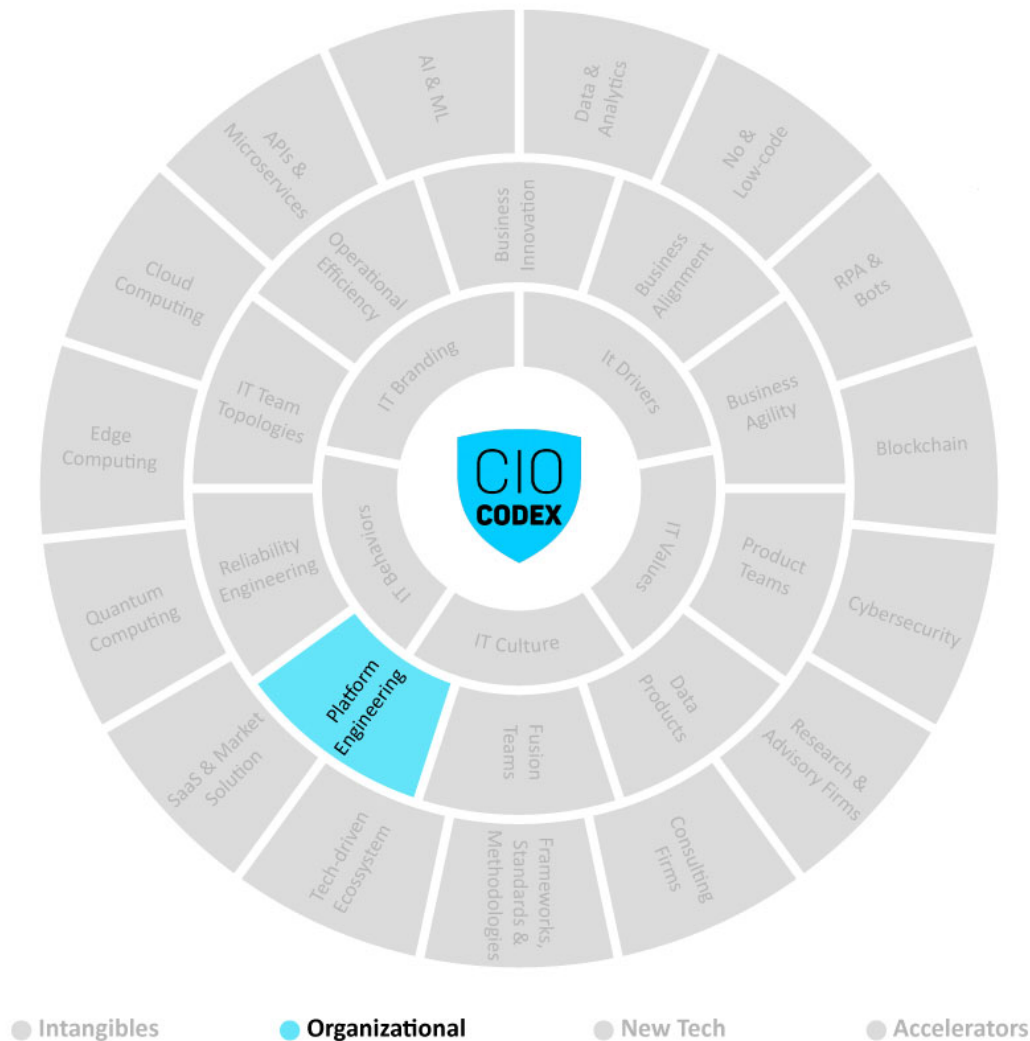




How IT can be successful

CIO Codex Agenda Framework



Platform Engineering é um conceito inovador na camada Organizational que reflete a evolução do desenvolvimento de sistemas e operações de TI.

Este modelo redefine as responsabilidades das equipes de desenvolvimento, ampliando seu escopo para incluir não só a criação, mas também a operação contínua dos sistemas em ambientes de produção.

O conteúdo aborda a abordagem de Platform Engineering e como ela promove uma mentalidade de “você constrói, você opera”, incentivando os desenvolvedores a considerarem aspectos operacionais desde o início do ciclo de vida do desenvolvimento.

A integração das responsabilidades de desenvolvimento e operação sob o mesmo teto visa a otimização do desempenho dos sistemas e a maximização da qualidade do serviço.

Este conteúdo explora as práticas de Platform Engineering, destacando como esta abordagem pode melhorar a colaboração entre as equipes, aumentar a eficiência operacional e garantir a implementação de soluções mais robustas e confiáveis.

É dada atenção especial às metodologias e ferramentas que suportam o Platform Engineering, como automação de infraestrutura, integração contínua, entrega contínua (CI/CD) e monitoramento em tempo real.

A discussão inclui como essas práticas são essenciais para a criação de plataformas que são resilientes, escaláveis e que podem ser mantidas com eficiência a longo prazo.

O conteúdo também enfrenta os desafios associados à adoção do Platform Engineering, como a necessidade de uma mudança cultural que abrace a propriedade integral do ciclo de vida do produto e o ajuste nos processos tradicionais de desenvolvimento e operações de TI.

São apresentadas estratégias para superar a resistência à mudança, gerenciar a complexidade e educar as equipes para uma nova forma de trabalhar.

A engenharia de plataforma, à primeira vista, pode parecer uma repaginação das práticas estabelecidas de DevSecOps.

O grande diferencial do Platform Engineering está na ideia de acelerar o ciclo de desenvolvimento e segurança operacional através de uma estrutura mais organizada e eficiente, bastante inspirada nos conceitos do “Team Topologies”.

Esta abordagem foca em aumentar a produtividade e a qualidade por meio da especialização dos times.

Times dedicados ao desenvolvimento de software são apoiados por equipes de plataforma, que são especialistas em fornecer os componentes estruturais necessários, como arquitetura e pipelines automatizados de DevSecOps.

Este modelo é essencialmente uma extensão lógica do que já é praticado em DevSecOps, onde a segurança é integrada ao ciclo de vida do desenvolvimento de software desde o início.

O que a engenharia de plataforma introduz é uma camada adicional de especialização e suporte, permitindo que as equipes de desenvolvimento se concentrem mais na lógica de negócios e menos na infraestrutura e na conformidade de segurança, que são gerenciadas por uma equipe dedicada.

Sob a perspectiva de escala, a engenharia de plataforma faz muito sentido.

Ela oferece uma maneira estruturada de lidar com as crescentes demandas por software em um ambiente empresarial cada vez mais dependente de tecnologia.

Ao criar equipes especializadas que podem focar exclusivamente em suas áreas de expertise, reduzimos o gargalo que frequentemente ocorre quando os desenvolvedores precisam navegar entre as complexidades da infraestrutura e da segurança.

Além disso, a engenharia de plataforma pode aumentar a agilidade das organizações ao permitir uma entrega mais rápida e eficiente de software.

A integração e entrega contínuas (CI/CD), fundamentais para o DevOps e DevSecOps, são aprimoradas sob o modelo de engenharia de plataforma, pois as ferramentas e processos são padronizados e otimizados por equipes que entendem profundamente as nuances e requisitos técnicos.

A adoção geral de práticas de engenharia de plataforma no mercado parece não só sensata, mas inevitável.

Empresas que buscam escalar suas operações de desenvolvimento sem sacrificar a qualidade ou a segurança verão na engenharia de plataforma uma solução atraente.

Isso se alinha com a tendência geral de especialização e divisão de trabalho, onde equipes altamente focadas podem atingir objetivos específicos com maior eficiência e eficácia.

Além disso, em um mercado que valoriza a rapidez e a adaptabilidade, a capacidade de responder rapidamente às mudanças, ao mesmo tempo em que se mantém a integridade e a segurança do sistema, é mais crucial do que nunca.

A engenharia de plataforma facilita esta dinâmica ao reduzir a carga sobre as equipes de desenvolvimento, permitindo-lhes manter o foco na inovação e na criação de valor.

Por fim, este conteúdo destaca a importância de estabelecer métricas de sucesso claras para as iniciativas de Platform Engineering, tais como a estabilidade do sistema, a frequência de lançamentos bem-sucedidos e a satisfação do usuário final.

O objetivo é prover uma visão clara sobre como o Platform Engineering pode ser um elemento transformador no modelo operacional de TI, entregando sistemas e plataformas que não apenas atendem às necessidades atuais, mas são projetados para a adaptabilidade e sucesso futuros.

Visão prática

Na prática, a engenharia de plataforma (Platform Engineering) representa um divisor de águas na maneira como as organizações desenvolvem e operam sistemas.

Sua premissa central - “you build it, you run it” - redefine a responsabilidade dos times de desenvolvimento, atribuindo-lhes o dever de operar e sustentar os sistemas que criam.

Essa abordagem traz benefícios claros: maior agilidade, confiabilidade operacional e alinhamento com as demandas de negócio.

Plataformas como Alicerces da Inovação

A prática de Platform Engineering destaca a criação de plataformas de autoatendimento como um dos pilares fundamentais.

Essas plataformas oferecem ferramentas, pipelines automatizados e serviços padronizados que permitem aos desenvolvedores provisionarem recursos e lançar aplicações sem depender de outras equipes.

Por exemplo, um catálogo de serviços em nuvem que possibilita a criação de ambientes de teste em minutos, sem burocracia, reduz drasticamente o tempo de espera e acelera o desenvolvimento.

Isso não só promove eficiência como também melhora a experiência dos desenvolvedores, permitindo que eles se concentrem em entregar valor ao negócio.

Transformação Operacional na Prática

Implementar Platform Engineering requer a adoção de práticas modernas e ferramentas robustas. Entre os elementos-chave da transformação prática estão:

- Automação Extensiva: Infraestrutura como Código (IaC) e pipelines de CI/CD garantem consistência e reduzem erros manuais.
- Monitoramento e Observabilidade: Ferramentas especializadas são essenciais para fornecer insights em tempo real sobre o

desempenho dos sistemas.

- **Segurança Integrada (DevSecOps):** Políticas de segurança automatizadas e verificações contínuas garantem a conformidade sem comprometer a velocidade.
- **Testes de Resiliência:** Simulações de falhas utilizando práticas como Chaos Engineering ajudam a validar a robustez e a resiliência dos sistemas.
- **Integração Contínua com Feedback:** Loops contínuos de feedback entre desenvolvimento e operações permitem que aprendizados operacionais retroalimentem o design de sistemas.

Essas práticas criam um ciclo de desenvolvimento mais eficiente e resiliente, transformando a TI em um parceiro estratégico confiável para o negócio.

Cultura de Colaboração: A Base do Sucesso

Na prática, a engenharia de plataforma só é bem-sucedida quando sustentada por uma cultura colaborativa.

Desenvolvedores, operadores, especialistas em segurança e stakeholders de negócios precisam trabalhar juntos para garantir que os sistemas atendam às necessidades do negócio enquanto mantêm altos padrões de qualidade e segurança.

Um exemplo prático dessa colaboração é a criação de times multifuncionais inspirados no conceito de “Team Topologies”.

Equipes de plataforma, como “enabling teams”, oferecem suporte e capacitação para que as equipes de desenvolvimento possam operar de forma autônoma, mas dentro de padrões e diretrizes estabelecidos.

Superando Desafios no Mundo Real

Embora a engenharia de plataforma traga inúmeros benefícios, sua implementação não está isenta de desafios.

Entre os mais comuns estão:

- **Resistência à Mudança:** Times acostumados a operar de forma segmentada podem resistir à integração das responsabilidades. Abordagens como treinamento contínuo e mentorias ajudam a superar essa resistência.
- **Complexidade Operacional:** A gestão de múltiplas plataformas requer um planejamento meticuloso e ferramentas integradas para evitar redundâncias e inconsistências.
- **Cultura de Propriedade:** Incorporar o senso de propriedade integral nos desenvolvedores exige uma transformação cultural que valorize aprendizado e accountability.
- **Monitoramento e Escalabilidade:** Garantir que as plataformas sejam escaláveis e tenham observabilidade suficiente para evitar problemas antes que impactem os usuários.

Uma estratégia prática para lidar com esses desafios é a adoção de indicadores de maturidade que avaliem continuamente o progresso, identificando lacunas e oportunidades de melhoria.

Indicadores de Sucesso na Prática

O sucesso da engenharia de plataforma deve ser medido por métricas claras que reflitam seu impacto operacional e estratégico. Exemplos incluem:

- **Tempo médio para recuperação (MTTR):** Mede a eficiência na resolução de problemas operacionais.
- **Frequência de lançamentos:** Reflete a capacidade de iterar rapidamente com qualidade.
- **Adesão às plataformas de autoatendimento:** Indica o nível de autonomia e adoção pelos times de desenvolvimento.
- **Satisfação dos desenvolvedores:** Avaliada por pesquisas internas, como o Developer Net Promoter Score (DNPS).
- **Conformidade com padrões de segurança:** Percentual de builds

aprovados em verificações de segurança automatizadas.

Esses indicadores permitem que as organizações monitorem o impacto real da engenharia de plataforma e ajustem continuamente suas estratégias.

Criando o Futuro com Platform Engineering

A prática de Platform Engineering está pavimentando o caminho para um futuro em que as operações de TI são mais ágeis, resilientes e alinhadas às demandas de negócios.

Ao adotar essa abordagem, as empresas não só aumentam sua eficiência operacional, mas também habilitam seus desenvolvedores a inovar de forma mais rápida e consistente.

Com plataformas robustas, uma cultura de colaboração e ferramentas modernas, a engenharia de plataforma posiciona as organizações para navegar com sucesso no cenário digital acelerado e competitivo de hoje.

Evolução Cronológica

O conceito de Platform Engineering é uma disciplina emergente no domínio da Tecnologia da Informação, reconhecendo a crescente complexidade dos sistemas e a necessidade de uma abordagem holística que abrange tanto a criação quanto a operação de sistemas em ambientes de produção.

A seguir é explorada a evolução cronológica do Platform Engineering, destacando como essa abordagem tem sido desenvolvida e ajustada ao longo do tempo para enfrentar os desafios de um ambiente de negócios em constante evolução.

1) - Início e Evolução do Platform Engineering (Anos 2000 - 2010)

- **Origem e Primeiros Passos:** No início dos anos 2000, a necessidade de alinhar desenvolvimento e operações começou a ganhar força. Inicialmente, as equipes de desenvolvimento e operações trabalhavam de forma segmentada, o que gerava

atrasos e ineficiências. Com o tempo, a metodologia DevOps começou a emergir, estabelecendo as bases para o conceito de Platform Engineering. Esta fase marcou o início da conscientização sobre a importância da sinergia entre desenvolvimento e operações.

- Primeiras Experiências: As primeiras iniciativas focaram em integrar processos de desenvolvimento e operações, promovendo a colaboração e a comunicação contínua entre as equipes. A premissa de “you build it, you run it” começou a ser implementada, incentivando os desenvolvedores a assumirem responsabilidades operacionais, aumentando a conscientização sobre as implicações de suas decisões de design e código.

2) - Expansão e Maturidade do Platform Engineering (Anos 2010 - 2020)

- Consolidação da Mentalidade DevOps: Durante os anos 2010, a metodologia DevOps consolidou-se, estabelecendo princípios fundamentais que deram suporte ao Platform Engineering. A necessidade de uma abordagem holística que abrange a criação e operação de sistemas tornou-se evidente, e as organizações começaram a investir em ferramentas e plataformas que suportassem a automação e a orquestração eficientes.
- Desenvolvimento de Plataformas Autosserviço: Nesta fase, as plataformas autosserviço começaram a ganhar destaque, permitindo que os desenvolvedores provisionassem recursos, configurassem ambientes e implementassem aplicações com autonomia. Isso não apenas acelerou o ciclo de vida de desenvolvimento de software, mas também aumentou a confiança nas soluções implementadas. A infraestrutura como código e a orquestração de contêineres tornaram-se elementos chave, permitindo que os sistemas fossem iterados, escalados e mantidos com confiança.

3) - Implementação e Consolidação do Platform Engineering (2020 - Presente)

- Mudança de Mentalidade e Ferramentas Avançadas: A

implementação eficaz do Platform Engineering requer uma mudança de mentalidade significativa. Os desenvolvedores devem internalizar e priorizar a operacionalização durante todas as fases do desenvolvimento do sistema. Ferramentas avançadas para monitoramento, telemetria, segurança e compliance tornaram-se essenciais, garantindo que os sistemas sejam sustentáveis e resilientes no longo prazo.

- **Alinhamento de Criação e Operação:** Ao alinhar de perto a criação e operação de sistemas, as organizações ganham em agilidade, qualidade e desempenho. A adoção de práticas como DevSecOps e a infraestrutura como código tornou-se vital, garantindo que a segurança e a rastreabilidade das mudanças sejam incorporadas desde o início. A governança técnica sólida e uma abordagem de FinOps asseguram que os recursos tecnológicos sejam utilizados de maneira eficiente, sustentável e econômica.

4) - Reflexões e Desafios Futuros do Platform Engineering

- **Transformação Contínua:** A transição para o modelo de Platform Engineering é menos uma questão de “se” e mais uma de “quando” e “como”. As organizações devem estar preparadas para enfrentar os desafios da integração profunda entre desenvolvimento e operações, investindo em formação e desenvolvimento contínuo das equipes.
- **Capacitação e Inovação:** O futuro pertence às organizações que conseguem efetivamente integrar suas equipes de desenvolvimento e operações, capitalizando o conhecimento e a criatividade dispersos por toda a empresa, e transformando esses recursos em inovação e agilidade. A capacidade de resposta rápida às mudanças do mercado será um diferencial competitivo crucial.

Platform Engineering está redefinindo o ecossistema de TI, pois ao alinhar de perto a criação e operação de sistemas, as organizações ganham em agilidade, qualidade e desempenho.

Com o desenvolvedor no centro desta transformação, a prática está se tornando um componente vital para empresas que buscam inovar e competir em um mundo digital acelerado.

As preocupações operacionais agora são uma consideração primária e não uma reflexão tardia, garantindo que a tecnologia não apenas atenda às necessidades atuais, mas também seja sustentável e resiliente no longo prazo.

Conceitos e Características

A Engenharia de Plataforma é uma disciplina emergente que responde à crescente complexidade dos sistemas modernos e à necessidade de alinhar criação e operação de software em um único ciclo contínuo.

Seu princípio central “you build it, you run it” incentiva desenvolvedores a assumirem também a responsabilidade operacional, promovendo maior consciência sobre as implicações de design e código em ambientes de produção.

Essa abordagem redefine a forma como as equipes de TI trabalham, unindo desenvolvimento e operações em torno de plataformas que oferecem autoatendimento, padronização e automação, ao mesmo tempo em que preservam governança, segurança e eficiência.

Mais do que um conjunto de ferramentas, a Engenharia de Plataforma é uma estratégia organizacional que possibilita maior agilidade, resiliência e qualidade, colocando o desenvolvedor no centro da transformação digital.

A Platform Engineering está redefinindo o ecossistema de TI, uma vez que ao integrar criação e operação em torno de plataformas robustas e autosserviço, proporciona agilidade, eficiência, resiliência e sustentabilidade no longo prazo.

Para empresas que buscam competir em um mundo digital acelerado, ela se tornou uma disciplina vital, alinhando inovação, governança e responsabilidade compartilhada.

A seguir, estão os principais conceitos e características que fundamentam o Platform Engineering.

Definição e Propósito

Engenharia de Plataforma é a prática de desenvolver e manter uma base comum de

código, ferramentas e processos que podem ser reutilizados por várias equipes de desenvolvimento. O objetivo é reduzir redundância, otimizar recursos e facilitar a escalabilidade das aplicações.

Componentes Reutilizáveis e Autoatendimento

Um dos pilares do modelo é a criação de bibliotecas, APIs, microserviços e interfaces que possam ser reutilizados em diferentes projetos. As plataformas de autoatendimento permitem que desenvolvedores provisionem recursos e configurem ambientes sem depender de suporte operacional, aumentando autonomia e velocidade.

Infraestrutura como Código (IaC)

A prática de gerenciar infraestrutura por meio de código substitui processos manuais, trazendo precisão, rastreabilidade e automação. O IaC suporta pipelines de CI/CD, permitindo que mudanças em infraestrutura sejam versionadas, testadas e entregues com a mesma fluidez que o código de aplicação.

Padronização e Governança

A Engenharia de Plataforma define padrões claros para desenvolvimento, segurança, monitoramento e operações. Esses padrões garantem consistência e qualidade em toda a organização. A governança assegura que tais práticas sejam aplicadas, equilibrando inovação com conformidade regulatória e política interna.

Cultura de DevOps e Colaboração

Platform Engineering fortalece a cultura de DevOps, unindo desenvolvedores, operadores e stakeholders em torno de uma colaboração contínua. Ferramentas e práticas integradas ao ciclo de vida do software facilitam entregas rápidas, aprendizado contínuo e resolução eficiente de problemas.

Monitoramento Contínuo e Automação

A observabilidade e o monitoramento contínuo são fundamentais para manter a saúde da plataforma. Aliados à automação, permitem detectar e corrigir falhas rapidamente, além de antecipar riscos e prevenir incidentes antes que afetem o ambiente produtivo.

Segurança Integrada

Na Engenharia de Plataforma, a segurança é incorporada desde o início do ciclo de desenvolvimento. Políticas automatizadas, testes contínuos e práticas de DevSecOps asseguram que a proteção dos sistemas seja tratada como parte essencial do processo, e não como uma etapa posterior.

Propósito e Objetivos

O propósito do Platform Engineering é estabelecer um modelo operacional no qual as equipes de desenvolvimento assumem uma responsabilidade integral pelo ciclo de vida dos sistemas que criam, desde a concepção até a produção e manutenção contínua.

Este modelo busca harmonizar o processo de desenvolvimento de software com as operações, resultando numa maior eficiência e qualidade dos serviços de TI fornecidos.

Objetivos do Platform Engineering:

- **Automação e Orquestração:** Implementar ferramentas e práticas que automatizem o processo de desenvolvimento e operações, reduzindo o tempo de colocação no mercado e os erros manuais.
- **Desenvolvimento Sustentável:** Garantir que os sistemas sejam projetados com sustentabilidade operacional em mente, otimizando recursos e garantindo a escalabilidade.
- **Colaboração Multidisciplinar:** Fomentar uma cultura de colaboração entre desenvolvedores, operadores e outros stakeholders, para garantir uma visão compartilhada e alinhamento de objetivos.
- **Monitoramento Contínuo:** Estabelecer um sistema de monitoramento proativo que permita antecipar e resolver problemas antes que eles impactem os usuários.
- **Qualidade e Confiabilidade:** Assegurar que todos os sistemas e serviços atendam a padrões rigorosos de qualidade e confiabilidade desde o início.

- **Feedback e Melhoria Contínua:** Criar um loop de feedback entre as equipes de desenvolvimento e operações, para a melhoria contínua do produto e do processo.
- **Capacitação e Conhecimento Compartilhado:** Desenvolver programas de treinamento e documentação que permitam o compartilhamento de conhecimento e melhores práticas.
- **Gestão de Configuração:** Implementar práticas de gestão de configuração que garantam a consistência dos ambientes de desenvolvimento, teste e produção.
- **Preparação para a Mudança:** Construir sistemas e processos que possam se adaptar rapidamente a novos requisitos, tecnologias e condições de mercado.
- **Segurança Integrada:** Integrar a segurança no processo de desenvolvimento, garantindo que as considerações de segurança sejam uma parte contínua do ciclo de vida do software.

Adotar a engenharia de plataforma é uma estratégia para criar uma base sólida que apoia a inovação, a agilidade e a excelência operacional, reconhecendo que o sucesso a longo prazo no mercado de software depende tanto da qualidade dos produtos quanto da eficiência e eficácia das operações.

Roadmap de Implementação

Para o tema de Platform Engineering dentro do contexto organizacional, a implementação de um roadmap robusto que guie a transformação operacional é de suma importância.

O foco deve estar na criação de sistemas robustos e na garantia de sua eficácia operacional após a implementação. A abordagem deve ser holística, considerando não apenas o desenvolvimento, mas também a operacionalização contínua e sustentável dos sistemas.

A seguir, detalham-se as etapas propostas para o Roadmap de Implementação, ancoradas nos novos conceitos de responsabilidades ampliadas dos times de

desenvolvimento.

Platform Engineering busca harmonizar a criação e a gestão de sistemas, enfatizando a importância de considerar todo o ciclo de vida do produto desde a concepção.

Este paradigma propõe que as equipes não sejam apenas construtoras, mas também guardiãs dos sistemas, garantindo operação e manutenção eficientes.

Principais Etapas da Implementação:

Definição de Responsabilidades

- Estabelecer claramente as responsabilidades dos times de desenvolvimento, que devem abranger tanto a criação quanto a manutenção dos sistemas.

Capacitação e Cultura

- Investir em treinamento para fomentar uma cultura de responsabilidade completa sobre os sistemas, incluindo aspectos operacionais e de produção.

Infraestrutura e Ferramentas

- Prover as ferramentas necessárias para o desenvolvimento e a operação de sistemas, como plataformas de integração contínua e entrega contínua (CI/CD).

Desenvolvimento Guiado por Operações

- Incentivar práticas de desenvolvimento que levem em conta a operacionalização desde o início, como configuração como código e infraestrutura imutável.

Monitoramento e Feedback

- Implementar sistemas de monitoramento que forneçam feedback em tempo real sobre a performance e a saúde dos sistemas.

Iteração e Melhoria Contínua

- Criar ciclos de revisão e iteração para a melhoria contínua dos sistemas, aplicando aprendizados operacionais ao processo de desenvolvimento.

Alinhamento Estratégico

- Assegurar que a engenharia da plataforma esteja alinhada com a estratégia global da empresa, garantindo que os sistemas suportem os objetivos de negócio.

Governança e Compliance

- Definir políticas para governança, risco e compliance, garantindo que a plataforma e os sistemas sejam seguros e estejam em conformidade com regulamentos.

Escalabilidade e Resiliência

- Preparar sistemas para serem escaláveis e resilientes, capazes de se adaptar e se recuperar rapidamente de falhas.

Avaliação de Desempenho

- Estabelecer métricas para avaliar o desempenho das plataformas e a eficácia das equipes, levando em conta tanto critérios técnicos quanto de negócios.

Integração e Colaboração

- Fomentar a integração e a colaboração entre as equipes de desenvolvimento e operações, criando um ambiente de trabalho sinérgico.

Adoção de Práticas de SRE E DevSecOps

- Adotar práticas e disciplinas de Site Reliability Engineering (SRE) e DevSecOps para garantir que a operacionalização dos sistemas seja tão robusta quanto sua criação.

Este roadmap deve ser dinâmico, permitindo ajustes conforme evoluem as tecnologias e as necessidades operacionais.

A visão é criar não apenas um conjunto de sistemas funcionais, mas uma plataforma integrada e viva que respalde os objetivos de negócios e propicie um ciclo contínuo de inovação e melhoria.

Melhores Práticas de Mercado

Na vanguarda das práticas organizacionais, o Platform Engineering tem emergido como um pilar essencial na interseção da criação e da operação de sistemas.

Este domínio, essencial na camada Organizacional, abarca a responsabilidade integral dos times de tecnologia em todo o ciclo de vida dos sistemas, da concepção à produção.

As melhores práticas recomendadas pelo mercado atual são fundamentais para garantir a eficiência, a estabilidade e a inovação contínua dos serviços de TI.

Platform Engineering é uma disciplina que se concentra em estruturar equipes de TI para assumirem responsabilidades ampliadas, abrangendo tanto o desenvolvimento quanto a operação de sistemas.

Esta abordagem promove uma maior compreensão e atenção às necessidades de operação desde o início do desenvolvimento, assegurando que a transição de sistemas para produção seja suave e alinhada com os objetivos de negócios.

Práticas Recomendadas:

- Desenvolvimento Orientado para Operações (Ops-First Development): As equipes devem desenvolver com um foco preventivo em operações, considerando aspectos como monitoramento, escalabilidade e manutenção desde o início do desenvolvimento.
- Automação de Infraestrutura como Código (IaC): Implementar

infraestrutura como código para gerenciar e provisionar recursos de TI através de scripts automatizados, melhorando a consistência e a eficiência operacional.

- **Integração e Entrega Contínuas (CI/CD):** Adotar práticas de CI/CD para acelerar o ciclo de lançamento de software enquanto mantém a qualidade e a estabilidade dos sistemas.
- **Monitoramento e Observabilidade Avançados:** Utilizar ferramentas sofisticadas para monitorar sistemas em produção, proporcionando insights profundos e permitindo uma resposta rápida a incidentes.
- **Cultura de SRE (Site Reliability Engineering):** Incorporar práticas de SRE para equilibrar a necessidade de lançar novos recursos rapidamente com a importância de manter a confiabilidade do sistema.
- **Escalabilidade e Performance Design:** Assegurar que os sistemas são projetados para serem escaláveis e performáticos, atendendo às crescentes demandas sem comprometer a experiência do usuário.
- **Testes de Carga e Simulação de Falhas (Chaos Engineering):** Realizar testes de carga regulares e simulações de falhas para garantir a resiliência e a recuperação rápida dos sistemas.
- **Documentação e Conhecimento Compartilhado:** Manter uma documentação abrangente e atualizada, facilitando a transferência de conhecimento e a colaboração entre equipes.
- **Segurança Integrada (DevSecOps):** Integrar segurança em todas as fases do ciclo de vida do desenvolvimento, garantindo que as considerações de segurança sejam uma prioridade constante.
- **Foco na Experiência do Desenvolvedor (DX):** Criar ambientes de desenvolvimento que incentivem a inovação, com ferramentas e processos que simplificam e aceleram o trabalho dos desenvolvedores.
- **Feedback Operacional no Desenvolvimento:** Encorajar uma

retroalimentação contínua entre as equipes de operações e desenvolvimento para melhorar continuamente os produtos.

- **FinOps e Governança de Custo:** Implementar práticas de FinOps para otimizar os custos de operação, garantindo a eficiência financeira sem sacrificar o desempenho ou a escalabilidade.
- **Parcerias com Fornecedores de Cloud e Software:** Trabalhar estreitamente com fornecedores para garantir que as soluções escolhidas estejam alinhadas com as melhores práticas e tendências da indústria.

As práticas de Platform Engineering são essenciais para organizações que buscam não apenas desenvolver, mas operar sistemas de TI que sejam resilientes, escaláveis e alinhados com os objetivos estratégicos da empresa.

Ao adotar essas práticas recomendadas, as organizações podem assegurar que estão não apenas acompanhando, mas liderando na corrida pela inovação tecnológica e excelência operacional.

Desafios Atuais

No atual panorama organizacional, a engenharia de plataformas, conhecida como Platform Engineering, surge como um pilar crítico para a sustentação de iniciativas de transformação digital.

Ela representa uma evolução do pensamento em desenvolvimento e operações de sistemas, refletindo a crescente complexidade e a necessidade de soluções robustas, escaláveis e continuamente adaptáveis.

A engenharia de plataformas busca endereçar o desafio de integrar e alinhar desenvolvimento, operações e negócios. A responsabilidade compartilhada em todo o ciclo de vida do sistema – do design à operação – requer novas competências, ferramentas e mentalidades.

A seguir são explorados alguns dos principais desafios atuais:

Cultura de Propriedade Completa

- Criar uma cultura onde os desenvolvedores são responsáveis não apenas pelo desenvolvimento, mas também pela operação e manutenção dos sistemas.

DevOps e SRE

- Integrar práticas de DevOps e Site Reliability Engineering (SRE) para garantir a confiabilidade, a performance e a eficiência das plataformas em produção.

Observabilidade

- Desenvolver sistemas com observabilidade desde o início, para que as equipes possam ter visibilidade e insights sobre o desempenho e a saúde dos sistemas em tempo real.

Automação

- Ampliar a automação para abranger não só a integração e entrega contínuas, mas também a recuperação automática e a gestão de infraestrutura.

Segurança Integrada

- Incorporar segurança no pipeline de desenvolvimento para garantir que as plataformas sejam seguras por design.

Plataformas Como Produto

- Tratar as plataformas como produtos, com uma equipe dedicada e um ciclo de vida gerenciado, requerendo uma abordagem focada no usuário final.

Escalar com Controle

- Permitir que as plataformas escalem de acordo com as necessidades do negócio, mantendo um controle rigoroso sobre a

qualidade e a governança.

Habilidades e Treinamento

- Assegurar que as equipes tenham as habilidades necessárias para abraçar esta nova forma de trabalho, o que pode exigir uma requalificação significativa.

Decisões Baseadas em Dados

- Empregar dados e métricas para informar decisões de engenharia e operacionais, visando aprimorar continuamente a eficiência e a experiência do usuário.

Colaboração Multifuncional

- Fomentar uma colaboração efetiva entre as equipes de engenharia, operações e negócios para garantir que as plataformas atendam às necessidades estratégicas da organização.

Estes desafios realçam a importância de uma visão holística que transcenda as fronteiras tradicionais entre desenvolvimento e operações, exigindo uma sinergia entre estratégia, tecnologia e execução.

Confrontar estes desafios é fundamental para que as organizações se posicionem de maneira competitiva e resiliente no mercado dinâmico atual.

Tendências para o Futuro

Na vanguarda da inovação organizacional, o tema da Engenharia de Plataforma representa um campo vital para as tendências futuras de TI, especialmente no que concerne a responsabilidade integral dos times de desenvolvimento.

As tendências para o futuro neste domínio são delineadas por uma crescente ênfase na integração das fases de criação e operação de sistemas, refletindo um modelo

operacional onde a visão end-to-end é essencial.

A seguir, encontram-se as tendências projetadas para a Engenharia de Plataforma:

- Integração DevOps: Ampliação da adoção de práticas DevOps, promovendo uma cultura de colaboração contínua entre desenvolvedores e operadores de TI.
- Automação e Orquestração: Investimento em ferramentas de automação e orquestração para facilitar a entrega contínua e a gestão de infraestrutura como código.
- Plataformas como Produto: Tratamento de plataformas como produtos, com equipes dedicadas à sua evolução e manutenção, semelhante à gestão de produtos de software.
- Engenharia de Confiabilidade do Site (SRE): A implementação de práticas de Engenharia de Confiabilidade do Site (SRE) será mais difundida, garantindo alta disponibilidade e desempenho dos sistemas em produção.
- Foco na Experiência do Usuário: Aumento do foco na experiência do usuário final, inclusive no estágio de design da plataforma, para garantir que os sistemas sejam intuitivos e eficientes.
- Microserviços e Arquitetura Sem Servidor: Progressão para arquiteturas baseadas em microserviços e sem servidor para promover escalabilidade e agilidade operacional.
- Monitoramento e Observabilidade: Desenvolvimento de capacidades avançadas de monitoramento e observabilidade para permitir insights rápidos e ação proativa na operação de sistemas.
- Segurança Integrada: Incorporação de segurança em todas as fases do ciclo de vida do desenvolvimento de software, do planejamento à operação.
- Inteligência Artificial e Aprendizado de Máquina: Utilização crescente de IA e aprendizado de máquina para otimizar a operação de plataformas e prever problemas antes que eles ocorram.

- **Cultura de Propriedade de Código:** Promoção de uma cultura onde os desenvolvedores assumem a propriedade total do código, desde a escrita até a produção e além.
- **Gestão de Configuração Baseada em Políticas:** Implementação de gestão de configuração baseada em políticas, permitindo que as práticas de governança sejam aplicadas de forma automática e consistente.

Estas tendências apontam para um futuro em que os times de Engenharia de Plataforma operam não apenas como construtores de sistemas, mas como proprietários integrais do ciclo de vida do produto.

Isso exigirá uma revisão profunda dos modelos operacionais atuais, uma mudança de mentalidade dos desenvolvedores e uma maior colaboração interdepartamental.

A convergência de desenvolvimento e operação facilitará a entrega contínua de valor, a otimização do desempenho operacional e a alavancagem estratégica das capacidades tecnológicas.

KPIs Usuais

Platform Engineering, como conceito, implica uma responsabilidade end-to-end sobre os sistemas de tecnologia, desde a sua criação até a operação em produção.

Essa abordagem integral exige que as equipes de desenvolvimento e operações trabalhem em estreita colaboração, orientadas por métricas que garantam a eficiência, a estabilidade e a adaptabilidade dos sistemas.

Os Key Performance Indicators (KPIs) seguintes são fundamentais para gerenciar efetivamente essa responsabilidade compartilhada e assegurar que as práticas de Platform Engineering estejam alinhadas com os objetivos organizacionais:

- **Tempo de Lead para Mudanças (Lead Time for Changes):** Mede o tempo desde a concepção até a implementação efetiva de uma mudança no sistema, refletindo a agilidade da equipe.
- **Frequência de Implantação de Releases:** Indica a capacidade da equipe de entregar novas funcionalidades e atualizações de forma

consistente e confiável.

- Taxa de Falhas de Implantação (Deployment Failure Rate): Monitora a proporção de deployments que resultam em falhas, necessitando rollback ou ação corretiva imediata.
- Tempo Médio para Recuperação (Mean Time to Recovery, MTTR): Reflete a rapidez com que a equipe pode restaurar o serviço após uma falha ou interrupção.
- Disponibilidade e Uptime: Avalia a proporção do tempo em que o sistema está operacional e acessível aos usuários, um indicador chave de confiabilidade.
- Taxa de Erros em Produção (Error Rate): Quantifica a frequência de erros ou bugs que ocorrem em um ambiente de produção.
- Eficiência de Processamento de Dados: Mede a capacidade do sistema de processar dados de maneira eficiente, um fator crítico para sistemas baseados em dados.
- Nível de Adoção de Práticas de DevOps: Avalia a integração e aplicação de práticas de DevOps, fundamentais para a colaboração entre desenvolvimento e operações.
- Automatização de Testes: Percentual de testes automatizados em relação ao total, uma métrica importante para a qualidade contínua do código.
- Índice de Performance de Aplicação (Application Performance Index, Apdex): Mede a satisfação dos usuários com o tempo de resposta e a performance do sistema.
- Custo por Ticket de Suporte: Analisa a eficiência da equipe ao lidar com tickets de suporte, considerando o custo associado.
- Contribuição para a Redução de Custos Operacionais: Quantifica o impacto das melhorias de plataforma na redução de custos operacionais gerais.
- Índice de Automação de Infraestrutura: Mede o grau de automação na infraestrutura, indicando o nível de maturidade e

eficiência operacional.

- **Engajamento e Satisfação da Equipe de Platform Engineering:** Reflete o moral e a motivação da equipe, fundamentais para a inovação contínua e o sucesso do projeto.
- **Conformidade com Padrões de Segurança e Regulamentações:** Garante que os sistemas estejam em conformidade com as normas de segurança da indústria e regulamentações governamentais.

Estes KPIs fornecem uma estrutura robusta para monitorar e melhorar continuamente os sistemas de tecnologia sob a responsabilidade dos times de Platform Engineering.

Eles permitem que as organizações avaliem o sucesso de sua abordagem operacional, garantindo que os sistemas não apenas atendam às necessidades atuais, mas também estejam preparados para a evolução futura, oferecendo flexibilidade, escalabilidade e segurança em um mundo tecnológico que muda rapidamente.

Exemplos de OKRs

Para o tema Platform Engineering da camada Organizational, os OKRs devem focar na construção e otimização de plataformas tecnológicas que habilitem desenvolvimento rápido e eficaz, além de suportar operações escaláveis.

Aqui estão alguns exemplos de OKRs que podem ser implementados:

Objetivo 1: Desenvolver e manter plataformas de engenharia robustas para suportar crescimento escalável.

- **KR1:** Aumentar a automação na integração e entrega contínua (CI/CD) em 30% até o final do trimestre.
- **KR2:** Implementar uma nova arquitetura de microserviços para dois produtos principais, melhorando a modularidade.
- **KR3:** Garantir 99,9% de tempo de atividade para todas as plataformas internas de engenharia.

Objetivo 2: Melhorar a eficiência do desenvolvimento de software através de plataformas de engenharia.

- KR1: Reduzir o tempo de ciclo de desenvolvimento de novas funcionalidades em 25%.
- KR2: Aumentar a reutilização de código em 40% com a implementação de um repositório de serviços compartilhados.
- KR3: Realizar treinamentos trimestrais sobre as melhores práticas de plataforma para 100% das equipes de engenharia.

Objetivo 3: Assegurar a conformidade e segurança em todas as plataformas de engenharia.

- KR1: Realizar auditorias de segurança mensais e remediar todas as vulnerabilidades críticas dentro de uma semana após a detecção.
- KR2: Implementar políticas de conformidade automatizadas que reduzam os riscos de não conformidade em 50%.
- KR3: Alcançar zero violações de segurança de dados nos próximos dois trimestres.

Objetivo 4: Fomentar a inovação por meio das plataformas de engenharia.

- KR1: Lançar um programa de hackathon interno semestral para explorar novas tecnologias e abordagens.
- KR2: Implementar 5 novas tecnologias emergentes nas plataformas existentes que contribuam para os objetivos de negócios.
- KR3: Criar um fundo de inovação que permita a experimentação rápida de ideias promissoras por parte das equipes de engenharia.

Objetivo 5: Maximizar a satisfação e a produtividade do desenvolvedor com as plataformas de engenharia.

- KR1: Melhorar a pontuação de satisfação do desenvolvedor em relação às ferramentas e plataformas em 20%.
- KR2: Reduzir o número de interrupções operacionais impactando desenvolvedores em 50%.
- KR3: Introduzir autoatendimento em ambientes de desenvolvimento e teste, aumentando a agilidade do desenvolvedor em 30%.

Esses OKRs ajudam a orientar o foco da plataforma de engenharia para a melhoria contínua, qualidade, e alinhamento com os objetivos estratégicos da empresa, garantindo que as plataformas ofereçam valor, segurança e suporte eficaz para as necessidades de negócios e tecnologia.

Critérios para Avaliação de Maturidade

Abaixo verificam-se critérios de maturidade para avaliar a implementação de Platform Engineering na camada Organizational, usando uma estrutura inspirada no CMMI.

Aqui estão cinco critérios para cada um dos cinco níveis de maturidade: Inexistente, Inicial, Definido, Gerenciado e Otimizado no contexto de Platform Engineering:

Nível de Maturidade: Inexistente

- Falta de Estratégia de Platform Engineering: A organização não possui uma estratégia formal para o desenvolvimento e operação de plataformas tecnológicas.
- Ausência de Padrões de Engenharia de Plataforma: Não existem padrões ou melhores práticas definidos para a engenharia de plataformas tecnológicas.
- Infraestrutura Descentralizada: A infraestrutura é gerenciada de forma descentralizada, sem considerar a consolidação de recursos.
- Monitoramento Limitado: Não há sistemas eficazes de

monitoramento e gerenciamento de plataformas tecnológicas.

- Falta de Automação de Tarefas: As tarefas de engenharia de plataforma são executadas manualmente, sem automação.

Nível de Maturidade: Inicial

- Conscientização sobre Platform Engineering: A organização reconhece a importância da engenharia de plataforma, mas a implementação é esporádica.
- Experimentação com Padrões: Iniciativas de engenharia de plataforma estão em estágios iniciais de experimentação e aplicação de padrões.
- Papéis e Responsabilidades Iniciais: Papéis e responsabilidades iniciais para a equipe de engenharia de plataforma são definidos.
- Colaboração Emergente: A colaboração entre equipes de desenvolvimento e plataforma começa a surgir, mas ainda é ad hoc.
- Monitoramento Básico: A organização inicia a implementação de sistemas de monitoramento básico de plataforma.

Nível de Maturidade: Definido

- Estratégia de Platform Engineering: A organização define uma estratégia clara para o desenvolvimento e operação de plataformas tecnológicas.
- Padrões e Melhores Práticas Estabelecidos: Padrões e melhores práticas são estabelecidos para a engenharia de plataforma.
- Equipe de Engenharia de Plataforma: Uma equipe dedicada é responsável pelo desenvolvimento e operação de plataformas tecnológicas.
- Colaboração Estruturada: A colaboração entre equipes de desenvolvimento e plataforma é formalizada com processos e

ferramentas definidos.

- Monitoramento Avançado: Sistemas avançados de monitoramento e gerenciamento de plataforma são implementados.

Nível de Maturidade: Gerenciado

- Melhoria Contínua: A organização busca continuamente melhorar a eficácia e eficiência da engenharia de plataforma.
- Avaliação de Desempenho: Métricas de desempenho são usadas para avaliar o sucesso da engenharia de plataforma.
- Automação de Processos: Processos de engenharia de plataforma são automatizados para maior eficiência.
- Segurança e Conformidade: A segurança e conformidade são integradas às práticas de engenharia de plataforma.
- Integração com a Estratégia Organizacional: A engenharia de plataforma está totalmente alinhada com a estratégia global da organização e contribui significativamente para os objetivos organizacionais.

Nível de Maturidade: Otimizado

- Inovação em Platform Engineering: A organização promove a inovação constante nas práticas e processos de engenharia de plataforma.
- Análise de Dados Estratégicos: A análise de dados é usada para tomar decisões estratégicas relacionadas à engenharia de plataforma.
- Evolução Contínua da Estratégia: A estratégia de engenharia de plataforma é adaptada de acordo com as mudanças nas necessidades do mercado.
- Cultura de Excelência: A cultura organizacional favorece a excelência na engenharia de plataforma como um componente

crítico de operações.

- Liderança de Vanguarda: Líderes de engenharia de plataforma demonstram liderança de vanguarda na implementação das melhores práticas.

Esses critérios fornecem uma estrutura para avaliar a maturidade da implementação de Platform Engineering na camada Organizational, permitindo que a organização avance em direção a práticas mais eficazes na criação e operação de plataformas tecnológicas para apoiar seus objetivos estratégicos.