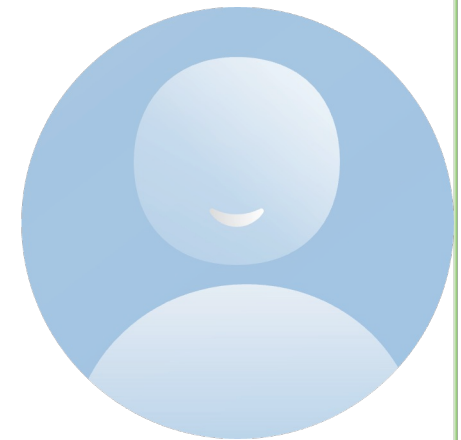
permite que as organizações obtenham o máximo de valor de negócios



ajuda equipes de engenharia, finanças e negócios

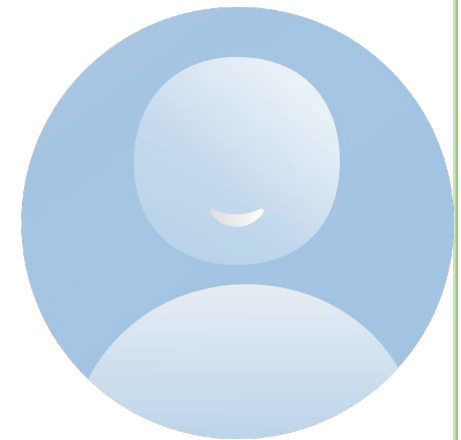


melhora a colaborar em decisões de gastos baseadas em dados



Princípios FinOps

1. As equipes precisam colaborar
2. Todo mundo toma posse para o seu uso da nuvem
3. Uma equipe centralizada impulsiona FinOps
4. Os relatórios devem ser sempre acessíveis, no tempo certo
5. As decisões são impulsionadas pelo valor dos negócios da nuvem
6. Aproveite o modelo de custo variável da nuvem



Personas envolvidas



Praticante FinOps



Executivos



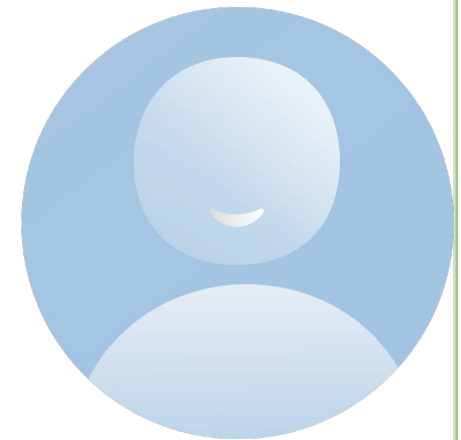
Donos de produtos/Gestores



Finanças/Compras



Times de engenharia/TI



Domínios

**Compreensão do
uso e o custo da
nuvem**

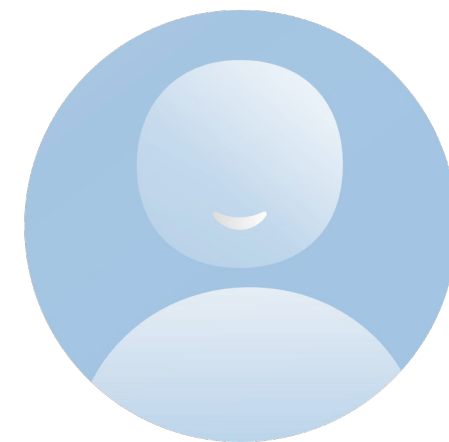
**Rastreamento de
desempenho &
benchmarking**

**Tomada de decisão
em tempo real**

**Otimização da taxa
de nuvem**

**Otimização do uso
de nuvem**

**Alinhamento
Organizacional**

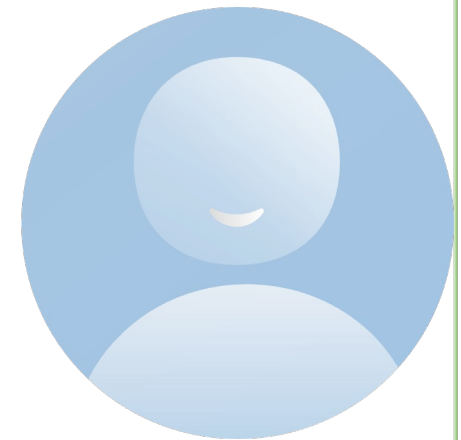


Jornada da adoção do FinOps



FinOps – Para saber mais

- Instituto FinOps
 - <https://www.finops.org>
- Calculadora AWS
 - <https://calculator.aws/#/>
- Calculadora Azure
 - <https://azure.microsoft.com/en-us/pricing/calculator/>





PUC Minas
Virtual