COBERTURA DOS CRITÉRIOS POTENCIAIS-USOS E A CONFIABILIDADE DO SOFTWARE

ADALBERTO NOBIATO CRESPO[1]
ALBERTO PASQUINI[2]
MÁRIO JINO[1]
JOSÉ CARLOS MALDONADO[3]

[1] FEEC-UNICAMP
Av. Alberto Einstein, 400
CEP: 13083-970, C.P. 6101
Campinas -SP, Brasil
{crespo,jino}@dca.fee.unicamp.br

[3] ICMSC-USP - São Carlos
Rua Dr. Carlos Botelho, 1465,
CEP: 13560-970, C.P. 668
São Carlos - SP, Brasil
jcmaldon@icmsc.usp.br

[2] ENEA - Roma - Itália
PASQUINI_A@casaccia.enea.it

Resumo

O objetivo principal deste artigo é apresentar os resultados de um experimento[1] realizado para se investigar a relação entre a cobertura do código e a confiabilidade do software. Outras relações também foram investigadas: cobertura do código e dados de teste, cobertura do código e defeitos removidos, confiabilidade do software e defeitos removidos e finalmente confiabilidade do software e dados de teste. O experimento foi realizado para se investigar essas relações, aplicando-se a família de critérios de teste estruturais *Potenciais-Usos*. Os dados de teste foram gerados aleatoriamente, condicionados ao perfil operacional do usuário. A cobertura dos elementos requeridos dos critérios foi calculada utilizando-se a ferramenta de teste POKETOOL e a confiabilidade do software foi estimada utilizando-se um algoritmo específico.

Estes *resultados* mostram-nos que pode valer a pena investigar o uso de informação sobre cobertura de elementos requeridos nos modelos de crescimento de confiabilidade de software.
Palavras-chave: confiabilidade, cobertura, citérios de teste.

Abstract

The main purpose of this paper is to present the results of an experiment carried out to investigate the relationship between code coverage and software reliability. Other relationships were also investigate: code coverage versus test data, code coverage and removed faults, software reliability versus removed faults and software reliability versus test data. Criteria from the potencial-uses family of structural testing criteria were applied. The data set was randomly selected conditioned to a user operational profile. Coverage of required elements of the criteria was determined using the POKE-TOOL testing tool and software reliability was calculated by a specific algorithm. Results from the experiment show the existence of a strong correlation between code coverage and software reliability. These results show us that it may be worthwhile to investigate the use of information on coverage of required elements in software reliability growth models.
Keywords: Reliability, coverage, test criteria.

1 Introdução

1.1 - Fundamentação

1 - Experimento realizado no centro de pesquisa ENEA - "Ente per le Nuove tecnologia, LÉnergie e L'Ambiente em Roma - Itália, num intercâmbio científico com o Departamento de Computação e Automação Industrial da Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas DCA/FEE/UNICAMP

PDF compression, OCR, web optimization using a watermarked evaluation copy of CVISION PDFCompressor

No processo de desenvolvimento de produtos de software, assim como em outras engenharias, obter o compromisso entre a *qualidade* e o *custo* constitui o maior objetivo dos projetistas. A confiabilidade é um dos fatores da qualidade que tem sido extensivamente considerado na análise da qualidade do software e, por isso, têm chamado a atenção dos pesquisadores, sendo uma área relativamente nova para a comunidade científica.

A preocupação com a confiabilidade de software teve início por volta de 1967 com Hudson [11] e, a partir dos anos 70, fundamentados na teoria sobre confiabilidade de hardware surgiram os primeiros estudos e, conseqüentemente, os primeiros modelos de crescimento de confiabilidade [2] [6] [9]. Na década de 80, os estudos se ampliaram e surgiram vários outros modelos. Atualmente na literatura pode-se obter uma estimativa de 40 modelos de confiabilidade de software [18]. A modelagem sobre confiabilidade de software consiste em criar um modelo matemático capaz de fornecer uma descrição probabilística precisa da confiabilidade baseada nas suposições a priori sobre os fatores que afetam a confiabilidade e, também, baseada nos resultados obtidos pelos dados experimentais. Basicamente, um modelo de confiabilidade é um modelo matemático que representa falhas como um processo aleatório que é caracterizado pelo *tempo entre falhas* ou pelo *número de falhas* num período de tempo. Um modelo é construído assumindo-se uma adequada função densidade de probabilidade para as variáveis aleatórias de interesse, estimando-se então os parâmetros livres por meio de procedimentos de inferência estatística aplicados aos dados coletados. A maioria dos modelos de confiabilidade de software concentra-se na descrição da *evolução da confiabilidade* durante a fase de teste do software.

1.2 - Limitações dos Modelos e Necessidade de Melhorias

Apesar do grande esforço de pesquisa na área de modelagem, os modelos podem somente fornecer uma estimação grosseira da confiabilidade do software. As limitações dos modelos de confiabilidade e as razões para sua insuficiente qualidade preditiva têm sido apontada por vários autores [3] [4]. Até mesmo os mais recentes modelos sobre confiabilidade utilizam somente dados sobre falhas obtidos através do teste funcional, ou teste "caixa preta", com a simples suposição de que a confiabilidade cresce automaticamente com o progresso do teste [5]. O resultado concreto é que um modelo pode se enquadrar bem para um programa mas falhar para outro programa. No procedimento sobre a modelagem de confiabilidade as informações sobre a cobertura de elementos requeridos, considerando-se um teste "caixa branca", não é utilizada em nenhuma das abordagens. Mesmo sabendo-se que a ocorrência de falhas, na maioria das vezes, está relacionada com o exercício de elementos requeridos do teste. Uma outra restrição nos modelos de confiabilidade de software está nos métodos de teste. Toda estratégia de teste tem um limite na sua capacidade em revelar defeitos em um programa [15]. Quando uma estratégia de teste atinge seu limite, não se consegue detectar mais nenhum defeito com essa estratégia e, assim, o tempo entre falhas é dramaticamente aumentado. Logo, a estimativa da confiabilidade, produzida pelos modelos baseados no domínio do tempo, cresce sem que haja a remoção de defeitos. Os testadores que não estiverem conscientes do nível de saturação da estratégia de teste podem obter uma superestimação da confiabilidade. Outro fator importante no uso dos modelos de confiabilidade é a estimação do perfil operacional. Porém, o perfil operacional pode ser difícil de ser estimado, principalmente em softwares utilizados em controle de processos. Em outros casos, um único perfil operacional de um software pode não ser suficiente para os diferentes usuários deste software. Além disso, um perfil operacional pode ser alterado durante o

PDF compression, OCR, web optimization using a watermarked evaluation copy of CVISION PDFCompressor

processo de manutenção do software. Todas essas causas podem conduzir a erros na estimação do perfil operacional que afetam a sensibilidade dos modelos de crescimento de confiabilidade, [16]. Os modelos baseados no domínio do tempo não consideram a estratégia de teste utilizada. Experimentos realizados evidenciam que diferentes técnicas de teste resultam em diferentes estimativas da confiabilidade, o que evidentemente afeta o desempenho desses modelos, [10]. Desta forma, os modelos apenas tentam simular a forma de uma função matemática que representa o crescimento da confiabilidade. Não se tem considerado as razões do fenômeno representado por aquela forma funcional. Além dessas limitações, existem problemas referentes às suposições que são feitas em muitos modelos de crescimento de confiabilidade e que não podem ser consideradas em aplicações práticas [7] [14]. Uma outra restrição com referência ao *tempo*, utilizado como variável de controle do teste, é que não se pode garantir que o esforço do teste está sendo adequado se as entradas selecionadas do domínio não executam pelo menos as principais funções do software em teste. Imaginando-se que fosse possível a realização de um teste onde o tempo $t \rightarrow \infty$, de nada adiantaria o esforço desse teste se as entradas fossem de tal forma que exercitassem sempre apenas algumas funções do software, ou seja, sempre os mesmos elementos, no caso de um teste estrutural. Neste ponto, ressalta-se a importância de se observar a cobertura de elementos requeridos, quando se utiliza um teste "caixa branca", para garantir que pelo menos uma certa porcentagem dos elementos requeridos seja exercitada. Neste contexto, o tempo não é importante. Muitos modelos de confiabilidade consideram o *tempo* como variável de controle do teste. Talvez essa grande utilização seja devido à herança da teoria sobre confiabilidade de hardware, onde a variável *tempo* para o teste é de grande significância.

Calcada nessas evidências, surge uma nova tendência: a de se utilizar informações que têm alguma relação com a confiabilidade para o efeito de modelagem. Com isso, surgem os estudos para se investigar a existência de alguma relação entre a cobertura do código e a confiabilidade do software. Acredita-se que a cobertura do teste estrutural e a confiabilidade estejam estreitamente relacionadas [1] [5]. Essa abordagem, onde se utiliza a cobertura do código como informação relacionada com a confiabilidade, pode ser uma alternativa para a tradicional abordagem "caixa preta" de teste, onde não se considera a estrutura do código em teste, para a estimação de confiabilidade. Alguns estudos já foram feitos na direção desta nova abordagem tais como os de Ramsey e Basili [8], Adams [13], Garg [1], Veevers and Marshall [20], Malaiya e outros [12]. O mais recente experimento com teste baseado em fluxo de controle e fluxo de dados foi realizado por Frate, Garg, Mathur and Pasquini [19]. Os resultados evidenciaram a existência de uma relação entre cobertura e confiabilidade. Chen [10] também utilizou a cobertura do teste estrutural para criar um fator de compressão que deveria ser usado nos tradicionais modelos de confiabilidade de software baseados no tempo.

Este artigo mostra os resultados de um experimento realizado para se investigar a existência de alguma relação entre a confiabilidade do software e a cobertura do código, utilizando-se uma família de critérios estruturais denominada "Potenciais-Usos", mais especificamente os critérios *Todos-potenciais-usos*, *Todos-potenciais-usos/du* e *Todos-potenciais-du-caminhos*. A Seção 2 trata da descrição do experimento e dos critérios de teste utilizados. A Seção 3 trata da metodologia utilizada no experimento, onde se descreve o software escolhido para teste e o procedimento de geração dos dados de teste. Na Seção 4 apresenta-se a análise dos resultados. Finalmente, na Seção 5 apresentam-se as conclusões.

PDF compression, OCR, web optimization using a watermarked evaluation copy of CVISION PDFCompressor

2 Descrição do Experimento

Neste experimento, investigaram-se algumas relações envolvendo as variáveis confiabilidade do software e cobertura dos elementos requeridos aplicando-se um critério de teste estrutural mais forte do que os critérios utilizados em [19]. Trata-se da família dos critérios Potenciais-usos , mais especificamente os critérios Todos-potenciais-usos, Todos-potenciais-usos/du e Todos-potenciais-du-caminhos [17]. Os critérios baseados em fluxo de controle Todos os Nós e Todos os Arcos serão também aplicados para que os resultados sejam utilizados apenas como referência, por serem critérios mais fracos. A realização deste experimento permitiu investigar as seguintes relações:
cobertura do código x defeitos revelados, para cada um dos critérios;
cobertura do código x dados de teste, para cada um dos critérios;
cobertura do código x confiabilidade do software, para cada um dos critérios;
confiabilidade do software x defeitos removidos;
confiabilidade do software x dados de teste;

Acredita-se que a realização deste experimento seja mais uma contribuição para o entendimento da relação entre a confiabilidade do software e a cobertura de elementos requeridos de um teste estrutural. Os critérios Potenciais-usos são critérios baseados em fluxo de dados muito fortes. Daí o interesse em verificar a existência e o entendimento dessa relação na presença desses critérios. A hierarquia entre os critérios do experimento é a seguinte: Todos os Nós; Todos os Arcos; Todos-potenciais-usos; Todos-potenciais-du-caminhos; Todos-potenciais-usos/du,onde o critério Todos os Nós é o menos exigente (o mais fraco) e o critério Todos-potenciais-usos/du é o mais exigente (o mais forte).

3 Metodologia

O programa escolhido para o experimento é um software denominado "space", utilizado para calcular e fornecer parâmetros como entrada a um outro software; este, por sua vez, é utilizado para calcular a melhor distribuição física de antenas utilizadas em aplicações espaciais. O software em teste foi desenvolvido pela European Space Agency - ESA, em linguagem C e consiste de aproximadamente 10.000 linhas de código, sendo 6.100 linhas de código executável. O sistema consiste de um programa principal e mais 134 rotinas interligadas por passagem de parâmetros. Durante o procedimento do teste de integração e do uso operacional do software, 33 defeitos foram descobertos e registrados.

Os dados de teste foram gerados por um processo aleatório condicionado às probabilidades do perfil operacional utilizado. O perfil operacional do programa "space" foi calculado entrevistando-se o usuário e atribuindo-se probabilidades de execução às funções do programa. Neste procedimento, 20.000 dados de teste foram gerados para serem utilizados na atividade de detecção das falhas e remoção dos defeitos. Assim, o processo de geração dos dados de teste foi automático por intermédio de um programa escrito em linguagem C.

A Tabela 1 mostra os resultados da remoção dos defeitos e do cálculo da cobertura, para cada intervalo de dados de teste e para cada critério utilizado.

Para a observação da relação da confiabilidade com outras variáveis, o cálculo foi efetuado toda vez que houve a detecção de uma falha por um dado de teste e, consequentemente, a remoção do defeito que causou a falha.

A Tabela 2 ilustra a ordem do dado de teste, a ordem da falha e o número acumulado de defeitos removidos do programa "space". Ilustra, também, o total de execuções necessárias e o total de falhas obtidas no programa para se atingir a convergência do algoritmo de cálculo da confiabilidade e, finalmente, a confiabilidade em cada ponto onde houve uma falha.

PDF compression, OCR, web optimization using a watermarked evaluation copy of CVISION PDFCompress

Tabela 1: Cobertura Atingida nos Dados de Teste

Dados de Teste	Ordem Def. Removidos	C R I T É R I O S				
		Nós	Arcos	PU	PDU	PUDU
1 - 1	1	0,30684932	0,21296296	0,1670922	0,07406531	0,15177305
1 - 5	2	0,39931507	0,25396825	0,1951773	0,08897302	0,17361702
1 - 8	3	0,42123288	0,27513228	0,20680851	0,0989115	0,18978723
1 - 12	4	0,4739726	0,32407407	0,24028369	0,11736867	0,22184397
1 - 15	5	0,48493151	0,33333333	0,24794326	0,12091813	0,22921986
1 - 24	6	0,50890411	0,3505291	0,26269504	0,13274965	0,24539007
1 - 44	7	0,53835616	0,38359788	0,28567376	0,15286323	0,26921986
1 - 71	8, 9, 10	0,57465753	0,42328042	0,31829787	0,16919072	0,29531915
1 - 78	11	0,5869863	0,43386243	0,32680851	0,17628964	0,3035461
1 - 100	12, 13	0,64041096	0,47751323	0,3577305	0,19309039	0,3293617
1 - 108	14	0,68082192	0,51587302	0,40992908	0,20894463	0,36453901
1 - 126	15	0,68356164	0,51984127	0,41163121	0,21060104	0,36624113
1 - 1216	16	0,71575342	0,58068783	0,45390071	0,24917179	0,41134752
1 - 1332	17	0,71575342	0,58068783	0,4541844	0,25011832	0,41163121
1 - 1901	18	0,71780822	0,58465608	0,46212766	0,25224799	0,41560284

Legenda: PU: Todos-potenciais-usos; PDU: Todos-potenciais-usos/du; PUDU: Todos-potenciais-du-caminhos.

A Tabela 3 mostra a ordem dos defeitos removidos, a correspondente confiabilidade do programa "space" e a cobertura dos critérios aplicados.

Tabela 2: Confiabilidade do programa "space"

Remoção de Defeitos			Estimação da Confiabilidade		
Dado de Teste	Ordem da Falha	Ordem Def. Removidos	Número de Execuções	Número de Falhas	Confia-bilidade
—	0	0	5199	3127	0,398654
1	1	1	4037	2410	0,403022
5	2	2	3100	1170	0,622581
8	3	3	2051	647	0,684544
12	4	4	1971	567	0,712329
15	5	5	1758	359	0,795791
24	6	6	3051	303	0,900688
44	7	7	2942	296	0,899388
71	8	8, 9, 10	4345	350	0,919448
78	9	11	2023	93	0,954029
100	10	12, 13	2307	100	0,956640
108	11	14	1177	45	0,961767
126	12	15	2145	19	0,991142
1216	13	16	2376	8	0,996633
1332	14	17	3426	3	0,999124
1901	15	18	3840	2	0,999479

4 Análise dos Resultados

Os resultados do experimento se caracterizam pela análise feita nos dados tabulados e nas observações dos gráficos que representam o comportamento das variáveis em estudo. São apresentados separadamente em cada um dos itens seguintes.

PDF compression, OCR, web optimization using a watermarked evaluation copy of CVISION PDFCompressor

4.1 - Relação Entre Cobertura do Código e Defeitos Removidos

Os valores da cobertura atingida e os defeitos removidos estão ilustrados na Tabela 1. Ressalta-se que o cálculo da cobertura é efetuado sempre após a remoção do defeito. Assim, por exemplo, o valor 0,21296296 representa a cobertura atingida no critério Todos os Arcos, após a remoção do primeiro defeito.

A Figura 1 ilustra o comportamento do crescimento da cobertura em função dos defeitos removidos, para cada um dos critérios aplicados. Com o gráfico da Figura 1, observa-se claramente a hierarquia existente entre os critérios: Todos Nós, Todos Arcos, PU, PUDU e PDU. Esta relação hierárquica está em conformidade com a demonstração teórica feita em [17]. No limite, pode-se observar a diferença da cobertura atingida entre os critérios Todos os Nós (o mais fraco) e PDU (o mais forte), quando há a remoção de um defeito. Na remoção dos 18 defeitos apenas 25,22% de cobertura foi atingida com o critério PDU, contra 71,78% com o critério Todos os Nós. Isto dá uma noção do poder de cada critério de teste em revelar defeitos, se o critério fosse adotado como estratégia para seleção dos dados de teste. Ainda na Figura 1, pode-se observar o comportamento do crescimento de cobertura em cada critério. Por ser um critério mais forte, PDU apresenta um crescimento menos acentuado em relação aos demais critérios. Aparentemente, numa análise superficial, todos os critérios utilizados mostram uma relação linear entre a cobertura e o número de defeitos removidos, possibilitando uma grosseira estimativa de cobertura atingida com a remoção do próximo defeito.

Numa análise mais detalhada, pode-se observar a existência de um fator de crescimento de cobertura mais acentuado quando houve a remoção de certos defeitos. Esse comportamento acontece na remoção dos 4°, 7°, 14° e 16° defeitos, em todos os critérios. Isto, obviamente caracteriza alguma particularidade nestes defeitos, como por exemplo a importância da sua remoção.

Tabela 3: Cobertura dos Critérios e a Confiabilidade

Ordem Def. Removidos	Confia-bilidade	C R I T É R I O S				
		Nós	Arcos	PU	PDU	PUDU
0	0,398654	0	0	0	0	0
1	0,403022	0,33424657	0,21164021	0,1670922	0,07406531	0,15177305
2	0,622581	0,39931507	0,25396825	0,1951773	0,08897302	0,17361702
3	0,684544	0,42123288	0,27513228	0,20680851	0,0989115	0,18978723
4	0,712329	0,4739726	0,32407407	0,24028369	0,11736867	0,22184397
5	0,795791	0,48493151	0,33333333	0,24794326	0,12091813	0,22921986
6	0,900688	0,50890411	0,3505291	0,26269504	0,13274965	0,24539007
7	0,899388	0,53835616	0,38359788	0,28567376	0,15286323	0,26921986
8, 9, 10	0,919448	0,57465753	0,42328042	0,31829787	0,16919072	0,29531915
11	0,954029	0,5869863	0,43386243	0,32680851	0,17628964	0,3035461
12, 13	0,956640	0,64041096	0,47751323	0,3577305	0,19309039	0,3293617
14	0,961767	0,68082192	0,51587302	0,40992908	0,20894463	0,36453901
15	0,991142	0,68356164	0,51984127	0,41163121	0,21060104	0,36624113
16	0,996633	0,71575342	0,58068783	0,45390071	0,24917179	0,41134752
17	0,999124	0,71575342	0,58068783	0,4541844	0,25011832	0,41163121
18	0,999479	0,71780822	0,58465608	0,46212766	0,25224799	0,41560284

Legenda: PU: Todos-potenciais-usos; PDU: Todos-potenciais-usos/du; PUDU: Todos-potenciais-du-caminhos.

Os resultados aqui apresentados estão compatíveis com os resultados obtidos no experimento realizado em [19] onde foram aplicados os critérios de teste Todos os Blocos, Todas as Decisões e Todos os Usos. O comportamento da cobertura dos critérios Todos os Nós e Todos os Arcos nesse experimento está semelhante ao comportamento da cobertura dos

PDF compression, OCR, web optimization using a watermarked evaluation copy of CVISION PDFCompressor

critérios Todos os Blocos e Todas as Decisões no experimento realizado em [19]. Numa análise conjunta envolvendo os critérios de teste de ambos os experimentos o comportamento da cobertura alcançada em função dos defeitos removidos obedece perfeitamente à hierarquia existente entre os critérios, analisada em [17]. A coerência nos dados de ambos os experimentos reforça as conclusões obtidas sobre os resultados.

4.2 - Relação Entre Cobertura do Código e os Dados de Teste

Os resultados obtidos sobre cobertura do código e aplicação dos dados de teste são observados, também, na Tabela 1. As Figuras 2a e 2b ilustram graficamente o comportamento da cobertura em função do número de dados de teste aplicados. A Figura 2b é um complemento da Figura 2a.

Pode-se observar claramente em todos os critérios o efeito do crescimento da cobertura na aplicação dos primeiros dados de teste. Até a aplicação do vigésimo dado de teste, o crescimento da cobertura é bastante acentuado, observando-se, evidentemente, a hierarquia dos critérios. Isto é, mais acentuada no critério mais fraco - Todos os Nós, e não tanto acentuada no critério mais forte - PDU. A cobertura continua com um crescimento acentuado, porém com um fator menor, até a aplicação do dado de teste numero 108. Após isto, a cobertura cresce de uma forma bem mais suave, em todos os critérios.

O crescimento acentuado com a aplicação dos primeiros dados de teste, parece óbvio desde que esse fato ocorre qualquer que seja o critério de seleção de dados utilizado.

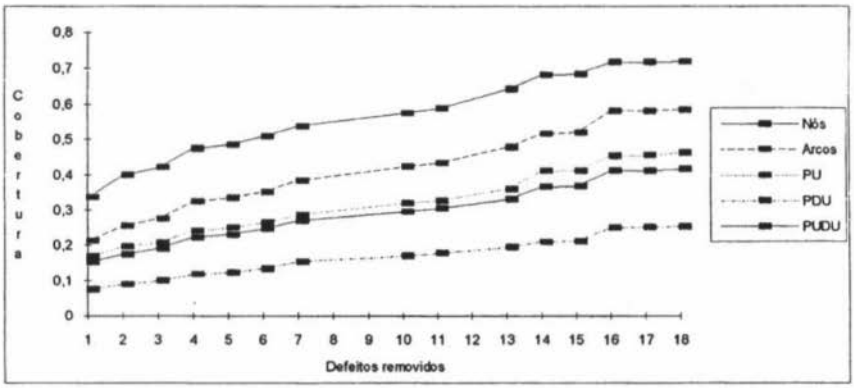

Figura 1 *Relação Entre Cobertura e Defeitos Removidos*

O crescimento suavizado a partir de um certo ponto e a tendência à estabilização no final da aplicação dos dados está explicado pela estratégia de seleção adotada.
A estratégia de seleção dos dados de teste adotada foi a aleatória, condicionada ao perfil operacional do usuário.

Isto também explica o maior número de falhas detectadas com a aplicação dos primeiros 100 dados de teste.

PDF compression, OCR, web optimization using a watermarked evaluation copy of CVISION PDFCompressor

Devido ao uso do perfil operacional na seleção dos dados, os defeitos situados na parte do código com baixa probabilidade de ser exercitado tornam-se mais dificeis de serem revelados.

Isto conduz à aplicação de um maior número de dados de teste para se detectar uma falha causada por um defeito com essa característica. Assim, o comportamento da cobertura lustrada nos gráficos das Figuras 2a e 2b parece estar em conformidade com a estratégia de seleção adotada.

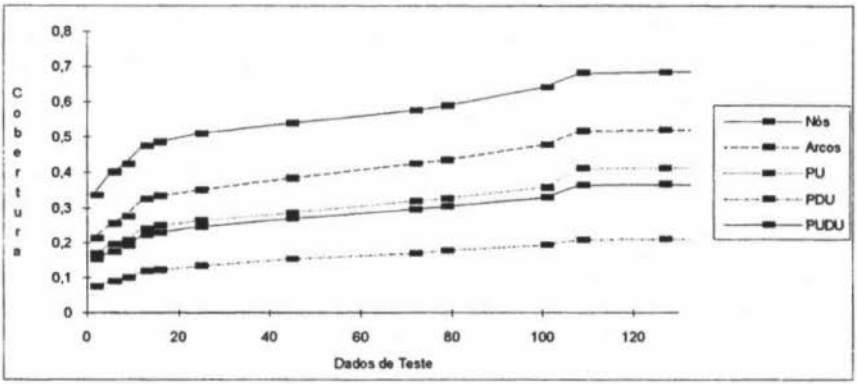

Figura 2a: *Relação Entre Cobertura e Dados de Teste*

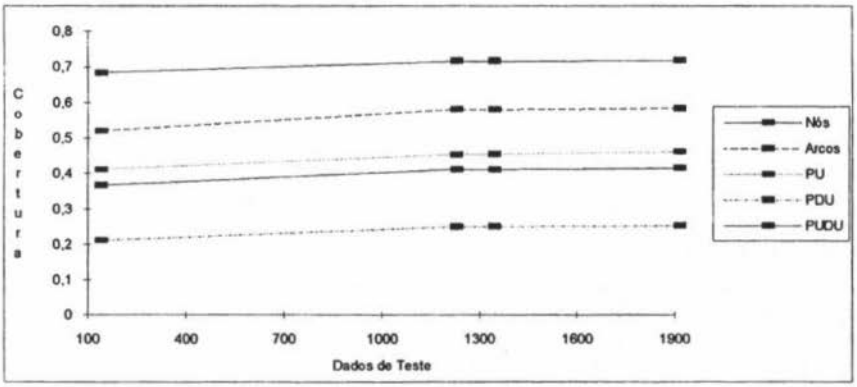

Figura 2b: *Relação Entre Cobertura e Dados de Teste* (complemento)

PDF compression, OCR, web optimization using a watermarked evaluation copy of CVISION PDFCompressor

4.3 - Relação Entre Cobertura do Código e Confiabilidade do Software

Os resultados obtidos sobre cobertura e confiabilidade podem ser observados na Tabela 3. As Figuras 3 e 4 ilustram o comportamento gráfico das relações entre a confiabilidade e a cobertura dos critérios de fluxo de controle - Todos os Nós e Todos os Arcos, respectivamente.

Tabela 4: Correlação Entre Confiabilidade e Cobertura

	COBERTURA DOS CRITÉRIOS				
Correlação	Nós	Arcos	PU	PDU	PUDU
Confiabil.	0,93381474	0,89146152	0,86618407	0,87546915	0,88310049

Legenda: PU: Todos-potenciais-usos; PDU: Todos-potenciais-usos/du; PUDU: Todos-potenciais-du-caminhos

As Figuras 5, 6 e 7 ilustram o comportamento da confiabilidade e a cobertura atingida referentes aos critérios Todos-potenciais-usos, Todos-potenciais-usos/du e Todos-potenciais-du-caminhos, respectivamente. Os dados sobre a medida de correlação estatística, existente entre a confiabilidade e a cobertura dos critérios utilizados, encontram-se na Tabela 4.

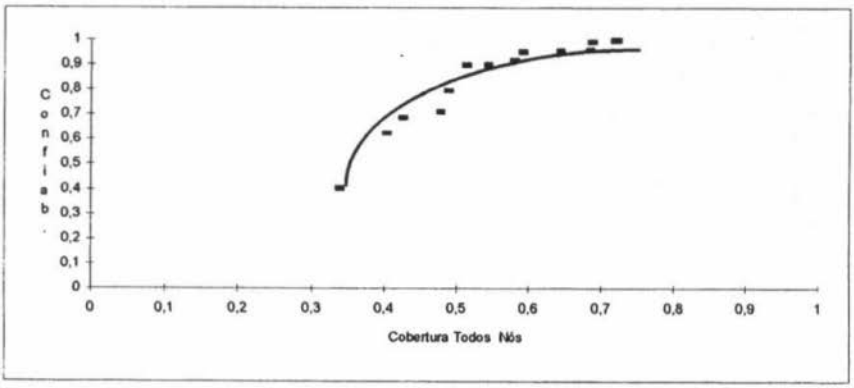

Figura 3: *Relação Entre Cobertura e Confiabilidade - Critério Todos Nós*

Observando-se os valores da correlação na Tabela 4 pode-se notar a existência de uma forte correlação da confiabilidade com a cobertura dos critérios utilizados. Todos os valores são significativamente altos. Portanto, neste experimento, fica comprovado que a cobertura pode ser uma medida que contribui para a estimativa da confiabilidade. Assim, faz sentido pensar na existência de uma relação matemática entre as variáveis confiabilidade e cobertura.

Numa análise visual dos gráficos nas Figuras 3 a 7, observa-se a existência de uma relação não linear entre a confiabilidade e a cobertura dos critérios aplicados. Pode ser notado em cada uma das figuras que a relação empírica existente entre a confiabilidade e a cobertura difere sensivelmente para cada um dos critérios.

Essa diferença observada entre as relações inviabiliza a existência de uma relação entre a confiabilidade e a cobertura, sem considerar as características do critério estrutural que está sendo utilizado.

PDF compression, OCR, web optimization using a watermarked evaluation copy of CVISION PDFCompressor

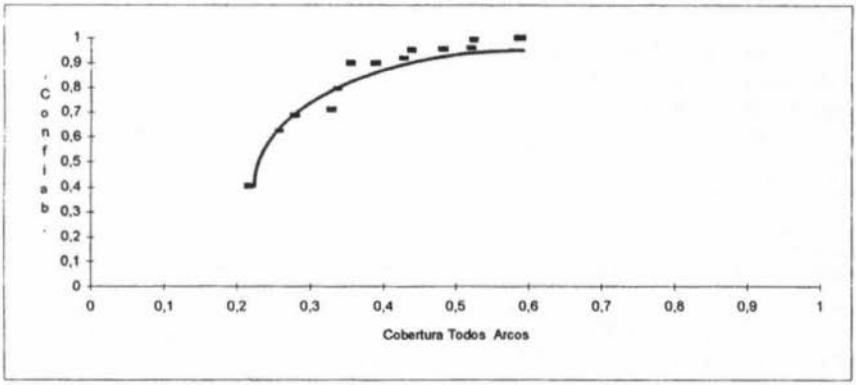

Figura 4: *Relação Entre Cobertura e Confiabilidade - Critério Todos Arcos*

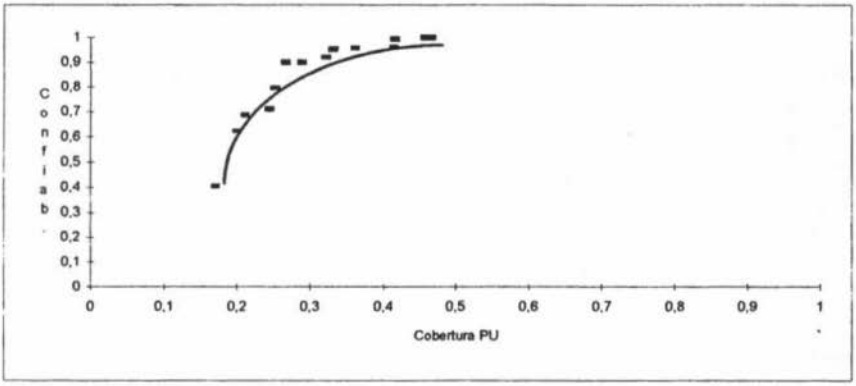

Figura 5: *Relação Entre Cobertura e Confiabilidade - Critério PU*

índice de cobertura atingido na aplicação dos dados para a detecção de uma falha pode ser uma indicação do poder de detecção do teste. Esse poder de detecção do teste, evidentemente, caracteriza a relação entre a confiabilidade e a cobertura.

Assim, é viável pensar na existência de uma relação matemática, entre a confiabilidade e a cobertura, que inclua um parâmetro representando o poder de detecção do teste estrutural que está sendo utilizado

Baseado nestas evidências, é razoável pensar na existência de uma relação que varia conforme o peso atribuído ao teste utilizado. Assim, a Figura 8 traz uma noção gráfica de como poderia ser caracterizada a relação matemática em questão.

PDF compression, OCR, web optimization using a watermarked evaluation copy of CVISION PDFCompressor

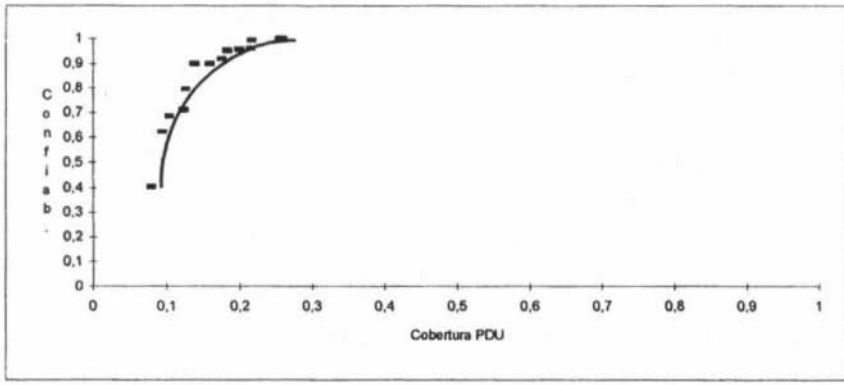

Figura 6: *Relação Entre Cobertura e Confiabilidade - Critério PDU*

Na utilização de um critério fraco, possivelmente, atinge-se a cobertura total 100% do código, sem que a confiabilidade chegue ao seu valor limite. Uma situação oposta pode ocorrer na presença de um critério forte, isto é, possivelmente a confiabilidade pode atingir o seu valor máximo sem que a cobertura do código atinja seu limite.

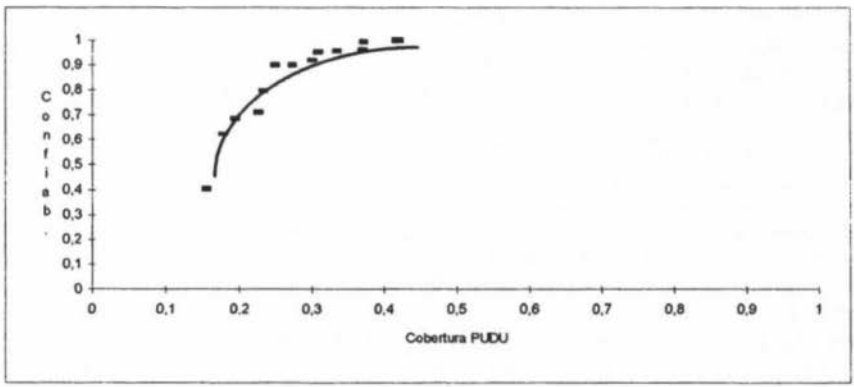

Figura 7: *Relação Entre Cobertura e Confiabilidade - Critério PUDU*

Esses resultados podem ser comparados com os resultados obtidos no experimento realizado em [19], referente ao programa "space". Naquele experimento a relação de não linearidade entre a cobertura e a confiabilidade também acontece de uma forma bastante suave, quando se observam os resultados nos gráficos.

Comparando-se os resultados dos dois experimentos percebe-se que a não linearidade aumenta à medida em que o critério de teste utilizado é mais forte. Os índices de cobertura atingidos pelos critérios comuns aos dois experimentos são praticamente os mesmos.

PDF compression, OCR, web optimization using a watermarked evaluation copy of CVISION PDFCompressor

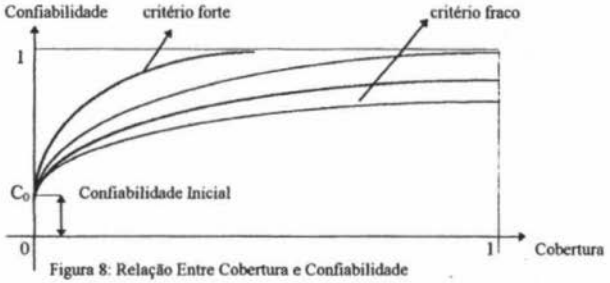

Figura 8: Relação Entre Cobertura e Confiabilidade

4.4 - Relação Entre Confiabilidade do Software e Defeitos Removidos

Os resultados obtidos sobre confiabilidade e defeitos removidos podem ser observados na Tabela 2. A Figura 9 apresenta a visualização gráfica do comportamento da confiabilidade com os defeitos removidos. Deve ser ressaltado que o cálculo da confiabilidade é sempre efetuado após a remoção do defeito. Como exemplo, o valor 0,622581 na Tabela 3 representa a confiabilidade do programa "space" após a remoção do segundo defeito. Observando-se o gráfico na Figura 9 percebe-se um rápido crescimento na confiabilidade após a remoção dos primeiros defeitos. Fato observado principalmente até a remoção do 2º. defeito. Após a remoção do 6º. defeito, a confiabilidade tem um crescimento menos acentuado, sendo bastante suave na remoção dos últimos defeitos. O crescimento acentuado, logo no início, indica que a remoção desses defeitos interfere intensamente no comportamento da taxa de falhas. São defeitos que se situam numa parte do código com alta probabilidade de execução. Observando-se os resultados na Tabela 3, a confiabilidade decresce após a remoção do 7º. defeito, significando que o programa tende a falhar mais. Este decréscimo de confiabilidade pode ocorrer por algumas razões. Neste caso específico, ocorre devido ao fato que o 7º. defeito estava mascarando a ocorrência da próxima falha.

O crescimento suave da confiabilidade na detecção dos últimos defeitos, tendendo à estabilização, identifica o baixo índice de detectabilidade desses defeitos, por estarem localizados em regiões do código com baixa probabilidade de execução. Comparando-se os resultados com o experimento realizado em [19] observa-se que o comportamento final da confiabilidade do software é idêntico em ambos os experimentos. Isto é, em ambos a confiabilidade cresce rapidamente na remoção dos primeiros defeitos, tende à estabilização na remoção dos últimos defeitos e converge para o mesmo valor. Porém, observa-se que as trajetórias das confiabilidades nos dois experimento são distintas. Essa diferença nas trajetórias é explicada pela diferença entre os conjuntos de dados de teste gerados aleatoriamente nos dois experimentos, apesar das estratégias de seleção de dados serem as mesmas.

4.5 - Relação Entre Confiabilidade do Software e Dados de Teste

Os resultados obtidos sobre confiabilidade e dados de teste, podem ser vistos na Tabela 2. As Figuras 10a e 10b ilustram o comportamento gráfico da confiabilidade em função dos dados de teste aplicados. A Figura 10b é um complemento da Figura 10a.

PDF compression, OCR, web optimization using a watermarked evaluation copy of CVISION PDFCompressor

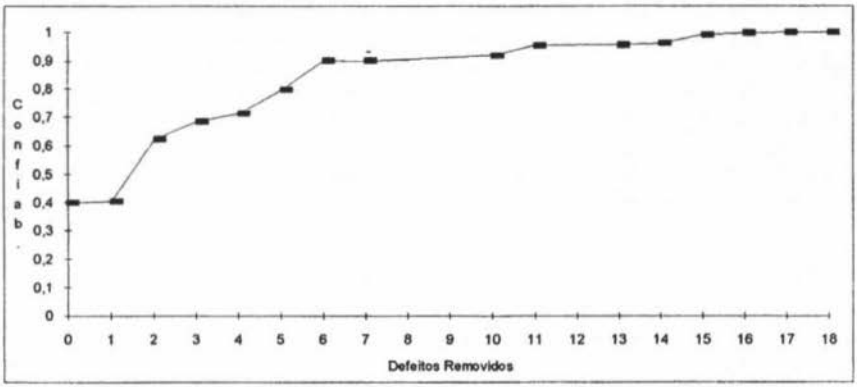

Figura 9: *Relação Entre Confiabilidade e Defeitos Removidos*

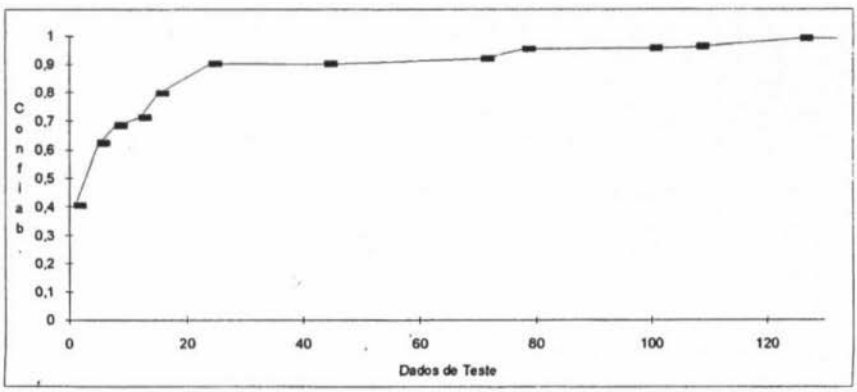

Figura 10: *Relação Entre Confiabilidade e Dados de Teste*

Observando-se os dados na Tabela 2 e a Figura 10, nota-se, claramente, o rápido crescimento da confiabilidade com a aplicação dos primeiros 24 dados de teste.

Após a aplicação do 24º. dado de teste, a confiabilidade cresce numa forma mais suave, tendendo à estabilização após a aplicação do 100º. dado de teste.

O comportamento da confiabilidade está bastante coerente com a estratégia de seleção de dados adotada. O rápido crescimento no início, revela o poder de detecção de falhas existente nos dados de teste selecionados. Isto acontece devido à estratégia de seleção adotada - aleatória, condicionada ao perfil operacional do usuário. Isto revela que os defeitos que causaram essas falhas estão situados na parte do código com alta probabilidade de

PDF compression, OCR, web optimization using a watermarked evaluation copy of CVISION PDFCompressor

execução. O crescimento da confiabilidade numa forma mais suave, tendendo à estabilização, revela que os defeitos que causaram essas falhas se encontram em partes do código com baixa probabilidade de execução. O grande número de dados de teste necessários para se detectar os últimos defeitos revela o nível de saturação existente na estratégia de seleção adotada. Ressalta-se que 20.000 dados de teste foram aplicados e após o 1901º nenhum defeito novo foi detectado, indicando assim a saturação da estratégia. Certamente, uma estratégia de seleção baseada na cobertura de qualquer teste estrutural conduziria à utilização de um menor número de dados de teste necessários para a detecção desses defeitos.

5 Conclusão

Analisando-se os resultados obtidos, alguns pontos merecem destaque:

- Existe uma relação entre a cobertura do código atingida pelo critério de teste e os defeitos removidos do software. O crescimento da cobertura do código provocado pela remoção de um defeito ocorre em todos os critérios, com exceção de um mesmo caso observado nos critérios Todos os Nós e Todos os Arcos. Essa exceção é explicada pela presença de dois defeitos sobrepostos situados num mesmo nó que por sua vez não é um nó de decisão. Observa-se que o crescimento da cobertura obedece rigorosamente à hierarquia existente entre os critérios utilizados. Esse comportamento pode ser observado no gráfico da Figura 1. Os resultados da relação entre cobertura do código e defeitos removidos estão compatíveis com os resultados obtidos no experimento realizado em [19] onde foram aplicados os critérios de teste Todos os Blocos, Todas as Decisões e Todos os Usos. O comportamento da cobertura dos critérios Todos os Nós e Todos os Arcos nesse experimento está semelhante ao comportamento da cobertura dos critérios Todos o Blocos e Todas as Decisões no experimento realizado em [19]. Numa análise conjunta envolvendo os critérios de ambos os experimentos o comportamento da cobertura em função dos defeitos removidos obedece perfeitamente à hierarquia entre os critérios, analisada em [17].
- Em relação ao número de dados de teste aplicados e a cobertura do código atingida pelo critério de teste, os 20 primeiros dados de teste provocou um rápido crescimento na cobertura. Entre os 21º e 108º dados de teste a cobertura cresce de uma forma mais suave. Após o 108º dado de teste a cobertura ainda cresce mas de uma forma bastante suave, tendendo à estabilização. O crescimento acentuado no início parece óbvio uma vez que esse fato deve ocorrer qualquer que seja o critério de seleção de dados de teste adotado. O crescimento suavizado, a partir de um certo ponto, tendendo à estabilização mostra-se compatível com a estratégia de seleção de dados adotada; aleatória condicionada ao perfil operacional do usuário.

  Um maior número de falhas foi detectado com a aplicação dos primeiros 100 dados de teste. Devido ao uso do perfil operacional na estratégia de seleção dos dados as falhas causadas pelos defeitos situados na parte do código com baixa probabilidade de ser exercitada tornam-se mais dificeis de serem reveladas. Isto explica a necessidade de se aplicar um maior número de dados de teste para se revelar as falhas causadas por defeitos com essa característica. Esse comportamento pode ser observado nas Figuras 2a e 2b.
- As medidas de confiabilidade do software e a cobertura do código atingida pelos critérios podem ser observadas na Tabela 3 e nos gráficos das Figuras 3, 4, 5, 6 e 7. As medidas de correlação entre as variáveis confiabilidade e cobertura atingida pelos critérios podem ser observadas na Tabela 4. Observa-se que a correlação entre as variáveis cobertura e a confiabilidade é significativa. Uma análise visual nos gráficos revela a existência de uma

PDF compression, OCR, web optimization using a watermarked evaluation copy of CVISION PDFCompressor

relação ligeiramente não linear entre as variáveis cobertura e a confiabilidade. A não linearidade na relação entre essas variáveis torna-se mais acentuada na medida em que o critério aplicado é mais forte. Essas evidências confirmam que uma relação matemática entre as variáveis cobertura e confiabilidade deve constar de um parâmetro que indique a ponderação atribuída ao critério de teste adotado. Os resultados comprovam que a medida de cobertura atingida pelo critério é uma variável que contribui para a estimação da confiabilidade. Os resultados da relação entre confiabilidade e dados de teste podem ser comparados com os resultados obtidos no experimento realizado em [19], referente ao programa "space". Naquele experimento a relação de não linearidade entre a cobertura e a confiabilidade também acontece de uma forma bastante suave, quando se observa graficamente os resultados. Comparando-se os resultados dos dois experimentos percebe-se que a não linearidade aumenta à medida em que o critério de teste utilizado é mais forte. Os índices de cobertura atingidos pelos critérios comuns aos dois experimentos são praticamente os mesmos.

- Os dados sobre confiabilidade e defeitos removidos podem ser observados na Tabela 3 e no gráfico da Figura 9. Observa-se que há um rápido crescimento da confiabilidade após a remoção dos primeiros defeitos e tende à estabilização a partir de um certo ponto. Esse comportamento indica a alta detectabilidade dos primeiros defeitos removidos e a baixa detectabilidade dos últimos defeitos removidos. Observa-se ainda na Tabela 3 que a confiabilidade decresce após a remoção do 7º defeito indicando que o software pode falhar mais. Esse decréscimo pode ocorrer por algumas razões. Nesse caso específico ocorre devido ao fato que o 7º defeito mascara a ocorrência da próxima falha. Assim, após a remoção do 7º defeito a ocorrência de falhas aumenta, pois poderia ser causada por qualquer um dos defeitos 8, 9 ou 10. Comparando-se os resultados com o experimento realizado em [19] observa-se que o comportamento final da confiabilidade é idêntico em ambos os experimentos. Isto é, em ambos a confiabilidade cresce rapidamente na remoção dos primeiros defeitos, tende à estabilização na remoção dos últimos defeitos e finalmente converge para um mesmo valor. Porém, observa-se que as trajetórias das confiabilidades nos dois experimentos são distintas. Essa diferença nas trajetórias é explicada pela diferença entre os conjuntos de dados de teste gerados aleatoriamente nos dois experimentos, apesar das estratégias de seleção de dados serem as mesmas.
- A relação entre a confiabilidade e número de dados de teste pode ser observada nos dados da Tabela 2 e nos gráficos das Figuras 10a e 10b. Nota-se um rápido crescimento da confiabilidade com a aplicação dos primeiros 24 dados de teste. Após isto, a confiabilidade cresce numa forma mais suave tendendo à estabilização após o 100º dado de teste. O rápido crescimento no início revela o poder de detecção de falhas existente nos dados de teste gerados. Isto acontece devido a estratégia de seleção adotada - aleatória, condicionada ao perfil operacional do usuário. Esse comportamento revela que os defeitos que causaram as primeiras falhas estão situados em regiões do código com alta probabilidade de execução e defeitos que causaram as últimas falhas estão situados em regiões do código com baixa probabilidade de execução. O grande número de dados de teste necessário para se revelar as últimas falhas indica o nível de saturação existente na estratégia de seleção de dados adotada. Ressalta-se que 20.000 dados de teste foram aplicados e após o 1901º nenhuma falha foi detectada. Certamente uma estratégia de seleção de dados baseada na cobertura de qualquer critério de teste estrutural conduziria à utilização de um menor número de dados de teste para a detecção dessas falhas.

PDF compression, OCR, web optimization using a watermarked evaluation copy of CVISION PDFCompressor

O experimento citado em [19] foi realizado com a aplicação de outros critérios de teste, utilizando-se a ferramenta de teste ATAC. Alguns resultados publicados em [19] puderam ser comparados com alguns resultados deste experimento. As relações entre cobertura do código e defeitos removidos, cobertura do código e confiabilidade, confiabilidade e defeitos removidos comparadas nos dois experimentos mostraram se coerentes. As concordâncias nos dados de ambos experimentos comprovam a veracidade dos resultados e reforçam as conclusões obtidas.

Referências

[1] Praerit Garg, "Investigating coverage - reliability relationship and sensitivity of reliability to errors in operational profile", Technical Report - Department of Computer Sciences, Purdue University, West Lafayette, IN 47907, May 1994.

[2] Z. Jelinski and P. Moranda, "Software reliability research", in Statistical Computer Performance Evaluation. W. Freiberger, Ed. New York: Academic, pp. 465 - 485, 1972.

[3] D. Hamlet, "Are We testing for true reliability", IEEE Software, Vol. 9, No. 4, July 1992.

[4] B. Littlewood, L. String, "Validation of ultrahigh dependability for software-based systems", Communication of the ACM, Vol. 36, No.1, Jan. 1993.

[5] Gulyan S. Varadan, "Trends in Reliability and Test Strategies," IEEE Software, pp. 10, May 1995.

[6] B. Littlewood, J. L. Verral, "A Bayesian reliability growth model for computer software", Applied Statistics, vol. 22, pp. 332-346, 1973.

[7] J. Tian, P. Lu and J. Palma, "Test-execution-based reliability measurement and modeling for large commercial software,"IEEE Trans. Soft. Eng., vol 21, no. 5, pp 405-414, May 1995.

[8] J. Ramsey and V. R. Basili, "Analyzing the Test Process Using Structural Coverage," Proc. ICSE'85, pp. 306-312, 1985.

[9] M. L. Shooman, "Probabilistic models for software reliability prediction," in Statistical Computer Performance Evaluation, W. Freiberg, Ed. New York: Academic, pp.485-502, 1972.

[10] M. Chen, A. P. Mathur and V. J. Rego, "Effect of Testing Technique on Software Reliability Estimates Obtained Using A Time- Domain Model," IEEE Transactions on Reliability, vol. 44, no. 1, pp. 97-103 March 1995.

[11] A. Hudson, "Program errors as a Birth and Death Process", Technical Report SP - 3011, Santa Monica, Cal.: Systems Development Corporation. 1967

[12] Y. K. Malaiya, N. Li, J. Bieman, R. Karcick, B. Skibe, "The Relationship Between Test Coverage and Reliability,"Proceedings of the Fifth International Symposiun on Software Reliability Engineering, Monterey, CA, pp. 186-195, November 6-9, 1994

[13] E. N. Adams, "Minimizing Cost Impact of Software Defects," IBM Research Division, Report RC 8228(35669), 1980

[14] J. Tian, "Integrating Time Domain and Input Domain Analysis of Software Reliability Using Tree-Based Models," IEEE Trans. Soft. Eng., vol 21, no. 12, pp. 945-958, December 1995.

[15] M. Chen, "Tools and Tchniques for Testing Based Software Reliability Estimation," Ph. D. Thesis, Purdue University, Aug. 1994.

[16] A. N. Crespo, P. Matrella and A. Pasquini, "Sensitivity of reliability growth models to operational profile errors,".

[17] J. C. Maldonado, "Critérios Potenciais Usos: Uma Contribuição ao Teste Estrutural de Software." Tese de Doutorado, DCA/FEE/UNICAMP - Campinas, SP, Julho 1991.

[18] R. C. Tausworthe, M. R. Lyu, "A Generalized Technique for Simulating Software Reliability," IEEE Software, pp. 77-88, March 1996.

[19] F. D. Frate, P. Garg, A. P. Mathur and A. Pasquini, "Experiments to Investigate the Correlation Between Code Coverage and Software Reliability." SERC-TR-162-P, Software Engineering Research Center, Purdue University, West Lafayette, Indiana 47907, April, 1995.

[20] A. Veevers and A. Marshall, "A Relationship Between Software Coverage Metrics and Reliability," Software Testing, Verification and Reliability, vol. 4, pp. 3-8, 1994.

[21] M. R. Lyu, "Handbook of Reliability Engineering", McGraw-Hill, 1966.

PDF compression, OCR, web optimization using a watermarked evaluation copy of CVISION PDFCompressor