Testes de Software

Eficiência

15/0078692	Caio César de Almeida Beleza
16/0070287	Érico Maximiano Bandeira
16/0122180	Geovana Ramos Sousa Silva
15/0035756	Geovanne Santos Saraiva
16/0123186	Guilherme Guy de Andrade
13/0122254	Lucas Midlhey Cardoso Naves

Sumário

1. Eficiência

A eficiência se refere à quantificação da performance do software, ou seja, mede a qualidade do desempenho do programa. Em palavras mais técnicas, se refere à quantidade de recursos e código necessários para que o programa realize uma tarefa, como poder de processamento, memória, espaço em disco e energia. (NAIK; TRIPATHY, 2008).

Em 2011, a Organização Internacional de Normalização (ISO) definiu a norma ISO/IEC 25010, a qual define alguns novos conceitos de qualidade de software. Nesta norma, “eficiência” é renomeada para "eficiência de desempenho". Além disso um novo subindicador é acrescentado. De acordo com a norma 25010, os sub indicadores de eficiência passam a ser:

Subindicador	**Descrição**
Comportamento em tempo real	Grau em que os tempos de resposta e processamento e as taxas de rendimento de um produto ou sistema, ao executar suas tarefas, atendem aos requisitos.
Utilização de recursos	Grau em que as quantidades e tipos de recursos utilizados por um produto ou sistema, ao executar suas tarefas, atendem aos requisitos.
Capacidade	Grau em que os limites máximos de um produto ou parâmetro do sistema atendem aos requisitos

Esses subindicadores podem ser avaliados individualmente no processo de teste de software que indicará a eficiência do software. Dessa forma, o teste pode ser modelado para testar eficiência com vários enfoques, permitindo uma análise mais profunda e completa. Um exemplo seria a realização de testes em páginas CMSs, em que pode-se eleger características específicas do software para uma análise ainda mais esmiuçada. Os subindicadores podem ser representados pelas seguintes características (HONORATO, 2014):

Característica do Software	**Subindicador Correspondente**

Tempo de carregamento da página	Comportamento em tempo real
Total de Requisições	Utilização de recursos
Tamanho da Página	Utilização de recursos
Total de arquivos JS	Utilização de recursos
Total de arquivos CSS	Utilização de recursos

Tendo-se em mãos esses dados, o desenvolvimento de testes é facilitado, pois a partir de um conceito abstrato e de alto nível - Eficiência - foram geradas várias características técnicas presentes no software e que são testáveis. Esse processo de transformação de requisitos não funcionais em características técnicas deve ser feito antes do início dos testes, para que o fator de qualidade seja testado com acurácia.

2. Testes de Eficiência

A seguir, tem-se a descrição e casos de uso de testes que medem, direta ou indiretamente, a eficiência do software e são utilizados na indústria.

2.1 Teste de Carga

O teste de carga analisa e verifica quanto o software suporta relacionando volume de transação, acessos simultâneos e usuários. Através desse teste é possível definir a estabilidade do sistema definindo o período de grande acesso e estabelecendo um limite de operação, medir desempenho do software através da sua eficiência em grande fluxo de dados e informações, evidenciar como sistema comporta de maneira progressiva analisando o fluxo de entrada de usuários e evidenciando falhas e bugs que por ventura podem ocorrer, analisar tempo de resposta para requisições necessárias e desta forma o sistema possa ser automatizado e melhorado.

O processo de execução do teste de carga é subdividido em 5 etapas, são elas:

- Definição dos objetivos: incluem ampla variedade de medições, tempo de resposta esperado, numero de usuarios que cada atividade deve suportar e o que pode acontecer em momentos de pico.
- Configuração de ambiente para teste: O ambiente deve simular fielmente o ambiente de produção real, Isso inclui questões relacionadas à configuração e perfis da máquina, arquitetura de rede, balanceadores de carga, firewall, bases de dados, entre outras.

- Criação de cenários de carga: A criação de cenários de carga é feita tanto através dos registros de atividades dos usuários, programação ou, como na maioria dos casos, da combinação dos dois. Cenários de teste de carga incluirão pontos de validação, transações e medições.
- Execução de Teste: executa diferentes simulações a fim de analisar os dados gerados em condições reais na obtenção dos resultado.
- Análise e resultados: diferentes de outros testes, para análise do teste de carga é necessário uma atenção maior, pois analisa a quantidade de usuarios relacionados horario de pico e atividades no sistema, isso quer dizer o resultado esperado depende da funcionalidade do sistema e o crescimento do uso do software.

2.2 Teste de Estresse

O teste de estresse busca avaliar o comportamento do sistema em condições anormais, ou seja, testa se o sistema continua executando suas funções de forma correta mesmo em situação de sobrecarga da máquina em que está executando ou de uso intenso do próprio sistema. Este teste propõe que se o sistema se comporta adequadamente nessas situações, também se comporta adequadamente em situações normais (SANTANA, 2007).

O teste de estresse é geralmente executado de forma automatizada, escolhendo-se algumas funcionalidades para rodar simultaneamente, ou a mesma funcionalidade executando em paralelo consigo mesma em grande quantidade.

É esperado que não haja a perda, nem modificação de informações e que as funções terminem adequadamente sua execução e, caso retorne algum valor, retorná-lo sem variação do resultado esperado, sem causar prejuízo a estabilidade do sistema.

Os testes de estresse muitas vezes são executados em conjunto com testes de desempenho e também são confundidos com estes testes. Entretanto o teste de desempenho possui foco em verificar como o sistema lida com a carga normal e planejada de trabalho, enquanto o teste de estresse coloca o sistema em condições anormais, que se encontram fora do planejado para um sistema específico (SANTOS; NETO; RESENDE, 2010).

Um exemplo de uso prático dos testes de estresse se encontra no universo dos sistemas gerenciadores de bases de dados (SGBD), neste universo existem ferramentas especializadas em executar testes de estresse, como DTM DB Stress, JMeter e outros. Estas ferramentas executam diversas operações simultâneas nos

SGBD para garantir que eles executam corretamente, mesmo em condições anormais (PIERAZO, 2013).

2.3 Teste de Longa Duração ou Resistência

Ao contrário dos testes de carga e stress, que são analisados por curtos períodos de tempo, os testes de resistência consistem em observar a integridade do sistema, sendo submetido por uma alta carga de acessos e requerimentos de usuários por algumas horas, ou até dias.

Com esse tipo de teste é possível descobrir vários erros imprevisíveis, como por exemplo, vazamentos de memória, que acontece quando a memória alocada não está sendo devidamente liberada, o que consome a memória disponível, prejudicando a performance do sistema e em casos mais avançados, causando falhas nesse sistema.

2.4 Teste de Subida Rápida

Este tipo de teste é realizado através de uma simulação, uma subida rápida e grande de usuários simultâneos em curto período de tempo. O objetivo é determinar se o sistema em teste irá falhar ou será bem sucedido ao lidar com uma mudança brusca na carga de usuários.

2.5 Teste de Configuração

Testes de configuração verificam uma determinada aplicação em ambientes de software e hardware diferenciados. Há muita diferença entre várias especificações no mercado, desde de sua conexão a rede até seu banco de dados e também um ambiente de software que pode se diferencia de máquina para máquina, causando uma diferenciação no produto a ser analisado.

Nesse teste é possível enxergar erros que acontecem fora do ambiente do desenvolvedor que seria, muitas vezes, considerado um cenário ideal. Sendo este tipo de teste uma maneira de validar a aplicação com uma mistura de software e hardware ativas no ambiente.

Num teste aplicado por K. Stone (1999) ele fez diversos testes como: Utilizar scripts de teste de Integração, fechar e abrir várias aplicações da máquina, executar transações selecionadas para simular um uso do dia-a-dia do usuário e repetia o processo. Querendo alcançar o objetivo de validar e verificar se a aplicação funciona corretamente no ambiente que o cliente prescreveu.

2.6 Teste de Resposta

Representa a percepção do usuário de quão rápido o sistema reage para uma “Request” ou “Query”. A reação pode ser lenta ou rápida com base no tipo de atividade e o tempo necessário para processar a requisição. A consistência do tempo de resposta é medida em vários ciclos de teste, se a performance é calculada especificamente em termos de tempo de resposta temos o Teste de Throughput, que mede o throughput de um servidor em um sistema baseado em Web. Ele é uma medida do número de bytes enviados por unidade de tempo. Throughput de vários servidores em que a arquitetura de sistema pode ser medido em kilobits por segundo, consultas em banco de dados por minuto, transações por hora, ou qualquer outra característica vinculada ao tempo.
Quando se fala em teste de tempo de resposta temos mais uma ferramenta que é o teste de capacidade, mede a capacidade global do sistema e determina até que ponto a resposta tempo e throughput torna-se inaceitável. Teste de Capacidade é realizado com carga normal para determinar a capacidade extrema, onde o Stress é determinado por sobrecarregar o sistema até que ele falhe, o que também é chamado de carga de estresse, para determinar a capacidade máxima de um sistema.
Alguns defeitos de software que podem causar atraso nas respostas: alto consumo de recursos(CPU, memória, banda), padrões de design de tabelas não são seguidos e baixa qualidade nas lógicas de “*queries*”.

3. Resultados

Aluno	**Tópico**	**Descrição**	**Resultados esperados**	**Referências**
Geovana	Eficiência	Introdução e definição do tema	-	[2], [3], [5]
Lucas	Teste de Carga	Analisar o comportamento do software mediante ao fluxo de transações, tempo de resposta esperado, número de usuários que cada atividade deve suportar e o que pode acontecer em momentos de pico.	Bom tempo de resposta do servidor, fluxo de usuário consistente independente do pico de uso	

Guilherme	Teste de Estresse	Analisar o comportamento do software em situações anormais de uso, como memória limitada, sobrecarga e outros.	O software deve continuar a operar normalmente, produzindo os resultados esperados em situações normais. Por exemplo: manter o retorno de uma função independente da situação de estresse dele.	[6], [7], [8]
Caio	Teste de Resistência	Analisar a integridade do software, quando submetido à condições de carga normais por um longo período de tempo.	O software deve manter a integridade de seus dados e permanecer sem erros significativos em sua base, independenteme nte da quantidade de tempo e de carga a que esteja sendo submetido.	[4],[12]
Geovanne	Teste de Resposta	Mede a capacidade global do sistema e determina até que ponto a resposta tempo e throughput torna-se inaceitável.	A indústria apresenta as seguintes normas: para um sistema multimídia interativo, o tempo de resposta deveria ser 0.1 segundo ou menos, 90% do tempo, já para sistemas online onde usuários fazem múltiplas tarefas	[11]

			simultaneamente, o tempo de resposta deveria ser 1 segundo ou menos, 90% do tempo.	
Érico	Teste de Configuração	Analisar o software em diferentes ambientes, tanto de hardware como de software, propriamente dito.	Validar e verificar o software do ambiente desejado pelo cliente, para que não haja imprevistos em sua aplicação.	[10]

4. Referências

[1] DELAMARO, Marcio ; JINO, Mario; MALDONADO, José. **Introdução ao teste de software.** [*S. l.*]: Elsevier, 2016.

[2] HONORATO, Thiago Silveira. **Uma proposta de método de escolha para sistemas CMSs.** 2014. 166 f. TCC (Graduação) - Curso de Engenharia de Software, Universidade de Brasília, DF, 2014.

[3] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO). **25010**: **Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models**. Londres, 2011.

[4] KUNHUA, Zhu; JUNHUI, Fu; YANCUI, Li. **Research the performance testing and performance improvement strategy in web application.** 2nd International Conference on Education Technology and Computer. 2010.

[5] NAIK, Kshirasagar; TRIPATHY, Priyadarshi. **SOFTWARE TESTING AND QUALITY ASSURANCE:** Theory and Practice. Canadá: Wiley, 2008.

[6] PIERAZO, Cynthia et al. **Análise Comparativa de Ferramentas de Teste para Aplicações em Banco de Dados**. e-RAC, v. 3, n. 1, 2013.

[7] SANTANA, André Aguiar. **Metodologia de teste para acelerar o desenvolvimento de sistemas de processamento paralelo**. 2007. Tese de Doutorado. Universidade de São Paulo.

[8] SANTOS, Ismayle Sousa; NETO, Pedro Alcântara Santos; RESENDE, Rodolfo. **Geração de Testes de Desempenho e Estresse a partir de Testes Funcionais**. Revista de Informática Teórica e Aplicada, v. 17, n. 2, p. 174-192, 2010.

[9] SILVA, O. J., Borges, C. A. Salviano, C. F., Crespo, A. N., Roullier, A. C.; **Aplicação da ISO/IEC TR 15504 na Melhoria do Processo de Desenvolvimento de Software de uma Pequena Empresa**; Anais do Simpros 2003: Simpósio Internacional de Melhoria de Processo de Software, Recife, Brasil, Novembro 2003

[10] STONE, Kerry. **Course Registration System Test Plan for the Architectural Prototype,** WyIT432, V1.0, 1999, Wylie College IT.

[11] Teste de desempenho: Conceitos, Objetivos e Aplicação Disponível em: <http://www.linhadecodigo.com.br/artigo/3256/teste-de-desempenho-conceitos-objetivos-e-aplicacao-parte-1.aspx>. Acesso em: 26 de abr. 2019.

[12] XU, Z.; Zhang, J.; XU, Z.; **Memory Leak Detection Based on Memory State Transition Graph.** 18th Asia-Pacific Software Engineering Conference. 2011