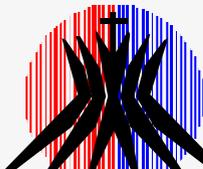


Compreensão de Programas com Localização de Casos de Uso

Marcelo Mendes Marinho
mmarinho13@brturbo.com

Nicolas Anquetil
anquetil@ucb.br



Agenda

- Compreensão de Programas
 - Localização de Implementação
 - Proposta
 - Definição da solução
 - Conclusão
-

Agenda

- ☐ Compreensão de Programas
 - ☐ Localização de Implementação
 - ☐ Proposta
 - ☐ Definição da solução
 - ☐ Conclusão
-

Introdução

- Compreensão de Programas
 - 40% a 90% do tempo gasto na modificação de um software

“programas que não podem ser compreendidos, não podem ser mudados”

(Rajlich; Wilde, 2002)

Introdução - Cenário

- Usuário: “Agora, o cálculo da taxa de serviço deve aparecer na nota fiscal”
 - Programador:
 - Qual parte do código deve ser alterada?
 - Onde começar a procurar?
 - Quanto tempo vai levar?
 - Que testes devem ser realizados no fim?
-

Compreensão de Programas

- Adquirir conhecimento suficiente sobre um artefato de software para executar uma tarefa (CANFORA; CIMITILE, 1998)
 - Explicar:
 - O comportamento
 - A estrutura
 - O mapeamento do domínio da aplicação com o código-fonte do programa
-

Agenda

- Compreensão de Programas
 - Localização de Implementação
 - Proposta
 - Definição da solução
 - Conclusão
-

Localização de Implementação

□ Taxa de serviço

■ Regras de aplicação

■ Calculo

□ Nota fiscal

■ Data emissão

■ Valor

■ Descritivo

■ Cliente

GET-YES-NO-ANSWER.

IF ANSWER IS EQUAL TO 'y', MOVE 'Y' TO ANSWER.

IF ANSWER IS EQUAL TO 'n', MOVE 'N' TO ANSWER.

IF ANSWER IS NOT EQUAL TO 'Y'

IF ANSWER IS NOT EQUAL TO 'N'

DISPLAY 'You must answer either Y or N: '

WITH NO ADVANCING;

ACCEPT ANSWER .

DELETE-ORDER.

CALL 'ACCDL' USING BY DESCRIPTOR ERROR-
CODE, 'IMMEDIATE', BY DESCRIPTOR SALES-
DS.

MOVE SPACES TO STORE-CODE, PAY-TERMS

MOVE ZEROS TO QTY, ORDER-DATE.

950-OUT.

EXIT.

Domínio de aplicação
(dimensão humana)

Código fonte
(dimensão "matemática")

Abordagens de Localização

□ Análise Estática

- Comando “grep” (busca de palavras chaves)
- LSI (análise do texto dos comentários e identificadores)

□ Análise Dinâmica (Software *Reconnaissance* (WILDE, 1994))

- Rastros de execução
 - Casos de teste que ativam ou não uma funcionalidade
-

Feature location

- Localização de funcionalidade
 - Funcionalidades não têm formato definido
 - “Caso de teste”, “cenário”, “*feature*”
 - Métodos não relacionados com o dia-a-dia da manutenção/desenvolvimento
 - Funcionalidades de baixo nível
 - Ex: Abrir arquivo, Copiar/Colar, Imprimir
-

Agenda

- Compreensão de Programas
 - Localização de Implementação
 - Proposta
 - Definição da solução
 - Conclusão
-

Proposta

- Localização de implementação baseada em casos de uso
 - Associar cada passo do caso de uso ao código que o implementa
 - Aproveitar a estrutura dos casos de uso
 - Aproveitar a popularidade dos casos de uso
-

Casos de Uso

- Formato para a especificação de funcionalidades
 - Seqüência de ações (entrada/saida)
 - Especificam fluxos (agrupa várias seqüências relacionadas)
 - casos de teste
 - Resultado de valor observável (significado no negócio)
-

Casos de Uso – Exemplo

Caso de Uso: Realizar Saque em Caixa Eletrônico

Fluxo Básico

- a1: O caso de uso se inicia quando o correntista passa o cartão no leitor do caixa eletrônico
 - a2: O sistema lê e armazena as informações do cartão (tipo do cartão, agência e número da conta corrente)
 - a3: O sistema apresenta para o correntista a tela com as opções de operação disponíveis [...]
-

Casos de Uso – Exemplo (cont.)

Fluxo Alternativo: Senha incorreta

A1_1: Se no passo a8 a senha informada está incorreta o sistema apresenta uma tela com uma mensagem de erro ("Senha incorreta") com opções para cancelar a operação ou introduzir nova senha

A1_2: Se o usuário deseja entrar nova senha o sistema volta para o passo a7, caso contrário encerra a operação e volta para a tela inicial

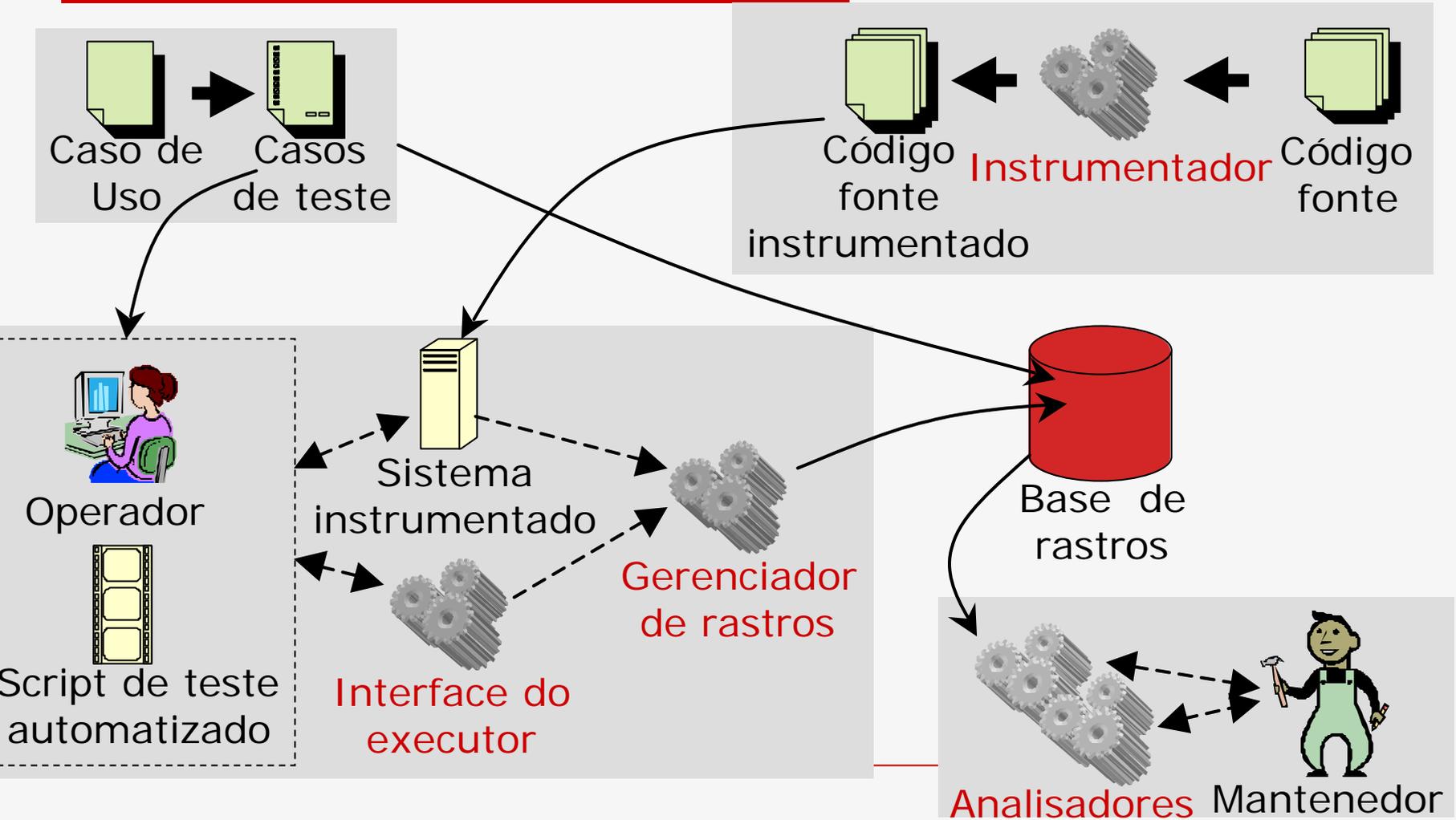
Agenda

- Compreensão de Programas
 - Localização de Implementação
 - Proposta
 - Definição da solução
 - Conclusão
-

Método

- Instrumentar o código fonte e re-compilar o sistema
 - Gerar casos de teste
 - Executar casos de teste (geração de rastros)
 - Análises dos rastros marcados
-

Arquitetura



Instrumentador

- Implementado em Java para código fonte Java
 - Rastreia chamadas de métodos
 - Uso da pilha de execução para determinar que método chamou que método
-

Interface do executor

- ❑ Leia um caso de teste no banco
 - ❑ Apresenta cada passo do caso de teste ao usuário (em ordem)
 - ❑ Avisar o Gerenciador de rastros quando o usuário informa a execução de um passo
 - ❑ Pode ser usado por ferramentas de teste (interface programada)
-

Gerenciador de rastros

- Recebe mensagens
 - Sistema instrumentado
 - Interface do executor
 - “Log” de cada evento dentro do banco de rastros
 - Rastros marcados pelos passos do caso de teste
 - Implementado em Java, usa RMI
-

Analísadores

- Leiam o banco de rastros
 - Descrição dos casos de uso
 - Descrição dos casos de teste → associados aos casos de uso (fluxos de caso de uso)
 - Rastros marcados → associados aos casos de teste, com indicação dos passos
-

Analísadores

- “Execução post-mortem”
 - Passo a passo do caso de teste
 - Todas as chamadas de método

Mozilla Firefox

file:///C:/temp/public_html/index.html

file:///C:/temp...html/index.html | file:///C:/temp.../SimMatrix.html | file:///C:/te...ml/index.html

3 +[com.sun.j2ee.blueprints.signon.web.ProtectedResource].getURLPattern

6. Cliente escolhe as opcoes de transporte e submete para o sistema.

2 +[com.sun.j2ee.blueprints.waf.controller.web.MainServlet].doPost

2 +[com.sun.j2ee.blueprints.waf.controller.web.MainServlet].doProcess

2 +[com.sun.j2ee.blueprints.waf.controller.web.MainServlet].getURLMapping

2 +[com.sun.j2ee.blueprints.waf.controller.web.URLMapping].isTransactional

2 +[com.sun.j2ee.blueprints.waf.controller.web.RequestProcessor].processReq

2 +[com.sun.j2ee.blueprints.waf.controller.web.html.HTMLActionSupport].perform

2 +[com.sun.j2ee.blueprints.consumerwebsite.actions.CartHTMLAction].perform

2 +[com.sun.j2ee.blueprints.consumerwebsite.actions.CartHTMLAction].upda

2 +[com.sun.j2ee.blueprints.consumerwebsite.AdventureComponentManag-

2 +[com.sun.j2ee.blueprints.consumerwebsite.Cart].setAdventureDays

```
import com.sun.j2ee.blueprints.catalog.  
  
/**  
 * This class manages components used i  
 *  
 */  
public class AdventureComponentManager  
    implements HttpSessionLis  
  
    *  
    * specialize that needs stia  
    *  
    */  
public void sessionCreated(HttpSess  
    se.getSession().setAttribute(Ad
```

Analísadores

□ Filtro simples

- Lista dos métodos chamados
- Cada método apenas uma vez por passo

Mozilla Firefox

file:///C:/temp/public_html/index.html

so de Uso: ABUC01 - Olhar Catalogo

Fluxo Basico

1. Esse caso de uso se inicia quando o Cliente acessa a tela principal do sistema pelo Fluxo Basico
 - > [\[com.sun.j2ee.blueprints.signon.web.SignOnFilter\].doFilter](#)
 - > [\[com.sun.j2ee.blueprints.consumerwebsite.AdventureComponentManager\].set](#)
 - > [\[com.sun.j2ee.blueprints.consumerwebsite.SignOnNotifier\].attributeAdded](#)
 - > [\[com.sun.j2ee.blueprints.consumerwebsite.SignOnNotifier\].processEvent](#)
 - > [\[com.sun.j2ee.blueprints.consumerwebsite.AdventureComponentManager\].get](#)
 - > [\[com.sun.j2ee.blueprints.consumerwebsite.AdventureComponentManager\].get](#)
 - > [\[com.sun.j2ee.blueprints.consumerwebsite.AdventureComponentManager\].get](#)
 - > [\[com.sun.j2ee.blueprints.consumerwebsite.AdventureComponentManager\].get](#)
 - > [\[com.sun.j2ee.blueprints.signon.web.ProtectedResource\].getProtectedResource](#)
2. Cliente seleciona uma categoria de aventura.

```
/* Copyright 2004 Sun Microsystems, Inc.  
http://adventurebuilder.dev.java.net/LICENSE.html  
$Id: SignOnNotifier.java,v 1.3 2004/05/20 16:08:00 jduffy Exp $
```

```
package com.sun.j2ee.blueprints.consumerwebsite;
```

```
import java.beans.Beans;  
import java.util.*;  
import javax.servlet.http.*;  
import javax.servlet.*;
```

```
private void processEvent(HttpSession session,  
HttpSession session = session.get
```

Analísadores

- Similaridade entre casos de uso
 - Similaridade entre Ações de “Escolher Pacote” e “Rastrear Pedido”

Escolher Pacote	Rastrear Pedido	Simil.
Fluxo 1 – Passo 1	Fluxo 1 – Passo 1	73.9%
Fluxo 1 – Passo 2	Fluxo 1 – Passo 1	85.0%
Fluxo 1 – Passo 3	Fluxo 1 – Passo 1	89.5%
Fluxo 5 – Passo 1	Fluxo 1 – Passo 1	94.4%
Fluxo 6 - Passo 1	Fluxo 1 – Passo 1	89.5%

Agenda

- Compreensão de Programas
 - Localização de Implementação
 - Proposta
 - Definição da solução
 - Conclusão
-

Conclusão

- Uma infra-estrutura para rastreamento de casos de uso
 - Algumas análises implementadas
 - Trabalhos futuros
 - Extensão para outras linguagens de programação
 - Outras análises
 - “Solução” pronta (integração com ferramentas existentes: IDE, teste)
-

Novas Análises

- Visualização em vários níveis:
 - Rastros de casos de uso
 - Rastros de fluxos (casos de teste)
 - Rastros de passos
-

Analísadores

- Uso de Unidades Computacionais
 - Aparece em vários casos de uso?
 - É única de um caso de uso?
 - De uma ação?
-

Analísadores

- Comparação entre Casos de Uso e entre Ações
 - A implementação dos Casos de Uso tem partes em comum?
 - Corresponde a descrições semelhantes?
-