



Henrique Feliciano Prange

**Uma Avaliação Empírica de um Ambiente Favorável para o
Desenvolvimento Dirigido por Testes**

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: Prof. Arndt von Staa

Rio de Janeiro
Março de 2007



Henrique Feliciano Prange

Uma Avaliação Empírica de um Ambiente Favorável para o Desenvolvimento Dirigido por Testes

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.

Prof. Arndt von Staa

Orientador
PUC-Rio

Prof. Renato Fontoura de Gusmão Cerqueira

Departamento de Informática - PUC-Rio

Prof. Simone Diniz Junqueira Barbosa

Departamento de Informática - PUC-Rio

Prof. Carlos José Pereira de Lucena

Departamento de Informática - PUC-Rio

Prof. José Eugenio Leal

Coordenador(a) Setorial do Centro Técnico Científico - PUC-Rio

Rio de Janeiro, 21 de março de 2007

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

Henrique Feliciano Prange

Graduou-se em Informática pela PUC-Rio em 2004. Participou do Laboratório de Engenharia de Software (LES) da PUC-Rio onde fez pesquisas focadas na área de Engenharia de Software. Desde julho de 2003 trabalha na empresa Moleque de Idéias onde teve a oportunidade de desenvolver projetos usando a tecnologia Java e implantar um ambiente favorável ao desenvolvimento dirigido por testes. É um aficionado por qualidade de software e está sempre em busca de formas práticas de se obter os melhores resultados nesta área.

Ficha Catalográfica

Prange, Henrique Feliciano

Uma avaliação empírica de um ambiente favorável para o desenvolvimento dirigido por testes / Henrique Feliciano Prange; orientador: Arndt Von Staa. – 2007.

117 f. : il. ; 29,7 cm

Dissertação (Mestrado em Informática)–Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2007.

Inclui bibliografia

1. Informática – Teses. 2. Desenvolvimento dirigido por testes. 3. Extreme programming. 4. Equipes pequenas. 5. Infra-estrutura. 6. Ferramentas. 7. Boas práticas. I. Staa, Arndt Von. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Para o meu pai, o maior incentivador e responsável por eu ter chegado até aqui.

Agradecimentos

Ao meu orientador Professor Arndt von Staa pelo estímulo e apoio para a realização deste trabalho.

Aos meus amigos Guga e Rodrigo Paes pelo suporte nos primeiros períodos do mestrado e por me oferecerem a oportunidade de ingressar nessa empreitada.

Aos professores que participaram da Comissão examinadora.

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

Aos meus amigos Nilton Lessa e Luís Eugênio, por sempre confiarem no meu trabalho.

A todos os meus amigos da empresa Moleque de Idéias que tornaram possível a realização desta pesquisa.

Ao meu amigo Pérez pelas piadas, os questionamentos e a revisão do trabalho.

Em especial, à minha mãe, à Amalia Ponce e à minha namorada, Bruma, que me deram atenção, carinho e amor em todos os momentos.

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

Resumo

Prange, Henrique Feliciano; Staa, Arndt von. **Uma Avaliação Empírica de um Ambiente Favorável para o Desenvolvimento Dirigido por Testes.** Rio de Janeiro, 2007. 117p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Test Driven Development (TDD) é uma das práticas de *eXtreme Programming* (XP) mais fáceis de entender e ao mesmo tempo uma das mais difíceis de executar. Para que o TDD seja usado apropriadamente, é preciso empregar práticas complementares, utilizar ferramentas adequadas e tomar algumas precauções para que seja feito de forma correta. Este trabalho de mestrado apresenta um estudo baseado na experiência real – realizada em uma pequena empresa – na qual foi elaborada uma infra-estrutura favorável ao desenvolvimento dirigido por testes. Quais as vantagens e desvantagens de cada uma das práticas? Como introduzir essas práticas no dia-a-dia de uma pequena empresa? Que tipo de infra-estrutura deve ser montada? Quais as ferramentas? Quanto tempo e qual o tipo de investimento necessário para a implantação dessas melhorias? Estas e outras perguntas são respondidas no decorrer do trabalho.

Palavras-chave

Desenvolvimento dirigido por testes; *eXtreme Programming*; equipes pequenas; infra-estrutura; ferramentas; boas práticas

Abstract

Prange, Henrique Feliciano; Staa, Arndt von. **An Empirical Evaluation of an Environment Designed for Test Driven Development.** Rio de Janeiro, 2007. 117p. MSc Dissertation - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Test Driven Development (TDD) is one of the eXtreme Programming's (XP) easiest practices to understand but at the same time difficult to implement. It is necessary to use complementary practices, appropriate tools, and follow carefully some rules for achieving good results. A real experiment creating an adequate environment for TDD was conducted in a small company. This study will show the results obtained. What are the advantages and disadvantages of each one of the practices? How to establish these practices in small company daily operations? What type of environment has to be built? Which tools? How much time and investment for implementing this kind of enhancement would be required? This work will present answers to these questions.

Keywords

Test Driven Development; eXtreme Programming; small teams; infrastructure; tools; best practices

Sumário

1 Introdução	11
1.1. Motivação	12
1.2. Estrutura da Dissertação	13
2 Medição e Acompanhamento	15
2.1. Ferramentas	19
2.1.1. JIRA	19
2.1.2. Trac	20
2.1.3. Bugzilla	21
2.1.4. GForge	21
2.1.5. Confluence	21
2.1.6. JDepend	22
2.1.7. JavaNCSS	24
2.1.8. Cobertura	25
2.1.9. Clover	25
2.2. Precauções	26
3 Gerenciamento	28
3.1. <i>Build</i> Automatizado	28
3.2. Ferramentas	30
3.2.1. Ant	31
3.2.2. Maven	32
3.3. Precauções	35
4 Integração	38
4.1. <i>Continuous Integration</i>	38
4.2. Coletividade do Código	40
4.3. Padrões de Desenvolvimento	41
4.4. Ferramentas	41
4.4.1. Cruise Control	42

4.4.2. Continuum	42
4.4.3. CVS	42
4.4.4. Subversion	43
4.4.5. Checkstyle	43
4.4.6. Jalopy	43
4.5. Precauções	44
5 Organização	48
5.1. <i>Pair Programming</i>	48
5.2. Infra-estrutura	52
5.3. Precauções	53
6 <i>Test Driven Development</i>	57
6.1. Ciclo de Desenvolvimento	63
6.2. Qualidades de um Bom Caso de Teste	68
6.3. Critérios para Seleção de Casos de Teste	69
6.4. <i>Refactoring</i>	71
6.5. <i>Mock Objects</i>	74
6.6. Ferramentas	75
6.6.1. JUnit	76
6.6.2. TestNG	76
6.6.3. Eclipse	77
6.6.4. IntelliJ IDEA	78
6.6.5. EasyMock	78
6.7. Extensões do JUnit	79
6.7.1. DBUnit	79
6.7.2. XMLUnit	79
6.7.3. HttpUnit	80
6.8. Precauções	80
7 Avaliação Prática	87
7.1. Planejamento	88
7.2. Proposta	89
7.3. Preparação do Ambiente	91

7.4. Treinamento Sobre TDD	93
7.5. Manutenção	96
7.6. Dificuldades	97
7.6.1. Ambiente de Desenvolvimento	97
7.6.2. Tempo	99
7.6.3. Métricas	99
7.6.4. Recursos	100
7.6.5. Testar Aplicações Específicas	101
7.6.6. Código Legado	102
7.6.7. Constante Evolução	103
7.7. Avaliação e Críticas	104
7.7.1. Dados Coletados	105
7.7.2. Vantagens	106
7.7.3. Críticas	108
7.7.4. Resultado	110
8 Conclusão	112
8.1. Trabalhos Futuros	113
9 Referências Bibliográficas	115