

**FACULDADE 7 DE SETEMBRO – FA7**

**CURSO DE PÓS-GRADUAÇÃO EM DESENVOLVIMENTO ÁGIL DE SOFTWARE**

**TESTE DE ACEITAÇÃO COM WEBDRIVER**

**JOSÉ RILDO ROCHA LESSA**

**FORTALEZA - 2015**

JOSÉ RILDO ROCHA LESSA

TESTE DE ACEITAÇÃO COM WEBDRIVER

Monografia apresentada à Faculdade 7 de Setembro como requisito parcial para obtenção do título de Especialista em Desenvolvimento Ágil de Software.

Orientador: Prof. M. Sc. Wagner Costa.

FORTALEZA - 2015

**TESTE DE ACEITAÇÃO COM WEBDRIVER**

Monografia apresentada à Faculdade 7 de Setembro como requisito parcial para obtenção do título de Especialista em Desenvolvimento Ágil de Software.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

JOSÉ RILDO ROCHA LESSA

Monografia aprovada em: \_\_\_\_\_\_ / \_\_\_\_\_\_ / \_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. M. Sc. Wagner Costa

1º Examinador: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2º Examinador: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. M. Sc. Ciro Coelho

Coordenador do Curso

agradecimentos

Em primeiro lugar à Deus, pela força, coragem e discernimento para enfrentar todas as dificuldades e contratempos durante esta grande batalha.

À minha família que como sempre me apoiou e contribui para que esse grande desafio fosse conquistado da maneira mais amena possível.

Ao meu orientador, professor e um grande profissional na área de tecnologia Wagner Costa, que fez com excelência o seu papel como educador e mediador desse trabalho com maestria.

Aos meus amigos, que testemunharam minhas angústias, incertezas e felicidades durante o desenvolvimento deste trabalho, contribuindo de várias formas na elaboração desse trabalho.

Minha noiva Rianna que mesmo em meados do nosso casamento contribuiu diretamente para a conclusão dessa pesquisa.

RESUMO

Esse trabalho tem por objetivo apresentar uma visão geral sobre testes de aceitação usando o webdriver. Será feita uma explanação sobre a problemática de entregas de produtos de softwares com defeitos e o que isso acarreta na relação com cliente, uma introdução sobre alguns testes de softwares com ênfase em testes de aceitação e por fim a aplicação desse teste usando o webdriver salientando suas vantagens obtidas pela automatização desse tipo de testes.

**SUMÁRIO**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **1** | **INTRODUÇÃO ............................................................................................** | **10** |
|  | 1.1 | Motivação .................................................................................................... | 10 |
|  | 1.2 | Objetivo Geral ............................................................................................. | 11 |
|  | 1.2.1 | Objetivos Específicos .................................................................................. | 11 |
|  |  |  |  |
|  | **2** | **SERVIÇO ORIENTADO A ARQUITETURA ...............................................** | **13** |
|  | 2.1 | Modelo de Referencia SOA ........................................................................ | 13 |

**1 INTRODUÇÃO**

**1.1 Motivação**

Cada vez mais os softwares se mostram presentes e necessários em vários âmbitos do dia a dia das pessoas, sejam no entretenimento, estudos, saúde e principalmente no mundo dos negócios. Com essa relação dos softwares com todos os setores vem a complexidade e a intenção de serem feitos produtos de qualidade e que agregue valor de negócio a essa necessidade de mercado.

Existe uma recorrência com usuários dos sistemas supracitados, vários depoimentos que os softwares não funcionam, não atendem ao negócio, são pouco intuitivos ou simplesmente geram resistências por não ajudarem muito o “operacional” dos clientes [1]. Esses descontentamentos de clientes muitas vezes acontecem mesmo na fase de produção dessas ferramentas, onde a equipe de desenvolvimento não consegue captar do cliente sua real necessidade, assim entregando no primeiro momento módulos sem valor de negócio, com erros e gerando valores não previstos no projeto.

Uma tentativa de acertos nas entregas e captação de novos clientes é a entrega de um produto de qualidade e que esteja “pronto” pra ser usado, essa é a intenção dos testes de softwares que tentam reduzir ao máximo uma entrega de módulos com erros ou imprecisões assim evitando que vá para produção e que o cliente só enxergue os erro após homologação do produto, pois quanto mais tarde os erros forem detectados maior o custo de concertá-los, segundo a regra 10 de Myers.[2].

Rocha (2001) defende que a engenharia de software se mostra objetiva e concisa quando prima pelo estabelecimento e uso de princípios no desenvolvimento de softwares de maneira sistêmica e econômica. A fim de ter um produto que atenda o negócio e com um mínimo de qualidade ela abrange três pilares fundamentais: método, ferramenta e procedimentos. (falar sobre esse três???)

Durante o processo de desenvolvimento de software alguns componentes são responsáveis pela qualidade do projeto, dentre eles: revisões de contrato, desenvolvimento de planos de qualidade, revisões de artefatos, teste de software, treinamento adequado e certificação da equipe envolvida no projeto, ações corretivas e preventivas, controle de documentação, geração e análise de métricas de qualidade, dentre outros [11].

Falar sobre testes ---introdução

**1.2 Tipos de testes**

Para Coward (1997) os testes se classificam em testes de caixa preta e teste de caixa branca. O primeiro que também é conhecido como teste funcional tem com objetivo avaliar o comportamento externo do software sem se preocupar com seu cerne ou codificação. Aqui o sistema recebe os dados “crus”, esses recebem um processamento e a sua saída tem que atender ao negócio envolvido.

Já o teste de caixa branca testa o software de maneira mais crítica, trata de sua estrutura e concisão de código. Para Pressman (2006) o engenheiro de testes pode derivar os casos de teste que garantam que todos os caminhos independentes dentro de um módulo tenham sido exercitados pelo menos uma vez.

**1.2 Níveis de testes (Talvez mude para depois de processo de testes)**

No entendimento de Peters et al. (2001) os testes tem níveis durante o processo de desenvolvimento de software, que são dispostos de maneira a alcançar objetivos distintos para cada fase.

O nível de unidade contempla a implementação do software e seu objetivo e assegurar que dentre outras coisas que a lógica esteja completa e que os componentes ou módulos trabalhe conforme foi projetado.

O nível de integração que a junção dos módulos escritos trabalhem em harmonia e suas dependências sejam respeitadas assim garantindo o produtos completo.

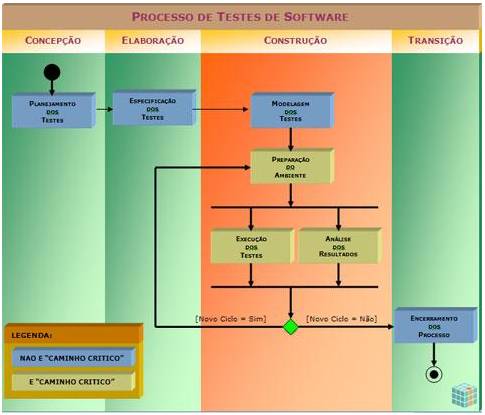
Já no sistêmico garante o software na sua totalidade como entidade, aqui é verificado se o que está sendo entregue atende os requisitos operacionais.

Um teste bastante interessante e que as empresas estão aderindo é o teste de aceitação que é feito com um grupo restrito de usuários chaves do sistema. Esses simulam operações de rotina da ferramenta para verificar se seu comportamento está de acordo com o que foi contratado [3], ou seja baseia-se no método caixa-preta que visa demonstrar conformidade ou não com os requisitos sem saber como é feito internamente.

**1.3 Processo de testes**

A etapa práticas de testes se dar em duas fase: A primeira que é projetar a escrita dos testes, elaborando os casos de teste, nessa é idealizado quais pontos são críticos para serem testados, normalmente esses cenários são imaginados em cima de regra de negócio e parte críticas da função da aplicação. Na segunda são executados esses passos planejados nos casos de testes, aqui a atividade se torna bem repetitiva, pois sua execução será seguida por esses casos e caso seja encontrado alguma não conformidade terá que repetir todos os passos novamente no intuito de encontrar efeitos colaterais dessa correção, tornando esse trabalho enfadonho e desgastante para o tester.

-- Aqui cabe falar mais sobre o quanto é exaustivo testar manual



Fonte: imasters

Testes de Aceitação têm por objetivo validar o produto, ou seja, verificar se este atende aos requisitos especificados. Eles são executados em ambiente o mais semelhante possível ao ambiente real de execução. Os testes de aceitação podem ser divididos em testes funcionais e não funcionais; Testes de Aceitação têm por objetivo validar o produto. conforme proposto pela IEEE segundo Poston (1996).

2.6.2 TESTES DE ACEITAÇÃO

Os testes de aceitação são testes de alto nível que devem ser realizados desde a fase inicial dos testes de software. Eles são divididos em dois itens principais, os testes funcionais e os testes não funcionais.

2.6.2.1TESTES FUNCIONAIS

Para Ramirez (1999), os testes funcionais são desenhados para verificar a consistência entre o produto implementado e os respectivos requisitos funcionais. A completeza e a precisão da especificação dos requisitos do software são fundamentais para assegurar a qualidade desses testes. Os testes funcionais têm como suas principais características a análise do valor limite, testes de comparação e testes de tempo real.

A análise do valor limite é uma técnica para a seleção de casos de teste que exercitam os limites. O emprego dessa técnica deve ser complementar ao emprego da partição de equivalência. Assim, em vez de se selecionar um elemento aleatório de cada classe de equivalência, selecionam-se os casos de teste nas extremidades de cada classe.

O foco dessa pesquisa é o teste de aceitação, esse tem função de verificar se o sistema está em conformidade com a necessidade do usuário e constatar se o mesmo está de acordo com o que foi elicitados nos requisitos. Normalmente esses testes são realizados por um grupo restrito de usuários finais e de preferência em um ambiente parecido com a produção que o sistema irá rodar. (MYERS, 2004).

**1 AQUI FALAREI SOBRE DIFICULDADES DESSES TESTES MANUAIS**

**2 MENCIONAREI SOBRE AUTOMATIZAÇÃO DE TESTES DE ACEITAÇÃO**

**3 FALAREI SOBRE A EVOLUÇÃO: SELENIUM FIREFOX (PLUGIN), SELENIUM 2, WEBDRIVER**

**4 FAREI UMA BREVE COMPARAÇÃO ENTRE ALGUMAS FERRAMENTAS DE TESTES DE ACEITAÇÃO**

**5 IREI FALAR DO CASE (EMPRESA QUE FAZ O TESTE MANUAL)**

**6 IMPLATAÇÃO E TREINAMENTO DO WEBDRIVE**

Esse teste permite aos sistemas entrar em funcionamento, com a segurança de que os processos críticos funcionam corretamente.  
Assegura que as aplicações são funcionais e estáveis e se integram corretamente nos processos de negócios cruciais. Assegura que todas as transações são corretamente integradas no processo de negócio.  
Nos dias de hoje, já não é aceitável esperar que seja a empresa ou os seus utilizadores a desempenharem o papel de profissionais de Teste de Aceitação na fase de entrada em funcionamento.  
O risco para os processos de negócio é igualmente importante. Os problemas que eventualmente surjam em sistemas críticos podem resultar em paragens dispendiosas e prolongadas, em novo desenvolvimento e teste. O Teste de Aceitação corretamente implementado irá atenuar estes riscos e reduzir os custos de novo desenvolvimento e colocação no mercado.  
Embora o Teste de Aceitação (UAT - User Acceptance Testing) constitua uma das etapas finais do Ciclo de Vida do Produto, o seu envolvimento ao longo de todo o processo de desenvolvimento é essencial. Os processos de negócio, bem como os requisitos, devem ser considerados fundamentais neste processo. O UAT oferece aos utilizadores finais do sistema a oportunidade de o testarem eficazmente, antes da sua entrada em funcionamento.  
Enquanto os programadores executam testes em relação à especificação do sistema e à documentação técnica e verificam se estas se adequam aos objetivos, os técnicos de testes de Aceitação testam em relação aos requisitos do negócio, verificando, também, a sua adequação aos objetivos de negócio

**1.2 Objetivo Geral**

Apresentar a automatização de teste de aceitação utilizando o webdriver criando suítes de testes em conjunto com demandas em fábrica de softwares.

**1.3 Objetivos Específicos**

* Entender problemas de entrega de produtos sem testes consistentes.
* Compreender a importância de testes de aceitação em projetos de softwares.
* Automatização de testes de aceitação utilizando o webdriver
* Comparações de bugs em entregas antes e depois de aplicado o teste de aceitação.
* Repassar para o cliente a importância e vantagens de ter esse tipo de teste no projeto.

**Organização do Trabalho**

**É importante falar dos testes de aceitação sim, tem que dar uma contextualizada boa dos testes para tentar mostrar a importância deles, depois tenta mostrar o trabalho que é fazê-los manualmente, aí pega o link para os automatizados, chegando no Selenium. O Selenium também pode ser utilizado para outros tipos de testes como o de integração e regressão.**

**##################################################**

**2 Definição**

**##################################################**

**3 Tecnologias aplicadas**

**##################################################**

**3 Conclusão**

**##################################################**

BIBLIOGRAFIA - FALTA FORMATAR

**[1]** **BSTQB – Brazilian Software Testing Qualifications Board**. <http://www.bstqb.org.br/uploads/docs/syllabus_2007br.pdf>.

**[ 2]** **Base de Conhecimento em Teste de Software.** 2ª Edição. Rios, Emerson; Cristalli, Ricardo; Moreira, Trayahú & Bastos, Aderson. – S. Paulo, Martins Fontes, 2007.

[3] Pressman , 1995

[11] GALIN D. Software Quality Assurance – From Theory to Implementation. Pearson, -, 2004.

Alexandre Bartie. “Fábrica de Testes Orientada a Resultados” http://imasters.uol.com.br/artigo/4665/des\_de\_software/fabrica\_de\_testes\_e\_orientada\_a\_resultados/ Publicado em 05/09/2006. figura do

MYERS, Glenford J., John Wiley & Sons, *The Art of Software Testing*, 2, Nova Jérsei: 2004. ISBN [ISBN 0-471-46912-2](http://pt.wikipedia.org/wiki/Especial:Fontes_de_livros/0471469122)

BEIZER, B. Software Testing Techniques. Thomson Computer Press, 2nd edition, 1990.

2 - Com a preocupação de reduzir os custos da atividade de testes, a automação deste processo tem um importante papel (Leow et al. 2004).

LEOW, W.; KHOO, S.; SUN, Y. Automated Generation of Test Programs From

Closed Specifications of Classes and Test Cases. In proceedings of 26th International Conference on Software Engineering. 2004.

3 - Tipos de teste - HARROLD, M. Reduce, Reuse, Recycle, Recover: Techniques for Improved Regression Testing. IEEE International Conference on Software Maintenance. 2009.

4 - 11] Alexandre Bartie. “Fábrica de Testes Orientada a Resultados” http://imasters.uol.com.br/artigo/4665/des\_de\_software/fabrica\_de\_testes\_e\_orientada\_a\_resultados/ Publicado em 05/09/2006. figura do

5 - CAETANO, Cristiano. Base de Conhecimento em Teste de Software: Módulo 2 -

Planejando os Testes, 2010. Testes de aceitação

6 - BASTOS, Aderson; et al. Base de conhecimento em teste de software. São Paulo:

Martins, 2007 - Testes de aceitação

POSTON, R.M. Automating specification-based software testing. IEEE computer

society. Chicago: Books, 1996.

RAMIREZ, Jaime Arturo. Teste de software, Belo Horizonte, dez. [1990]. Disponível em: <http://ead1.eee.ufmg.br/~renato/engsoft/Teste\_Soft.pdf>. Acesso em: 13 fev. 2001.