

Vamos Falar de Testes de Software!

Janaina Jamarino

Agenda

1 - Perguntas
Frequentes

1.1 O que testar

1.2 O que é
garantia de
qualidade

1.3 Porque é
importante testar

2 - Problemas de
Software

2.1 Erros

2.2 Defeitos

2.3 Falhas

3 - Os 7 princípios de
Teste de Software

4 - Técnicas e
tipos de teste

5 - Metodologias

5.1 TDD (Test
Driven-Developm
ent)

5.2 BDD
(Behavior
Driven-Developm
ent)

O que é teste e o que testar?

- O teste de software é uma investigação conduzida para fornecer informações sobre a qualidade do software produto ou serviço em teste aos interessados.
- Um procedimento destinado a estabelecer a qualidade, desempenho ou confiabilidade de algo, especialmente antes de ser levado para o usuário final..
- O teste de software também pode fornecer uma visão objetiva e independente do software para permitir que a empresa aprecie e entenda os riscos da implementação de software.
- Seu principal objetivo é revelar bugs.
- Dividido em verificação e validação.

O que é teste e o que testar?

- Atende aos requisitos que nortearam sua concepção e desenvolvimento.
- Responde corretamente a todos os tipos de entradas.
- Desempenha suas funções em um tempo aceitável.
- É suficientemente utilizável.
- Pode ser instalado e executado nos ambientes de utilização pelo usuário final..
- Alcança o resultado geral desejado pelos stakeholders.

O que é garantia de qualidade?

- A garantia da qualidade do software consiste em um meio de monitorar o processos e métodos de engenharia de software usados para garantir qualidade.
- Os métodos pelos quais isso é realizado são muitos e variados, e pode incluir a garantia de conformidade com um ou mais padrões, como ISO 9000 (ISO / IEC 9126 intitulado como “Engenharia de Software -Qualidade do produto”) ou um modelo como o CMMI(Modelo de Capacidade e Maturidade Integrado).
- Definição de requisitos de conformidade IEEE.

Diferenças entre: QA

QA - Quality Assurance é o que chamamos de Garantia de Qualidade e é a forma de prevenir erros e defeitos de fabricação de evitar problemas ao entregar produtos ou serviços para clientes.

QE

QE - Quality Engineer, é o que chamamos de Engenharia de qualidade e em desenvolvimento de software, é a gestão, desenvolvimento, operação e manutenção de sistemas de TI e arquiteturas empresariais com alto padrão de qualidade.

QC

QC - Quality Control é o que chamamos de controle de qualidade, e é um processo pelo qual as entidades revisam a qualidade de todos os fatores envolvidos na Produção.

Por que é importante testar?

- Testes rigorosos de componentes e software e a documentação associada pode ajudar a reduzir os riscos de ocorrência de defeitos.
- Quando os defeitos são detectados e posteriormente corrigidos, há uma contribuição para a qualidade dos componentes ou sistemas.
- Além disso, o teste de software também pode ser necessário para atender requisitos contratuais ou legais ou padrões específicos da indústria.
- Reduz e previne defeitos.

Todo mundo já teve experiência com defeitos de softwares.

- Erros são causados por alguém.
- O início de um defeito pode ser uma deficiência, imperfeição ou falha que resultará em uma falha.
- A falha é um bug.
- O software que não funciona corretamente pode causar muitos problemas, incluindo perda de dinheiro, tempo ou reputação comercial e até mesmo ferimento ou morte.

Bugs famosos

- Bug do Milênio (1999), custo: \$500 bilhões.
- Para economizar espaço de armazenamento de computador, softwares legados muitas vezes armazenavam anos para datas com números de dois dígitos, como 99 para 1999. Esses softwares também interpretavam 00 para significar 1900, em vez de 2000, por isso, quando o ano de 2000 veio, bugs apareceriam.
- Tratamento de Câncer (2000), custo: 8 pessoas mortas e 20 seriamente feridas.
- O software de radiação da empresa Multidate calculou mal a dosagem de radiação que deveria ser enviada, expondo pacientes a níveis letais de radiação. Os físicos que foram indicados para checar as máquinas foram condenados a morte.

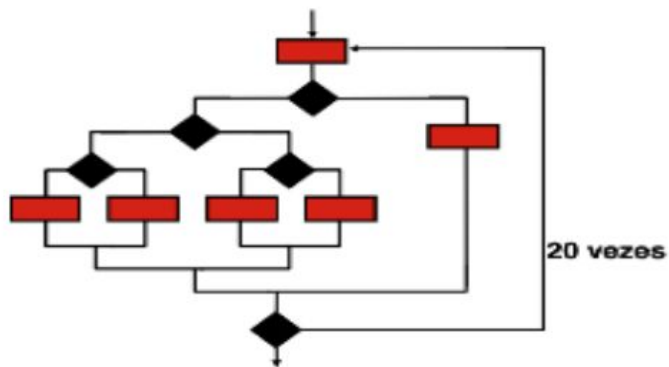
E testes tem princípios?
Sim senhores!

Os 7 princípios de teste são:

1. Os testes mostram a presença de defeitos.
1. Testes exaustivos não são possíveis (em efetivos).
1. Testar desde o início do ciclo de vida do projeto.
1. Agrupar defeitos.
1. Paradoxo dos pesticidas
1. Testes dependem do contexto.
1. Ausência de erros é uma falácia.

Os sete princípios de teste

Teste exaustivo

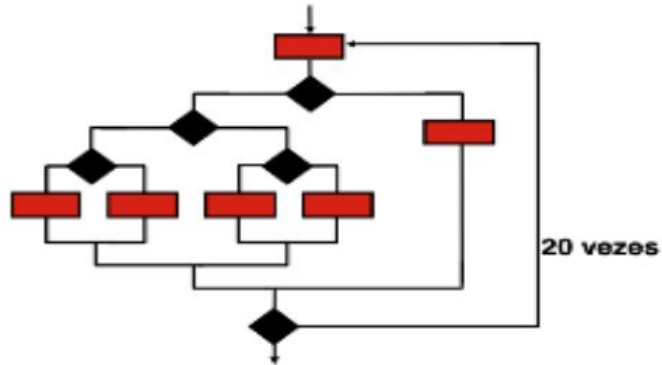


Quantos caminhos possíveis existem?

Os 7 princípios de testes

Teste exaustivo

$$5^{20} = 95367431640625$$



Quantos caminhos possíveis existem?

Tipos de teste

Primeiro, precisamos entender o que é White, black and grey-box

White -Box

O teste de caixa branca é um método de teste de software que testa interno estruturas ou funcionamento de um inscrição.

Black-box

O teste de caixa preta é um método de teste de software que examina o funcionalidade de um aplicativo sem olhar para o seu interior estruturas ou funcionamento.

Grey-Box

O teste de caixa cinza é uma mistura entre caixa-branca e caixa-preta.

Testes funcionais e não-funcionais

Testes Funcionais:

Testam a 'Funcionalidade' de um software ou um aplicativo.

Ele testa o comportamento do software baseado no documento de requisitos.

Testes não-funcionais:

Alguns aspectos complexos, como Desempenho de um aplicativo, etc, devem ser verificados e considerados para garantia de qualidade do software testado..

A qualidade depende principalmente de tempo, precisão, estabilidade, correção e durabilidade de um produto sob circunstâncias adversas.

Tipos de teste

Teste de unidade

Na programação de computadores, o teste de unidade é um método pelo qual unidades individuais de código-fonte são testadas para determinar se são apto para uso. Uma unidade é a menor parte testável de um aplicativo.

Testes de integração

Teste de integração (às vezes chamado de integração e teste, abreviado I&T) é a fase de teste de software em que módulos de software individuais são combinados e testados como um grupo.

Teste End-2-End

O teste ponta a ponta é uma metodologia usada para testar se o fluxo de um aplicativo está funcionando conforme projetado do início ao fim.

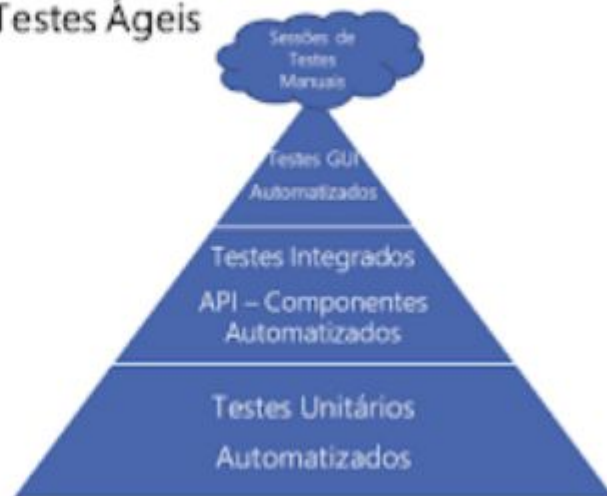
Quando a gente fala de unidade, integração e ponta-a-ponta, a gente está falando de granularidade

Pirâmide de Testes Ideal

Tradicional



Testes Ágeis



Técnicas de teste de caixa branca



- Teste de API - teste do aplicativo interna e externamente
- Cobertura de código - criação de testes para satisfazer alguns critérios de código cobertura (por exemplo, o designer de teste pode criar testes para fazer com que todas as instruções no programa sejam executadas pelo menos uma vez).
- Métodos de injeção de falha - introdução intencional de falhas no código para avaliar a eficácia das estratégias de teste.
- Teste de mutação.
- Teste estático.
- Teste de hipótese de erro.
- Teste de lista de verificação.
- Teste baseado na experiência.

Técnicas de teste de caixa preta

- O particionamento de equivalência ou particionamento de classe de equivalência é uma técnica de teste de software que divide os dados de entrada de um software unidade em partições de dados equivalentes a partir dos quais os casos de teste podem ser construídos.
- A análise de valor limite é uma técnica de teste de software em que testes são projetados para incluir representantes de valores limite em um intervalo.
- O teste da mesa de decisão é uma técnica de teste de software usada para testar comportamento do sistema para diferentes combinações de entrada. Também chamado de Causa efeito.
- O teste de transição de estado é definido como a técnica de teste de software em que mudanças nas condições de entrada causam mudanças de estado no Aplicativo em Teste (AUT).
- Teste exploratório.
- Teste baseado na experiência.

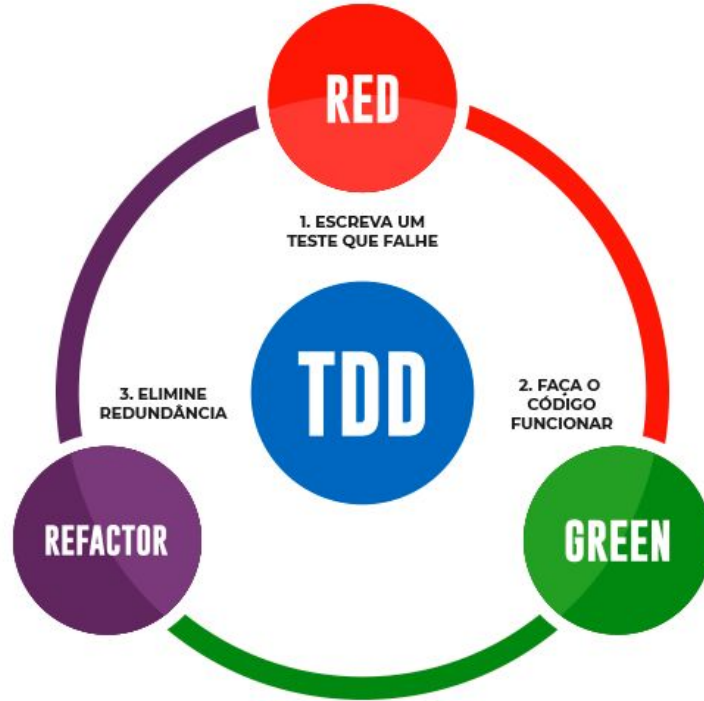
TDD

O que é TDD - Test Driven-Development?

Desenvolvimento orientado à teste: É um processo de desenvolvimento de software que depende da repetição de um ciclo de desenvolvimento muito curto: requisitos são transformados em casos de teste muito específicos, então o software é aprimorado para que os testes passem.

Como aplicar?

1. Adicionar um teste de um requisito
2. Execute todos os testes e veja se o novo teste falha.
3. Escreva um código básico para fazer o teste passar
4. Execute testes novamente para ver se ele passa
5. Código de refatoração
6. Faça tudo de novo



BDD

O que é BDD- Behavior Driven-Development?

Na engenharia de software, o desenvolvimento orientado por comportamento (BDD) é um Processo ágil de desenvolvimento de software que incentiva a colaboração entre desenvolvedores, controle de qualidade e participantes não técnicos ou de negócios em um projeto de software.

Os três amigos:

1. Negócios
1. Desenvolvimento
1. Testes

Obrigada!

Perguntas?

girls
support
girls