



Teste de Regressão

Alunos: José Eustáquio
João Vítor

Teste de regressão

- É uma estratégia importante para reduzir **“efeitos colaterais”**. Execute testes de regressão toda vez que for feita uma **alteração grande** no software (incluindo a integração de novos componentes).
- Você sempre pode executar testes de regressão para verificar se as mudanças no programa não introduziram **novos bugs**.



- O teste de regressão é **muito caro** e geralmente **impraticável** quando um sistema é **testado manualmente**, pois os **custos com tempo e esforço são muito altos**.
- Em tais situações, você precisa tentar escolher os **testes mais relevantes** para executar novamente, e é fácil perder testes importantes.

- **Teste de regressão.** Cada vez que um novo **módulo** é acrescentado como parte do **teste de integração**, o software **muda**. Novos caminhos de fluxo de dados são estabelecidos, podem ocorrer novas **entradas e saídas**, e nova lógica de controle é chamada. Os **efeitos colaterais** associados a essas **alterações** podem causar **problemas** em funções que antes funcionavam corretamente.

- O **teste de regressão** ajuda a garantir que as alterações (devido ao teste ou por outras razões) não introduzam **comportamento indesejado** ou **erros adicionais**. O **teste de regressão** pode ser executado **manualmente**, reexecutando um subconjunto de todos os casos de teste ou usando ferramentas **automáticas** de captura/reexecução. ***Ferramentas de captura/reexecução*** permitem que o engenheiro de software capture casos de teste e resultados para reexecução e comparação subsequente.

- No entanto, **testes automatizados**, fundamentais para o desenvolvimento **test-first**, **reduzem** drasticamente os **custos** com testes de regressão.
- Os **testes existentes** podem ser executados novamente de **forma rápida e barata**. Após se fazer uma mudança para um sistema em desenvolvimento test-first, **todos os testes** existentes devem ser executados com **êxito** antes de qualquer funcionalidade ser adicionada.

- O teste de regressão pode ser **executado** para **garantir** que **não tenham** sido introduzidos **novos erros**.
- À medida que os componentes do software vão sendo **mantidos ou adaptados**, interações inesperadas causam efeitos colaterais involuntários em outros componentes. É importante ter um conjunto de casos de **teste de regressão** pronto para verificar o comportamento do sistema depois que alterações forem aplicadas a um produto de software.
- O **teste de regressão** pode ser executado para garantir que não tenham sido introduzidos novos erros.

Regressão automatizados

- Técnicas como os testes de regressão automatizados são **úteis** quando são feitas **alterações no sistema**. As alterações podem ser expressas como histórias de usuário e o envolvimento do cliente pode priorizar as mudanças necessárias em um sistema funcional. Em suma, simplesmente, a evolução envolve a continuação do processo de desenvolvimento ágil.

- O **teste de regressão** é a reexecução do mesmo subconjunto de testes que já foram executados, para assegurar que as alterações não tenham propagado efeitos colaterais indesejados.

***O conjunto de teste de regressão
contém três classes diferentes de casos
de teste:***

Uma amostra representativa dos testes que usam todas as funções do software.

- **Objetivo:** Este conjunto de testes visa cobrir uma gama representativa de funcionalidades do software.
- **Razão:** Garante que, mesmo com mudanças no código, as funcionalidades existentes não foram inadvertidamente quebradas. Ele fornece uma cobertura geral do software para garantir que as alterações não tenham efeitos colaterais não desejados em partes não relacionadas do sistema.

Testes adicionais que focalizam as funções de software que podem ser afetadas pela alteração.

- **Objetivo:** Estes testes estão concentrados nas funções específicas do software que podem ser impactadas pela alteração recente.
- **Razão:** Foca nas áreas mais propensas a problemas devido à mudança, permitindo uma detecção precoce de problemas potenciais. Se uma alteração afeta uma parte específica do código, é crucial testar intensivamente essa área para garantir que as modificações não causem problemas inesperados.

Testes que focalizam os componentes do software que foram alterados.

- **Objetivo:** Este conjunto de testes concentra-se nos componentes específicos do software que foram alterados recentemente.
- **Razão:** Dá uma atenção especial às partes do sistema que foram diretamente modificadas. Isso é fundamental para verificar se as alterações foram implementadas corretamente e se não introduziram novos bugs ou falhas. Esses testes garantem que as alterações planejadas foram executadas com sucesso nos componentes designados.

- À medida que o **teste de integração** progride, o número de **testes de regressão** pode **crescer muito**. Portanto, o conjunto de testes de regressão deve ser projetado de forma a incluir **somente aqueles testes** que tratam de uma ou mais classes de erros em cada uma das funções principais do programa.

FERRAMENTAS DO SOFTWARE

- *TestWorks*, desenvolvida pela Software Research
<http://www.testworks.com/stwhome.html>),
é uma série completa de ferramentas de teste automáticas que ajudam no projeto de casos de teste para software desenvolvido em C/C++ e Java e proporciona suporte para teste de regressão.

- 1. [Appsurify TestBrain](#) – Melhor para otimização de testes orientados por IA
- 2. [Katalon Studio](#) — Melhor kit de ferramentas completo para cobertura completa de testes
- 3. [testRigor](#) — Melhor para converter testes manuais em testes automatizados
- 4. [IBM Rational Functional Tester \(RFT\)](#) — Melhor para simplificar a virtualização e edição de testes usando capturas de tela
- 5. [Micro Focus Unified Functional Testing \(UFT\)](#) — Melhor para automatizar aplicativos do Windows

Teste de Regressão

Sistema Bancário:

a. **Caso de Teste:** Transferência de Fundos.

Pré-condição: Contas válidas e fundos suficientes.

Ação: Realizar uma transferência entre contas.

Verificação: Confirmar que o saldo é atualizado corretamente nas contas de origem e destino.

b. **Caso de Teste:** Pagamento de Conta.

Pré-condição: Conta válida e conta de pagamento cadastrada.

Ação: Efetuar um pagamento de conta.

Verificação: Verificar se o saldo é ajustado após o pagamento.

Livro

- Engenharia de software uma abordagem profissional 8ª edição Roger S.Pressman
- Ian SommerVille engenharia software a 9ª edição
- <https://theqalead.com/tools/best-regression-testing-tools/>
- <https://www.tecnisys.com.br/testes-de-regressao/>