

Tecnologia em Análise e Desenvolvimento de Sistemas

ENGENHARIA DE SOFTWARE III

Testes de Software

Prof. Claudemir Santos Pinto
claudemir.santos2@fatec.sp.gov.br

TESTES DE SOFTWARE

Um processo geral de testes, começa com testes individuais de unidades do programa, como funções ou objetos. Na sequência, testa-se a integração destes aos sistemas e subsistemas e finalmente, após a conclusão do sistema são feitos testes de release, para verificar se o sistema funciona conforme especificado.

TESTES DE SOFTWARE

O objetivo do estágio de testes de componente é descobrir defeitos por meio de testes de componentes individuais do programa, que podem ser funções, objetos ou componentes reusáveis.

TESTES DE SOFTWARE

O teste de integração é a atividade inicial de teste de sistema, na qual são testados os componentes integrados para detectar defeitos.

O teste de releases concentra-se em testar releases do cliente e em validar se o sistema a ser liberado atende aos requisitos.

TESTES DE SOFTWARE

Algumas diretrizes que aumentam a probabilidade de que os testes de defeitos sejam bem-sucedidos:

- 1 – Escolher entradas que forcem o sistema a gerar todas as mensagens de erros
- 2 – Projetar entradas que causem overflow dos buffers.
- 3 – Repetir a mesma entrada ou série várias vezes
- 4 – Forçar a geração de saídas inválidas
- 5 – Forçar os resultados de cálculos a serem muito grandes ou muito pequenos

TESTES DE SOFTWARE

Ao testar sistemas, você deve tentar 'quebrar' o sistema usando sua experiência e diretrizes para escolher os tipos de casos de teste eficazes na descoberta de defeitos em outros sistemas.

TESTES DE SOFTWARE

Testes de software são divididos em alguns tipos, de acordo com seu objetivo particular:

- **Teste de configuração**

Testa se o software funciona no hardware que ele será instalado.

- **Teste de instalação**

Testa se o software instala como planejado em diferentes hardwares e sob diferentes condições como pouco espaço de memória, interrupções de rede, interrupções na instalação, etc.

TESTES DE SOFTWARE

- **Teste de integridade**

Testa a resistência do software à falhas (robustez).

- **Teste de Segurança**

Testa se o sistema e os dados são acessados de maneira segura apenas pelo autor das ações.

- **Teste funcional**

Testa os requisitos funcionais, os casos de uso. A aplicação faz o que deveria fazer?

TESTES DE SOFTWARE

- **Teste de unidade**

Testa um componente isolado ou classe do sistema.

- **Teste de integração**

Testa se um ou mais componentes combinados funcionam de maneira satisfatória

TESTES DE SOFTWARE

■ **Teste de volume**

Testa o comportamento do sistema operando com o volume “normal” de dados e transações envolvendo o banco de dados durante um longo período de tempo.

TESTES DE SOFTWARE

■ **Teste de performance**

O teste de performance se divide em 3 tipos:

✓ Teste de carga

Testa o software sob as condições normais de uso. Ex.: tempo de resposta, número de transações por minuto, usuários simultâneos, etc.

✓ Teste de stress

Testa o software sob condições extremas de uso. Grande volume de transações e usuários simultâneos. Picos excessivos de carga em curtos períodos de tempo.

TESTES DE SOFTWARE

✓ Teste de estabilidade

Testa se o sistema se mantém funcionando de maneira satisfatória após um período de uso.

■ **Teste de usabilidade**

Teste focado na interação do usuário com o sistema, consistência da interface (como conduz o usuário), layout, acesso às funcionalidades, mecanismos de interação, etc.

TESTES DE SOFTWARE

- **Testes de caixa branca e caixa preta**

Basicamente, teste de caixa branca envolve o código e o de caixa-preta, não.

- **Teste de regressão**

Reteste de um sistema ou componente para verificar se alguma modificação recente causou algum efeito indesejado e para certificar que o sistema ainda atende aos requisitos.

TESTES DE SOFTWARE

- **Teste de manutenção**

Testa se a mudança de ambiente não interferiu no funcionamento do sistema.

Projeto de casos de teste

É a parte do teste de sistemas e de componentes, no qual você projeta os casos (entradas e saídas esperadas) que testam o sistema. A meta é criar um conjunto de casos de testes eficazes para descobrir defeitos do programa e demonstrar que o sistema atende aos requisitos.

Para projetar um caso de teste, selecione uma característica do sistema ou do componente que você vai testar. Depois selecione um conjunto de entradas para executar aquela característica, documente as saídas esperadas e verifique se coincidem com as saídas reais.

Projeto de casos de teste

Uma abordagem para a elaboração do projeto de casos de teste é o teste estrutural, que baseia-se na análise de um programa para determinar seus caminhos e no uso dessa análise para ajudar na seleção dos casos de teste, de forma que nenhuma parte do sistema seja esquecida.

Exemplo de criação de caso de teste baseado em caso de uso

[UC07]	
Nome:	Cadastrar usuário
Atores:	Vendedor
Prioridade:	Essencial
Requisitos associados:	<ul style="list-style-type: none"> [RF02] Cadastro de usuários
Entradas:	<ul style="list-style-type: none"> Nome, CPF, Data de Nascimento Login (exceto para funcionário) Senha (exceto para funcionário) Crédito, data inicial, final, permissão para comprar em dinheiro, dias da semana (apenas funcionário)
Pré-condições:	<ul style="list-style-type: none"> O vendedor deve estar logado no sistema O usuário não deverá estar cadastrado no sistema
Pós-condições:	<ul style="list-style-type: none"> O usuário deve estar cadastrado no banco de dados
Fluxos de eventos	
Fluxo Normal:	<ol style="list-style-type: none"> O usuário seleciona a opção "Cadastrar Usuário". [Fluxo Excepcional 1] O usuário seleciona se o cadastro a ser feito é de funcionário, gerente ou vendedor. O usuário fornece os dados da pessoa a ser cadastrada. O sistema verifica se todas as informações obrigatórias foram fornecidas. [Fluxo Excepcional 2] O sistema valida as informações fornecidas. O sistema mostra uma tela para confirmação do cadastro. O usuário confirma o cadastro. Os dados são armazenados na base de dados.
Fluxo Excepcional 1:	<ol style="list-style-type: none"> A opção "Voltar" é selecionada. Todos os dados informados são descartados. A tela inicial do sistema é mostrada.
Fluxo Excepcional 2:	<ol style="list-style-type: none"> Uma ou várias informações obrigatórias não são informadas ou o formato da informação não é o adequado. O sistema mostra uma tela informando que falta uma ou mais informações obrigatórias. O sistema retorna para a tela "Cadastrar Pessoa" com os dados informados nos seus respectivos lugares.

Automação de testes

Como já foi dito, o teste é uma fase dispendiosa e trabalhosa do processo de software. Sendo assim, as ferramentas de teste estão entre as primeiras ferramentas de software a serem desenvolvidas.

A automação reduz os custos de teste pelo apoio ao processo de teste com uma variedade de ferramentas de software.

TESTES DE SOFTWARE

7 princípios do teste de software:

1 – Teste demonstra a presença de defeitos.

...PORÉM nunca poderão indicar a ausência absoluta deles!
Os testes reduzem a probabilidade que erros desconhecidos permaneçam no sistema, mas mesmo que nenhum defeito seja encontrado isso não é prova de conformidade.

TESTES DE SOFTWARE

7 princípios do teste de software:

2 – Teste exaustivo é impossível.

Mesmo com auxílio da automação, o número de combinações possíveis de cenários de teste numa aplicação é gigantesco, inviabilizando a possibilidade de se afirmar que TUDO foi testado.

Ok, não iremos testar tudo. Mas então o quanto devemos testar? Essa escolha deve ser feita com base no risco que cada funcionalidade oferece para o negócio. Quanto maior o risco, maior o foco que deve ser dado a esta parte do sistema.

TESTES DE SOFTWARE

7 princípios do teste de software:

3 – Testes devem iniciar o quanto antes e erros encontrados tarde custam mais para corrigir.

Testes não devem ser vistos como uma etapa final ou um simples "polimento" do sistema.

Iniciando o mais cedo possível no ciclo de vida do desenvolvimento do software, diminuámos o custo das correções e possibilitamos que erros de design, requisitos e arquitetura sejam encontrados no momento ideal.

TESTES DE SOFTWARE

7 princípios do teste de software:

4 – Agrupamento de defeitos

80% dos defeitos são causados por 20% do código. Ao identificar essas áreas sensíveis, os testes podem priorizá-las, enquanto ainda procuram por erros nas demais regiões.

Um aspecto importante deste princípio é a necessidade de analisar cuidadosamente as causas-raízes de determinado agrupamento. Na maioria das vezes, a análise destas causas levam a melhorias no processo de desenvolvimento da empresa.

TESTES DE SOFTWARE

7 princípios do teste de software:

5 -Paradoxo do Pesticida

Caso os mesmos testes sejam aplicados repetidamente, em determinado momento eles deixam de ser úteis, ou seja, não conseguem encontrar nenhum novo defeito. Por isso, os testes precisam ser revisitados com frequência. Na nossa metáfora, testar outros princípios ativos.

7 princípios do teste de software:

6 – Teste é dependente do contexto

Nem toda aplicação tem o mesmo nível de risco e nem todo defeito traz o mesmo impacto quando ocorre. Não há sentido algum em testar uma ferramenta de treinamento da mesma maneira que se testa um site de e-commerce. E é igualmente absurdo imaginar que este último deve ser testado da mesma maneira que um sistema de controle aéreo.

Diferentes tipos de aplicações exigem a aplicação de técnicas diferentes de teste.

TESTES DE SOFTWARE

7 princípios do teste de software:

7 – A ilusão da ausência de defeitos

De nada adianta eliminar um sem-número de defeitos se o sistema não foi projetado com foco nas necessidades do negócio. Em casos como este, os testes só terão contribuído para fazer com que o sistema funcione perfeitamente da maneira que o cliente NÃO precisa.

TESTES DE SOFTWARE

A busca constante por antecipar cada vez mais as possíveis falhas da aplicação e assegurar que o sistema entregue atenda as reais necessidades do cliente, agregarão valor ao seu negócio.

Fixação – escreva com suas palavras:

1. Cite 3 dicas que aumentam as chances dos testes de defeitos serem bem sucedidos.
2. Exemplifique 3 tipos de testes de software
3. Diferencie teste de carga, teste de stress e teste de estabilidade
4. O que é um caso de teste ?
5. Como projetar os casos de teste ?

Pesquisar :

1. 5 ferramentas automatizadas para testes.
2. Qual a diferença de testes de caixa-branca e caixa preta ?
3. Segundo a **IEEE 829**, quais os documentos necessários para o projeto de testes de software?

Atividade Complementar – PI:

1. Elaborar Casos de Testes de 5 funcionalidades do seu Proj Interdisciplinar.
2. Elaborar Glossário do PI (lista de termos técnicos ou específicos do negócio)

Atividade para ser entregue como tarefa no Teams