

TESTE FORA DE SINCRONIZAÇÃO NO POST 1 DO PROD

Contents

TESTE FORA DE SINCRONIZAÇÃO NO PROD POST 1 TESTE FORA DE SINCRONIZAÇÃO NO PROD POST 1 TESTE FORA DE SINCRONIZAÇÃO NO PROD POST 1 TESTE FORA DE SINCRONIZAÇÃO NO PROD POST 1

Tipos com base no objetivo do teste

Teste funcional

[O teste funcional](#) verifica o que o sistema faz garantindo que os recursos e as funcionalidades funcionem de acordo com os requisitos especificados.

- [Ensaio De Fumos](#): Executa verificações básicas em funcionalidades críticas para garantir que o aplicativo esteja estável para testes posteriores.
- [Teste de sanidade](#): Verifica funcionalidades específicas após pequenas alterações ou correções de bugs para garantir que elas funcionem conforme esperado.
- [Teste de integração](#): Testa a interação entre diferentes módulos para garantir que eles funcionem em conjunto corretamente.
- [Teste do sistema](#): Testa o aplicativo completo como um todo para validar a funcionalidade fim-a-fim.
- [Teste de Regressão](#): Testa novamente as funcionalidades existentes após alterações ou correções de bugs para garantir que nenhum novo defeito seja introduzido.
- [Teste de aceitação do usuário \(UAT\)](#): Realizado por usuários finais ou clientes para verificar se o software atende aos requisitos de negócios.

Teste não funcional

[O teste não funcional](#) avalia o desempenho do sistema em termos de usabilidade, desempenho, segurança e compatibilidade.

- [Teste de usabilidade](#): Avalia a facilidade de uso, a interface de usuário e a experiência geral do usuário do aplicativo.
- [Teste de compatibilidade](#): Verifica se o aplicativo funciona corretamente em diferentes

navegadores, dispositivos e sistemas operacionais.

- [Teste de desempenho](#): Mede a velocidade, a capacidade de resposta, a estabilidade e o comportamento do aplicativo em diferentes cargas de trabalho.
- [Teste de segurança](#): Identifica vulnerabilidades e garante a proteção de dados confidenciais e recursos do sistema.

Processo de teste manual

O diagrama a seguir fornece uma representação detalhada das etapas no processo de teste manual.

Processo de teste manual

- **Análise de requisitos**: compreenda e analise os requisitos do projeto, as expectativas do usuário e as necessidades comerciais para identificar o que precisa ser testado.
- **Criação do plano de teste**: crie um plano de teste que defina o escopo de teste, os objetivos, os recursos, a programação e a abordagem de teste.
- **Criação de caso de teste**: projete casos de teste detalhados com etapas de teste, resultados esperados e dados de teste para validar a funcionalidade do aplicativo.
- **Configuração do ambiente de teste**: prepare o hardware, o software, as ferramentas e os dados de teste necessários para simular o ambiente de produção.
- **Execução de caso de teste**: execute os casos de teste e compare os resultados reais com os resultados esperados para identificar quaisquer problemas.
- **Registro de Defeitos**: registre os defeitos com detalhes completos, como descrição, gravidade, etapas de reprodução e evidência de suporte.
- **Correção e reavaliação de defeitos**: verifique se os defeitos relatados foram corrigidos corretamente e confirme se o problema não existe mais.
- **Teste de regressão**: teste novamente afetado e funcionalidades relacionadas para garantir que alterações recentes não introduziram novos defeitos.
- **Relatório de encerramento do teste**: prepare um relatório final resumindo as atividades de teste, os resultados, o status do defeito e as principais lições aprendidas.

Ferramentas usadas para teste manual

- **TestRail**: uma ferramenta de gerenciamento de teste usada para organizar, executar e rastrear casos de teste manuais com recursos de relatório detalhados.
- **Xray (para Jira)**: uma ferramenta de teste integrada ao Jira que suporta gerenciamento de teste manual, rastreabilidade e rastreamento de defeito.
- **Qase**: uma plataforma de gerenciamento de testes baseada em nuvem que simplifica a criação, a execução e a colaboração em equipe do caso de teste.

- Zephyr: uma solução de gerenciamento de testes que ajuda a gerenciar atividades de testes manuais com forte integração e geração de relatórios Jira.
- Tuskr: Uma ferramenta leve de gerenciamento de testes baseada em nuvem para criar, gerenciar e rastrear casos de testes manuais de forma eficiente.

Sobre esta tradução

A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês ([link fornecido](#)) seja sempre consultado.