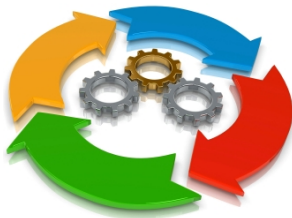


UNIVERSIDADE FEDERAL DE SANTA MARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA - PPGI

Automação de Testes em Aplicações de BPMS: um Relato de Experiência

Jéssica Lasch de Moura e Andrea Schwertner Charão

- BPM e BPMS;
- Verificação e testes = desafio;
 - Falta de automação nos testes pode levar a problemas.



■ Ciclo de vida BPM

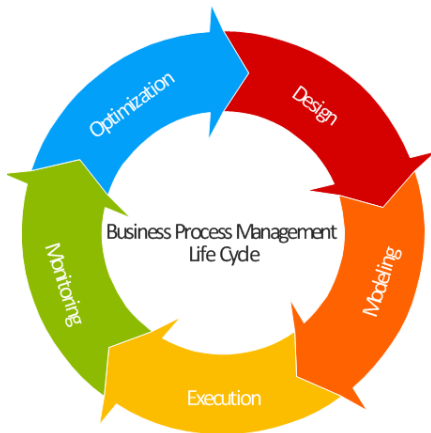


Figura: Ciclo de vida BPM. Fonte: <https://conceptdraw.com>

Aplicação alvo dos testes

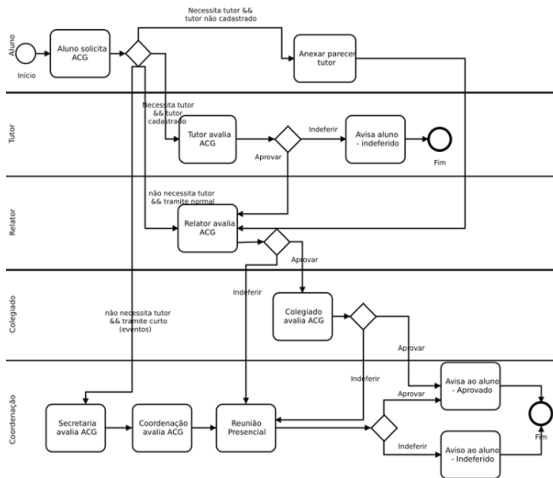


Figura: Processo alvo dos testes: Solicitação de Atividade Complementar de Graduação

Aplicação alvo dos testes

- A aplicação foi submetida a testes funcionais realizados manualmente;
- BPMS utilizado não possuía suporte a nenhum tipo de teste automatizado;
- Buscou-se outros BPMS com licenças open source ou freeware, que pudessem implementar o processo em questão e que oferecessem suporte a testes.



Figura: BPMS analisados

- 1 Testes de carga: um tipo de teste de desempenho, visando avaliar o comportamento do sistema frente a um grande número de solicitações;
- 2 Testes funcionais: verificar as saídas do sistema produzidas a partir de entradas pré-definidas.



Testes de carga

- Realizados gerando-se múltiplas requisições HTTP ao servidor, de forma controlada;
- Exemplos de ferramentas: JMeter, The Grinder e Gatling

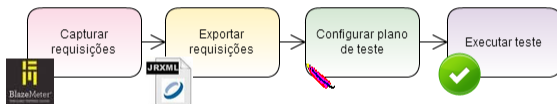


Figura: Etapas para teste de carga utilizando o JMeter

Testes de carga

- As requisições HTTP carregam chaves identificando usuários e processos;
 - Bonita = chave identificadora de sessão gerada no momento em que o usuário acessa o sistema e outra chave identificadora de instância.

Extractor de Expressão Regular	
Nome:	Extractor de Expressão Regular
Comentários:	
Quais amostras testar	
<input checked="" type="radio"/> Amostras principais e sub-amostras <input type="radio"/> Somente Amostras principais <input type="radio"/> Somente Sub-amostras <input type="radio"/> JMeter Variable <input type="text"/>	
Campo da Resposta a ser verificado	
<input type="radio"/> Corpo (body) <input type="radio"/> Corpo (body) - não escapado <input type="radio"/> Body as a Document <input checked="" type="radio"/> Cabeçalhos (headers) <input type="radio"/> URL <input type="radio"/> Código de Resposta	
Nome de Referência:	AUTHTOKEN
Expressão Regular	identityKey=(.+)
Modelo:	\$1\$
Número para Combinação (0 para aleatório)	
Valor Padrão:	Não encontrou

Figura: Obtendo chave identificadora utilizando o Extrator de Expressão Regular

Resultados dos Testes de Carga

Usuários	Login	Pág. Inicial	Seleção Processo	Form. Inicial	Enviar form.
1	126	32	38	80	73
50	597	191	179	368	152
100	1972	571	552	760	694
200	10.149	3.239	934	2.122	1.918

Tabela: Tempos médios de resposta, em milissegundos

Testes funcionais

- Para executar testes funcionais em aplicações Web, pode-se utilizar ferramentas livres como Selenium, Watir ou Geb;
- Para este trabalho, escolheu-se a ferramenta Selenium, aliada ao Cucumber-JVM para descrição dos testes.



Figura: Etapas para teste funcional utilizando o Selenium e Cucumber-jvm

- Para a execução dos testes com sucesso, foram necessarias algumas alterações nos codigos gerados.

Código gerado pelo Selenium IDE

```
@Test
public void testESeleniumActiviti() throws Exception {
    driver.get(baseUrl + "/activiti-explorer/ui/");
    driver.findElement(By.cssSelector("div.login-button")).click();
}
```

Código adaptado

```
public void login(){
    driver.get(baseUrl + "ui/");
    driver.switchTo().frame(driver.findElement(By.name("PID6")));
    driver.findElement(By.name("username")).clear();
    driver.findElement(By.name("username")).sendKeys("kermix");
    driver.findElement(By.name("password")).clear();
    driver.findElement(By.name("password")).sendKeys("kermix");
    driver.findElement(By.cssSelector("div.login-button")).click();
}
```

Figura: Exemplo de alteração no código gerado pelo Selenium

Resultados dos testes funcionais

	Bonita	Activiti
Componentes Web	HTML, CSS, Ajax	HTML, CSS, Ajax
Captura da interação do usuário utilizando o Selenium	Total	Parcial (necessitou de inserção manual de alguns campos)
Foi possível exportar o código gerado pelo Selenium?	Sim	Sim
Reconhecimento de todos os campos capturados SEM alteração de código	Parcial	Parcial
Reconhecimento de todos os campos capturados COM alteração de código	Total	Total
Foi possível criar o cenário e executar o teste?	Sim	Sim

Tabela: Resumo comparativo sobre o teste funcional

Considerações finais

- Teste de carga: útil para explicar falhas observadas no trabalho anterior;
 - Pode ser trabalhoso ou até inviável;
- Teste funcional: maior sucesso na execução dos testes;
 - Menor dependência dos BPMS (em comparação com o teste de carga);
 - Também pode ser trabalhoso dependendo do processo;
- Sob certas condições, é viável testar aplicações de BPMS com ferramentas de teste para sistemas Web;
- Pôde-se notar que o suporte a testes automatizados é pouco explorado em BPMS;
 - Seria uma funcionalidade bem-vinda.