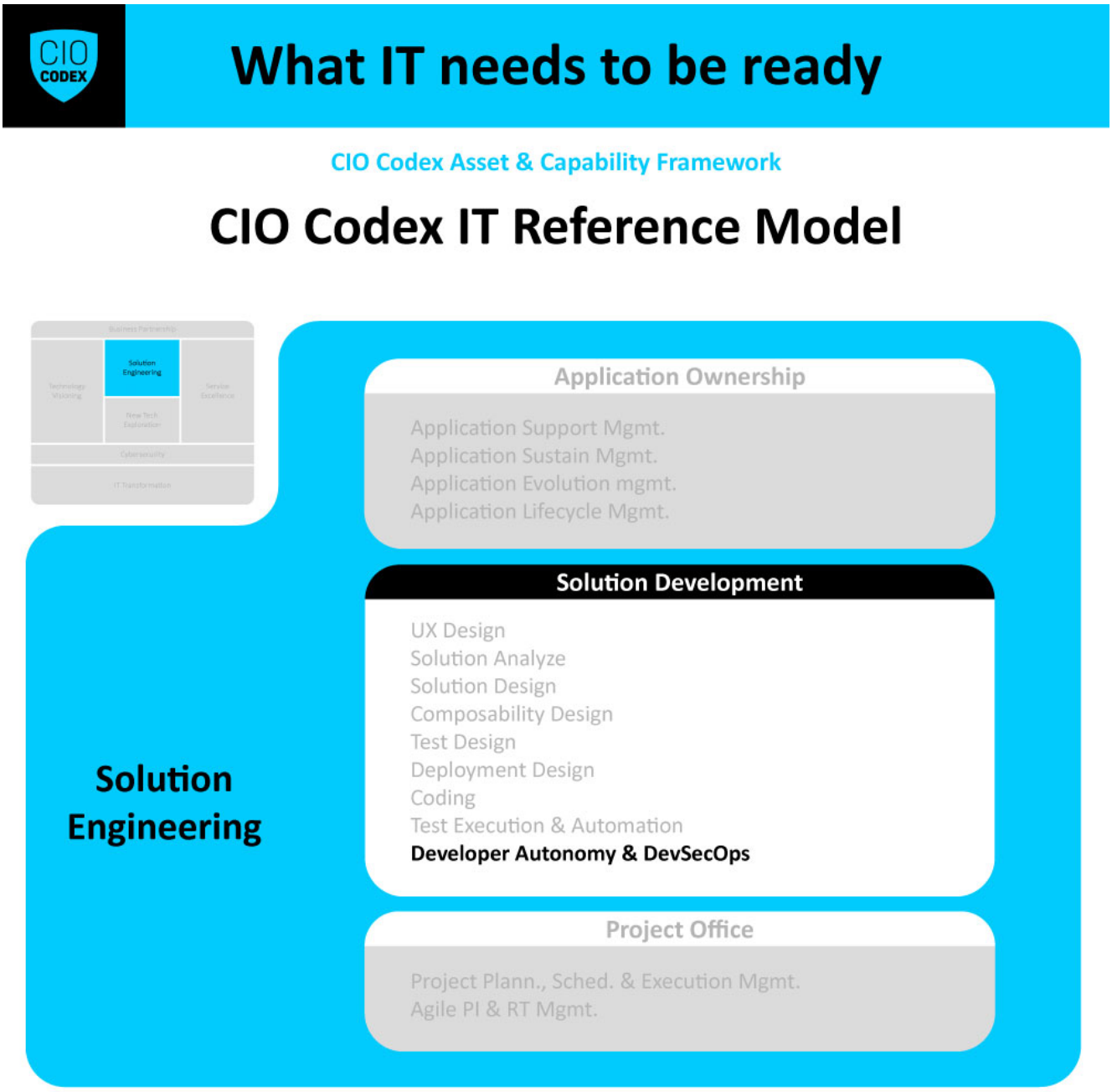

Dentro do escopo do CIO Codex Capability Framework, a capability Developer Autonomy & DevSecOps é reconhecida como um pilar central na construção de soluções de TI resilientes, seguras e adaptáveis.

Tal capability habilita os desenvolvedores a exercerem maior autonomia e responsabilidade, especialmente em relação à integração de práticas de segurança no ciclo de desenvolvimento do software, refletindo a essência da abordagem DevSecOps.

A autonomia do desenvolvedor é a liberdade concedida aos profissionais de TI para tomar decisões e realizar ações de forma independente, um aspecto que impulsiona a agilidade, a criatividade e a inovação no desenvolvimento de software.

Esta autonomia, entretanto, não ocorre de forma isolada, mas sim dentro de um ecossistema estruturado que oferece processos claros, ferramentas adequadas e suporte contínuo para tomada de decisão embasada e segura.

Em consonância, o DevSecOps é uma metodologia que incorpora segurança como um elemento fundamental desde o início do desenvolvimento operacional de software, integrando equipes de desenvolvimento (Dev), segurança (Sec) e operações (Ops).

Tal integração garante que a segurança não seja apenas uma camada adicional, mas um componente intrínseco ao processo de desenvolvimento, promovendo soluções mais robustas e menos suscetíveis a vulnerabilidades.

Neste cenário, as práticas de Integração Contínua (CI) e Entrega Contínua (CD) são vitais, automatizando a construção, teste e implantação de software, assegurando assim atualizações frequentes, confiáveis e alinhadas às diretrizes de segurança e conformidade.

A cultura de colaboração, promovida por esta capability, dissolve os silos operacionais tradicionais, incentivando uma interação estreita e contínua entre as equipes de desenvolvimento, operações e segurança para garantir a entrega de software eficaz, ágil e seguro.

Uma evolução significativa desta capability é a adoção de recursos baseados em inteligência artificial no processo de desenvolvimento, que vêm ampliando de forma substancial a autonomia, a qualidade e a produtividade das equipes técnicas.

Essas plataformas de IA atuam como assistentes contextuais ao longo do ciclo de desenvolvimento, sugerindo soluções, automatizando tarefas repetitivas, identificando potenciais falhas de segurança e promovendo a conformidade com padrões de codificação e arquitetura.

O uso disciplinado e estratégico desses recursos permite acelerar a geração de valor, reduzindo a ocorrência de erros e promovendo a consistência nas entregas técnicas.

Além disso, ao fornecer insights contextuais durante a codificação, testes e revisões, essas tecnologias possibilitam que os desenvolvedores tenham maior clareza nas decisões técnicas, sem depender exclusivamente de especialistas para cada etapa do processo.

Isso representa um avanço importante no empoderamento técnico, uma vez que permite ampliar o escopo de atuação dos desenvolvedores e reduzir gargalos operacionais nas esteiras de entrega.

As características proeminentes desta capability incluem a disponibilização de ferramentas avançadas de automação que permitem a execução de integração e

entrega contínuas, bem como a implementação de testes de segurança automatizados e avaliações dinâmicas de risco.

Programas de treinamento e capacitação são oferecidos aos desenvolvedores para habilitá-los nas práticas DevSecOps, assim como nas competências emergentes relacionadas ao uso seguro e ético de tecnologias baseadas em IA no desenvolvimento de sistemas.

Um fluxo constante de feedback entre os desenvolvedores, equipes de operações e segurança é vital para o aperfeiçoamento contínuo dos processos e das práticas adotadas.

Além disso, a promoção de uma cultura organizacional onde a segurança é prioridade e onde o uso de novas tecnologias é guiado por governança responsável, contribui para o desenvolvimento de software mais seguro, sustentável e alinhado com os objetivos estratégicos da organização.

No contexto do CIO Codex, o propósito da Developer Autonomy & DevSecOps é criar um ambiente onde os desenvolvedores são capacitados com as ferramentas, informações e processos necessários para incorporar considerações de segurança e eficiência desde o início do ciclo de desenvolvimento.

Isso não só promove a qualidade e a produtividade no desenvolvimento de software, mas também garante que as soluções sejam seguras por design e entregues com maior velocidade e confiabilidade.

Os objetivos desta capability são claros: empoderar os desenvolvedores com a autonomia necessária para a tomada de decisões técnicas e de segurança, integrar práticas de segurança de forma contínua ao longo do desenvolvimento de software, implementar automação inteligente para testes e validações, promover a cultura DevSecOps e elevar os níveis de eficiência, qualidade e segurança nas entregas de software.

O impacto da Developer Autonomy & DevSecOps nas dimensões tecnológicas é profundo. Requer uma infraestrutura capaz de suportar automações inteligentes, pipelines integrados de CI/CD e ambientes que sustentem a colaboração contínua entre áreas multidisciplinares.

Influencia diretamente a arquitetura da solução, que deve prever a integração de práticas de segurança desde o design, assim como prever pontos de auditoria e testes automatizados ao longo do ciclo de vida.

Garante a funcionalidade adequada dos sistemas com foco em resiliência operacional e mitigação proativa de vulnerabilidades cibernéticas.

Além disso, influencia o modelo operacional ao fomentar uma mentalidade colaborativa, de melhoria contínua e aprendizado permanente, características fundamentais para sustentar a inovação tecnológica com responsabilidade e segurança.

A capability Developer Autonomy & DevSecOps foi concebida para prover uma visão abrangente e detalhada de sua relevância dentro do CIO Codex Framework, destacando seu papel estratégico na criação de soluções tecnológicas seguras, modernas e sustentáveis.

Ela não só permite que as equipes de TI atinjam excelência nos processos de desenvolvimento, como também estabelece um novo patamar de referência para o uso responsável de tecnologias emergentes no ecossistema de engenharia de software, em prol do sucesso e da competitividade organizacional.

Conceitos e Características

A capability de Developer Autonomy & DevSecOps capacita desenvolvedores a integrar segurança desde o início do ciclo de desenvolvimento, promovendo a abordagem DevSecOps como uma disciplina estruturante da entrega moderna de software.

Ela se fundamenta em conceitos como autonomia técnica, práticas de integração e entrega contínuas, uso inteligente de tecnologias emergentes e uma cultura de colaboração ampliada, com o objetivo de garantir que o software seja desenvolvido com segurança, eficiência e adaptabilidade.

Nesse contexto, a adoção de ferramentas cognitivas baseadas em inteligência artificial reforça ainda mais o papel estratégico desta capability, viabilizando maior fluidez, assertividade e responsabilidade ao longo do ciclo de desenvolvimento.

Conceitos

- Developer Autonomy: Refere-se à capacidade dos desenvolvedores de tomar decisões e realizar ações independentes, com base em dados, boas práticas e ferramentas que apoiem sua atuação técnica. Essa autonomia impulsiona a agilidade e estimula a inovação, reduzindo a dependência de ciclos hierárquicos longos e favorecendo a entrega contínua de valor ao

negócio.

- DevSecOps: É uma abordagem que integra a segurança (Sec) desde o início (Dev) até a operação (Ops) do ciclo de vida do software, transformando a segurança em um componente nativo do processo. Ao incorporar práticas preventivas desde as fases iniciais, evita-se que vulnerabilidades se propaguem até os ambientes produtivos, otimizando a qualidade e confiabilidade das soluções entregues.
- Integração Contínua e Entrega Contínua (CI/CD): São práticas fundamentais que automatizam a construção, testes, validação e implantação de software. Facilitam a realização de ciclos curtos e frequentes de entrega, permitindo identificar falhas mais rapidamente e garantir atualizações confiáveis, com menor risco de regressão ou impacto ao negócio.
- Cultura de Colaboração: Promove uma atuação sinérgica e contínua entre os times de desenvolvimento, segurança e operações, eliminando barreiras organizacionais e fomentando um ambiente onde a responsabilidade compartilhada pela segurança e pela qualidade é uma premissa.
- Inteligência Artificial Aplicada ao Desenvolvimento: Representa o uso de algoritmos avançados e modelos de linguagem para apoiar o desenvolvedor em decisões técnicas, revisões de código, sugestão de padrões, identificação de inconsistências e reforço da segurança desde a escrita inicial. A IA, integrada ao ciclo de desenvolvimento, permite ampliar a visão contextual do desenvolvedor e aumentar sua produtividade com responsabilidade.

Características

- Ferramentas de Automação Inteligente: Disponibiliza um conjunto de ferramentas que vão além da automação tradicional. Incluem recursos capazes de interpretar contexto, sugerir melhorias e executar ações com base em padrões de segurança, performance e aderência arquitetural. Tais ferramentas impulsionam a consistência e a escalabilidade do processo de entrega.
- Treinamento e Capacitação Contínua: Oferece programas estruturados de capacitação técnica que abrangem tanto as práticas fundamentais de

DevSecOps quanto o uso seguro e eficiente de tecnologias emergentes, como inteligência artificial aplicada ao desenvolvimento. A formação contínua prepara os times para operar em um ambiente dinâmico, orientado por inovação e resiliência.

- Avaliação de Riscos Dinâmica: Realiza avaliações contínuas de riscos ao longo de todo o ciclo de desenvolvimento, incorporando mecanismos automatizados para detecção de vulnerabilidades em tempo real e utilizando heurísticas e modelos de aprendizado para antecipar possíveis brechas ou desvios de conformidade.
- Feedback Contínuo e Contextualizado: Estabelece canais estruturados de feedback entre desenvolvedores, equipes de operações e segurança, possibilitando ajustes imediatos nas soluções e na estratégia de entrega. A introdução de insights baseados em IA auxilia na retroalimentação qualificada, reduzindo ruídos de comunicação e aumentando a eficácia da melhoria contínua.
- Cultura de Segurança Integrada: Fomenta uma cultura organizacional onde a segurança é responsabilidade compartilhada, não apenas uma obrigação funcional. A mentalidade de segurança como valor transversal é sustentada por práticas colaborativas, automações inteligentes e visibilidade em tempo real do estado de conformidade das aplicações em desenvolvimento.
- Ambiente Orientado por Dados e Inteligência: Promove a criação de um ecossistema técnico onde as decisões são suportadas por dados confiáveis, extraídos de métricas operacionais, análises de desempenho e inferências automatizadas. Essa orientação baseada em evidências fortalece a governança e a eficácia do processo de engenharia de software.

Propósito e Objetivos

A capability de Developer Autonomy & DevSecOps, ou Autonomia do Desenvolvedor e DevSecOps, é essencial para habilitar uma abordagem moderna, ágil e resiliente no desenvolvimento de software.

Seu propósito central é empoderar os desenvolvedores com autonomia, ferramentas,

processos inteligentes e visão sistêmica, permitindo que considerações de segurança sejam incorporadas desde os estágios iniciais do ciclo de vida do desenvolvimento.

Essa capability sustenta a abordagem DevSecOps como uma prática estratégica, em que o desenvolvimento seguro e eficiente torna-se uma responsabilidade compartilhada, contínua e integrada à cultura organizacional.

No contexto atual, caracterizado por esteiras de desenvolvimento aceleradas, grande complexidade arquitetural e exigências de conformidade regulatória, esta capability ganha ainda mais relevância ao incorporar recursos de inteligência artificial como elementos estruturantes da automação, da segurança preditiva e da ampliação da autonomia técnica.

Ao transformar o processo de desenvolvimento em um ecossistema de aprendizado contínuo, assistido por algoritmos capazes de identificar riscos, sugerir melhorias e antecipar desvios, viabiliza-se um novo patamar de qualidade e velocidade nas entregas digitais.

Objetivos

Dentro do escopo do CIO Codex Capability Framework, os objetivos estratégicos da Developer Autonomy & DevSecOps incluem:

- Empoderar Desenvolvedores: Capacitar os desenvolvedores com autonomia decisória, sustentada por processos bem definidos, governança leve e ferramentas inteligentes que possibilitem tomadas de decisão técnicas com responsabilidade, qualidade e segurança.
- Integração Contínua da Segurança: Assegurar que as práticas de segurança estejam embutidas de forma nativa e contínua em todas as fases do ciclo de desenvolvimento, desde o planejamento e modelagem até a entrega e sustentação em produção.
- Automação Inteligente de Segurança: Implementar automações que realizem não apenas testes e verificações, mas que sejam capazes de aprender com o histórico de incidentes, adaptar-se a contextos específicos e promover alertas e recomendações preventivas durante o desenvolvimento.
- Cultura DevSecOps: Estabelecer e nutrir uma cultura organizacional centrada na colaboração entre os times de desenvolvimento, segurança e operações, com ênfase na responsabilidade compartilhada, ciclos curtos de feedback e melhoria contínua.

- Eficiência e Qualidade Sustentáveis: Elevar os padrões de produtividade e qualidade do software entregue, promovendo a antecipação da detecção de falhas, reduzindo retrabalho e minimizando riscos operacionais e de segurança em ambientes críticos.
- Incorporação Responsável de IA no Desenvolvimento: Estimular o uso disciplinado de recursos baseados em inteligência artificial para suportar o raciocínio técnico, revisão de código, verificação de conformidade e geração de insights contextuais, aumentando a maturidade técnica da equipe.

Impacto na Tecnologia

A Developer Autonomy & DevSecOps tem impacto direto e multifacetado nas principais dimensões tecnológicas da organização. Seu alcance vai além das práticas de desenvolvimento, afetando modelos de arquitetura, operações, segurança cibernética e estrutura organizacional de TI:

- Infraestrutura: Requer uma infraestrutura moderna, escalável e automatizável que sustente esteiras de CI/CD, ferramentas de análise estática e dinâmica, mecanismos de observabilidade e, cada vez mais, engines baseados em IA capazes de oferecer suporte contextual e contínuo ao processo de desenvolvimento.
- Arquitetura: Influencia as decisões arquiteturais ao exigir componentes modulares, segregação lógica de domínios críticos, interfaces seguras por padrão e observabilidade nativa. A arquitetura deve estar preparada para suportar padrões de automação e segurança contínua desde o design.
- Sistemas e Ciclo de Vida de Software: Afeta diretamente o modelo de construção de sistemas, orientando o desenvolvimento para práticas como security by design, shift-left testing e revisão contínua por pares e mecanismos automatizados. Isso resulta em soluções mais estáveis, seguras e sustentáveis ao longo do tempo.
- Cybersecurity: Assume papel central na mitigação de riscos cibernéticos, ao antecipar falhas e vulnerabilidades por meio de práticas automatizadas e aprendizado constante. A capacidade de realizar varreduras dinâmicas e contextualizadas durante o desenvolvimento torna-se um diferencial estratégico.
- Modelo Operacional: Impacta o modelo operacional da TI ao promover

estruturas mais colaborativas, multidisciplinares e orientadas por objetivos compartilhados. A sinergia entre Dev, Sec e Ops permite respostas mais rápidas a incidentes, maior confiabilidade das entregas e maior alinhamento com as metas estratégicas da organização. Introduz um novo paradigma de governança onde os desenvolvedores passam a atuar como protagonistas também na segurança e qualidade das entregas. O uso de assistentes inteligentes reforça a tomada de decisão responsável, alinhada às boas práticas e políticas corporativas.

Essa capability, ao aliar visão moderna de engenharia de software, automação contínua e inteligência aplicada, torna-se um alicerce para a transformação digital responsável, elevando o patamar da TI organizacional frente às exigências do mercado, da segurança regulatória e da excelência operacional.

Roadmap de Implementação

A capability de Developer Autonomy & DevSecOps desempenha papel fundamental na camada de Solution Engineering, sendo um catalisador para o desenvolvimento de soluções seguras, escaláveis e de alta qualidade desde as fases iniciais do ciclo de vida do software.

Sua implementação representa uma transformação significativa na forma como os times de desenvolvimento, segurança e operações interagem, promovendo uma cultura onde a autonomia, a colaboração e a inteligência aplicada se tornam fundamentos operacionais.

A seguir, um roadmap estruturado de implementação, alinhado aos princípios do CIO Codex Capability Framework, destacando os principais marcos, práticas recomendadas e oportunidades de uso de tecnologias emergentes, especialmente recursos de IA, como aceleradores desta jornada:

- Avaliação da Maturidade Atual: Realize uma avaliação abrangente da maturidade dos processos atuais de desenvolvimento, identificando pontos fortes e lacunas em relação a práticas de CI/CD, segurança integrada, autonomia técnica e uso de ferramentas inteligentes. Inclua nessa etapa um diagnóstico do nível de adoção de automações baseadas

em IA, tanto em tarefas de codificação quanto em verificação de segurança e qualidade.

- Definição de Objetivos Estratégicos: Estabeleça objetivos claros e mensuráveis para a adoção da capability, garantindo alinhamento com os direcionadores estratégicos da organização. Os objetivos devem abranger não apenas a integração da segurança, mas também o empoderamento técnico, a redução do tempo de resposta a vulnerabilidades e o fortalecimento da inteligência situacional dos desenvolvedores.
- Treinamento e Capacitação Técnica: Implemente trilhas de aprendizado contínuo para os times de desenvolvimento, com foco em práticas DevSecOps, engenharia segura de software, uso ético de IA e ferramentas de suporte inteligente à codificação e testes. O preparo da equipe é determinante para uma adoção bem-sucedida das tecnologias envolvidas, garantindo autonomia com responsabilidade.
- Integração de Segurança no Ciclo de Vida: Identifique pontos críticos do ciclo de desenvolvimento onde a segurança deve ser introduzida de forma sistemática: revisões de código, análises estáticas, modelagem de ameaças e testes dinâmicos automatizados. Incorpore também recursos de IA que ofereçam sugestões contextuais durante a escrita do código, ampliando a prevenção de falhas em tempo real.
- Ferramentas de Automatização e IA Assistiva: Avalie e implemente um portfólio de ferramentas que ofereçam cobertura abrangente em testes, verificação de conformidade e identificação de vulnerabilidades. Priorize aquelas que utilizam inteligência artificial para identificação de padrões suspeitos, análise de comportamento do código e sugestões baseadas em boas práticas acumuladas.
- Definição de Métricas e KPIs Inteligentes: Crie um conjunto de métricas que capturem não apenas a eficácia da segurança integrada, mas também a evolução da autonomia dos times e a eficácia das automações inteligentes. Indicadores como lead time de correção de falhas, taxa de detecção precoce de vulnerabilidades e uso efetivo de recomendações de IA são exemplos relevantes.
- Cultura de Colaboração Ampliada: Estabeleça canais permanentes de comunicação entre desenvolvimento, segurança e operações, com cadência clara para revisões colaborativas. Incentive práticas como pair programming, security champions e sessões conjuntas de análise de código com apoio de ferramentas cognitivas.

- Automação de Testes de Segurança: Integre testes de segurança automatizados diretamente nas esteiras de CI/CD, garantindo que cada build seja validado com base em critérios de segurança pré-definidos. A automação, apoiada por IA, pode adaptar-se dinamicamente ao contexto do código, priorizando testes de maior risco e ampliando a cobertura com mais inteligência.
- Feedback Contínuo e Inteligente: Implemente mecanismos de feedback contínuo com alertas acionáveis, relatórios em linguagem clara e recomendações embasadas por aprendizado de máquina. Esses recursos aumentam a capacidade de resposta dos times e reforçam o aprendizado técnico em tempo real.
- Avaliação de Riscos Contínua e Contextualizada: Mantenha processos ativos de avaliação de riscos ao longo de todo o ciclo de desenvolvimento. Ferramentas analíticas com IA podem correlacionar fatores de risco técnico, histórico de incidentes e mudanças recentes para priorizar os pontos mais críticos em cada ciclo.
- Auditoria e Conformidade Automatizada: Implemente rotinas de auditoria contínua com apoio de regras automatizadas e mecanismos de conformidade em tempo real. Esse modelo garante aderência aos padrões regulatórios e evita desvios de governança que possam comprometer a segurança ou a integridade da aplicação.
- Melhoria Contínua com Análise de Incidentes: Adote uma abordagem sistemática para análise de falhas e incidentes passados, identificando pontos de fragilidade recorrentes. As plataformas baseadas em IA podem auxiliar na geração de insights e aprendizado organizacional, refinando continuamente os processos e as proteções adotadas.
- Avaliação de Impacto e Valor Gerado: Meça o impacto da capability em termos de redução de riscos, aceleração do desenvolvimento, satisfação do usuário final e maturidade de segurança. Combine métricas operacionais com percepções qualitativas para demonstrar o valor estratégico da Developer Autonomy & DevSecOps.
- Compartilhamento de Conhecimento e Boas Práticas: Crie repositórios vivos de boas práticas, aprendizados, playbooks e estudos de caso para uso por toda a organização. Incentive a criação de comunidades técnicas internas que promovam o intercâmbio de conhecimento e o uso avançado de inteligência aplicada ao desenvolvimento seguro.

A implementação estruturada da Developer Autonomy & DevSecOps habilita a construção de um ambiente de desenvolvimento altamente eficaz, onde a qualidade, segurança e velocidade coexistem de forma sinérgica.

Com a incorporação progressiva de tecnologias inteligentes, as equipes de desenvolvimento passam a operar com maior autonomia e precisão, entregando soluções robustas, auditáveis e preparadas para os desafios de um mercado cada vez mais exigente e regulado.

Melhores Práticas de Mercado

A capability Developer Autonomy & DevSecOps, posicionada dentro da macro capability Solution Development e ancorada na camada de Solution Engineering, exerce um papel essencial na modernização das práticas de desenvolvimento, promovendo uma jornada segura, eficiente e ágil desde os estágios iniciais do ciclo de vida do software.

Sua abordagem visa combinar autonomia técnica, integração contínua de segurança, automação inteligente e colaboração interdisciplinar, apoiando-se cada vez mais em recursos de inteligência artificial aplicada para acelerar decisões e garantir conformidade com padrões elevados de qualidade e proteção.

A seguir, uma lista estendida e atualizada de melhores práticas de mercado, compatíveis com o contexto do CIO Codex Capability Framework:

- Empoderar Desenvolvedores com Autonomia Técnica e Suporte Inteligente: Capacitar os desenvolvedores para tomar decisões informadas e seguras de forma autônoma, com suporte de ferramentas que analisam contexto, sugerem boas práticas e reforçam a qualidade do código em tempo real.
- Integração Contínua de Segurança desde o Design: Inserir controles de segurança desde as fases iniciais de modelagem de sistemas e requisitos, promovendo práticas como security by design, threat modeling e validações automatizadas durante todas as etapas do desenvolvimento.
- Automatização Inteligente de Segurança: Utilizar ferramentas que automatizam testes de segurança, identificação de vulnerabilidades e validação de padrões de codificação, com capacidade adaptativa baseada

em inteligência artificial e aprendizado contínuo dos padrões históricos de risco.

- Cultura de Colaboração Estendida (Dev+Sec+Ops+AI): Estabelecer ambientes colaborativos onde o diálogo técnico entre desenvolvimento, segurança, operações e especialistas em dados e IA ocorra de forma contínua, promovendo a troca de conhecimento, alinhamento de objetivos e resolução ágil de conflitos técnicos.
- Ferramentas de Automação Avançadas e Cognitivas: Adotar ferramentas modernas que combinem automação com inteligência analítica, capazes de prover sugestões em tempo real, realizar análises semânticas e antecipar riscos baseados em padrões de uso e comportamento do sistema.
- Treinamento, Capacitação e Educação sobre IA e Segurança: Oferecer programas estruturados de capacitação para que os desenvolvedores compreendam tanto os fundamentos do DevSecOps quanto o uso responsável e eficaz de assistentes de IA, práticas de engenharia segura e automações inteligentes no ciclo de vida de aplicações.
- Avaliação de Riscos Dinâmica e Contextualizada: Implementar práticas contínuas de gestão de riscos que considerem não apenas aspectos técnicos e regulatórios, mas também fatores como complexidade arquitetural, criticidade da solução e análise preditiva suportada por IA.
- Feedback Contínuo com Inteligência Adaptativa: Estabelecer um fluxo constante e inteligente de feedback, apoiado por motores cognitivos que sintetizem alertas, correlacionem eventos e recomendem ações corretivas de forma personalizada e priorizada.
- Cultura de Segurança Inerente à Organização: Fomentar um ambiente onde a segurança é percebida como valor essencial de todos os envolvidos, não apenas como responsabilidade de uma área específica. A segurança passa a ser parte integrante da cultura de produto, da estratégia e da entrega de valor ao cliente.
- Monitoramento Contínuo e Resposta a Incidentes com Apoio de IA: Adotar ferramentas de observabilidade e monitoramento que utilizem algoritmos de detecção de anomalias, identificação de ameaças em tempo real e suporte à decisão durante situações de crise, otimizando o tempo de resposta e contenção de incidentes.
- Testes de Penetração, Red Team e Simulações Automatizadas: Executar testes regulares com suporte de automações que simulem

comportamentos maliciosos, combinando técnicas tradicionais com modelos generativos e heurísticas adaptativas que elevam a eficácia das simulações ofensivas.

- Governança de Acessos e Privilégios com Auditoria Automatizada: Aplicar controles granulares de acesso baseados em identidade e contexto, auditados continuamente com apoio de mecanismos automatizados que detectem comportamentos fora do padrão e exceções não justificadas.
- Revisões de Código e Análise Estática Assistida por IA: Utilizar ferramentas de análise estática com capacidade de interpretação semântica e contextual, que vão além de regras fixas, ajudando a detectar padrões complexos de falhas e más práticas com maior precisão e proatividade.
- Padronização e Conformidade Dinâmica com Normas de Segurança: Estabelecer padrões de segurança alinhados às principais regulamentações da indústria e complementá-los com validações automáticas durante a esteira de desenvolvimento, ajustando regras com base em aprendizado contínuo e evolução regulatória.
- Gestão de Identidade, Autenticação e Criptografia com Inteligência Adaptativa: Implementar soluções modernas de identidade digital, autenticação multifatorial e criptografia automatizada, monitorando continuamente a eficácia dos controles e ajustando parâmetros com base em indicadores de risco.
- Auditorias de Segurança Automatizadas e Contínuas: Promover auditorias contínuas com ferramentas que analisam conformidade, detectam violações e geram relatórios estruturados em tempo real, permitindo ações corretivas imediatas e aprendizado organizacional.
- Planos de Resiliência e Recuperação com Simulações Inteligentes: Desenvolver e testar planos de continuidade com apoio de simulações automatizadas de cenários de desastre, priorizando sistemas críticos e testando respostas com base em modelos probabilísticos de impacto.
- Avaliação e Governança de Fornecedores de Segurança: Estabelecer critérios rigorosos para seleção, homologação e monitoramento contínuo de fornecedores, avaliando não apenas produtos, mas práticas e conformidade com os padrões de segurança da organização.
- Conscientização em Segurança com Experiências Imersivas: Promover programas de sensibilização que combinem treinamentos interativos, simulações práticas e insights baseados em incidentes reais, reforçando

comportamentos seguros com base em evidências e impacto tangível.

A adoção estruturada destas melhores práticas reforça o papel estratégico da capability Developer Autonomy & DevSecOps como pilar da entrega moderna de software.

Ao combinar autonomia técnica, automação inteligente e uma cultura de segurança enraizada, as organizações se tornam mais resilientes, inovadoras e preparadas para enfrentar os desafios da era digital.

Essas práticas são fundamentais para fortalecer a proteção dos ativos digitais, reduzir a superfície de ataque e garantir a excelência operacional em um ambiente cada vez mais exigente em agilidade, qualidade e conformidade.

Desafios Atuais

A capability Developer Autonomy & DevSecOps é peça-chave na transformação da engenharia de software moderna, ao promover a integração estruturada da segurança desde o início do ciclo de desenvolvimento, dentro de uma cultura colaborativa e contínua.

Contudo, sua adoção ampla e consistente nas organizações ainda esbarra em uma série de desafios técnicos, culturais e operacionais que exigem visão estratégica, resiliência institucional e atualização constante de competências.

No contexto do CIO Codex Capability Framework, esses desafios refletem tanto as barreiras estruturais herdadas de modelos tradicionais, quanto a complexidade de integrar práticas DevSecOps em ambientes que evoluem rapidamente, com tecnologias emergentes, times distribuídos e crescentes exigências regulatórias.

A seguir, os principais obstáculos enfrentados pelas organizações:

- Cultura Organizacional Fragmentada: Muitas organizações ainda operam sob estruturas segmentadas, onde desenvolvimento, segurança e operações são silos isolados. Essa fragmentação dificulta a fluidez do conhecimento e impede a adoção genuína de práticas DevSecOps, exigindo mudanças profundas na mentalidade e no modelo de gestão.
- Resistência à Mudança e Adoção Cultural Lenta: A transformação cultural

necessária para a internalização dos valores DevSecOps frequentemente encontra resistência em equipes habituadas a ciclos longos, validações centralizadas e baixa autonomia. A transição exige liderança ativa, patrocínio executivo e comunicação clara dos ganhos esperados.

- Integração Complexa de Ferramentas e Plataformas: O ecossistema de ferramentas necessário para suportar automação, testes de segurança, pipelines de CI/CD e visibilidade contínua tende a ser diversificado e tecnicamente heterogêneo. Alinhar essas tecnologias sem comprometer performance, segurança ou interoperabilidade representa um grande desafio.
- Gap de Capacitação Técnica: A insuficiência de conhecimento sobre práticas modernas de segurança, automação e uso de IA por parte dos desenvolvedores, arquitetos e operadores pode comprometer a eficácia da capability. O desafio não é apenas ensinar ferramentas, mas formar uma nova mentalidade de engenharia orientada à segurança e autonomia.
- Incorporação de Segurança desde o Design: A mudança de paradigma para incorporar segurança desde a concepção da solução ainda não é natural para muitos times de produto. Isso requer reformulação dos processos de discovery, planejamento técnico e análise de requisitos, além de maior envolvimento de especialistas em segurança nas fases iniciais dos projetos.
- Conciliar Compliance com Agilidade: Organizações atuando em setores regulados enfrentam o desafio de conciliar práticas ágeis com exigências de conformidade (LGPD, PCI-DSS, SOX, etc.). Manter auditoria contínua, rastreabilidade e evidências ao mesmo tempo em que se mantém velocidade exige maturidade em automação e governança.
- Monitoramento e Observabilidade de Segurança: A implementação de práticas de observabilidade em tempo real para detectar anomalias, comportamentos suspeitos e indicadores de risco demanda investimento em infraestrutura, dados e ferramentas inteligentes que consigam interpretar eventos em escala e gerar alertas acionáveis.
- Gerenciamento Eficaz de Vulnerabilidades: Em ambientes dinâmicos e com múltiplos pipelines, manter um processo eficaz e automatizado de detecção, priorização e correção de vulnerabilidades é um desafio crítico. A complexidade aumenta à medida que se incluem aplicações legadas, múltiplos fornecedores e times distribuídos.
- Colaboração Interfuncional Sustentada: Estabelecer uma colaboração

contínua e com propósito entre desenvolvimento, operações e segurança requer mais do que cerimônias ágeis. Requer confiança institucional, alinhamento de incentivos e visibilidade compartilhada de riscos e prioridades, além do suporte de ferramentas colaborativas.

- Feedback Contínuo Automatizado e Relevante: Criar ciclos de feedback rápidos, úteis e integrados exige automação inteligente, integração com plataformas de versionamento e testes, além de modelos que saibam interpretar o contexto da aplicação. O excesso de alertas irrelevantes é contraproducente e pode minar a confiança no sistema.
- Interpretação e Adoção Ética de Inteligência Artificial: A inclusão de plataformas baseadas em IA no ciclo de desenvolvimento também traz desafios: desde o entendimento correto de suas sugestões, até o risco de dependência cega de recomendações automatizadas. É preciso equilibrar autonomia técnica com responsabilidade e supervisão humana criteriosa.

Superar esses desafios requer uma abordagem estratégica, que alinhe tecnologia, cultura e capacitação contínua.

A capability Developer Autonomy & DevSecOps, quando corretamente implementada, oferece os instrumentos, processos e princípios necessários para uma jornada segura e ágil de desenvolvimento, alicerçada em:

- Automação inteligente e responsável;
- Colaboração transversal entre as áreas;
- Governança leve com foco em valor e mitigação de riscos;
- Formação contínua de talentos técnicos com visão ampliada.

Essa capability não se limita a ferramentas ou metodologias, ela representa uma mudança de paradigma, onde o desenvolvedor deixa de ser um executor isolado e passa a ser um agente ativo da qualidade, segurança e inovação.

Vencer esses desafios é condição essencial para que a TI avance rumo à excelência digital com resiliência, responsabilidade e relevância estratégica.

Tendências para o Futuro

A capability Developer Autonomy & DevSecOps ocupa um lugar estratégico na jornada de modernização da engenharia de software, ao possibilitar a integração estruturada da segurança desde os primeiros estágios do ciclo de desenvolvimento.

Baseada em pilares como autonomia técnica, integração contínua, entrega ágil e cultura de colaboração multidisciplinar, essa capability não apenas fortalece a eficiência operacional, como também estabelece as bases para soluções seguras por design.

À medida que o cenário tecnológico se torna mais dinâmico e complexo, com ameaças cibernéticas evoluindo rapidamente e a pressão por entregas rápidas se intensificando, novas tendências estão redesenhando o papel da Developer Autonomy & DevSecOps dentro do contexto do CIO Codex Capability Framework.

Essas tendências refletem uma convergência entre automação avançada, inteligência adaptativa e modelos operacionais resilientes. A seguir, os principais vetores de evolução:

- Automatização de Segurança Baseada em IA e Aprendizado Contínuo: A automação de segurança irá além das regras estáticas, incorporando mecanismos de inteligência artificial que aprendem com padrões históricos de risco, contexto de uso e comportamento do sistema. Isso permitirá uma detecção mais precisa e contextualizada de ameaças, reduzindo falsos positivos e acelerando a resposta.
- Segurança Antecipada e Orientada por Contexto (Shift-Left & Predictive Security): A estratégia de shift-left security será amplificada por mecanismos preditivos, permitindo que a segurança seja integrada não apenas antecipadamente, mas também com base em inferências sobre o impacto potencial das decisões arquiteturais e do comportamento do código em tempo de execução.
- Segurança como Código e Infraestrutura Imutável: A consolidação da prática de Security as Code permitirá que políticas, regras e permissões sejam definidas, versionadas e auditadas como parte da base de código da aplicação. Isso trará ganhos expressivos em padronização, rastreabilidade e integração com esteiras automatizadas.
- Inteligência de Ameaças Integrada aos Pipelines de CI/CD: Os pipelines de integração e entrega contínua passarão a consumir dados de

inteligência de ameaças em tempo real, permitindo que decisões automatizadas de bloqueio, alerta ou reforço de proteções sejam tomadas antes mesmo da implantação da aplicação.

- Automação de Conformidade e Auditoria Cognitiva: Processos de auditoria e conformidade deixarão de ser eventos manuais e periódicos, tornando-se contínuos e assistidos por IA. Dashboards de conformidade em tempo real, alimentados por varreduras automáticas e validações cruzadas, serão essenciais para ambientes regulados.
- Adoção Ampliada de Arquiteturas Zero Trust: O modelo Zero Trust será cada vez mais aplicado ao desenvolvimento de software, com validações constantes de identidade, permissões e contexto antes de qualquer transação ser autorizada. Isso exigirá que a segurança seja tratada como um subsistema operacional dentro das soluções.
- Análise Contínua e Autônoma de Vulnerabilidades: A verificação contínua de vulnerabilidades em código, bibliotecas e containers passará a ser uma prática-padrão, com apoio de plataformas que oferecem sugestões de correção automáticas e priorização baseada em impacto real sobre o ambiente operacional.
- Capacitação Contínua e Customizada em Segurança para Desenvolvedores: Programas de formação em segurança serão adaptativos, baseados em trilhas personalizadas por perfil, função e nível de maturidade. Plataformas de aprendizado digital incluirão simulações práticas, desafios interativos e integração com ferramentas do dia a dia dos desenvolvedores.
- Integração Nativa entre DevSecOps e Plataformas de Observabilidade: A telemetria gerada durante o desenvolvimento e a execução das aplicações será analisada de forma contínua para detectar padrões de risco, gargalos de performance e violações de política. Essa integração permitirá ações corretivas automatizadas e mais alinhadas ao contexto de produção.
- Cultura de Segurança Inteligente e Distribuída: A cultura de segurança evoluirá para um modelo distribuído e proativo, onde cada papel técnico é corresponsável pela proteção da informação e pela resiliência da solução. O uso de assistentes inteligentes embutidos nas ferramentas de desenvolvimento facilitará a internalização de práticas seguras como parte do fluxo natural de trabalho.

As tendências projetadas para o futuro da capability Developer Autonomy & DevSecOps apontam para um cenário onde segurança, agilidade e inteligência convergem de forma orgânica.

À medida que os modelos operacionais se tornam mais adaptativos e orientados por dados, o papel da automação inteligente e da autonomia assistida se intensifica — transformando a maneira como o software é concebido, desenvolvido e protegido.

Essa evolução posiciona a Developer Autonomy & DevSecOps não apenas como um facilitador técnico, mas como um pilar estratégico na construção de soluções digitais seguras, escaláveis e sustentáveis, preparando as organizações para enfrentar com confiança os desafios da economia digital.

KPIs Usuais

A capability Developer Autonomy & DevSecOps representa uma evolução na forma como segurança, agilidade e qualidade são incorporadas ao ciclo de vida do desenvolvimento de software.

Para garantir a governança efetiva dessa capability, é essencial a definição e o monitoramento contínuo de Indicadores-Chave de Desempenho (KPIs) que reflitam não apenas a adoção técnica, mas também a maturidade cultural, a eficiência operacional e o retorno estratégico das iniciativas.

No contexto do CIO Codex Capability Framework, os KPIs devem ser relevantes, mensuráveis e orientados à ação, proporcionando visibilidade objetiva sobre o progresso, os riscos e os impactos da Developer Autonomy & DevSecOps.

Com o avanço das práticas de automação inteligente, da integração de IA ao ciclo de desenvolvimento e da cultura de melhoria contínua, os indicadores a seguir representam um conjunto atualizado e abrangente de métricas a serem consideradas:

- Taxa de Automação de Testes de Segurança (Security Test Automation Rate): Proporção de testes de segurança realizados de forma automatizada em relação ao total de testes aplicados. Reflete o nível de maturidade da esteira CI/CD em segurança.
- Tempo Médio para Integração de Segurança (Average Security Integration Time): Tempo médio necessário para inserir controles de segurança (como análise estática, validações de dependências e

escaneamento de vulnerabilidades) no ciclo de desenvolvimento, desde a fase de planejamento até a primeira execução funcional.

- Quantidade de Vulnerabilidades Identificadas (Vulnerabilities Identified Count): Volume total de vulnerabilidades encontradas nas fases de desenvolvimento, validação e pre-release. Pode ser segmentado por criticidade e origem (ex: bibliotecas de terceiros, código interno, configuração).
- Tempo Médio para Correção de Vulnerabilidades (Average Vulnerability Resolution Time): Tempo médio entre a detecção e a correção efetiva de uma vulnerabilidade, mensurado por tipo e nível de severidade. Serve como indicador direto da agilidade e capacidade de resposta da equipe.
- Taxa de Adoção das Práticas DevSecOps (DevSecOps Practices Adoption Rate): Percentual de times de desenvolvimento que seguem as práticas estabelecidas de DevSecOps, como uso de ferramentas integradas, automação de testes, revisões seguras e versionamento de políticas.
- Quantidade de Treinamentos de Segurança Realizados (Security Training Count): Número total de treinamentos, workshops ou certificações oferecidas aos desenvolvedores com foco em segurança de software e práticas DevSecOps. Também pode ser avaliada a taxa de conclusão e aplicabilidade prática.
- Taxa de Falhas de Segurança em Produção (Security Failures in Production Rate): Proporção de falhas de segurança detectadas após o software estar em produção. Altas taxas indicam fragilidades nas fases de validação anteriores e deficiências no processo de integração de segurança.
- Nível de Colaboração entre Equipes (Team Collaboration Level): Métrica qualitativa e quantitativa que avalia o grau de interação entre desenvolvimento, segurança e operações. Pode ser derivada de ferramentas de colaboração, frequência de cerimônias DevSecOps e engajamento em fluxos de melhoria contínua.
- Taxa de Conformidade com Padrões de Segurança (Security Standards Compliance Rate): Percentual de aderência às políticas internas de segurança, frameworks regulatórios (ex: OWASP, NIST) e normas externas. Pode incluir conformidade automatizada validada por mecanismos de auditoria em tempo real.
- Quantidade de Código Revisado por Pares (Code Reviewed by Peers Count): Volume de commits ou pull requests submetidos à revisão

colaborativa, especialmente com foco em aspectos de segurança. Um bom indicador da cultura de qualidade técnica e responsabilidade coletiva.

- Eficiência do Processo de Correção (Correction Process Efficiency): Mede a taxa de sucesso e o tempo envolvido em ações corretivas, incluindo reprovações evitadas, falhas reincidentes e tempo médio para reteste de vulnerabilidades após correções aplicadas.
- Taxa de Retorno sobre o Investimento em Segurança (Security ROI Rate): Indicador financeiro que mede o retorno obtido com investimentos em práticas, ferramentas e processos de segurança. Pode considerar redução de incidentes, custos evitados com retrabalho ou penalidades regulatórias.
- Quantidade de Vulnerabilidades Corrigidas Antes do Lançamento (Pre-Launch Vulnerabilities Fixed Count): Número absoluto de vulnerabilidades resolvidas antes da liberação da aplicação em produção. Um KPI direto da eficácia da segurança shift-left.
- Satisfação dos Desenvolvedores com Ferramentas de Segurança (Developer Satisfaction with Security Tools Level): Índice de satisfação baseado em pesquisas regulares ou análises comportamentais que avaliam a experiência do desenvolvedor com as ferramentas de segurança disponíveis. Baixas notas podem indicar atritos que comprometem a adoção.
- Tempo Médio de Implantação Contínua (Average Continuous Deployment Time): Tempo médio para realizar uma nova implantação via pipeline de CI/CD. Serve como termômetro para medir o impacto das validações de segurança no fluxo de entregas.
- Taxa de Sugestões de IA Aceitas (AI-Supported Suggestions Acceptance Rate): Indicador emergente que mede a taxa de sugestões feitas por assistentes inteligentes (ex: análise de código, revisão de políticas) que foram efetivamente aceitas e aplicadas pelos desenvolvedores.
- Volume de Alertas de Segurança Validados com Êxito (Validated Security Alerts Volume): Mede a quantidade de alertas de segurança automatizados que, ao serem investigados, foram confirmados como ameaças legítimas. É um KPI importante para avaliar a qualidade das ferramentas de detecção.

O acompanhamento estruturado e estratégico dos KPIs da Developer Autonomy & DevSecOps permite que as organizações tenham uma visão clara da evolução técnica,

do retorno sobre os investimentos em segurança e da maturidade cultural das equipes envolvidas.

Mais do que controlar, os indicadores funcionam como insumos valiosos para a tomada de decisão baseada em evidências, possibilitando ajustes táticos e priorizações conforme o ciclo de vida da capability avança.

Com o avanço da inteligência aplicada, esses KPIs poderão evoluir para modelos preditivos, análises em tempo real e recomendações automatizadas de melhoria — reforçando o valor da capability como ativo estratégico de desenvolvimento seguro e eficiente.

Exemplos de OKRs

A capability de Developer Autonomy & DevSecOps, crucial no CIO Codex Capability Framework, capacita desenvolvedores a integrar aspectos de segurança desde o início do ciclo de desenvolvimento, adotando a abordagem DevSecOps.

Esta capability enfatiza a autonomia do desenvolvedor, integração contínua, entrega contínua e uma cultura colaborativa para garantir desenvolvimento de software seguro e eficiente.

Seguem exemplos de Objetivos e Resultados-Chave (OKRs) para efetivar essa capability:

Empoderamento e Autonomia dos Desenvolvedores

Objetivo: Fortalecer a autonomia dos desenvolvedores em decisões de segurança durante o desenvolvimento de software.

- KR1: Aumentar em 30% a adoção de decisões autônomas de segurança pelos desenvolvedores.
- KR2: Reduzir em 20% o tempo de resposta para mudanças relacionadas à segurança no ciclo de desenvolvimento.
- KR3: Implementar 3 novas ferramentas de automação que suportem a autonomia do desenvolvedor.

Integração Contínua de Segurança

Objetivo: Integrar considerações de segurança de forma contínua durante todo o ciclo de desenvolvimento.

- KR1: Aumentar a cobertura de testes de segurança em 40%.
- KR2: Reduzir em 25% as vulnerabilidades de segurança identificadas após a implantação.
- KR3: Implementar análises de segurança automatizadas em 100% dos projetos.

Automação de Segurança

Objetivo: Automatizar a execução de testes de segurança e a verificação de vulnerabilidades.

- KR1: Automatizar 60% dos testes de segurança rotineiros.
- KR2: Diminuir em 30% o tempo de identificação e correção de vulnerabilidades.
- KR3: Aumentar em 50% a frequência dos ciclos de revisão de segurança automatizados.

Cultura de DevSecOps

Objetivo: Fomentar uma cultura organizacional que valorize a colaboração entre desenvolvimento, operações e segurança.

- KR1: Realizar 5 workshops interdepartamentais por ano para promover práticas de DevSecOps.
- KR2: Aumentar em 20% a participação de membros de segurança em projetos de desenvolvimento.
- KR3: Implementar um sistema de feedback contínuo entre desenvolvimento, segurança e operações.

Eficiência e Qualidade do Desenvolvimento

Objetivo: Melhorar a eficiência do desenvolvimento de software e garantir a qualidade e segurança do software produzido.

- KR1: Reduzir em 15% o tempo de desenvolvimento de novas funcionalidades.
- KR2: Atingir uma taxa de 95% de conformidade com padrões de qualidade e segurança de software.
- KR3: Diminuir em 40% os incidentes de segurança no software após a implantação.

Esses OKRs ressaltam a importância da Developer Autonomy & DevSecOps no desenvolvimento de soluções tecnológicas seguras e eficientes.

Implementar esses OKRs contribui para melhorar a autonomia dos desenvolvedores, integrar a segurança ao longo do ciclo de desenvolvimento, automatizar processos de segurança, promover uma cultura de DevSecOps e assegurar a eficiência e qualidade do desenvolvimento de software.

A abordagem DevSecOps é essencial para enfrentar desafios de segurança modernos, garantindo ao mesmo tempo agilidade e inovação no desenvolvimento de software.

Critérios para Avaliação de Maturidade

A capability Developer Autonomy & DevSecOps é componente central da engenharia de soluções resilientes, sustentáveis e seguras dentro do CIO Codex Capability Framework.

Seu papel vai além da integração técnica de ferramentas de segurança; ela promove uma cultura que combina autonomia técnica responsável, integração contínua de práticas seguras e colaboração interfuncional, desde os primeiros momentos do ciclo de vida do software.

Para medir sua efetividade ao longo do tempo e identificar oportunidades de evolução, é fundamental a adoção de critérios de avaliação de maturidade baseados em estágios estruturados.

Inspirado no modelo CMMI (Capability Maturity Model Integration), este framework define cinco níveis progressivos de maturidade para a Developer Autonomy & DevSecOps, com ênfase em processos, cultura, automação e aprendizado contínuo:

Nível de Maturidade Inexistente

- A segurança não é reconhecida como parte essencial do desenvolvimento de software.
- Não há processos, ferramentas ou cultura voltada à segurança ou à autonomia do desenvolvedor.
- Nenhuma ação estruturada é realizada para integrar práticas DevSecOps.
- Ausência completa de políticas ou padrões técnicos relacionados à segurança no ciclo de vida do software.
- Treinamento e conscientização sobre segurança inexistem.
- Estado de alerta: organização exposta a riscos significativos sem qualquer mitigação preventiva.

Nível de Maturidade Inicial

- A segurança começa a ser reconhecida, mas com ações reativas e não estruturadas.
- Algumas práticas básicas são aplicadas de forma pontual e dependem do comprometimento individual.
- A abordagem DevSecOps é discutida, mas ainda não está sistematizada ou aplicada de forma consistente.
- Existem políticas documentadas, porém mal comunicadas ou pouco compreendidas pelas equipes.
- Treinamentos sobre segurança ocorrem de forma esporádica e geralmente após falhas ou incidentes.
- Momento de transição, onde a consciência emerge, mas o impacto ainda é limitado pela informalidade e ausência de governança.

Nível de Maturidade Definido

- Práticas de segurança são formalmente incorporadas aos processos de desenvolvimento.
- A estratégia DevSecOps é estabelecida, documentada e divulgada organizacionalmente.
- Desenvolvedores possuem clareza sobre seu papel na proteção do código e da infraestrutura.
- Políticas de segurança estão integradas às rotinas de trabalho e

auditáveis.
- A formação em segurança passa a ser regular e alinhada com os papéis técnicos dos colaboradores.
- Organização com fundamentos sólidos, estruturando uma cultura DevSecOps replicável e consistente.

Nível de Maturidade Gerenciado

- Os processos de segurança são otimizados e sustentados por mecanismos de controle e automação.
- A estratégia DevSecOps é evolutiva, sendo refinada com base em métricas, feedback e análise de incidentes.
- Métricas de qualidade e segurança são coletadas continuamente, analisadas e utilizadas para ações corretivas.
- Treinamento em segurança é contínuo, adaptativo e parte do plano de desenvolvimento profissional das equipes.
- A colaboração entre desenvolvimento, segurança e operações é fluida e orientada por objetivos compartilhados.
- Organização madura, com práticas consistentes, visão integrada e cultura de aprendizado contínuo.

Nível de Maturidade Otimizado

- A capability é referência na indústria em termos de excelência, inovação e inteligência aplicada à segurança.
- A automação de segurança é inteligente, contextual e acionável incorporando recursos baseados em IA, análise preditiva e validação em tempo real.
- A abordagem DevSecOps é transversal, impactando decisões de arquitetura, design, governança e cultura.
- A cultura de segurança é orgânica e profundamente enraizada, com responsabilidade compartilhada e atitude preventiva em todos os níveis.
- A organização atua como formadora de práticas de mercado, contribuindo com comunidades e padrões emergentes.
- Estado de excelência, onde a segurança não é apenas uma função técnica,

mas um ativo estratégico e um diferencial competitivo institucionalizado.

Os critérios de avaliação de maturidade da Developer Autonomy & DevSecOps servem como bússola para orientar a evolução contínua da capability nas organizações.

Eles permitem identificar lacunas críticas, consolidar boas práticas e priorizar investimentos em processos, cultura, automação e ferramentas em prol do aumento da qualidade e produtividade do desenvolvimento.

Convergência com Frameworks de Mercado

A capability Developer Autonomy & DevSecOps, parte da macro capability Solution Development e inserida na camada Solution Engineering do CIO Codex Capability Framework, desempenha um papel central na integração de práticas de segurança ao longo de todo o ciclo de desenvolvimento de software, promovendo uma abordagem proativa e estruturada no enfrentamento dos riscos inerentes à engenharia de soluções modernas.

Esta capability reforça o princípio de empoderamento dos desenvolvedores, fornecendo-lhes autonomia, recursos e diretrizes que permitam práticas de DevSecOps eficazes, sustentáveis e aderentes à realidade de ambientes de desenvolvimento ágeis e altamente dinâmicos.

O conceito de autonomia aqui transcende o mero acesso a ferramentas: trata-se da capacidade de os times de desenvolvimento atuarem de maneira independente, porém alinhada, com ciclos rápidos de entrega e integrando segurança como um requisito fundamental e não acessório.

Um vetor adicional que vem potencializando de forma significativa essa capability é a adoção estratégica de plataformas baseadas em inteligência artificial ao longo do processo de desenvolvimento de software.

Estas plataformas atuam como copilotos cognitivos, assistindo desenvolvedores em atividades como geração de código, validação de padrões de segurança, elaboração de testes automatizados, revisão de código e até mesmo na documentação técnica.

Tais capacidades ampliam a autonomia individual, aumentam a consistência entre os membros da equipe e elevam o padrão de qualidade técnica desde os estágios iniciais

da concepção das soluções.

Além disso, o uso dessas ferramentas baseadas em IA fomenta um ciclo de aprendizado contínuo, já que sugestões, alertas e refatorações são geradas em tempo real, alinhadas com boas práticas de engenharia segura e moderna.

Isso cria um ambiente fértil para que o desenvolvedor atue simultaneamente como executor e como aprendiz, em um modelo de aprendizado ativo que impulsiona a maturidade técnica ao mesmo tempo em que promove produtividade exponencial.

A seguir, é apresentada a análise da convergência dessa capability com um conjunto de dez frameworks de mercado amplamente reconhecidos:

COBIT

- Nível de Convergência: Médio
- Racional: O COBIT foca fortemente em governança de TI, inclusive nos aspectos de segurança e conformidade. A capability Developer Autonomy & DevSecOps contribui nesse contexto ao integrar segurança desde o início do desenvolvimento. A convergência é reforçada quando se considera o uso de IA para promover controles automatizados, auditoráveis e consistentes em linha com políticas de governança, mesmo que o COBIT não se aprofunde diretamente na autonomia técnica do desenvolvedor.

ITIL

- Nível de Convergência: Médio
- Racional: Embora centrado na gestão de serviços de TI, o ITIL se beneficia da adoção de práticas DevSecOps para fortalecer a confiabilidade e segurança dos serviços entregues. A introdução de IA no desenvolvimento eleva ainda mais o nível de controle e rastreabilidade das mudanças, o que se alinha à gestão do ciclo de vida do serviço promovido pelo ITIL.

SAFe

- Nível de Convergência: Alto
- Racional: O SAFe valoriza a agilidade empresarial com governança. Developer Autonomy & DevSecOps é pilar essencial na entrega contínua e segura em escala, com forte sinergia com o SAFe. A automação orientada por IA acelera ainda mais os ciclos de entrega, promovendo segurança e qualidade com maior granularidade e menor dependência de validações manuais.

PMI

- Nível de Convergência: Baixo
- Racional: O PMI aborda projetos sob uma ótica mais tradicional, com foco em escopos, cronogramas e entregas. A autonomia do desenvolvedor e as práticas DevSecOps intensificadas pela automação via IA fogem da abordagem preconizada pelo PMI, que não contempla diretamente tais práticas no seu escopo metodológico.

CMMI

- Nível de Convergência: Médio
- Racional: O CMMI promove a maturidade de processos, sendo alinhado à adoção de práticas contínuas de melhoria e controle. Developer Autonomy & DevSecOps, ampliado pelo uso de IA, eleva o nível de maturidade ao permitir mecanismos de autocorreção, validação contínua e documentação assistida, fomentando melhoria incremental com rastreabilidade robusta.

TOGAF

- Nível de Convergência: Baixo
- Racional: TOGAF é voltado à arquitetura corporativa em níveis macro. Embora a segurança seja uma preocupação inerente, a autonomia do desenvolvedor e as práticas operacionais de DevSecOps têm pouca sobreposição direta com o framework. A IA aplicada ao desenvolvimento

também não é foco do TOGAF, embora possa se integrar à camada de arquitetura de soluções de forma indireta.

DevOps SRE

- Nível de Convergência: Alto
- Racional: Fortemente alinhado. A cultura de confiabilidade, automação e melhoria contínua é o cerne do SRE, casando-se perfeitamente com os princípios de Developer Autonomy & DevSecOps. O uso de IA multiplica os efeitos desejados, reduzindo o tempo médio de entrega, acelerando testes, prevenindo incidentes e promovendo observabilidade inteligente.

NIST

- Nível de Convergência: Médio
- Racional: NIST define padrões e controles de segurança e privacidade, aos quais o DevSecOps busca aderir. O uso de inteligência artificial permite conformidade contínua e detecção preditiva de violações, o que amplia a efetividade da implementação dos controles definidos pelo NIST, mesmo que este não prescreva diretamente tal abordagem.

Six Sigma

- Nível de Convergência: Baixo
- Racional: A metodologia Six Sigma foca em controle estatístico e melhoria de processos. Embora compartilhe com o DevSecOps a busca por qualidade, o escopo da capability inclusive quando ampliado pela IA não encontra sinergia direta com o Six Sigma, que é mais centrado em métricas operacionais tradicionais do que em práticas de engenharia segura e autônoma.

Lean IT

- Nível de Convergência: Médio

- Racional: Lean IT prioriza a eliminação de desperdícios e ganho de eficiência, objetivos igualmente perseguidos pelo DevSecOps e amplificados com o uso de IA. A inteligência artificial, ao reduzir retrabalho e antecipar problemas, é um catalisador de eficiência lean, embora o alinhamento conceitual dependa de como os princípios são operacionalizados.

A capability Developer Autonomy & DevSecOps demonstra convergência mais robusta com frameworks que priorizam agilidade, automação, qualidade contínua e governança em tempo real, como SAFe e DevOps SRE.

Ao incorporar o uso estratégico de inteligência artificial no ciclo de desenvolvimento, essa capability expande significativamente seu impacto, promovendo uma nova fronteira de engenharia segura, autônoma e inteligente.

Ao transformar tarefas repetitivas em ciclos assistidos por IA, libera-se tempo criativo dos desenvolvedores, potencializa-se a assertividade das entregas e fortalece-se o alinhamento com padrões de conformidade e segurança.

Esse novo paradigma não apenas melhora a produtividade e qualidade, mas também reforça o papel do desenvolvedor como protagonista na construção de soluções tecnológicas resilientes e alinhadas aos objetivos de negócio.

Assim, o CIO Codex reconhece o valor emergente de tais práticas como parte essencial do modelo de capabilities de uma TI moderna, reforçando que o caminho para o futuro passa necessariamente por autonomia consciente, segurança nativa e inteligência aplicada ao desenvolvimento.

Processos e Atividades

Develop DevSecOps Strategy

Desenvolver uma estratégia de DevSecOps é um passo crítico para integrar a segurança em todo o ciclo de desenvolvimento de software.

Este processo envolve a criação de uma abordagem abrangente que incorpora práticas de segurança desde o início do desenvolvimento até a operação e manutenção das aplicações.

A estratégia deve alinhar-se com os objetivos de negócio da organização e garantir que todos os aspectos de segurança sejam considerados em cada fase do desenvolvimento.

Isso inclui a definição de políticas de segurança, a escolha de ferramentas adequadas (com capacidades inteligentes), a automação de testes de segurança, e a promoção de uma cultura DevSecOps apoiada por IA.

O plano estratégico deve ser claro, mensurável e capaz de evoluir com as mudanças tecnológicas, ameaças de segurança emergentes e inovações em assistências baseadas em IA.

- PDCA focus: Plan
- Periodicidade: Anual

#	Nome da Atividade	Descrição	Inputs	Outputs	RACI	DARE
1	Assess Current Security	Avaliar o estado atual das práticas de segurança dentro do ciclo de desenvolvimento. Incluir análises assistidas por IA para detectar padrões e vulnerabilidades com maior profundidade e velocidade.	Relatórios de auditoria, políticas de segurança atuais	Avaliação de segurança	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Cybersecurity; Informed: IT Governance & Transformation	Decider: Solution Engineering & Development; Advisor: Cybersecurity; Recommender: IT Governance & Transformation; Executer: Solution Engineering & Development

2	Define Security Policies	Definir políticas de segurança que serão integradas ao ciclo de desenvolvimento. Alinhar políticas com modelos adaptativos de aprendizado de ameaças suportados por IA.	Avaliação de segurança, requisitos de negócio	Políticas de segurança definidas	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Cybersecurity; Informed: Architecture & Technology Visioning	Decider: Solution Engineering & Development; Advisor: Cybersecurity; Recommender: Architecture & Technology Visioning; Executer: Solution Engineering & Development
3	Select DevSecOps Tools	Selecionar ferramentas de DevSecOps que suportem a automação e integração contínua de segurança. Priorizar ferramentas com recursos de IA para análise de código, recomendações de correção e triagem inteligente.	Políticas de segurança definidas, pesquisa de ferramentas	Ferramentas selecionadas	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: IT Infrastructure & Operation; Informed: Data, AI & New Technology	Decider: Solution Engineering & Development; Advisor: IT Infrastructure & Operation; Recommender: Data, AI & New Technology; Executer: Solution Engineering & Development
4	Develop Training Programs	Desenvolver programas de treinamento para capacitar a equipe nas práticas de DevSecOps. Utilizar IA para personalizar conteúdos conforme os perfis dos desenvolvedores.	Políticas de segurança, ferramentas selecionadas	Programas de treinamento	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: IT Governance & Transformation; Informed: Architecture & Technology Visioning	Decider: Solution Engineering & Development; Advisor: IT Governance & Transformation; Recommender: Architecture & Technology Visioning; Executer: Solution Engineering & Development

5	Create Implementation Plan	Criar um plano de implementação detalhado para integrar as práticas de DevSecOps. Incorporar milestones para adoção de IA progressiva e automações cognitivas.	Programas de treinamento, ferramentas selecionadas	Plano de implementação	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: IT Governance & Transformation; Informed: Architecture & Technology Visioning	Decider: Solution Engineering & Development; Advisor: IT Governance & Transformation; Recommender: Architecture & Technology Visioning; Executer: Solution Engineering & Development

Identify Autonomy Opportunities

Identificar oportunidades para aumentar a autonomia e implementação de automatizações e suporte por AI aos desenvolvedores é essencial para promover uma cultura de inovação e eficiência.

Este processo envolve a análise das práticas de desenvolvimento atuais para identificar áreas onde os desenvolvedores podem ser capacitados a tomar decisões de forma independente e com o suporte metodológico e de ferramental de AI adequado.

Isso pode incluir a adoção de ferramentas de automação, a criação de guidelines claras para o desenvolvimento seguro e a promoção de uma cultura de responsabilidade e confiança.

O objetivo é permitir que os desenvolvedores atuem de maneira mais ágil e eficiente, reduzindo a necessidade de supervisão constante e aumentando a velocidade de entrega de software.

Aumentar a autonomia dos desenvolvedores também requer a implementação de mecanismos de feedback e suporte contínuos para garantir que eles tenham os recursos necessários para tomar decisões informadas.

Este processo visa identificar e fomentar oportunidades que aumentem a autonomia dos desenvolvedores com o suporte de IA.

Envolve análise das práticas atuais de desenvolvimento para apontar áreas que podem ser assistidas ou automatizadas, reduzindo dependências e aumentando a agilidade e qualidade.

- PDCA focus: Plan
- Periodicidade: Semestral

#	Nome da Atividade	Descrição	Inputs	Outputs	RACI	DARE
1	Analyze Development Practices	Analisar as práticas de desenvolvimento atuais para identificar áreas de melhoria. Usar IA para analisar repositórios, identificar padrões de retrabalho, duplicidade e dependências excessivas.	Documentação de processos de desenvolvimento	Relatório de análise	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: IT Governance & Transformation; Informed: Architecture & Technology Visioning	Decider: Solution Engineering & Development; Advisor: IT Governance & Transformation; Recommender: Architecture & Technology Visioning; Executer: Solution Engineering & Development
2	Identify Bottlenecks	Identificar gargalos e áreas onde a autonomia dos desenvolvedores pode ser aumentada. Mapear pontos de fricção onde a assistência cognitiva pode apoiar a tomada de decisões.	Relatório de análise	Lista de gargalos	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: IT Infrastructure & Operation; Informed: Data, AI & New Technology	Decider: Solution Engineering & Development; Advisor: IT Infrastructure & Operation; Recommender: Data, AI & New Technology; Executer: Solution Engineering & Development

3	Propose Autonomy and Automation Enhancements	Propor melhorias para aumentar a autonomia dos desenvolvedores, incluindo ferramentas e práticas recomendadas. Recomendar adoção de assistentes de desenvolvimento para sugestões contextuais.	Lista de gargalos, feedback dos desenvolvedores	Propostas de melhoria	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: IT Governance & Transformation; Informed: Architecture & Technology Visioning	Decider: Solution Engineering & Development; Advisor: IT Governance & Transformation; Recommender: Architecture & Technology Visioning; Executer: Solution Engineering & Development
4	Develop Autonomy Guidelines	Desenvolver guidelines claras para o desenvolvimento seguro e eficiente com maior autonomia. Criar diretrizes com exemplos práticos de uso de IA no ciclo seguro de desenvolvimento.	Propostas de melhoria	Guidelines de autonomia	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Cybersecurity; Informed: Architecture & Technology Visioning	Decider: Solution Engineering & Development; Advisor: Cybersecurity; Recommender: Architecture & Technology Visioning; Executer: Solution Engineering & Development
5	Implement Support Mechanisms	Implementar mecanismos de suporte contínuos para desenvolvedores, incluindo treinamentos e feedback. Incorporar canais inteligentes de suporte (chatbots técnicos, FAQs dinâmicos, mentoring assistido).	Guidelines de autonomia, feedback dos desenvolvedores	Mecanismos de suporte	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: IT Governance & Transformation; Informed: Data, AI & New Technology	Decider: Solution Engineering & Development; Advisor: IT Governance & Transformation; Recommender: Data, AI & New Technology; Executer: Solution Engineering & Development

Implement DevSecOps Practices

A implementação das práticas de DevSecOps conforme planejado é fundamental para garantir que a segurança seja integrada de forma contínua e eficiente ao ciclo de desenvolvimento de software.

Este processo envolve a aplicação das políticas e ferramentas definidas na estratégia DevSecOps, a automação de testes de segurança e a integração contínua.

A implementação eficaz requer a colaboração estreita entre as equipes de desenvolvimento, operações e segurança para garantir que todos os aspectos de segurança sejam considerados e aplicados de maneira uniforme.

Treinamentos e workshops são realizados para capacitar a equipe e garantir que todos estejam alinhados com as novas práticas.

A implementação de DevSecOps promove uma abordagem proativa à segurança, reduzindo riscos e aumentando a qualidade do software.

- PDCA focus: Do
- Periodicidade: Contínua

#	Nome da Atividade	Descrição	Inputs	Outputs	RACI	DARE
1	Apply Security Policies	Aplicar as políticas de segurança definidas no ciclo de desenvolvimento de software. Integrar mecanismos de validação baseados em IA para reforçar políticas.	Políticas de segurança definidas	Segurança aplicada	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Cybersecurity; Informed: IT Governance & Transformation	Decider: Solution Engineering & Development; Advisor: Cybersecurity; Recommender: IT Governance & Transformation; Executer: Solution Engineering & Development

2	Automate Security Tests	Automatizar testes de segurança para garantir a verificação contínua de vulnerabilidades. Implementar testes dinâmicos adaptativos baseados em IA.	Ferramentas de automação, scripts de teste	Testes automatizados	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: IT Infrastructure & Operation; Informed: Data, AI & New Technology	Decider: Solution Engineering & Development; Advisor: IT Infrastructure & Operation; Recommender: Data, AI & New Technology; Executer: Solution Engineering & Development
3	Conduct Training	Realizar treinamentos e workshops para capacitar a equipe nas práticas de DevSecOps. Adoção de simuladores e treinamentos personalizados via assistêntes cognitivos.	Programas de treinamento, políticas de segurança	Equipe capacitada	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: IT Governance & Transformation; Informed: Architecture & Technology Visioning	Decider: Solution Engineering & Development; Advisor: IT Governance & Transformation; Recommender: Architecture & Technology Visioning; Executer: Solution Engineering & Development

4	Integrate CI/CD	Integrar práticas de integração e entrega contínua para garantir atualizações frequentes e seguras. Automatizar validações de qualidade com sugestões baseadas em machine learning.	Ferramentas de CI/CD, políticas de segurança	Integração contínua	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: IT Infrastructure & Operation; Informed: Architecture & Technology Visioning	Decider: Solution Engineering & Development; Advisor: IT Infrastructure & Operation; Recommender: Architecture & Technology Visioning; Executer: Solution Engineering & Development
5	Collaborate with Security Teams	Colaborar com as equipes de segurança para garantir que as práticas de DevSecOps estejam alinhadas com os requisitos de segurança. Utilizar insights preditivos gerados por IA para alinhamento prático entre desenvolvimento e segurança.	Políticas de segurança, feedback das equipes de segurança	Alinhamento de segurança	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Cybersecurity; Informed: IT Governance & Transformation	Decider: Solution Engineering & Development; Advisor: Cybersecurity; Recommender: IT Governance & Transformation; Executer: Solution Engineering & Development

Monitor DevSecOps Performance

Monitorar continuamente o desempenho das práticas de DevSecOps é crucial para garantir que os objetivos de segurança e eficiência estejam sendo atingidos.

Este processo envolve a coleta e análise de dados sobre a eficácia das práticas implementadas, a detecção de vulnerabilidades e a avaliação do desempenho geral das equipes.

Relatórios regulares são gerados para documentar o status atual e identificar áreas

que necessitam de melhorias.

O feedback das partes interessadas é coletado para ajustar e refinar as práticas de DevSecOps, garantindo que elas estejam alinhadas com os objetivos de negócio e as melhores práticas do setor.

O monitoramento contínuo promove uma cultura de melhoria contínua e garante que a segurança seja mantida em todos os estágios do ciclo de desenvolvimento.

- PDCA focus: Check
- Periodicidade: Mensal

#	Nome da Atividade	Descrição	Inputs	Outputs	RACI	DARE
1	Collect Performance Data	Coletar dados de desempenho das práticas de DevSecOps implementadas. Coletar dados de logs, testes e builds via ferramentas de análise automatizada.	Resultados de testes de segurança, métricas de CI/CD	Dados de desempenho coletados	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: IT Infrastructure & Operation; Informed: Data, AI & New Technology	Decider: Solution Engineering & Development; Advisor: IT Infrastructure & Operation; Recommender: Data, AI & New Technology; Executer: Solution Engineering & Development
2	Analyze Performance Data	Analisar os dados de desempenho para identificar padrões e anomalias. Aplicar IA para identificar anomalias e sugestões de melhoria em tempo real.	Dados de desempenho coletados	Relatórios de análise	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Cybersecurity; Informed: Architecture & Technology Visioning	Decider: Solution Engineering & Development; Advisor: Cybersecurity; Recommender: Architecture & Technology Visioning; Executer: Solution Engineering & Development

3	Generate Performance Reports	Gerar relatórios detalhados sobre o desempenho das práticas de DevSecOps. Utilizar geradores de relatórios com linguagem natural e visões executivas.	Relatórios de análise, feedback das partes interessadas	Relatórios de desempenho	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: IT Governance & Transformation; Informed: Architecture & Technology Visioning	Decider: Solution Engineering & Development; Advisor: IT Governance & Transformation; Recommender: Architecture & Technology Visioning; Executer: Solution Engineering & Development
4	Review with Stakeholders	Revisar os relatórios de desempenho com as partes interessadas para assegurar conformidade e qualidade. Disponibilizar dashboards interativos com insights analíticos gerados por IA.	Relatórios de desempenho, feedback das partes interessadas	Feedback consolidado	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Cybersecurity; Informed: Data, AI & New Technology	Decider: Solution Engineering & Development; Advisor: Cybersecurity; Recommender: Data, AI & New Technology; Executer: Solution Engineering & Development

5	Document Findings	Documentar as descobertas e recomendações com base nos resultados do desempenho. Consolidar aprendizados em repositório dinâmico com sugestões assistidas.	Relatórios de desempenho, feedback consolidado	Documentação de descobertas	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Architecture & Technology Visioning; Informed: IT Governance & Transformation	Decider: Solution Engineering & Development; Advisor: Architecture & Technology Visioning; Recommender: IT Governance & Transformation; Executer: Solution Engineering & Development

Optimize DevSecOps Practices

A otimização contínua das práticas de DevSecOps é vital para manter a eficácia e eficiência das operações de desenvolvimento e segurança.

Este processo envolve a revisão regular das práticas atuais, a análise de feedback de desempenho e a implementação de melhorias com base nas descobertas.

A otimização pode incluir a atualização de ferramentas, a melhoria dos scripts de automação, a redefinição de políticas de segurança e a capacitação contínua da equipe.

A colaboração entre as equipes de desenvolvimento, operações e segurança é essencial para garantir que as melhorias sejam eficazes e alinhadas com os objetivos da organização.

A otimização contínua assegura que as práticas de DevSecOps estejam sempre atualizadas com as melhores práticas do setor e as novas ameaças de segurança.

- PDCA focus: Act
- Periodicidade: Mensal

#	Nome da Atividade	Descrição	Inputs	Outputs	RACI	DARE

1	Review Current Practices	Revisar as práticas de DevSecOps atuais para identificar áreas de melhoria. Benchmark automatizado com práticas de referência assistido por IA.	Documentação de práticas de DevSecOps	Relatório de revisão	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: IT Governance & Transformation; Informed: Architecture & Technology Visioning	Decider: Solution Engineering & Development; Advisor: IT Governance & Transformation; Recommender: Architecture & Technology Visioning; Executer: Solution Engineering & Development
2	Gather Stakeholder Feedback	Coletar feedback das partes interessadas sobre a eficácia das práticas de DevSecOps. Aplicar algoritmos para consolidar e classificar feedbacks por relevância.	Relatório de revisão, feedback das partes interessadas	Lista de melhorias sugeridas	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Cybersecurity; Informed: Data, AI & New Technology	Decider: Solution Engineering & Development; Advisor: Cybersecurity; Recommender: Data, AI & New Technology; Executer: Solution Engineering & Development
3	Develop Improvement Plan	Desenvolver um plano de melhoria para as práticas de DevSecOps com base no feedback e na revisão realizada. Incluir melhorias baseadas em aprendizado preditivo.	Lista de melhorias sugeridas	Plano de melhoria documentado	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: IT Infrastructure & Operation; Informed: Data, AI & New Technology	Decider: Solution Engineering & Development; Advisor: IT Infrastructure & Operation; Recommender: Data, AI & New Technology; Executer: Solution Engineering & Development

4	Implement Improvements	Implementar as melhorias nas práticas de DevSecOps conforme o plano desenvolvido. Adoção de melhorias com controle automatizado de impacto.	Plano de melhoria documentado	Práticas de DevSecOps melhoradas	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: IT Governance & Transformation; Informed: Architecture & Technology Visioning	Decider: Solution Engineering & Development; Advisor: IT Governance & Transformation; Recommender: Architecture & Technology Visioning; Executer: Solution Engineering & Development
5	Validate Improvements	Validar as melhorias implementadas para garantir que estejam funcionando conforme esperado. Validar com ferramentas de simulação inteligente e KPIs autogerados.	Práticas de DevSecOps melhoradas	Melhorias validadas	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Cybersecurity; Informed: Data, AI & New Technology	Decider: Solution Engineering & Development; Advisor: Cybersecurity; Recommender: Data, AI & New Technology; Executer: Solution Engineering & Development