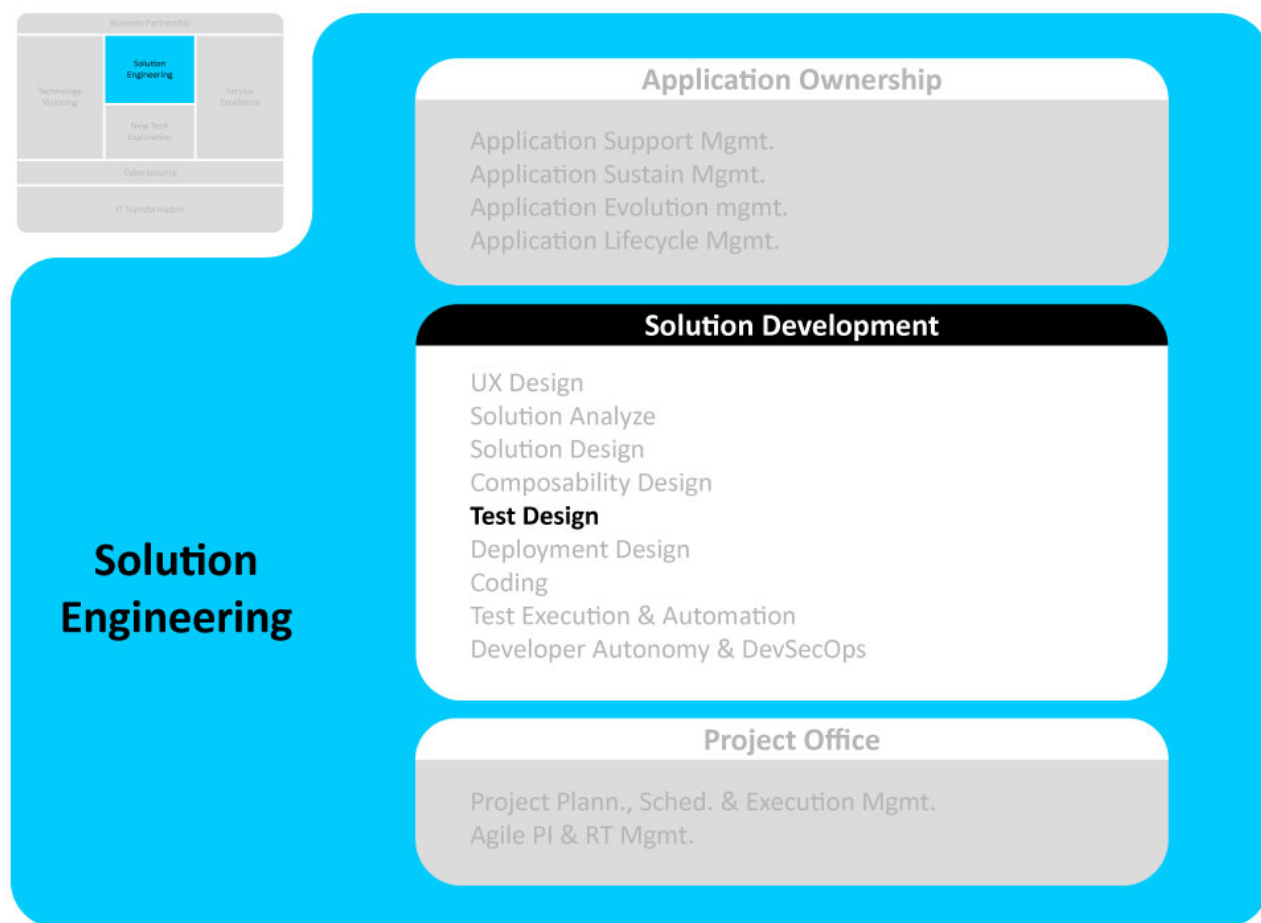




What IT needs to be ready

CIO Codex Asset & Capability Framework

CIO Codex IT Reference Model



Test Design, como parte integrante da macro capability Solution Development na camada Solution Engineering do CIO Codex Capability Framework, desempenha um papel fundamental na garantia da qualidade e confiabilidade das soluções de TI.

Esta capability envolve a criação de planos e casos de teste abrangentes, que são cruciais para identificar falhas e problemas antes do lançamento, economizando tempo e recursos, e assegurando que as soluções atendam aos padrões de qualidade estabelecidos.

No âmbito de Test Design, o plano de teste é um documento que delineia a estratégia global para testar uma solução, abordando objetivos, escopo, abordagem e cronograma.

Cada caso de teste descreve um cenário específico, com instruções detalhadas, entradas necessárias e critérios de sucesso.

Os testes funcionais verificam se as funcionalidades da solução atendem aos requisitos definidos, enquanto os testes de desempenho avaliam a capacidade da solução de lidar com cargas de trabalho específicas.

Além disso, os testes de usabilidade são conduzidos para garantir que a experiência do usuário seja intuitiva e eficiente.

As características chave do Test Design incluem detalhamento minucioso dos planos e casos de teste, cobrindo todas as funcionalidades e cenários relevantes.

A rastreabilidade é mantida para assegurar que cada requisito seja testado com sucesso. A automatização dos testes, quando apropriada, aumenta a eficiência e permite a reutilização.

A validação dos casos de teste é realizada com base em critérios específicos para garantir resultados confiáveis.

Além disso, relatórios detalhados são gerados após os testes, documentando os resultados e quaisquer problemas identificados.

O propósito fundamental da Test Design é criar planos e casos de teste que assegurem o funcionamento adequado das soluções.

Esta capability é essencial para identificar falhas e problemas antecipadamente, garantindo que as soluções atendam aos requisitos de qualidade e confiabilidade.

Dentro do CIO Codex Capability Framework, os principais objetivos da Test Design incluem garantir a qualidade, identificar falhas precocemente, assegurar a confiabilidade da solução, contribuir para a eficiência operacional e permitir abordagens de desenvolvimento mais ágeis.

Estes objetivos são cruciais para o desenvolvimento eficiente e confiável de soluções tecnológicas.

O impacto da Test Design em várias dimensões tecnológicas é significativo.

A infraestrutura pode necessitar de ambientes de teste específicos para simular diferentes cenários de uso da solução.

A arquitetura da solução é influenciada desde o início, considerando os requisitos de teste no processo de design.

Os sistemas são diretamente impactados, pois os casos de teste são executados nos sistemas em desenvolvimento.

Testes de segurança são inclusos para garantir a integridade e a confidencialidade dos dados.

Além disso, o modelo operacional da solução é afetado, já que os procedimentos de teste são uma parte integrante do ciclo de desenvolvimento.

Em resumo, Test Design é uma capability vital no CIO Codex Capability Framework, que não apenas capacita as equipes a desenvolver soluções de alta qualidade, mas também cria um processo de desenvolvimento mais ágil e eficiente.

Esta capability é crucial para enfrentar os desafios de um mercado em constante evolução, garantindo que as soluções de TI sejam confiáveis, seguras e atendam às necessidades e expectativas dos clientes.

Conceitos e Características

A capability de Test Design desempenha um papel crítico na garantia da qualidade e confiabilidade das soluções de TI.

Ao criar planos e casos de teste abrangentes, essa capability ajuda a identificar falhas e problemas antes do lançamento, economizando tempo e recursos, além de assegurar que a solução entregue atenda aos padrões de qualidade estabelecidos.

Conceitos

- **Plano de Teste:** Um plano que documenta a estratégia geral para testar uma solução, incluindo os objetivos, escopo, abordagem e cronograma dos testes.
- **Caso de Teste:** Descreve um cenário de teste específico, incluindo os passos a serem seguidos, as entradas necessárias e os critérios de sucesso.
- **Teste Funcional:** Concentra-se na verificação das funcionalidades da solução para garantir que elas atendam aos requisitos.
- **Teste de Desempenho:** Avalia o desempenho da solução, incluindo sua capacidade de lidar com cargas de trabalho e demandas específicas.
- **Teste de Usabilidade:** Visa avaliar a experiência do usuário, garantindo que a solução seja intuitiva e eficiente.

Características

- **Detalhamento:** Os planos e casos de teste são elaborados de forma minuciosa, cobrindo todas as funcionalidades e cenários relevantes.
- **Rastreabilidade:** Os testes são vinculados aos requisitos, permitindo rastrear se cada requisito foi testado com sucesso.
- **Automatização:** Quando apropriado, os testes podem ser automatizados para aumentar a eficiência e a reutilização.
- **Validação:** Os casos de teste são validados com base em critérios específicos, garantindo resultados confiáveis.
- **Relatórios:** São gerados relatórios detalhados após os testes, documentando os resultados e quaisquer problemas identificados.

Propósito e Objetivos

A capability de Test Design, ou Design de Teste, desempenha um papel crucial na garantia da qualidade e confiabilidade das soluções tecnológicas desenvolvidas.

Seu propósito fundamental é a criação de planos e casos de teste que assegurem que as soluções funcionem conforme esperado.

Essa capability é essencial para identificar falhas e problemas antes do lançamento, o que, por sua vez, assegura que as soluções atendam aos requisitos de qualidade e confiabilidade esperados.

Objetivos

Dentro do contexto do CIO Codex Capability Framework, os principais objetivos da Test Design incluem:

- **Garantir a Qualidade:** Desenvolver planos de teste abrangentes e casos de teste detalhados para garantir que a solução atenda aos padrões de qualidade estabelecidos.
- **Identificar Falhas Precocemente:** Detectar problemas e falhas de forma antecipada no ciclo de desenvolvimento, o que reduz custos e riscos associados à correção de problemas após o lançamento.

- **Assegurar a Confiabilidade:** Verificar a confiabilidade da solução, garantindo que ela funcione de forma consistente e previsível em diferentes cenários.
- **Eficiência Operacional:** Contribuir para a eficiência operacional ao reduzir o retrabalho e a necessidade de revisões extensivas após a implementação.
- **Inovação e Agilidade:** Permitir que a organização adote abordagens de desenvolvimento mais ágeis, pois problemas podem ser identificados e resolvidos rapidamente durante o ciclo de desenvolvimento.

Impacto na Tecnologia

A Test Design influencia diretamente diversas dimensões tecnológicas:

- **Infraestrutura:** Pode exigir ambientes de teste específicos para simular diferentes cenários de uso da solução.
- **Arquitetura:** Influencia a arquitetura da solução, considerando requisitos de teste desde o início do processo de design.
- **Sistemas:** Impacta diretamente o desenvolvimento de sistemas, uma vez que os casos de teste são executados nos sistemas em desenvolvimento.
- **Cybersecurity:** Inclui testes de segurança para garantir a integridade e a confidencialidade dos dados.
- **Modelo Operacional:** Afeta a operação da solução, uma vez que os procedimentos de teste são parte integrante do ciclo de desenvolvimento.

Roadmap de Implementação

A capability de Test Design, também conhecida como Design de Teste, é essencial para a garantia da qualidade e confiabilidade das soluções de TI.

Ela desempenha um papel crítico na identificação precoce de falhas e problemas, economizando tempo e recursos, além de assegurar que as soluções entregues estejam em conformidade com os padrões de qualidade estabelecidos.

Neste contexto, abaixo um roadmap de implementação para a capability de Test Design, considerando as melhores práticas e orientações do CIO Codex Capability

Framework:

- **Avaliação da Situação Atual:** Inicie com uma avaliação abrangente da situação atual da Test Design em sua organização. Identifique lacunas, pontos fortes e fraquezas na abordagem atual de teste.
- **Definição de Objetivos Claros:** Estabeleça objetivos claros para a implementação da Test Design, alinhados com os objetivos estratégicos da organização. Esses objetivos devem ser específicos, mensuráveis e relevantes.
- **Formação de Equipe Especializada:** Construa uma equipe de especialistas em Test Design, incluindo analistas de teste, engenheiros de qualidade e especialistas em automação de teste. Garanta que eles tenham as habilidades necessárias para desempenhar suas funções.
- **Padronização de Processos:** Desenvolva e documente processos padronizados para o design de testes. Isso inclui a definição de metodologias de teste, padrões de documentação e ferramentas a serem utilizadas.
- **Identificação de Requisitos de Teste:** Trabalhe em estreita colaboração com as equipes de desenvolvimento e negócios para identificar e documentar requisitos de teste claros e abrangentes.
- **Elaboração de Planos de Teste:** Crie planos de teste detalhados que abordem objetivos, escopo, abordagem de teste e cronograma. Certifique-se de que esses planos estejam alinhados com os requisitos de teste.
- **Design de Casos de Teste:** Desenvolva casos de teste específicos que detalhem cenários de teste, incluindo entradas, saídas esperadas e critérios de sucesso.
- **Automação de Teste:** Considere a automação de testes sempre que apropriado. Isso aumenta a eficiência e permite a execução de testes repetitivos de forma consistente.
- **Execução de Testes:** Realize testes funcionais, de desempenho e de usabilidade de acordo com os planos e casos de teste elaborados. Registre cuidadosamente os resultados.
- **Rastreabilidade de Requisitos:** Mantenha uma rastreabilidade rigorosa entre os requisitos e os casos de teste para garantir que todos os requisitos sejam testados.
- **Relatórios e Documentação:** Produza relatórios detalhados após a execução dos testes. Documente problemas identificados e forneça

informações claras sobre o status da qualidade.

- **Validação e Reteste:** Realize validações e reteste de problemas identificados para garantir que as correções sejam eficazes.
- **Integração Contínua:** Integre os processos de teste com os ciclos de desenvolvimento para permitir a identificação precoce de problemas e correções ágeis.
- **Feedback e Melhorias Contínuas:** Colete feedback das equipes de desenvolvimento e melhore constantemente os processos de Test Design com base nas lições aprendidas.

A implementação eficaz da capability de Test Design é fundamental para garantir a qualidade e confiabilidade das soluções de TI.

Um roadmap bem planejado, seguindo as etapas mencionadas acima, é essencial para o sucesso dessa capability. Isso resultará em soluções de maior qualidade, com identificação precoce de problemas e correções ágeis, contribuindo para a eficiência operacional e a satisfação do cliente.

Melhores Práticas de Mercado

A capability de Test Design, inserida na macro capability Solution Development e na camada Solution Engineering, desempenha um papel fundamental na garantia da qualidade e confiabilidade das soluções de TI.

Por meio da criação de planos e casos de teste abrangentes, essa capability ajuda a identificar falhas e problemas antes do lançamento, economizando tempo e recursos, além de assegurar que a solução entregue atenda aos padrões de qualidade estabelecidos.

- **Estratégia de Teste Claramente Definida:** Antes de iniciar qualquer atividade de teste, é crucial desenvolver uma estratégia de teste bem definida. Isso inclui estabelecer os objetivos, escopo, abordagem e cronograma dos testes. Uma estratégia sólida orienta todo o processo de Test Design.
- **Teste de Requisitos e Rastreabilidade:** Assegurar que cada requisito seja testado é fundamental. A rastreabilidade entre os requisitos e os casos de teste garante que nada seja deixado de lado, contribuindo para a

qualidade do produto final.

- **Abordagem de Teste Funcional e Não Funcional:** É importante adotar uma abordagem abrangente que inclua testes funcionais para verificar se a solução atende aos requisitos e testes não funcionais para avaliar desempenho, segurança, usabilidade e outros aspectos não funcionais.
- **Automatização de Testes:** A automatização de testes, sempre que viável, pode aumentar significativamente a eficiência do processo de Test Design. Testes automatizados podem ser reutilizados e executados repetidamente, economizando tempo e minimizando erros humanos.
- **Validação de Casos de Teste:** Antes da execução dos casos de teste, é fundamental validar se eles estão bem definidos e abordam adequadamente os requisitos. Isso evita retrabalho e resultados inconsistentes.
- **Ambiente de Teste Controlado:** Garantir que o ambiente de teste seja uma réplica precisa do ambiente de produção é vital. Isso ajuda a identificar problemas que podem surgir em condições reais.
- **Relatórios Detalhados:** Após a execução dos testes, gerar relatórios detalhados é uma prática essencial. Os relatórios documentam os resultados dos testes e quaisquer problemas identificados, fornecendo informações cruciais para tomada de decisões.
- **Teste de Regressão:** Implementar testes de regressão para garantir que as alterações recentes não afetem funcionalidades previamente testadas e funcionando corretamente.
- **Teste de Segurança:** Incluir testes de segurança para avaliar a integridade e confidencialidade dos dados. Isso é especialmente relevante em um cenário de crescentes preocupações com a cibersegurança.
- **Teste de Usabilidade:** Avaliar a usabilidade da solução, garantindo que ela seja intuitiva e eficiente para os usuários finais. A usabilidade desempenha um papel fundamental na satisfação do usuário.
- **Teste de Desempenho Escalonável:** Implementar testes de desempenho escalonáveis para avaliar como a solução se comporta sob diferentes cargas de trabalho e demandas crescentes.

Essas melhores práticas de mercado, inseridas no contexto da Test Design, são essenciais para assegurar a qualidade e confiabilidade das soluções tecnológicas desenvolvidas.

Ao seguir essas estratégias e abordagens, as organizações podem identificar

problemas de forma antecipada, reduzindo custos e riscos associados à correção de falhas após o lançamento.

Além disso, a abrangência dos testes, a automação quando aplicável e a atenção aos aspectos não funcionais garantem que as soluções atendam aos padrões de qualidade estabelecidos, promovendo a satisfação do usuário final e o sucesso do projeto.

Desafios Atuais

A capability de Test Design, ou Design de Teste, desempenha um papel fundamental na garantia da qualidade e confiabilidade das soluções de TI.

Ao criar planos e casos de teste abrangentes, essa capability ajuda a identificar falhas e problemas antes do lançamento, economizando tempo e recursos, além de assegurar que a solução entregue atenda aos padrões de qualidade estabelecidos.

No entanto, as organizações enfrentam desafios significativos ao adotar e integrar essa capability em seus processos de negócios e operações de TI, seguindo as melhores práticas do mercado.

Abaixo, os principais desafios atuais dentro do contexto do CIO Codex Capability Framework:

- **Complexidade dos Ambientes de Teste:** À medida que as soluções de TI se tornam mais complexas, a criação de ambientes de teste que reflitam fielmente os ambientes de produção se torna um desafio. Garantir a consistência e a confiabilidade dos ambientes de teste é crucial.
- **Automação de Testes:** A automação de testes é uma prática essencial para acelerar o processo de teste. No entanto, desenvolver scripts de automação robustos e manuteníveis requer conhecimento técnico especializado.
- **Integração Contínua e Entrega Contínua (CI/CD):** A integração de testes em pipelines de CI/CD é fundamental para garantir que as soluções sejam testadas automaticamente em todas as fases de desenvolvimento. Isso requer uma abordagem cuidadosa de integração de testes.
- **Testes de Desempenho em Escala:** Com o aumento das demandas de escalabilidade, os testes de desempenho em escala se tornam mais complexos. Garantir que uma solução possa lidar com cargas de trabalho

cada vez maiores é um desafio.

- **Testes de Segurança Avançados:** A segurança cibernética é uma preocupação constante. Realizar testes de segurança avançados para identificar vulnerabilidades e garantir a proteção dos dados é um desafio crítico.
- **Testes de Usabilidade:** Avaliar a usabilidade da solução, especialmente em um cenário de interfaces de usuário complexas, é desafiador. Garantir uma experiência do usuário intuitiva requer abordagens específicas de teste.
- **Testes de Integração Complexa:** À medida que as soluções se tornam mais integradas com sistemas externos e ecossistemas complexos, os testes de integração se tornam mais desafiadores. Garantir a interoperabilidade é crucial.
- **Gestão de Dados de Teste:** Gerenciar dados de teste realistas e confidenciais é um desafio, especialmente em ambientes regulamentados.
- **Cobertura de Teste Adequada:** Garantir que todos os aspectos críticos da solução sejam testados de maneira abrangente é um desafio. A definição de uma estratégia de cobertura eficaz é necessária.
- **Métricas e Relatórios Significativos:** Gerar métricas e relatórios que forneçam insights úteis sobre a qualidade da solução pode ser complexo. A interpretação dos resultados de teste e a comunicação eficaz dos problemas identificados são essenciais.

Superar esses desafios é essencial para garantir que a capability de Test Design cumpra seu propósito central de garantir a qualidade e confiabilidade das soluções de TI.

A abordagem minuciosa e detalhada na criação de planos e casos de teste é um componente crítico para o sucesso do desenvolvimento de software, permitindo que as organizações entreguem soluções que atendam aos padrões de qualidade e confiabilidade esperados.

Tendências para o Futuro

A capability de Test Design desempenha um papel essencial na garantia da qualidade e confiabilidade das soluções de TI.

A criação de planos e casos de teste abrangentes é fundamental para identificar falhas e problemas antes do lançamento, economizando tempo e recursos, além de assegurar que as soluções atendam aos padrões de qualidade estabelecidos.

À medida que olhamos para o futuro no contexto do CIO Codex Capability Framework, as principais tendências que moldarão o desenvolvimento dessa capability:

- Testes Autônomos com IA: A automação de testes será impulsionada pela inteligência artificial, permitindo a execução de testes autônomos que podem se adaptar dinamicamente às mudanças na solução.
- Testes de Segurança Avançados: Com a crescente preocupação com a segurança cibernética, os testes de segurança se tornarão mais avançados e abrangentes, incluindo análises de vulnerabilidades em tempo real.
- Testes de Experiência do Usuário (UX): A ênfase nos testes de usabilidade e experiência do usuário aumentará, garantindo que as soluções sejam não apenas funcionais, mas também intuitivas e eficientes.
- Testes de Desempenho em Escala: À medida que as soluções se tornam mais complexas e escaláveis, os testes de desempenho se expandirão para garantir que as soluções possam lidar com cargas de trabalho cada vez maiores.
- Testes de Compatibilidade: Com a proliferação de dispositivos e navegadores, os testes de compatibilidade se tornarão essenciais para garantir que as soluções funcionem corretamente em todas as plataformas.
- Testes de Integração Contínua: Os testes serão incorporados à prática de integração contínua, permitindo a detecção precoce de problemas à medida que o código é desenvolvido.
- Testes Baseados em Dados: A análise de dados orientará os testes, permitindo a identificação de cenários de teste relevantes com base no uso real da solução.
- Testes de Conformidade Regulatória: Com a evolução das regulamentações, os testes de conformidade se tornarão mais rigorosos para garantir que as soluções cumpram os requisitos legais.
- Testes de Resiliência: Com a crescente dependência de sistemas críticos, os testes de resiliência se tornarão mais importantes para garantir a disponibilidade contínua das soluções.
- Testes Baseados em Modelos de Negócios: Os testes serão alinhados com

os modelos de negócios das organizações, garantindo que as soluções atendam aos objetivos estratégicos.

Essas tendências refletem as expectativas do mercado em relação à evolução da capability de Test Design.

À medida que as organizações buscam soluções tecnológicas cada vez mais robustas e confiáveis, a capacidade de realizar testes abrangentes e adaptáveis desempenhará um papel fundamental.

A Test Design continuará a ser um pilar essencial na garantia da qualidade e confiabilidade das soluções de TI, contribuindo para o sucesso de projetos tecnológicos no futuro.

KPIs Usuais

A capability de Test Design desempenha um papel crítico na garantia da qualidade e confiabilidade das soluções de TI.

Ao criar planos e casos de teste abrangentes, essa capability ajuda a identificar falhas e problemas antes do lançamento, economizando tempo e recursos, além de assegurar que a solução entregue atenda aos padrões de qualidade estabelecidos.

Para avaliar o desempenho dessa capability, é fundamental monitorar os Indicadores-Chave de Desempenho (KPIs) adequados.

Abaixo, os principais KPIs usuais no contexto do CIO Codex Capability Framework para a capability de Test Design:

- Cobertura de Testes (Test Coverage): Mede a extensão em que os planos de teste cobrem os requisitos funcionais e não funcionais da solução.
- Taxa de Defeitos Identificados em Teste (Defects Identified in Testing Rate): Avalia a proporção de defeitos encontrados durante os testes em relação ao total de defeitos identificados no ciclo de desenvolvimento.
- Taxa de Reutilização de Casos de Teste (Test Case Reusability Rate): Mede a porcentagem de casos de teste que podem ser reutilizados em projetos subsequentes.
- Eficiência na Execução de Testes (Test Execution Efficiency): Calcula o tempo médio necessário para executar um caso de teste.

- Taxa de Casos de Teste Automatizados (Automated Test Cases Rate): Reflete a proporção de casos de teste que foram automatizados em relação ao total.
- Tempo Médio para Corrigir Defeitos (Average Time to Fix Defects): Avalia o tempo médio necessário para corrigir defeitos identificados nos testes.
- Taxa de Sucesso dos Testes (Test Success Rate): Mede a porcentagem de casos de teste que são bem-sucedidos sem encontrar defeitos.
- Quantidade de Falhas Relacionadas à Usabilidade (Usability-Related Failures Count): Contabiliza a quantidade de falhas identificadas durante os testes de usabilidade.
- Eficácia na Integração de Testes de Desempenho (Performance Testing Integration Effectiveness): Avalia a eficácia na incorporação de testes de desempenho nos planos de teste.
- Tempo Médio para Preparar Ambientes de Teste (Average Time to Prepare Test Environments): Calcula o tempo médio necessário para configurar ambientes de teste.
- Quantidade de Testes Bem-Sucedidos (Number of Successful Tests): Contabiliza o número de casos de teste que foram concluídos com sucesso.
- Taxa de Detecção de Defeitos Críticos (Critical Defects Detection Rate): Avalia a proporção de defeitos críticos identificados durante os testes em relação ao total de defeitos.
- Taxa de Casos de Teste Reaproveitados em Projetos Futuros (Test Cases Reused in Future Projects Rate): Reflete a porcentagem de casos de teste que foram reutilizados em projetos subsequentes.
- Quantidade de Falhas Relacionadas ao Desempenho (Performance-Related Failures Count): Contabiliza a quantidade de falhas identificadas durante os testes de desempenho.
- Tempo Médio para Gerar Relatórios de Teste (Average Time to Generate Test Reports): Calcula o tempo médio necessário para criar relatórios detalhados após a conclusão dos testes.

Esses KPIs são essenciais para avaliar o desempenho da capability de Test Design, assegurando que as soluções tecnológicas sejam testadas de forma abrangente e confiável.

O monitoramento regular desses indicadores ajuda a identificar áreas de melhoria e contribui para o sucesso na garantia da qualidade das soluções de TI.

Exemplos de OKRs

A capability de Test Design, dentro do contexto do CIO Codex Capability Framework, é fundamental para assegurar a qualidade e confiabilidade das soluções de TI.

Através da elaboração de planos e casos de teste abrangentes, esta capability identifica falhas e problemas antes do lançamento, economizando tempo e recursos, e garantindo que as soluções atendam aos padrões de qualidade exigidos.

Abaixo, são apresentados exemplos de Objetivos e Resultados-Chave (OKRs) para demonstrar a eficácia desta capability:

Garantia de Qualidade Através de Testes Abrangentes

Objetivo: Desenvolver e implementar um plano de teste abrangente para assegurar a qualidade da solução.

- KR1: Aumentar a cobertura de teste em 30% em todos os projetos.
- KR2: Reduzir a incidência de defeitos críticos em 40% antes do lançamento da solução.
- KR3: Implementar revisões regulares de casos de teste para manter a relevância e eficácia.

Deteção e Correção Antecipada de Falhas

Objetivo: Identificar e corrigir falhas nos estágios iniciais do ciclo de desenvolvimento.

- KR1: Aumentar a deteção precoce de falhas em 50%.
- KR2: Reduzir em 25% o tempo médio entre a identificação e a correção de falhas.
- KR3: Realizar testes de regressão após cada grande atualização para garantir a integridade do sistema.

Confiabilidade e Consistência da Solução

Objetivo: Verificar e garantir a confiabilidade consistente da solução em diversos cenários.

- KR1: Alcançar uma taxa de sucesso de 99% em testes de confiabilidade.
- KR2: Conduzir testes de estresse trimestrais para assegurar a robustez do sistema.
- KR3: Implementar processos de validação contínua para monitorar a estabilidade da solução.

Eficiência Operacional nos Testes

Objetivo: Melhorar a eficiência dos processos de teste.

- KR1: Reduzir o tempo de ciclo de testes em 20% através da automação.
- KR2: Aumentar a reutilização de casos de teste existentes em 30%.
- KR3: Desenvolver e implementar um sistema de rastreamento para monitorar a eficácia dos testes.

Promoção de Inovação e Agilidade em Testes

Objetivo: Adotar práticas de teste inovadoras para apoiar abordagens de desenvolvimento ágeis.

- KR1: Implementar novas ferramentas de teste que aumentem a agilidade do processo de desenvolvimento em 25%.
- KR2: Realizar workshops semestrais para explorar novas técnicas e tecnologias de teste.
- KR3: Aumentar a flexibilidade dos planos de teste para acomodar mudanças rápidas no escopo do projeto.

Esses OKRs ressaltam a importância da Test Design na criação de soluções tecnológicas que não apenas atendem, mas também superam os padrões de qualidade e confiabilidade.

A implementação desses OKRs contribui significativamente para o aprimoramento contínuo da qualidade, confiabilidade e eficiência das soluções desenvolvidas, garantindo sua eficácia e sustentabilidade no mercado competitivo atual.

Critérios para Avaliação de Maturidade

A capability Test Design, inserida na macro capability Solution Development, na camada Solution Engineering, desempenha um papel crítico na garantia da qualidade de soluções desenvolvidas.

Trata-se da criação de planos e casos de teste com o objetivo de assegurar que as soluções funcionem conforme esperado, identificando falhas e problemas antes do lançamento.

Para avaliar a maturidade desta capability, foram definidos critérios de acordo com o modelo CMMI, abrangendo cinco níveis de maturidade:

Nível de Maturidade Inexistente

- A organização não reconhece a necessidade de testes de qualidade.
- Não há processos ou estratégias estabelecidos para o design de testes.
- Não existe documentação relacionada ao design de testes.
- Os testes não são realizados em nenhuma fase do desenvolvimento.
- Não há consideração para a detecção precoce de problemas.

Nível de Maturidade Inicial

- A organização reconhece a importância dos testes, mas a implementação é reativa.
- Existem processos básicos, porém não formalizados, para o design de testes.
- A documentação relacionada ao design de testes é limitada e inconsistente.
- Os testes são realizados de forma ad hoc durante o desenvolvimento.
- Alguns esforços são feitos para estabelecer diretrizes de design de testes.

Nível de Maturidade Definido

- A organização possui processos formalizados para o design de testes.
- O design de testes é integrado às fases do ciclo de vida do desenvolvimento.
- A documentação sobre design de testes é completa e gerenciada de forma eficaz.
- Os testes são planejados e executados de acordo com procedimentos estabelecidos.
- As estratégias de design de testes são alinhadas com os objetivos do projeto.

Nível de Maturidade Gerenciado

- A organização demonstra um alto grau de maturidade no design de testes.
- Os processos de design de testes são otimizados e personalizados para projetos específicos.
- A documentação sobre design de testes é dinâmica e adaptável às mudanças.
- A execução de testes é monitorada e relatórios são gerados regularmente.
- A melhoria contínua é uma prática comum no design de testes.

Nível de Maturidade Otimizado

- A organização é líder na excelência do design de testes.
- Os processos de design de testes são altamente eficazes e promovem inovação constante.
- A documentação sobre design de testes é referência estratégica para a organização.
- A automação de testes é implementada de forma ampla e eficaz.
- Os resultados dos testes contribuem para a melhoria contínua dos processos de desenvolvimento.

Esses critérios de maturidade são fundamentais para avaliar a capacidade de uma organização em projetar e executar testes de qualidade eficazes.

A Test Design desempenha um papel vital na identificação precoce de problemas e na

garantia da qualidade das soluções desenvolvidas, contribuindo para a satisfação do cliente e o sucesso dos projetos de desenvolvimento de software.

Convergência com Frameworks de Mercado

A capability Test Design, inserida na macro capability Solution Development e na camada Solution Engineering do CIO Codex Capability Framework, desempenha um papel fundamental na criação de planos e casos de teste para garantir que as soluções desenvolvidas funcionem conforme esperado.

Esta capability é essencial para identificar falhas e problemas antes do lançamento, assegurando a qualidade e a confiabilidade da solução.

A seguir, é analisada a convergência desta capability em relação a um conjunto de frameworks de mercado reconhecidos e bem estabelecidos em suas respectivas áreas de expertise:

COBIT

- **Nível de Convergência: Médio**
- **Racional:** O COBIT, com foco na governança de TI, ressalta a importância de processos de teste robustos para a gestão de riscos e controle de qualidade. Test Design alinha-se com o COBIT ao prover metodologias estruturadas para testes, contribuindo para a governança eficaz de TI.

ITIL

- **Nível de Convergência: Médio**
- **Racional:** ITIL aborda a gestão de serviços de TI, onde Test Design desempenha um papel crítico na garantia da qualidade dos serviços. A capacidade de testar eficazmente apoia a entrega contínua de serviços de alta qualidade.

SAFe

- **Nível de Convergência:** Alto
- **Racional:** SAFe enfatiza a agilidade e qualidade em escala empresarial, onde Test Design é crucial. A capacidade de criar planos de teste eficientes e adaptáveis é fundamental para a entrega rápida e confiável em ambientes ágeis.

PMI

- **Nível de Convergência:** Médio
- **Racional:** No gerenciamento de projetos, conforme o PMI, Test Design é importante para assegurar que os resultados do projeto atendam aos padrões de qualidade e requisitos definidos, alinhando-se assim com as práticas do PMI.

CMMI

- **Nível de Convergência:** Alto
- **Racional:** CMMI, focado na melhoria de processos, se alinha estreitamente com Test Design. Processos de teste eficazes são essenciais para alcançar os níveis de maturidade do CMMI, garantindo qualidade e eficiência.

TOGAF

- **Nível de Convergência:** Médio
- **Racional:** No TOGAF, um framework de arquitetura empresarial, Test Design contribui para garantir que as arquiteturas desenvolvidas sejam funcionais e confiáveis, alinhando-se assim com as práticas do TOGAF.

DevOps SRE

- **Nível de Convergência:** Alto

- Racional: Em DevOps SRE, onde a confiabilidade e eficiência operacional são cruciais, Test Design é fundamental. Testes eficazes garantem a entrega contínua de software confiável e de alta qualidade.

NIST

- Nível de Convergência: Médio
- Racional: NIST, com foco em padrões e segurança, se beneficia de Test Design na implementação de padrões de segurança em software, contribuindo para soluções seguras e conformes.

Six Sigma

- Nível de Convergência: Médio
- Racional: Six Sigma foca na melhoria da qualidade e eficiência. Test Design é essencial para identificar e eliminar defeitos, alinhando-se com os princípios do Six Sigma.

Lean IT

- Nível de Convergência: Médio
- Racional: Lean IT visa a eficiência e eliminação de desperdícios. Test Design contribui para este objetivo ao garantir que os processos de desenvolvimento sejam eficientes e produzam resultados de alta qualidade.

Em resumo, Test Design exibe uma convergência variada com os frameworks de mercado.

Há um forte alinhamento com frameworks que enfatizam qualidade e eficiência no desenvolvimento de software, como SAFe e CMMI.

Há também uma convergência média com frameworks focados em governança de TI, gerenciamento de serviços e segurança.

Esta análise realça a importância do Test Design na garantia da qualidade e

confiabilidade das soluções de TI, um aspecto crucial para o sucesso das organizações no ambiente tecnológico atual.

Processos e Atividades

Develop Test Design Plans

A criação de planos detalhados para design de testes é um processo fundamental para garantir a eficácia e eficiência dos testes realizados em uma solução de TI.

Este processo envolve a definição da estratégia de teste, incluindo objetivos, escopo, abordagem e cronograma dos testes.

A elaboração de um plano de teste abrangente assegura que todas as áreas críticas e funcionalidades da solução sejam avaliadas de forma adequada.

O plano de teste deve contemplar diferentes tipos de testes, como funcionais, de desempenho e de usabilidade, além de definir os recursos necessários, as responsabilidades das equipes e os critérios de aceitação.

Este planejamento minucioso é essencial para prever desafios e alinhar expectativas, garantindo que a fase de execução dos testes seja realizada de maneira organizada e eficiente.

- PDCA focus: Plan
- Periodicidade: Ad-hoc

#	Nome da Atividade	Descrição	Inputs	Outputs	RACI	DARE
---	-------------------	-----------	--------	---------	------	------

1	Define Testing Strategy	Definir a estratégia geral de teste, incluindo tipos de testes, ferramentas e métodos.	Requisitos do projeto, documentação de arquitetura	Estratégia de teste documentada	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Architecture & Technology Visioning, IT Governance & Transformation; Informed: IT Infrastructure & Operation	Decider: Solution Engineering & Development; Advisor: Architecture & Technology Visioning; Recommender: IT Governance & Transformation; Executer: Solution Engineering & Development
2	Identify Test Objectives	Identificar os objetivos específicos do teste, focando nas áreas críticas da solução.	Estratégia de teste, documentação de requisitos	Objetivos de teste claros e definidos	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Data, AI & New Technology; Informed: IT Governance & Transformation	Decider: Solution Engineering & Development; Advisor: Data, AI & New Technology; Recommender: IT Governance & Transformation; Executer: Solution Engineering & Development

3	Scope Definition	Definir o escopo dos testes, detalhando quais partes da solução serão testadas.	Objetivos de teste, documentação do projeto	Documento de escopo de teste	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Architecture & Technology Visioning; Informed: IT Infrastructure & Operation	Decider: Solution Engineering & Development; Advisor: Architecture & Technology Visioning; Recommender: IT Infrastructure & Operation; Executer: Solution Engineering & Development
4	Resource Planning	Planejar os recursos necessários para a execução dos testes, incluindo equipe e ferramentas.	Documento de escopo, estratégia de teste	Plano de recursos para teste	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: IT Infrastructure & Operation; Informed: IT Governance & Transformation	Decider: Solution Engineering & Development; Advisor: IT Infrastructure & Operation; Recommender: IT Governance & Transformation; Executer: Solution Engineering & Development

5	Schedule Test Phases	Estabelecer um cronograma detalhado para a execução dos testes, alinhando com o cronograma geral do projeto.	Plano de recursos, documento de escopo	Cronograma de fases de teste	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 Testing Requirements

Identificação dos requisitos para a execução dos testes é um processo vital para garantir que todos os aspectos necessários sejam contemplados durante a fase de testes.

Este processo envolve a coleta e análise dos requisitos de teste a partir de várias fontes, incluindo especificações de projeto, documentação de requisitos do cliente e objetivos de negócio.

A identificação precisa dos requisitos de teste permite a criação de casos de teste específicos que cobrem todas as funcionalidades críticas e possíveis cenários de uso.

Além disso, esse processo ajuda a definir os critérios de aceitação para cada requisito, assegurando que a solução final atenda às expectativas e necessidades dos stakeholders.

A documentação clara dos requisitos de teste é fundamental para garantir a rastreabilidade e a cobertura completa durante a execução dos testes.

- PDCA focus: Plan
- Periodicidade: Ad-hoc

#	Nome da Atividade	Descrição	Inputs	Outputs	RACI	DARE
---	-------------------	-----------	--------	---------	------	------

1	Collect Requirements	Coletar requisitos de teste a partir de documentação do projeto e especificações do cliente.	Documentação de requisitos, especificações do cliente	Lista de requisitos de teste	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Data, AI & New Technology; Informed: IT Infrastructure & Operation	Decider: Solution Engineering & Development; Advisor: Data, AI & New Technology; Recommender: IT Infrastructure & Operation; Executer: Solution Engineering & Development
2	Analyze Requirements	Analisar os requisitos coletados para identificar áreas críticas e dependências.	Lista de requisitos de teste	Análise dos requisitos de teste	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

3	Define Acceptance Criteria	Definir critérios de aceitação para cada requisito de teste.	Análise dos requisitos de teste	Crítérios de aceitação documentados	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Data, AI & New Technology; Informed: IT Infrastructure & Operation	Decider: Solution Engineering & Development; Advisor: Data, AI & New Technology; Recommender: IT Infrastructure & Operation; Executer: Solution Engineering & Development
4	Document Testing Needs	Documentar as necessidades de teste específicas, incluindo recursos e ferramentas necessárias.	Crítérios de aceitação, análise dos requisitos de teste	Documento de necessidades de teste	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: IT Infrastructure & Operation; Informed: IT Governance & Transformation	Decider: Solution Engineering & Development; Advisor: IT Infrastructure & Operation; Recommender: IT Governance & Transformation; Executer: Solution Engineering & Development

5	Review and Approve	Revisar e aprovar os requisitos de teste e critérios de aceitação definidos.	Documento de necessidades de teste	Requisitos de teste aprovados	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Architecture & Technology Visioning, IT Governance & Transformation; Informed: IT Infrastructure & Operation	Decider: Solution Engineering & Development; Advisor: IT Governance & Transformation; Recommender: Architecture & Technology Visioning; Executer: Solution Engineering & Development
---	--------------------	--	------------------------------------	-------------------------------	--	---

Create Test Designs

A criação dos designs de testes conforme planejado é um processo central na garantia da qualidade de uma solução de TI.

Este processo envolve o desenvolvimento de casos de teste específicos que cobrem todos os cenários relevantes identificados durante a fase de planejamento.

Os casos de teste devem ser detalhados, incluindo passos a serem seguidos, entradas necessárias, critérios de sucesso e possíveis resultados esperados.

A criação de designs de testes robustos e abrangentes é essencial para assegurar que todas as funcionalidades da solução sejam avaliadas de maneira rigorosa e sistemática.

Este processo também pode incluir a criação de scripts de teste automatizados para aumentar a eficiência e a cobertura dos testes.

A qualidade dos designs de teste impacta diretamente a capacidade de identificar e corrigir problemas antes do lançamento da solução.

- PDCA focus: Do
- Periodicidade: Ad-hoc

#	Nome da Atividade	Descrição	Inputs	Outputs	RACI	DARE
---	-------------------	-----------	--------	---------	------	------

1	Develop Test Cases	Desenvolver casos de teste detalhados para cobrir todos os cenários identificados.	Requisitos de teste, critérios de aceitação	Casos de teste desenvolvidos	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Architecture & Technology Visioning; Informed: IT Infrastructure & Operation	Decider: Solution Engineering & Development; Advisor: Architecture & Technology Visioning; Recommender: IT Infrastructure & Operation; Executer: Solution Engineering & Development
2	Create Test Scripts	Criar scripts de teste, incluindo scripts automatizados, quando aplicável.	Casos de teste desenvolvidos	Scripts de teste criados	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Data, AI & New Technology; Informed: IT Governance & Transformation	Decider: Solution Engineering & Development; Advisor: Data, AI & New Technology; Recommender: IT Governance & Transformation; Executer: Solution Engineering & Development

3	Define Test Environment	Definir o ambiente de teste necessário para a execução dos testes.	Casos de teste, scripts de teste	Ambiente de teste definido	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
4	Prepare Test Data	Preparar os dados de teste necessários para a execução dos casos de teste.	Casos de teste, ambiente de teste	Dados de teste preparados	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Data, AI & New Technology; Informed: IT Governance & Transformation	Decider: Solution Engineering & Development; Advisor: Data, AI & New Technology; Recommender: IT Governance & Transformation; Executer: Solution Engineering & Development

5	Review Test Designs	Revisar os designs de teste para assegurar que cobrem todos os requisitos e cenários.	Casos de teste, scripts de teste, dados de teste	Designs de teste revisados e aprovados	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Architecture & Technology Visioning; Informed: IT Infrastructure & Operation	Decider: Solution Engineering & Development; Advisor: Architecture & Technology Visioning; Recommender: IT Infrastructure & Operation; Executer: Solution Engineering & Development
---	---------------------	---	--	--	---	---

Evaluate Test Designs

A avaliação contínua dos designs de testes com base nos resultados obtidos é crucial para garantir a qualidade e a eficácia dos testes.

Este processo envolve a análise dos resultados dos testes realizados para identificar quaisquer falhas ou áreas de melhoria nos designs de teste.

Durante a avaliação, são comparados os resultados esperados com os obtidos, verificando a cobertura dos testes e a adequação dos critérios de aceitação.

Esta etapa é fundamental para assegurar que os casos de teste sejam relevantes e eficientes na identificação de problemas.

A avaliação contínua permite ajustes nos designs de teste, aprimorando a capacidade de detectar falhas e garantindo que a solução de TI atenda aos requisitos de qualidade e desempenho esperados.

- PDCA focus: Check
- Periodicidade: Mensal

#	Nome da Atividade	Descrição	Inputs	Outputs	RACI	DARE
---	-------------------	-----------	--------	---------	------	------

1	Gather Test Results	Coletar os resultados dos testes realizados.	Resultados dos testes, logs de execução	Resultados 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 Test Coverage	Analisar a cobertura dos testes para verificar se todos os requisitos foram testados.	Resultados dos testes, critérios de aceitação	Relatório de cobertura de teste	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

3	Identify Issues	Identificar falhas e problemas nos resultados dos testes.	Resultados dos testes, relatório de cobertura de teste	Lista de problemas identificados	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Data, AI & New Technology; Informed: IT Infrastructure & Operation	Decider: Solution Engineering & Development; Advisor: Data, AI & New Technology; Recommender: IT Infrastructure & Operation; Executer: Solution Engineering & Development
4	Review and Validate	Revisar e validar os problemas identificados para determinar sua gravidade e impacto.	Lista de problemas identificados, resultados dos testes	Problemas validados	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Architecture & Technology Visioning, IT Governance & Transformation; Informed: IT Infrastructure & Operation	Decider: Solution Engineering & Development; Advisor: Architecture & Technology Visioning; Recommender: IT Governance & Transformation; Executer: Solution Engineering & Development

5	Report Findings	Reportar os achados da avaliação dos testes e recomendar ações corretivas.	Problemas validados, relatório de cobertura de teste	Relatório de avaliação de teste	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Data, AI & New Technology; Informed: IT Governance & Transformation	Decider: Solution Engineering & Development; Advisor: Data, AI & New Technology; Recommender: IT Governance & Transformation; Executer: Solution Engineering & Development
---	-----------------	--	--	---------------------------------	--	--

Update Test Designs

A revisão e atualização dos designs de testes com base nos feedbacks recebidos é um processo essencial para a melhoria contínua da qualidade dos testes.

Este processo envolve a análise dos feedbacks obtidos durante a execução dos testes e a implementação de melhorias nos casos de teste existentes.

As atualizações podem incluir a adição de novos cenários de teste, a modificação de critérios de aceitação ou a otimização de scripts de teste automatizados.

O objetivo é garantir que os designs de teste evoluam conforme a solução de TI se desenvolve, mantendo a relevância e a eficácia dos testes ao longo do tempo.

Este processo de atualização contínua assegura que os testes permaneçam alinhados com os requisitos de negócio e técnicos, proporcionando uma base sólida para a entrega de soluções de alta qualidade.

- PDCA focus: Act
- Periodicidade: Mensal

#	Nome da Atividade	Descrição	Inputs	Outputs	RACI	DARE
---	-------------------	-----------	--------	---------	------	------

1	Collect Feedback	Coletar feedbacks dos resultados de testes e das equipes envolvidas.	Feedbacks dos testes, relatórios de avaliação	Feedbacks 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 Feedback	Analisar o feedback coletado para identificar áreas de melhoria nos designs de teste.	Feedbacks coletados, resultados dos testes	Análise de feedback	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

3	Plan Updates	Planejar as atualizações necessárias nos designs de teste com base na análise de feedback.	Análise de feedback, resultados dos testes	Plano de atualização dos designs de teste	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Data, AI & New Technology; Informed: IT Infrastructure & Operation	Decider: Solution Engineering & Development; Advisor: Data, AI & New Technology; Recommender: IT Infrastructure & Operation; Executer: Solution Engineering & Development
4	Implement Updates	Implementar as atualizações nos designs de teste, incluindo novos cenários e critérios.	Plano de atualização dos designs de teste	Designs de teste atualizados	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: IT Infrastructure & Operation; Informed: IT Governance & Transformation	Decider: Solution Engineering & Development; Advisor: IT Infrastructure & Operation; Recommender: IT Governance & Transformation; Executer: Solution Engineering & Development

5	Validate Updates	Validar as atualizações implementadas para garantir que atendem aos novos requisitos.	Designs de teste atualizados, resultados dos testes	Designs de teste validados	Responsible: Solution Engineering & Development; Accountable: Solution Engineering & Development; Consulted: Architecture & Technology Visioning; Informed: IT Infrastructure & Operation	Decider: Solution Engineering & Development; Advisor: Architecture & Technology Visioning; Recommender: IT Infrastructure & Operation; Executer: Solution Engineering & Development
---	------------------	---	---	----------------------------	--	--