Sistema de Informação para Grupos de Pesquisa

Desenvolvedor - Jorge J. G. Leandro
Grupo 3

MAC0332 - 2011
Engenharia de Software
IME - USP

Casos de Teste Caixa Preta - **Testes de Unidade**
Prof: Marco Gerosa

São Paulo, 3 de outubro de 2011

Capítulo 1

Apresentação

O presente documento descreve o cumprimento da tarefa *Criar casos de teste*, mediante uma coleção de Casos de Teste Caixa Preta para Análise Funcional de classes POJO, por meio de dois tipos de testes:

- Testes Baseados em Execução (Testes de Unidade), conforme o documento de requisitos do sistema e de acordo com diretrizes de [Schach, 2007] e gabaritos do *Processo Unificado Aberto - OpenUP*, descritos nas Seção 2.1.1.
- Testes Não-Baseados em Execução, Revisões do tipo *Walkthrough*, descritos na Seção 2.2.

Capítulo 2

Casos de Teste Caixa Preta - Testes de Unidade - Análise Funcional

2.1 Testes Baseados em Execução

2.1.1 Testes de Unidade

As tabelas 2.1 a 2.5 resumem as Classes de Equivalência consideradas significativas para a Análise Funcional das classes aqui documentadas. Segundo [Schach, 2007], classes de equivalência são conjuntos de casos de teste, cujos elementos são tão bons quanto quaisquer outros. O uso das mesmas evita o número exponencial de casos de testes, mesmo para poucos parâmetros.

2.2 Testes Não-baseados em Execução

2.2.1 Revisão de Código - *Walkthrough*

A revisão de código do tipo *Walkthrough* é semelhante à revisão do tipo Inspeção, mas um tanto menos formal. Como resultado de uma tal revisão, são produzidas listas de *ítens não bem compreendidos* e/ou *ítens possivelmente errados*[Schach, 2007]. Neste documento, mesclamos ambas as listas em apenas uma, como segue.

Tabela 2.1: Teste de Unidade - Dados de Contribuinte e Classes de Equivalência

atributo	caso de teste	resultado
Nome	Marco Aurélio Gerosa	aceitável
Nome	Engenharia de Software	aceitável
Publicacao	Lista de publicações *mocked*	aceitável
Usuario 1	Usuário *mocked*	aceitável

Tabela 2.2: Teste de Unidade - Dados de Disciplina e Classes de Equivalência

atributo	caso de teste	resultado
Sigla	mac0332	aceitável
Nome	Engenharia de Software	aceitável
Ementa	Gerenciamento de projeto.	
	Análise e especificação de requisitos.	aceitável
Nome do grupo 1	Grupo de Engenharia de Software	aceitável
Linha de Pesquisa 1 do grupo 1	Métodos Ágeis	aceitável
Linha de Pesquisa 2 do grupo 1	Software Livre	aceitável
Nome do grupo 2	Grupo de Computação Gráfica	aceitável
Linha de Pesquisa 1 do grupo 2	High Quality Image Rendering	aceitável
Linha de Pesquisa 2 do grupo 2	Applied Discrete Geometry	aceitável

Tabela 2.3: Teste de Unidade - Dados de Filiação e Classes de Equivalência

atributo	caso de teste	resultado
Nome do projeto 1	Projeto Gamma	aceitável
Nome do projeto 2	Projeto Beta	aceitável
Financiamento do projeto 1	Fapesp	aceitável
Financiamento do projeto 2	CNPq	aceitável
Descrição do projeto 1	Metodos de Otimizacao	aceitável
Descrição do projeto 2	Metodos de Criptografia	aceitável
Nome do contribuinte	Marco Aurélio Gerosa	aceitável

Tabela 2.4: Teste de Unidade - Dados de Grupo e Classes de Equivalência

atributo	caso de teste	resultado
Nome	Grupo de Sistemas de Software	aceitável
Linha de Pesquisa 1	Métodos Ágeis	aceitável
Linha de Pesquisa 2	Software Livre	aceitável
Projeto 11 da linha 1	Métodos de Otimização	aceitável
Projeto 12 da linha 1	Uso eficaz de Métricas	aceitável
Projeto 21 da linha 2	Achmus	aceitável
Projeto 22 da linha 2	Arquimedes	aceitável
Nome do grupo 2	Grupo de Computação Gráfica	aceitável
Linha de Pesquisa 1 do grupo 2	High Quality Image Rendering	aceitável
Linha de Pesquisa 2 do grupo 2	Applied Discrete Geometry	aceitável

Tabela 2.5: Teste de Unidade - Dados de LinhaPesquisa e Classes de Equivalência

atributo	caso de teste	resultado
Descrição do Projeto 1	Métodos de Otimização	aceitável
Descrição do Projeto 2	Uso eficaz de Métricas	aceitável
Financiamento do projeto 1	Fapesp	aceitável
Financiamento do projeto 2	CNPq	aceitável
Nome do grupo 1	Grupo de Engenharia de Software	aceitável
Nome da linha de pesquisa 11 do grupo 1	Métodos Ágeis	aceitável
Nome da linha de pesquisa 12 do grupo 1	Software Livre	aceitável
Nome do grupo 2	Grupo de Computação Gráfica	aceitável
Nome da linha de pesquisa 21 do grupo 2	High Quality Image Rendering	aceitável
Nome da linha de pesquisa 22 do grupo 2	Applied Discrete Geometry	aceitável
Veiculo	Veiculo.JOURNAL	aceitável
Financiamento	Fapesp	aceitável
Titulo	Service-oriented middleware for the Future Internet: state of the art and research directions	aceitável
Autor	Marco Gerosa	aceitável
Data	25/05/2011	aceitável
Nome do grupo 1	Grupo de Engenharia de Software	aceitável
Nome da linha de pesquisa 11 do grupo 1	Métodos Ágeis	aceitável
Nome da linha de pesquisa 12 do grupo 1	Software Livre	aceitável
Nome do grupo 2	Grupo de Engenharia de Computação Gráfica	aceitável
Nome da linha de pesquisa 21 do grupo 1	High Quality Image Rendering	aceitável
Nome da linha de pesquisa 22 do grupo 1	Applied Discrete Geometry	aceitável

Tabela 2.6: Teste de Unidade - Dados de Projeto e Classes de Equivalência

atributo	caso de teste	resultado
Descricao	Metodos de Otimizacao	aceitável
Financiamento	Fapesp	aceitável
Filiacao 11	Filiacacao mocked	aceitável
Filiacao 12	Filiacao mocked	aceitável
Filiacao 21	Filiacao mocked	aceitável
Filiacao 22	Filiacao mocked	aceitável
Publicacao 1	Publicacao 1 mocked	aceitável
Publicacao 2	Publicacao 2 mocked	aceitável

Tabela 2.7: Teste de Unidade - Dados de Publicacao e Classes de Equivalência

atributo	caso de teste	resultado
Titulo	Service-oriented middleware for the Future Internet	aceitável
Veiculo	Journal	aceitável
Data 11	25/05/2011	aceitável
Contribuinte 1	Marco Aurelio Gerosa mocked	aceitável
Contribuinte 2	Valdemar Setzer mocked	aceitável

Tabela 2.8: Teste de Unidade - Dados de Usuario e Classes de Equivalência

atributo	caso de teste	resultado
Login	magerosa	aceitável
Senha	pressmanschach	aceitável
Avatar	File mocked	aceitável
Contribuinte	Marco Aurelio Gerosa mocked	aceitável

2.2.1.1 Classe Contribuinte

Testada e funcionando.

2.2.1.2 Classe Usuário

Testada e funcionando.

2.2.1.3 Classe Grupo

O caso mais frequente é o de grupos com muitas pessoas. A classe Grupo deveria ter então uma referência para uma lista de pessoas. Nesta iteração foi acrescentada uma referência para lista de filiações.

2.2.1.4 Classe Projeto

Na 1a iteração, dever-se-ia criar um atributo *Nome* para projeto, tendo em vista que usualmente os projetos são referenciados entre participantes pelo nome daquele. O tipo do atributo *Financiamento* foi alterado para *String*, pois pretende-se que este represente o nome da Agência de Fomento que financia o projeto em questão. Ambos os pontos foram corrigidos na 2a iteração.

2.2.1.5 Classe Publicacao

O caso mais frequente é o de publicações com muitos autores. A classe Publicacao deveria ter então uma referência para uma lista de autores, no lugar de um atributo Autor do tipo *String*. Isto foi corrigido na 2a iteração, acrescentando-se uma referência para lista de contribuintes.

2.2.1.6 Controllers

Até o final da 2a iteração, foram implementados testes de unidade para os seguintes controllers:

- DisciplinaController.java
- GrupoController.java
- ProjetoControllerTest.java

2.2.2 Comentários e Recomendações

2.2.2.1 Novos Testes de Unidade

Os métodos de todas as classes aqui testadas foram aprovados, uma vez que foram usados os mesmos tipos de dados para parâmetros reais que os parâmetros formais das assinaturas de cada método. No entanto, pelo método de Teste Não-baseado em Execução de Código(Seção 2.2) , denominado revisão do tipo *Walkthrough* [Schach, 2007], constatou-se que não há métodos de validação de tipos nestas classes.

Portanto, o código deve ser adaptado para passar por outros testes de unidade, em que os tipos dos parâmetros reais sejam diferentes dos parâmetros formais. Nesta 2a iteração foram iniciadas tarefas visando à implementação da validação de dados de entrada.

2.2.2.2 Ítens a serem testados

Ítens que ainda devem ser testadas todas as classes DAO e CONTROLLER. As classes DAOs devem ser testadas com testes de integração nas próximas iterações.

Referências Bibliográficas

Schach, S. R. (2007). *Engenharia de Software: Os Paradigmas Clássico Orientado a Objetos*. Mc Graw Hill. 3, 5, 9