

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE
MINAS GERAIS – CAMPUS V

Teste de Software

Gabriel José
Gustavo Mascarenhas
Matheus Bernardo
Vitor Bonfim

Divinópolis - MG

2015

SUMÁRIO

1.	Introdução	03
2.	Técnicas de Teste de Software	04
2.1.	Caixa-branca	04
2.2.	Caixa-preta	04
2.3.	Análise do Valor Limite	05
2.4.	Causa-efeito	05
2.5.	Particionamento em Classes de Equivalência	05
3.	Técnica Escolhida	06

1. Introdução

Teste de software é uma etapa do processo de desenvolvimento muito importante para verificar o produto que vem sendo trabalhado atingiu todas suas expectativas e seu desempenho necessário no ambiente previamente estabelecido. O seu objetivo principal é alertar e identificar para a equipe de desenvolvimento as falhas cometidas e erros gerados durante a execução a fim de que o programa a ser entregue ao cliente possua o melhor desempenho possível.

2. Técnicas de Teste de Software

No mercado atual existem diversas técnicas que vem sendo desenvolvidas para a realização do teste de software, entre elas: estrutural (caixa-branca), funcional (caixa-preta), análise do valor limite, causa-efeito e particionamento em classes de equivalência. Todos possuem métodos que os diferenciam um dos outro, mas buscam sempre um objetivo principal, identificar falhas no sistema.

2.1 Técnica Estrutural (caixa-branca)

É uma Técnica de Teste, que internamente avalia aspectos software como: teste de fluxo de dados, teste de caminhos lógicos, teste de ciclos e teste de condição. Outra característica dessa técnica é que ela trabalha de forma direta com o código fonte do componente de software.

Outros aspectos também podem ser avaliados, pois os aspectos avaliados dependem da tecnologia e da complexidade da construção do software. O testador tem a capacidade de construir códigos para ligar bibliotecas e seus componentes, pois ele tem acesso ao código fonte da aplicação.

2.2 Técnica Funcional (caixa-preta)

O Teste Funcional, mais conhecido como teste de caixa-preta, é uma técnica de teste que o componente de software a ser testado é tratado como se fosse uma caixa-preta, não levando em conta o comportamento interno do mesmo. É proporcionados dados de entrada, o teste é executado, e sendo assim, o resultado obtido é comparado a um resultado esperado previamente conhecido. Caso o resultado obtido for igual ao resultado esperado, o teste é dado como sucedido.

2.3 Análise do Valor Limite

É comum acontecer erros associados nos valores próximos às extremidades das classes, onde deveriam estar alocados no centro. O teste busca encontrar falhas em entre os limites de valor em cada entrada.

Caso uma entrada tenha seus limites estabelecidos entre X e Y os testes devem ser realizados com valores que coincidem com cada fronteira, incluindo os valores limites.

2.4 Grafo Causa-efeito

A técnica de grafo causa-efeito explora diferentes formas de combinação de dados onde são analisados as condições de entrada e os resultados gerados, os grafo são as especificações a serem utilizadas em uma tabela de decisão.

Primeiramente são analisadas as especificações e o grafo é gerado ligando as causas e os efeitos, após isso são adicionados anotações ao grafo informado sobre as limitações de combinações possíveis. A partir daí é gerado uma tabela de decisões e posteriormente transformada em casos de teste.

2.5 Particionamento em classes de equivalência

Esse método de teste de software consiste em checar a validação e consistência de suas classes. Ele se procede dividindo o domínio de entrada do programa em classes de equivalência (eventualmente, o domínio de saída também pode ser considerado), a partir das quais os casos de teste são derivados, tendo como objetivo minimizar o número de casos de teste. Como todos os elementos de uma classe devem se proceder da mesma forma, pelo menos a princípio, esse critério selecionará apenas um caso de teste de cada classe.

As classes de equivalência serão os meios para se fazer o teste de software. Elas representam um conjunto de dados válidos ou inválidos de um domínio de entrada ou saída. Esses domínios podem valores numéricos específicos, faixa de valores, conjunto de valores relacionados ou condições lógicas.

3. Técnica Escolhida

A técnica escolhida para o nosso projeto de conclusão de curso foi a caixa-branca, mais especificamente a ferramenta JUnit. A ferramenta JUnit foi escolhida, pois o software Somado para Somativa esta sendo desenvolvido na linguagem jsp.

O auxílio na redução dos problemas existentes por meio dessa ferramenta – sejam eles pequenos ou grandes – se deve pelos testes que avaliam os métodos das classes desenvolvidos na linguagem Java, como Teste da Unidade, que testa o menor dos componentes de forma isolada, e o Teste da Integração, que testa a interação entre os componentes do software. A partir daí é possível criar baterias de testes a desejo dos desenvolvedores, e os resultados obtidos são apresentados de forma clara e visível.