PEREIRA, F.L. Vincenzi, A. M. R. Avaliação do Teste Funcional Sistemático *versus* Teste Estrutural no Contexto de Programas Orientado a Objetos. In: CONGRESSO DE PESQUISA, ENSINO E EXTENSÃO DA UFG - CONPEEX, 2., 2006, Goiânia.

Avaliação do Teste Funcional Sistemático *versus* Teste Estrutural no Contexto de Programas Orientado a Objetos.

PEREIRA, Flávio Lúcio[1]; **VINCENZI**, Auri Marcelo Rizzo [2].

Palavras-chave: Engenharia de Software, Qualidade de Software, Teste de Software.

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 JaBUTi[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. Abaixo é apresentada uma tabela com estas informações para os programas da Maratona de 2004 e 2005.

Programas da Maratona da ACM		
	Maratona de 2004	**Maratona de 2005**
Aprovados	77	115
Não Aprovados	220	259
Total	297	374

Tabela 1: Programas da Maratona da ACM

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.

Na Tabela 2 segue os totais de códigos implementados nas linguagens anteriormente citadas e que foram qualificados como aprovados.

Programas Aprovados		
	Maratona de 2004	**Maratona de 2005**
Total em C	42	63
Total em Java	6	4
Total C++	29	48
Total programas	77	115

Tabela 2: Programas Aprovados

Na competição, os problemas propostos pela ACM para serem resolvidos pelos participantes, estão divididos e nos enunciados dos problemas constam exemplos dos dados. As Tabelas 3 e 4 contém informações a respeito dos problemas dos programas aprovados da Maratona de Programação de 2004 e de 2005 respectivamente. É identificada a letra o nome de cada problema, além de uma breve descrição.

Problema	**Nome**	**Descrição**
A	tax	Aproveitamento máximo de área em torno do lago
B	petanque	Jogo de arremesso com bolas
C	longnight	Melhor caminho para visita dos museus de Viena
D	magic	Mágicas com cartas de baralho
E	twostacks	Jogo de cartas-paciência
F	oyester	Evitar inundações
G	grandpa	Buscar o segundo lugar no ranking na IBA

Tabela 3: Programas de 2004

Problema	**Nome**	**Descrição**
A	tornado	Determinar menor número de madeira para fechar as fendas
C	pascal	Organização de uma biblioteca
D	icpc	Determinar salários dos empregados da empresa
E	fiber	Instalação de cabos ópticos em cidades
F	genes	Mapeamento genético
G	computer dj	Escolha das músicas de uma festa
H	pnets	Problema de otimização

Tabela 4: Programas de 2005

No contexto deste trabalho apenas os programas desenvolvidos em linguagem de programação JAVA (linguagem suportada pela ferramenta JaBUTi[1]) foram aproveitados. Uma grande dificuldade encontrada durante a realização do experimento foi a pequena quantidade de programas aprovados escritos em linguagem JAVA, embora não tenha tido influência relevante no que tange às conclusões finais da experimentação. Na Tabela 6 é mostrado uma tabela contendo a cobertura média dos requisitos de testes - nós, arcos, usos e Potenciais-usos - e a quantidade de programas avaliados utilizando a ferramenta de teste JaBUTi[1].

Programas	***Qdte***	***Nós***	***Arcos***	***Usos***	***Pusos***
Magic	3	96	95	89	92,6
Grandpa	3	96,6	96,6	91,4	84,2

Tabela 6: Tabela de cobertura média dos programas de 2004 na linguagem *Java*

Programas	***Qdte***	***Nós***	***Arcos***	***Usos***	***Pusos***
Tornado	2	100	100	100	100
Pascal	2	100	100	100	100

Tabela 7: Tabela de cobertura média dos programas de 2005 na linguagem Java

4.CONCLUSÃO

Analisando-se os programas do experimento mais detalhadamente, observa-se que os resultados das coberturas médias se deve pela simplicidade dos métodos presentes nos programas em termos de fluxo de controle e de dados. Assim sendo, no caso de programas OO, seriam necessário repetir o experimento para outros conjuntos de programas para obter um resultado mais confiável. Portanto, após o término do experimento e a análise dos resultados obtidos, pode-se concluir que o conjunto de casos de teste utilizados pela comissão julgadora da maratona de programação da ACM foi satisfatório, pelo menos em relação ao conjunto de programas utilizados. Um estudo de caso mais detalhado deve ser conduzido de modo a avaliar-se os resultados aqui obtidos se confirmaram para um número maior de programas.

5. REFERÊNCIAS BIBLIOGRÁFICAS

[1] Vincenzi, A. M. R. , Wong, W. E., Delamaro, N. E., Maldonado, J. C., JaBUTi: *A coverage analysis tool for Java programs::* In: XVII SBES – Simpósio Brasileiro de Engenharia de Software, p 79-84, Manaus, AM, Brasil, out. 2003.

[2] Maratona, Acm international collegiate programming contest'2004. Página WEB, 2004, Disponível em: http://maratona.ime.usp.br/ . Acesso em: 11/04/2006.

1. Bolsista de iniciação científica. Instituto de Informática/UFG - flucio@inf.ufg.br

2.Orientador/Instituto de Informática/UFG - auri@inf.ufg.br