

SQUAD 1: Andrey Felipe de Lima; Daniela Siana Pabis; Eduardo Neves de Souza; João Pedro de Jesus Perin.

Planejamento de Testes:

1 - Definir um Escopo Claro de Testes:

Estabelecer o escopo de testes é fundamental para evitar falsas expectativas e minimizar o risco de retrabalho ou conflitos. Um escopo bem definido ajuda a manter o controle sobre o que será testado, prevenindo mal-entendidos durante o processo de desenvolvimento.

Exemplo:

O cliente deduz que “testar login” inclui testar a recuperação de senha, mas isso não estava no escopo, gerando conflito e retrabalho.

O que deve haver:

- Funcionalidades incluídas e excluídas;
- Ambiente de teste;
- Tipos de testes a serem realizados.

2 - Compreender os Requisitos Funcionais e Não Funcionais:

É crucial entender o que o sistema deve fazer (requisitos funcionais) e como ele deve se comportar (requisitos não funcionais) para criar bons testes e garantir que o produto final atenda às expectativas.

Exemplo:

Requisito Funcional: “Usuário deve conseguir efetuar pagamento via PIX.”

Requisito Não Funcional: “A resposta da transação deve levar, no máximo, 5 segundos.”

3 - Critérios de Aceitação Claros (Hard Requirements):

Critérios de aceitação bem definidos garantem que os testes sejam objetivos e

evitem subjetividades. Eles ajudam a determinar quando o produto está pronto para ser entregue.

Exemplo:

Critério vago: O sistema deve ser rápido.

Critério claro: O sistema deve carregar a tela em até 2 segundos.

4 - Identificação de Riscos:

A identificação de riscos ajuda a priorizar os testes, garantindo que os testes mais críticos sejam feitos primeiro, considerando o impacto de falhas.

Exemplo:

Alta prioridade: Permitir pagamentos.

Média prioridade: Permitir notificações.

Baixa prioridade: Permitir troca de foto do perfil.

5 - Definir Cronograma e Esforço:

Definir um cronograma claro e estimar o esforço necessário para a execução dos testes ajuda a alinhar expectativas com clientes e gestores e a prever atrasos.

Exemplo:

2 testadores com 35 casos de teste (em média, 30 minutos por caso). Um prazo realista seria cerca de 2 dias.

6 - MVQ – Mínima Qualidade Viável:

O MVQ define o nível mínimo de qualidade necessário para que o produto seja aceito pelo cliente e possa ser lançado no mercado, sem comprometer a entrega.

Exemplo:

Aplicativo bancário: O usuário deve poder consultar saldo, extrato e fazer transações simples, sem falhas.

7 - Gestão da Comunicação (Quem comunica, como e quando):

Estabelecer uma estratégia de comunicação objetiva ajuda a evitar ruídos e atrasos nas decisões durante o processo de teste.

Exemplo:

Bug / problema crítico: comunicar imediatamente ao responsável.

Ajuste pequeno: comunicar na daily.

8 - Planejar o Tipo de Teste Necessário:

É importante planejar qual tipo de teste será necessário para cada sistema, considerando suas características e as áreas de maior risco.

Exemplo:

Aplicativo de jogos: Foco em testes de desempenho;

Sistema bancário: Foco em segurança, testes de regressão e carga.

9 - Planejar Automação e Testes Manuais:

A automação reduz esforços repetitivos e aumenta a velocidade, mas os testes manuais continuam sendo essenciais para explorar novas funcionalidades, sendo necessário balancear ambos.

Exemplo:

Automação: testes de regressão em sob funcionalidades antigas;

Manual: testes manuais de uma nova funcionalidade de "Transferência via PIX".

10 - Paradoxo do Pesticida:

Testes repetitivos não identificam novos erros. Atualizar os casos de teste e adicionar cenários alternativos é essencial para garantir que novas falhas sejam descobertas e resolvidas.

Exemplo:

Estabelecer cenários alternativos para testes.

11 - Indicadores de Qualidade (KPIs):

KPIs ajudam a medir a evolução da qualidade do software, fornecendo dados quantitativos para melhorar o processo de teste.

Exemplos:

Número de falhas encontradas;

Tempo médio para correção.

12 - ISO/IEC 25010 (Modelo de Qualidade):

O modelo ISO/IEC 25010 define os atributos essenciais para garantir a qualidade de um sistema, oferecendo uma base de avaliação do software.

Exemplos de atributos:

Funcionalidade / Adequação: Verifica se o sistema faz o que foi pedido de forma correta e completa;

Usabilidade: Facilidade de uso e aprendizado rápido;

Compatibilidade: Capacidade de funcionar com outros sistemas;

Confiabilidade: Estabilidade, poucos erros e recuperação de falhas;

Eficiência: Rapidez e bom uso de recursos;

Manutenibilidade: Facilidade de alterar e evoluir o sistema;

Portabilidade: Capacidade de rodar em diferentes ambientes;

Segurança: Proteção de dados, controle de acesso e prevenção de ataques.

13 - Regra 10 de Myers:

Encontrar defeitos o mais rápido possível é essencial, pois quanto mais tarde o defeito é identificado, maior será o custo de correção.

Exemplo:

Bug encontrado em desenvolvimento: 1x custo;

Bug encontrado em homologação: 10x custo;

Bug encontrado em produção: 100x custo.

14 - Princípios Fundamentais de Teste de Software:

Elementos importantes que devem estar alinhados ao profissional de qualidade e testador para o êxito do projeto.

Demonstração de Defeitos:

O teste demonstra a existência de defeitos, mas não pode provar que eles não existem.

Inviabilidade do Teste Exaustivo:

Testar tudo não é viável, exceto em casos triviais. O foco deve ser em riscos e prioridades.

Testes Paralelos ao Desenvolvimento:

Os testes devem ser realizados em paralelo ao desenvolvimento do software, não apenas após a conclusão.

Agrupamento de Defeitos:

Os defeitos normalmente se concentram em áreas específicas do sistema.

Contexto dos Testes:

Os testes devem ser realizados de forma diferente, conforme o contexto do projeto.

15 - Preparação do Ambiente de Testes:

Garantir que o ambiente de testes esteja adequadamente configurado, isolado e com dados representativos é fundamental para a confiabilidade dos resultados.

Adicional:

A conduta é um dos pontos-chave para que um novo trabalho em equipe tenha êxito. Nesse sentido, é importante ressaltar o comportamento do indivíduo para um bom relacionamento interpessoal. Embora não faça parte diretamente do planejamento de testes, vale-se destacar:

16 - Atitudes de um Profissional de Qualidade:

Um bom profissional de qualidade precisa demonstrar várias atitudes, como: foco, boa comunicação e capacidade de negociação. Essas características são essenciais e devem estar alinhadas para que o planejamento ocorra sem mais problemas.

Exemplos:

- Pressão Organizacional: Manter a qualidade com prazos menores;
- Comprometido vs Envolvido: Comprometido assume responsabilidades, envolvido apenas participa;
- Autogerenciamento: Organização, foco e controle do tempo;
- Comunicação: Clareza verbal e coerência não verbal;
- Negociação (ganha-ganha): Acordos que beneficiam todos;
- Produtividade: Priorizar tarefas e reduzir distrações;
- Fluxo Contínuo: Evitar interrupções e gargalos.