

## **AVALIAÇÃO DO TESTE FUNCIONAL SISTEMÁTICO EM PROGRAMAS PROCEDIMENTAIS.**

**SOUSA**, Jorge Augusto<sup>1</sup>; **VINCENZI**, Auri Marcelo Rizzo<sup>2</sup>.

Palavras-chave: Teste Funcional e Estrutural, Teste Funcional Sistemático.

### **1. INTRODUÇÃO**

O processo de desenvolvimento de software envolve uma série de atividades e mesmo com o uso de técnicas, métodos e ferramentas o produto desenvolvido ainda pode conter erros. Assim sendo, atividades agregadas sob o nome de Garantia de Qualidade de Software têm sido introduzidas ao longo de todo o processo de desenvolvimento, entre elas atividades de Verificação, Validação e Teste (VV&T), visando reduzir a ocorrência de erros e os riscos associados. A ênfase deste trabalho se dá aos critérios de testes da técnica funcional e estrutural. Critérios da técnica funcional (ou caixa-preta) têm por objetivo determinar se o programa satisfaz os requisitos funcionais e não-funcionais que foram especificados. O problema é que, em geral, a especificação existente é informal e, desse modo, a determinação da cobertura total da especificação que foi obtida por um dado conjunto de casos de teste também é informal. Entretanto, os critérios funcionais são independentes da linguagem de programação empregada, e podem ser usados em qualquer fase de teste sem a necessidade de modificação. Exemplos desses critérios são: particionamento em classe de equivalência, análise do valor limite, grafo de causa-efeito e teste baseado em estado.

### **2. METODOLOGIA**

A metodologia adotada para atingir os objetivos desejados consiste na realização de estudos empíricos, desenvolvendo conjuntos de testes funcionais, ou utilizando conjuntos já existentes melhorando-os, seguindo as diretrizes impostas pelo TFS e, posteriormente, avaliando a cobertura de tais conjuntos em relação aos critérios de testes estruturais implementados pela ferramenta *Poke-Tool*[1]. Com base nos resultados obtidos novas diretrizes podem ser integradas ao TFS de modo a melhorar a cobertura dos conjuntos funcionais gerados em relação aos critérios estruturais.

### **3. RESULTADOS E DISCUSSÃO**

Para a análise e coleta dos dados foram utilizados os programas da Maratona[2] de Programação de 2004 e 2005 da ACM, já que possuem um conjunto de casos de teste funcionais desenvolvidos para validar a implementação dos problemas. Porém nem todos os programas que foram implementados durante a maratona podem ser utilizados, já que os times inscritos durante a competição precisam resolver os problemas propostos pela organização em um tempo limitado e sobre forte pressão. Os programas foram então submetidos em um sistema de testes onde é feita uma comparação entre os resultados obtidos pelos competidores e os resultados esperados dos problemas propostos. Assim, os programas da

Maratona são qualificados em aprovados e não aprovados. No ano de 2004, dos 297 programas submetidos para o sistema de avaliação, 77 foram considerados aprovados e 220 não-aprovados. Já em 2005, dos 374 programas, 115 foram aprovados.

Uma outra característica desta competição é que os programadores podem implementar seus códigos em diferentes linguagens. Para os dois anos de maratona analisado nesse projeto, os participantes optaram por C, C++ e Java. Nos anos de 2004 e 2005, foram implementados em C, respectivamente, 42 e 63 programas.

Entretanto, mesmo considerando o total de programas C, um número menor de programas é que pode efetivamente ser utilizado nos experimentos. Existem dois motivos que ocasionaram essa redução. Primeiro alguns programas, apesar de possuírem uma identificação única, possuíam o código fonte idêntico, reduzindo o número de programas a serem analisados. Isso ocorre quando uma determinada equipe submete duas vezes o mesmo código fonte para o sistema de avaliação. Segundo, para outros programas o parser da ferramenta *Poke-Tool* [1] não foi capaz de instrumentar o código e assim não se consegue obter a cobertura dos casos de teste. Isso é explicado pelo fato da ferramenta operar com o código ANSI C e alguns dos programas desenvolvidos na Maratona de Programação não foram escritos de acordo com esse padrão. Do total foram utilizados 25 programas do ano de 2004 e 39 de 2005.

Logo abaixo são mostrados o total de requisitos, o número de cobertos, o número de não-executáveis, os executáveis e a porcentagem de cobertura obtida para todos os critérios dos diversos problemas utilizados, considerando a média das soluções para cada problema.

**Critério: Todos nós**

Programa	#Total de req	#Cobertos	#Nexecut	#Execut	%Cobertura
A-2005	143	143	0	0	100
C-2005	169	168	1	0	100
D-2004	49	49	0	0	100
D-2005	219	202	0	17	92,24
F-2004	77	76	1	0	100
F-2005	157	139	0	18	88,54
G-2004	258	258	0	0	100
G-2005	243	243	0	0	100
H-2005	104	104	0	0	100
<b>Totais</b>	<b>1419</b>	<b>1382</b>	<b>2</b>	<b>35</b>	<b>97,86</b>

**Critério: Todos arcos**

Programa	#Total de req	#Cobertos	#Nexecut	#Execut	%Cobertura
A-2005	67	65	1	1	98,48
C-2005	79	72	4	3	96
D-2004	29	29	0	0	100
D-2005	105	94	2	9	91,26
F-2004	3	3	0	0	100
F-2005	69	66	3	0	100
G-2004	147	142	5	0	100
G-2005	126	118	7	1	99,16
H-2005	50	45	4	1	97,83
<b>Totais</b>	<b>675</b>	<b>634</b>	<b>26</b>	<b>15</b>	<b>98,08</b>

**Critério: Todos usos**

Programa	#Total de req	#Cobertos	#Nexecut	#Execut	%Cobertura
A-2005	519	464	45	10	97,89
C-2005	459	401	58	0	100
D-2004	282	278	4	0	100
D-2005	820	731	5	84	89,69
F-2004	207	193	12	2	98,97
F-2005	571	492	21	58	89,45
G-2004	147	142	5	0	100
G-2005	796	719	73	4	99,45
H-2005	301	277	23	1	99,64
<b>Totais</b>	<b>4102</b>	<b>3697</b>	<b>246</b>	<b>159</b>	<b>97,23</b>

#### 4. CONCLUSÃO

Para os programas em C, apenas para 2 dos 9 problemas investigados, (D-2004 e G-2004), os casos de teste funcionais foram capazes de cobrir 100% dos requisitos dos critérios estruturais, conforme pode ser observado nas Tabelas 15, 16 e 17. Considerando apenas os critérios Todos-Nós, representados na Tabela 15, para 7 dos 9 problemas, a cobertura máxima foi obtida, indicando que os testes funcionais foram efetivos em executar ao menos uma vez cada comando das implementações analisadas. Apenas para os problemas D-2005 e F-2005, a cobertura foi abaixo dos 100%.

Observa-se que esse número de problemas com cobertura de 100% decai à medida que um critério de teste estrutural mais rigoroso é empregado. No caso do critério Todos-Arcos, 4 em 9 dos problemas obtiveram a cobertura máxima e para o critério Todos-Usos esse número cai para 3. Isso indica que embora os Testes Funcionais Sistemáticos, em geral, proporcionaram uma boa cobertura dos critérios estruturais, melhorias ainda precisam ser feitas de modo que mais casos de teste possam ser gerados durante o teste funcional para que se consiga cobrir 100% dos requisitos critérios estruturais analisados.

Entretanto, devido ao tempo exigido na análise dos dados, a tarefa de sugerir melhorias no Teste Funcional Sistemático não pode ser realizada, sendo adiada para um possível trabalho futuro.

#### 5. REFERÊNCIAS BIBLIOGRÁFICAS

[1] Chaim, M. L., *POKE-TOOL – Uma Ferramenta para Suporte ao Teste Estrutural de Programas Baseado em Análise de Fluxo de Dados*. Tese de Mestrado, DCA/FEEC/UNICAMP, Campinas, SP, abr. 1991.

[2] DSI/CGSA, Qualidade no setor de software brasileiro: 2001. Revista do SEPIN/MCT, 2001, Disponível em: [http://www.mct.gov.br/Temas/in\\_fo/Ds\\_i/Quali2001/Public2001.htm](http://www.mct.gov.br/Temas/in_fo/Ds_i/Quali2001/Public2001.htm). Acesso em: 05/04/2005.

1. Bolsista de iniciação científica. Instituto de Informática/UFG - [jorge@inf.ufg.br](mailto:jorge@inf.ufg.br)

2. Orientador/Instituto de Informática/UFG - [auri@inf.ufg.br](mailto:auri@inf.ufg.br)