

SISTEMAS DE INFORMAÇÃO

GESTÃO E QUALIDADE DE SOFTWARE

JOÃO VITOR GOMES PEREIRA - 82329432

MATEUS HENRIQUE SALVADOR - 82323463

FELIPE CARDOSO SILVA – 82326693

ATIVIDADE DE AULA

PROF.º DOCENTE – ROBSON CALVETTI

SÃO PAULO – SP

2025

SUMÁRIO

CONCEITOS E ESTRATÉGIAS DE TESTES DE SOFTWARE	3
CONCEITO DE TESTE DE SOFTWARE	3
ESTRATÉGIAS DE TESTES	3
VERIFICAÇÃO E VALIDAÇÃO	4
TIPOS DE TESTES	5
TESTE UNITÁRIO.....	5
TESTE DE INTEGRAÇÃO	5
TESTE DE SISTEMA	5
TESTE DE VALIDAÇÃO.....	6
DEPURAÇÃO (DEBUGGING)	6
CONCLUSÃO	7
REFERÊNCIAS	8

CONCEITOS E ESTRATÉGIAS DE TESTES DE SOFTWARE

A qualidade do software é um dos aspectos mais críticos no desenvolvimento de sistemas. Para garantir que um software atenda aos requisitos esperados, é essencial aplicar estratégias de teste bem definidas. Os testes são responsáveis por identificar falhas, verificar a conformidade do sistema com as especificações e validar se o produto final atende às necessidades dos usuários.

Conceito de Teste de Software

O teste de software é um processo sistemático utilizado para verificar e validar o comportamento de um sistema, garantindo que ele funcione corretamente sob diferentes condições. Seu objetivo não é apenas encontrar erros, mas também garantir que o software atenda aos requisitos funcionais e não funcionais, como desempenho, usabilidade e segurança.

Os testes são parte fundamental do ciclo de vida do desenvolvimento e ajudam a reduzir os custos de manutenção ao identificar problemas o mais cedo possível. Além disso, são essenciais para garantir que o software tenha um comportamento previsível antes de ser entregue ao usuário final.

Estratégias de Testes

As estratégias de teste são abordagens utilizadas para planejar e conduzir testes de software de forma eficiente. Elas incluem diferentes tipos de testes aplicados em diversas fases do desenvolvimento. Algumas das principais estratégias são:

- **Teste baseado em requisitos:** Foca em verificar se o software atende a todos os requisitos especificados.
- **Teste baseado em riscos:** Prioriza os testes de acordo com os componentes do sistema mais críticos ou propensos a falhas.
- **Teste baseado em erros passados:** Usa dados históricos sobre falhas anteriores para testar áreas do sistema que costumam apresentar problemas.
- **Teste incremental:** Os módulos são testados progressivamente à medida que são integrados ao sistema.
- **Teste exploratório:** Realizado sem um plano de testes fixo, sendo guiado pela experiência do testador para encontrar possíveis defeitos.

Verificação e Validação

A verificação e a validação são processos distintos, porém complementares, no contexto dos testes de software.

- **Verificação:** Avalia se o software está sendo desenvolvido corretamente, verificando se cada fase do desenvolvimento atende às especificações estabelecidas. Isso inclui revisões de código, inspeções e testes estáticos.
- **Validação:** Confirma que o software atende às necessidades do usuário e cumpre os requisitos esperados. Essa etapa ocorre após a implementação e envolve testes dinâmicos para garantir a funcionalidade correta do sistema.

A verificação responde à pergunta “Estamos construindo o software corretamente?”, enquanto a validação questiona “Estamos construindo o software certo?”.

TIPOS DE TESTES

Os testes de software podem ser classificados em diferentes níveis, cada um com um propósito específico dentro do processo de garantia da qualidade.

Teste Unitário

O teste unitário verifica o funcionamento correto de módulos ou componentes individuais do software. Ele é realizado pelos desenvolvedores e foca em validar pequenos blocos de código isoladamente, garantindo que cada função ou método cumpra seu papel corretamente.

Esse tipo de teste é automatizável e auxilia na detecção precoce de falhas, evitando que erros simples se propaguem para fases posteriores do desenvolvimento.

Teste de Integração

O teste de integração verifica a interação entre diferentes módulos do sistema, garantindo que eles funcionem corretamente quando combinados. Como os módulos individuais podem operar corretamente isoladamente, mas falharem ao serem integrados, esse teste é essencial para identificar problemas de comunicação entre os componentes.

Existem diferentes abordagens para testes de integração, como:

- Big Bang: Todos os módulos são integrados e testados de uma vez.
- Top-Down: A integração ocorre de forma hierárquica, dos módulos de nível superior para os inferiores.
- Bottom-Up: Começa pelos módulos de nível inferior e avança para os superiores.
- Sanduíche: Uma combinação das abordagens Top-Down e Bottom-Up.

Teste de Sistema

O teste de sistema avalia o software como um todo, verificando se ele atende aos requisitos especificados. Ele abrange aspectos funcionais e não funcionais, como desempenho, usabilidade, segurança e compatibilidade.

Esse teste é fundamental para identificar falhas que não seriam perceptíveis em testes unitários ou de integração, pois envolve a execução do sistema em um ambiente semelhante ao de produção.

Teste de Validação

O teste de validação tem como objetivo garantir que o software atenda às expectativas do cliente. Ele geralmente envolve testes de aceitação, nos quais os usuários finais executam o sistema e verificam se ele cumpre suas necessidades.

Os testes de validação podem ser conduzidos de duas formas:

- Teste de Aceitação do Usuário (UAT - User Acceptance Testing): Realizado pelos usuários finais para garantir que o sistema esteja pronto para uso.
- Teste Alfa e Beta: O teste alfa ocorre em um ambiente de desenvolvimento controlado, enquanto o teste beta é feito por usuários reais antes do lançamento oficial.

Depuração (Debugging)

A depuração é o processo de identificar e corrigir erros detectados durante os testes. Enquanto o teste revela a presença de falhas, a depuração investiga a causa raiz do problema e implementa uma solução.

O processo de depuração geralmente envolve:

1. Reprodução do erro: Os desenvolvedores tentam replicar a falha para entender seu comportamento.
2. Isolamento da causa: Identificação do trecho de código responsável pelo problema.
3. Correção: Implementação da solução para eliminar o erro.
4. Reteste: Reexecução dos testes para garantir que a correção não introduziu novos problemas.

A depuração é um aspecto essencial do desenvolvimento, pois ajuda a garantir que as falhas sejam eliminadas antes da entrega do software.

CONCLUSÃO

Os testes de software são fundamentais para garantir a qualidade, confiabilidade e segurança dos sistemas. A aplicação de diferentes tipos de testes em diversas fases do desenvolvimento permite identificar falhas precocemente e reduzir custos de manutenção.

Ao implementar estratégias de teste eficientes, os desenvolvedores podem assegurar que o software atenda às necessidades dos usuários e funcione corretamente em ambientes reais. Além disso, a verificação, validação e depuração são etapas cruciais para garantir que o produto final esteja livre de erros críticos.

REFERÊNCIAS

Pressman, R. S. "Engenharia de Software". Capítulos 13 e 14 abordam estratégias e técnicas de teste de software.

Sommerville, I. "Engenharia de Software". Capítulo 20 aborda os testes de software.

<https://www.revistaespacios.com/a13v34n10/13341014.html>

<https://pt.linkedin.com/pulse/hist%C3%B3ria-fundamentos-e-import%C3%A2ncia-dos-testes-de-software-garcia-wx9rf>