

Início Rápido em Teste QA: Resumo

Atividade Realizada na Semana 02

Seção 1: Sobre o curso

• Soft Skills

1. Habilidades Pessoais do Testador/QA
2. Habilidades Interpessoais do Testador/QA
3. Trabalho em Equipe
4. Débito Técnico

• Hard Skills

Hard Skills - Tecnologia

1. Programação → Lógica → Linguagem
2. Telecomunicação → Protocolos → Meios → Redes
3. Infraestrutura → Virtualização → Servidores → Indicadores → Equipamentos → Componentes
4. Banco de Dados → Relacionais → Não Relacionais

• Hard Skills - QA

1. Planejar
2. Analisar
3. Modelar
4. Implementar / Preparar
5. Executar → Automatizada → Manual

Seção 2: Introdução ao Teste de Software

O teste pode demonstrar a presença de defeitos, mas não podem provar sua ausência.

Testar tudo (todas as combinações de entradas e pré-condições) não é viável, exceto em casos triviais.

Em vez do teste exaustivo, **riscos** e **prioridades** são levados em consideração para dar foco aos esforços de teste.

Teste antecipado

Quando mais cedo você começar a testar e encontrar defeitos. Procurando aonde mais costuma dar problemas mais você obtém um retorno.

"Quanto mais cedo encontramos um defeito, mais barata será sua identificação e correção"

Diferenças entre Teste e QA

- O testador tem como foco principal **identificar defeitos** e validar se o produto atende aos requisitos estabelecidos.
- O QA atua de forma proativa, buscando otimizar **todo o processo de desenvolvimento**. O QA visa prevenir erros, implementar melhorias contínuas e assegurar que os padrões de qualidade sejam seguidos em todas as etapas, desde o planejamento até a entrega

IEC/ISO 25010 - padrão internacional que define um modelo de qualidade para sistemas e produtos de software

Adequação Funcional

Usabilidade

Compatibilidade

Confiança

Eficiência e Desempenho: se refere à capacidade do software de ser rápido que podemos dividir em 3 sub características:

1º (Performance) Que o sistema processe as requisições e retorne os resultados, sem longas esperas para o usuário

2º Como o sistema utiliza os recursos do sistema, como memória RAM, disco, processador e rede

3º À habilidade do sistema de lidar com um grande volume de usuários e transações, especialmente em picos de demanda.

Manutenibilidade : A facilidade de dar manutenção a um software. Muitas vezes o software já está em uso, por um grande número de pessoas, e não pode haver interrupções que afetem a disponibilidade. Se ocorrer uma falha, a correção precisa ser rápida. Se houver necessidade de mudança (e mudanças sempre acontecem), o sistema precisa ser capaz de reagir rapidamente para implementar melhorias ou correções. podemos dividir em 5

características:

1º Modularidade: o software é dividido em partes independentes, se uma parte falha ela pode ser isolada sem afetar o sistema

2º Reusabilidade: criar componentes ou funcionalidades de software, de modo que possam ser utilizados em diferentes contextos ou em outras partes do sistema.

3º Analisabilidade: Quando um programador abre o código, ele deve compreender sua lógica, estrutura e o funcionamento. Ou seja deve ser simples de se analisar.

4º Modificabilidade: facilidade de alterar o software.

5º Testabilidade: facilidade e viabilidade de testar um software.

Portabilidade: A capacidade do Software funcionar em diferentes ambientes. Que podem ser diferentes sistemas Operacionais (Windows, Linux,MacOs) ou navegadores (Edge, Chrome, Firefox)

Segurança: O sistema deve ser seguro contra invasões, manipulação e roubo de dados.

Podemos dividir em 5 sub características:

1. Confidencialidade: Somente pessoas autorizadas tem acesso
 2. Somente pessoas autorizadas podem modificar algumas coisa
 3. não repúdio: Uma garantia que nenhuma das partes em uma ação/transação pode negar ter a realizado
 4. Responsabilidade: É possível rastrear as atividade de um usuário dentro do sistema.
 5. Autenticidade: Legitimidade de determinada ação ou a identidade de um usuário dentro do sistema
- Testes Manuais x Testes automatizados

Testes Tradicionais:

Modelo antigo, no qual os testes eram realizados de forma monótona e com baixa participação da equipe.

Testes Ágeis:

Os testes são integrados ao desenvolvimento, com feedback contínuo e contato com a equipe.

Seção 3: Atitudes de um profissional da Qualidade

Pressão Organizacional: Os profissionais do teste enfrentam pressão devido ao fato de ter que entregar produtos dentro do prazo

Comprometido X Envolvido:

Autogerenciamento:

Comunicação Verbal e Não Verbal:

Negociação (Ganha-Ganha):

Produtividade:

Fluxo Contínuo:

Seção 4: Planejamento de Testes

Aumentar escopo → Aumentar prazo → Aumentar o custo

Reduzir o prazo → reduzir o escopo → aumentar o custo

Estrutura para o plano de teste

cronograma → prazos

orçamento

qualidade → indicadores → metas

riscos → O que pode dar errado no projeto → O que pode dar errado no produto

Mapa Mental:

Uma outra alternativa ao plano de testes de forma mais simples e visual é o mapa mental nele podemos representar:

- A ser desenvolvido
- A ser testado
- Necessidades
- Restrições

Requisitos Funcionais

Os requisitos funcionais são todos os problemas e necessidades que devem ser atendidos e resolvidos pelo software por meio de funções ou serviços.

Requisitos Não Funcionais

Os requisitos não funcionais são todos aqueles relacionados à forma como o software ou app web tornará realidade os que está sendo planejado. Ou seja, enquanto os requisitos funcionais estão focados no que será feito, os não funcionais descrevem como serão feitos.

Estórias dos Usuários

INVEST

Independente

Negociável

Valioso

Estimável

Pequeno

Testável

Sprint

Sprint Backlog → O que vamos realizar nesse ciclo

Épicos

Conjuntos de Estórias que formam uma grande entrega.

Features

Funcionalidades

Pode depende de várias estórias para existir.

Análise de Riscos → Impacto → Alto → Médio → Baixo | Prejuízo → Probabilidade → Alta → Média → Baixa

Histórico dos Riscos → Ajuda muito a tomar boas deccisões

Gráfico de Pareto

Teste pairwise

Seção 5: Análise, Modelagem e implementação