



Gestão e Qualidade de Software

Sistemas de Informação

Integrantes do grupo:

Kayky Cerquiaro Prado - 822155538

Enrico Aguiar Vrunski - 82210618

Thiago Ferreira Lima Gonçalves - 824156179

Matheus Tognon Siqueira - 824141731

Felipe Soares de Oliveira – 824156311

Conceitos e Estratégias de Teste de Software

O teste de software tem dois principais objetivos: garantir que o programa funcione conforme o esperado e identificar possíveis defeitos antes de seu uso. Para isso, o programa é executado com dados fictícios, e os resultados são analisados para detectar erros, anomalias ou informações sobre os atributos não funcionais do sistema.

O processo de teste tem duas finalidades essenciais. A primeira é demonstrar que o software atende aos requisitos do desenvolvedor e do cliente. No caso de softwares customizados, deve haver pelo menos um teste para cada requisito especificado. Para softwares genéricos, os testes devem abranger todas as características e combinações que serão incorporadas no produto final. A segunda finalidade é identificar falhas no comportamento do software. Isso inclui a detecção de bugs, interações indesejadas, processamentos incorretos e corrupção de dados. Os testes de defeitos são essenciais para eliminar esses problemas antes da entrega do software.

Com base nesses objetivos, os testes são divididos em duas abordagens principais. Os testes de validação verificam se o sistema executa corretamente os casos de teste que refletem seu uso esperado. Já os testes de defeitos são projetados para expor falhas e garantir que os erros sejam corrigidos antes da liberação do software.

Os processos de verificação e validação garantem que o software atenda às especificações e expectativas do usuário durante o desenvolvimento. A verificação tem como objetivo confirmar se o software atende a seus requisitos funcionais e não funcionais. A validação, por sua vez, assegura que o software realmente cumpre o que o cliente espera. O nível de confiança exigido no processo de validação depende do propósito do sistema, das expectativas dos usuários e do ambiente de mercado.

Durante o desenvolvimento, os testes ocorrem em três níveis de granularidade. O primeiro nível é o teste unitário, que avalia componentes individuais do programa, como funções e classes de objetos. Esse tipo de teste deve verificar a funcionalidade de cada método, cobrindo diferentes cenários de entrada e saída. O segundo nível é o teste de componentes, que avalia a integração entre múltiplos componentes do sistema. Ele foca na interação entre os módulos, verificando se a comunicação entre eles ocorre corretamente. O terceiro nível é o teste de sistema, que ocorre após a integração de todos os componentes. Esse teste tem como objetivo garantir que os módulos funcionem juntos corretamente e que os dados fluam entre eles sem falhas. Em algumas organizações, o teste de sistema pode envolver equipes independentes, separadas dos desenvolvedores.

Após a execução dos testes, pode ser necessário localizar e corrigir erros no código. Esse processo é chamado de depuração. O teste de software tem como função identificar a presença de defeitos, enquanto a depuração se concentra na análise e correção desses defeitos. O uso de ferramentas de depuração, como GDB para C e C++, PDB para Python e Chrome DevTools para JavaScript, auxilia os desenvolvedores na investigação das falhas e na aplicação das correções adequadas antes da entrega final do software.

Com a aplicação dessas práticas, os testes garantem que o software seja entregue com qualidade, atendendo aos requisitos técnicos e às expectativas dos usuários.

Bibliografia:

SOMMERVILLE, Ian. Engenharia de Software. 9. ed. São Paulo: Pearson Prentice Hall, 2011.