

Bruno Loureiro Rezende

**Um Framework para a
Automação de Testes com
Linguagens de Especificação
Configuráveis**

DISSERTAÇÃO DE MESTRADO

**DEPARTAMENTO DE INFORMÁTICA
Programa de Pós-graduação em
Informática**

Rio de Janeiro, agosto de 2009



Bruno Loureiro Rezende

**Um Framework para a Automação de Testes com Linguagens
de Especificação Configuráveis**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para
obtenção do título de Mestre pelo Programa de Pós-
Graduação em Informática da PUC-Rio.

Orientador: Arndt von Staa

Rio de Janeiro
Agosto de 2009



Bruno Loureiro Rezende

**Um Framework para a Automação de
Testes com Linguagens de Especificação
Configuráveis**

Dissertação apresentada como requisito parcial para obtenção do título de Mestre pelo Programa de Pós-Graduação em Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Arndt von Staa
Orientador e Presidente
PUC-Rio

Carlos José Pereira de Lucena
PUC-Rio

Simone Diniz Junqueira Barbosa
Puc-Rio

Prof. José Eugenio Leal
Coordenador Setorial do Centro
Técnico Científico - PUC-Rio

Rio de Janeiro, 13 de agosto de 2009

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Bruno Loureiro Rezende

Graduou-se em Engenharia Computação na Pontifícia Universidade Católica do Rio de Janeiro (Brasil, Rio de Janeiro).

Ficha Catalográfica

Rezende, Bruno Loureiro

Um framework para a automação de testes com linguagens de especificação configuráveis / Bruno Loureiro Rezende ; orientador: Arndt von Staa. – 2009.

82 f. : il. ; 30 cm

Dissertação (Mestrado em Informática)– Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2009.

Inclui bibliografia

1. Informática – Teses. 2. Desenvolvimento dirigido por testes. 3. Desenvolvimento dirigido por comportamentos. 4. Engenharia de Software. I. Staa, Arndt von. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Para meus pais, Otto e Fernanda.

Agradecimentos

Aos meus pais, Otto e Fernanda, pelo carinho e pelos incontáveis esforços, tão necessários para que fosse possível tornar real mais esse objetivo.

Ao meu orientador Professor Arndt von Staa e ao “co-orientador” João Magalhães, atenciosos em cada momento de dificuldade, sempre contribuindo para a construção de um trabalho a altura do título.

À CAPES e à PUC-Rio, pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.

À Minds at Work, pela colaboração fundamental para a execução deste trabalho.

À minha namorada, Raquel, pelo carinho, paciência e cumplicidade, tão necessários para enfrentar esse desafio.

Aos meus amigos, em especial ao pessoal Minds e ex-Minds e ao pessoal do CEFET/RJ, pela alegria e pelo apoio, combustíveis para os momentos de cansaço e dúvida.

Ao Fred e à Juliana, pelo apoio e paciência durante a realização deste trabalho.

Aos professores que participaram da Comissão examinadora.

A todos os professores e funcionários do Departamento pelos ensinamentos e pela ajuda.

A todos os familiares que de uma forma ou de outra me estimularam ou me ajudaram.

Resumo

Rezende, Bruno Loureiro; Staa, Arndt von. **Um framework para a automação de testes com linguagens de especificação configuráveis**. Rio de Janeiro, 2009. 82p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Nesse trabalho foi criado um *framework* para a realização de testes automatizados de acordo com as práticas de desenvolvimento dirigido por testes, o qual pode ser estendido com novas linguagens de especificação, utilizando idéias de desenvolvimento dirigido por comportamentos. Espera-se que, ao utilizar esse *framework*, equipes de projetos de *software* possam especificar testes em uma linguagem mais adequada para o domínio de sua aplicação, através do suporte de uma ferramenta que permita tal nível de personalização. O *framework* foi instanciado com a criação de uma linguagem baseada em máquinas de estado, utilizada para a aplicação da ferramenta em um projeto real. O objetivo deste trabalho é avaliar os impactos da aplicação dessas técnicas, através da experiência de utilização da ferramenta no projeto. A motivação para esse trabalho surgiu da experiência no desenvolvimento de sistemas de controle, usualmente reativos e com requisitos de tempo real, nos quais muitas vezes a descrição de seus comportamentos é feita através de máquinas de estado, que possivelmente é a melhor linguagem para este domínio.

Palavras-chave

Engenharia de software; Desenvolvimento dirigido por testes; desenvolvimento dirigido por comportamentos.

Abstract

Rezende, Bruno Loureiro; Staa, Arndt von (Advisor). **A framework for test automation with configurable specification languages**. Rio de Janeiro, 2009. 82p. MSc. Dissertation - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

In this work we create a framework for automated testing according to test driven development practices that can be extended to support new specification languages, following ideas from behavior driven development. We expect that, by using this framework, software development teams will be able to specify tests in more proper languages for the application domain, with the support of a tool that allows such level of customization. The framework was instantiated with the creation of a language based on state machines, and used on a real-life project. The goal of this work is that software project teams become able to specify tests in the most fitting language for their application domain, through the support of a tool that makes possible such level of customization. The motivation for this work comes from the experience with the development of control systems, usually with real-time requirements, which often have their behaviors described by state machines, possibly the best language for this domain.

Keywords

Software Engineering; Test-Driven Development; Behaviour-Driven Development.

| | |
|--|----|
| 1 Introdução | 14 |
| 2 Revisão Bibliográfica | 17 |
| 2.1. Desenvolvimento Dirigido por Comportamentos | 17 |
| 2.1.1. Movendo o Foco Para os Comportamentos | 17 |
| 2.1.2. Utilizando Linguagens Mais Próximas do Domínio da Aplicação | 18 |
| 2.2. O Futuro das Ferramentas de Testes | 20 |
| 2.2.1. Objetivos Relacionados à Escrita de Testes | 20 |
| 2.2.2. Objetivos Relacionados à Leitura de Testes | 21 |
| 2.2.3. Objetivos Relacionados à Execução de Testes | 21 |
| 2.2.4. Como Alcançar os Objetivos? | 22 |
| 2.3. Frameworks de Testes | 23 |
| 2.3.1. JBehave | 23 |
| 2.3.2. RSpec | 25 |
| 2.3.3. Instinct | 26 |
| 3 Ferramenta Proposta | 28 |
| 3.1. Objetivos | 28 |
| 3.2. Motivação | 29 |
| 3.3. Arquitetura | 30 |
| 3.4. Instância de Exemplo da Ferramenta | 35 |
| 3.4.1. A Linguagem | 35 |
| 3.4.2. <i>Scripts</i> e Módulo Específico de Testes | 39 |
| 4 Uma Linguagem Baseada em Máquinas de Estado | 41 |
| 4.1. A Linguagem | 41 |
| 4.2. Instanciação do Módulo de Linguagem | 43 |
| 5 Tecnologias Utilizadas | 47 |
| 5.1. Inversão de Controle e Injeção de Dependências | 47 |
| 5.2. Reflexão Computacional | 48 |
| 6 Aplicação da Ferramenta | 49 |

| | |
|---|----|
| 6.1. Projeto Alvo | 49 |
| 6.1.1. A Arquitetura | 50 |
| 6.1.2. Características de Escalabilidade e Balanceamento de Carga | 53 |
| 6.2. Contexto da Aplicação da Ferramenta | 54 |
| 6.2.1. Oportunidade para a Aplicação | 54 |
| 6.2.2. O Processo de Desenvolvimento | 55 |
| 6.3. A Aplicação da Ferramenta ao Projeto | 56 |
| 6.3.1. Mudanças no Processo de Desenvolvimento | 57 |
| 6.3.2. Especificação dos Testes | 58 |
| 6.3.3. Resultados da Aplicação | 64 |
| | |
| 7 Discussão dos Resultados | 67 |
| 7.1. A Ferramenta de Testes | 67 |
| 7.2. A Linguagem Baseada em Máquinas de Estado | 68 |
| 7.3. A Aplicação ao Projeto | 69 |
| | |
| 8 Conclusão | 73 |
| | |
| 9 Trabalhos Futuros | 76 |
| | |
| Referências Bibliográficas | 80 |

Lista de figuras

| | |
|--|----|
| Figura 1: Exemplo de caso de teste descrevendo comportamento | 19 |
| Figura 2: Exemplo de script de entrada para a ferramenta JBehave. | 24 |
| Figura 3: Exemplo de módulo de testes para a ferramenta JBehave. | 24 |
| Figura 4: Exemplo de especificação de comportamento para a classe da figura 5. | 25 |
| Figura 5: Classe descrita pelo exemplo da figura 4. | 26 |
| Figura 6: Exemplo de classe do <i>framework Instinct</i> descrevendo comportamentos. | 27 |
| Figura 7: Diagrama de interação dos elementos da ferramenta | 30 |
| Figura 8: Arquitetura do <i>framework</i> | 31 |
| Figura 9: Diagrama de Classes da Ferramenta | 32 |
| Figura 10: Diagrama de seqüência descrevendo o funcionamento da ferramenta. | 34 |
| Figura 11: Diagrama da ferramenta após a instanciação. | 34 |
| Figura 12: Exemplo de arquivo de configuração | 35 |
| Figura 13: Exemplo de mudança nos comandos no arquivo de configuração | 37 |
| Figura 14: Exemplo de configuração com dois módulos de saída. | 38 |
| Figura 15: exemplo de <i>script</i> de testes | 39 |
| Figura 16: Assinaturas dos métodos do módulo específico de testes correspondente ao <i>script</i> da figura 15 | 39 |
| Figura 17: Máquina de estados descrevendo o controle de alarmes de velocidade | 42 |
| Figura 18: Arquivo de configuração para a instância com linguagem de máquinas de estado | 44 |
| Figura 19: Diagrama da ferramenta após a instanciação com a linguagem de máquinas de estado. | 45 |
| Figura 20: Exemplo de configuração dos módulos utilizando | |

| | |
|--|----|
| injeção de dependências | 47 |
| Figura 21: Arquitetura do sistema alvo | 49 |
| Figura 22: Exemplo de máquina de estados: Alarmes de alta velocidade | 58 |
| Figura 23: Exemplo de máquina de estados: Alarmes de atraso | 59 |
| Figura 24: Exemplos de script: Alarmes de alta velocidade | 60 |
| Figura 25: Exemplos de script: Alarmes de atraso | 61 |
| Figura 26: exemplo de relatório de execução apontando falha | 63 |
| Figura 27: exemplo de relatório de execução sem falhas | 64 |

Lista de tabelas

Tabela 1: Métricas relativas ao tamanho do sistema alvo.

66