

Exercício Prático - Conceitos e Estratégias de Testes de Software

Nome do Grupo: InovaTech

Integrantes:

1. Gabriel Farah de Lima (RA: 822231424)
(farahzerafacul21@hotmail.com)
2. Webster Diogenes Rodrigues (RA:8222242764)
(rdiogenes.webster12@gmail.com)
3. Bianca Alves Ribeiro (RA: 8222240261)
(bialuno4@gmail.com)
4. Luiz Gustavo França de Abreu (RA: 823210075)
(luizgustavo_40@hotmail.com)
5. Fabrício de Barros Narbon (RA: 822227166)
(fabricionarbon50@gmail.com)
6. Rafael Rossetto Guitarrari (RA: 823158602)
(rafaelguitarrari@gmail.com)

Curso: Gestão e Qualidade de Software

Turma: (CCP1AN-MCD3)

Professor: Robson Calvetti

Introdução.

O teste de software é uma prática essencial para garantir a qualidade, funcionalidade e segurança de um sistema. Através de diferentes estratégias e abordagens, é possível identificar falhas e garantir que o software atenda às expectativas dos usuários e aos requisitos definidos. Este documento apresenta uma explanação sobre os conceitos e estratégias de testes de software, abordando temas essenciais para garantir a qualidade dos sistemas desenvolvidos.

Conceito de Teste.

O teste de software é um processo utilizado para avaliar a qualidade de um sistema, identificando defeitos e garantindo que ele atenda aos requisitos especificados. Ele permite a detecção precoce de erros e ajuda a evitar falhas em ambientes reais.

Estratégias de Testes.

As estratégias de testes são planos que definem como a qualidade do software será verificada. Eles incluem diferentes tipos de testes, como funcionais (que validam as funções do sistema) e não funcionais (que avaliam desempenho, segurança e usabilidade). Também podem envolver testes automatizados e integração contínua. Em suma, as estratégias de testes são essenciais para mitigar riscos, evitar falhas críticas no produto e assegurar que o software entregue ao usuário seja de alta qualidade e esteja em conformidade com as expectativas e necessidades do cliente.

Conceitos de Verificação e Validação.

Verificação:

A verificação é o processo de assegurar que o software está sendo desenvolvido de acordo com as especificações e requisitos estabelecidos. É uma atividade que foca em garantir que o produto está sendo construído corretamente, seguindo os padrões e as boas práticas de desenvolvimento.

Métodos:

- Inspeção
- Revisão de Produtos
- Walkthroughs

Validação:

A validação é o processo de avaliar se o produto de software atende aos requisitos e expectativas do usuário final. É uma atividade que foca em garantir que o produto

certo está sendo construído, ou seja, que ele faz o que deveria fazer de acordo com as necessidades do cliente.

Métodos:

A execução dos testes é feita em cenários reais. O objetivo da validação é assegurar que o software atende às necessidades do usuário e cumpre sua finalidade no ambiente real de operação.

Resumo Comparativo:

Verificação: Garante que o software foi construído corretamente, focando nos aspectos técnicos e de conformidade com as especificações.

Validação: Garante que o software certo foi construído, assegurando que atende às expectativas do usuário final e é funcional no mundo real.

Tipos de testes.

Teste de Software:

O teste de software, é um processo crucial para garantir que o software funcione conforme os requisitos especificados. Nos anos 1960 e 1970, os testes eram vistos como um "mal necessário", mas a partir dos anos 1980, eles passaram a ser tratados como um processo formal dentro do desenvolvimento de sistemas. O teste de software tem como objetivo verificar se o software realiza as funções desejadas e identificar defeitos para melhorar a qualidade do produto. A evolução das metodologias de teste ao longo dos anos reflete a crescente complexidade e importância dos testes, especialmente com o advento da internet e a necessidade de garantir segurança e performance.

Teste Unitário:

O teste unitário era uma das principais atividades dos desenvolvedores nas décadas de 1960 e 1970. Este tipo de teste foca em verificar a funcionalidade de componentes individuais do código, como funções ou métodos, de forma isolada. Os testes unitários são fundamentais para detectar erros na lógica interna do código durante a fase de desenvolvimento, permitindo que os desenvolvedores corrijam problemas antes de avançar para estágios mais complexos do teste.

Teste de Integração:

O teste de integração recebeu maior atenção a partir dos anos 1980, quando a

integração de diferentes partes do software começou a ser mais valorizada. Este tipo de teste se concentra em verificar a interação entre diferentes módulos ou componentes de um sistema. O objetivo é garantir que, quando combinados, esses módulos funcionem corretamente como um sistema coeso. O teste de integração é crucial para identificar problemas de interface ou comunicação entre componentes, assegurando que o sistema atenda aos requisitos funcionais e técnicos.

Teste de Sistema:

O teste de sistema é uma etapa no processo de desenvolvimento de software que ocorre após os testes de integração. Ele se concentra em verificar o sistema como um todo para garantir que todos os componentes e módulos funcionem juntos conforme os requisitos especificados. O objetivo é validar o comportamento do software em um ambiente que simula o uso real, testando aspectos funcionais e não funcionais, como usabilidade, desempenho e segurança. A partir dos anos 1980, com a crescente complexidade dos sistemas, o teste de sistema tornou-se uma prática mais formal e estruturada, refletindo a importância de garantir que o software funcione corretamente em seu ambiente final.

Depuração:

A depuração é um processo intimamente relacionado ao teste de software. Ela envolve identificar, isolar e corrigir defeitos ou erros no código. Durante as décadas de 1960 e 1970, quando a maior parte do esforço dos desenvolvedores era dedicada à codificação e aos testes unitários, a depuração era uma atividade crítica para corrigir problemas detectados durante esses testes. A depuração é um processo contínuo que ocorre ao longo do ciclo de desenvolvimento, especialmente durante os testes unitários e de integração, onde erros são mais facilmente identificados e corrigidos. Com o avanço das práticas de teste e a formalização dos processos de desenvolvimento a partir dos anos 1980, a depuração também evoluiu, utilizando ferramentas e técnicas mais sofisticadas para melhorar a qualidade do software.

REFERÊNCIAS

SOMMERVILLE, Ian. Engenharia de software. 10. ed. Pearson, 2011.

PRESSMAN, Roger S. Engenharia de Software: uma abordagem profissional. 8. ed. McGraw-Hill, 2014.