



Leandra Mara da Silva

**Uma Abordagem Sensível à História para
Detecção de Anomalias de Código**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do
grau de Mestre pelo Programa de Pós-graduação em Informática
do Departamento de Informática da PUC-Rio

Orientador : Prof. Carlos José Pereira de Lucena
Co-Orientador: Prof. Alessandro Fabricio Garcia

Rio de Janeiro
Agosto de 2010



Leandra Mara da Silva

**Uma Abordagem Sensível à História para
Detecção de Anomalias de Código**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática do Departamento de Informática do Centro Técnico Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Carlos José Pereira de Lucena

Orientador

Departamento de Informática — PUC-Rio

Prof. Alessandro Fabricio Garcia

Co-Orientador

Departamento de Informática — PUC-Rio

Prof. Arndt von Staa

Departamento de Informática — PUC-Rio

Prof. Soeli Teresinha Fiorini

Pesquisadora — PUC-Rio

Prof. José Eugênio Leal

Coordenador Setorial do Centro Técnico Científico — PUC-Rio

Rio de Janeiro, 23 de Agosto de 2010

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

Leandra Mara da Silva

Leandra Mara da Silva recebeu da Universidade Federal Fluminense (UFF) o título de bacharel em Ciência da Computação em 2007. Atualmente, participa do grupo de pesquisa OPUS que funciona junto ao Laboratório de Engenharia de Software da PUC-Rio (LES/PUC-Rio). Ela também atua como integrante do TecGraf, contribuindo no desenvolvimento do sistema v3o2 que auxilia na análise de dados sísmicos pertencentes a Petrobrás. Seus trabalhos estão relacionados a temas como manutenção de sistemas, métricas, avaliação de código, qualidade de sistemas e engenharia de software empírica.

Ficha Catalográfica

Silva, L. M.

Uma Abordagem Sensível à História para Detecção de Anomalias de Código / Leandra Mara da Silva; orientador: Carlos José Pereira de Lucena; co-orientador: Alessandro Fabricio Garcia. — Rio de Janeiro : PUC-Rio, Departamento de Informática, 2010.

v., 120 f. il. ; 29,7 cm

1. Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Inclui referências bibliográficas.

1. Informática – Tese. 2. Manutenção de Sistemas. 3. Anomalias de Modularidade. 4. Estratégias de Detecção. 5. Métricas. 6. Métricas Sensíveis à História. 7. Engenharia de Software Empírica. I. Lucena, C. J. P.. II. Garcia, A. F.. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. IV. Título.

À minha família.

Agradecimentos

Agradeço a Deus pela sua constante presença em minha vida. Agradeço pelas oportunidades de estudo, pelo conhecimento e amadurecimento adquiridos ao longo desses anos. Agradeço pelas conquistas obtidas, pelas pessoas presentes na minha vida e por este título.

Agradeço, cheia de emoção, à minha mãe, irmãs, irmão e sobrinhos. Agradeço por todo apoio e incentivo, bem como por toda admiração demonstrada a mim. Agradeço também por serem sempre tão compreensivos comigo, mesmo quando precisei ficar tantos meses sem visitá-los por conta das atividades acadêmicas.

Agradecimentos especiais a Gustavo pela maneira única que me ama e que se preocupa comigo. Também agradeço por seu apoio e presença em cada projeto que tenho assumido na minha vida e por sua paciência nos meus momentos de incerteza. Agradeço ainda por sua compreensão quando muitas vezes durante o mestrado eu não consegui lhe dar a devida atenção. Agradeço também por suas dicas e contribuições no desenvolvimento da ferramenta proposta neste trabalho.

Minhas considerações também aos amigos que constantemente me alegraram e me renovaram nos momentos de cansaço. O carinho, as brincadeiras e os sorrisos dessas pessoas sempre me trouxeram alegria. Inúmeros amigos poderiam ser citados, mas eu não teria espaço suficiente para isso. Dessa forma, como representante dos amigos de Niterói, agradeço à Luca e, como representante dos amigos de Cachoeiro, agradeço à Silvania.

Também conheci pessoas especiais na PUC, verdadeiros parceiros e companheiros de pesquisa, pessoas com quem aprendi muito. Agradeço aos colegas do LES, do Grupo de Pesquisa OPUS e do TecGraf. Agradeço à Soeli que me incentivou na realização dos primeiros estudos sobre medição de código. Ao Alessandro e ao Chico pela amizade, além de suas importantes contribuições nos trabalhos publicados. À Isela, por sua amizade e conhecimento compartilhado na área de métricas.

A profissionais como Cassino e Leonardo do TecGraf que me proporcionaram a possibilidade de aplicar os primeiros conhecimentos adquiridos sobre métricas em sistemas desenvolvidos para a Petrobrás. Ao Gustavo Carvalho, Daniele Coutinho e Vera Menezes pela atenção a mim concedida em todos os momentos em que deles precisei.

Também não poderia deixar de agradecer ao Glauco Carneiro (UFBA), Cláudio “Baiano” (UFBA) e Eduardo Figueiredo (UFMG). Por termos alcançado, juntamente com o Alessandro Garcia e o Manoel Mendonça, o prêmio de melhor artigo do CBSoft-SBES 2010.

Agradeço também a alguns parceiros de estudo nas disciplinas cursadas. Dentre eles, Hildebrando Trannin, Kelly Leal, Katia Canepa, Elisabeth Suescun, Amadeu Barbosa, Márcio Santos, Luana Lachtermacher, Sérgio Cerqueira e outros.

Agradeço ao departamento de informática e professores, em especial à secretária Teresa e ao professor Arndt. Ao professor Carlos Lucena, meu orientador, pelo seu apoio em todas as minhas propostas de estudo.

Finalmente, ficam registradas as minhas considerações ao Alessandro, meu co-orientador. Revelo minha admiração pelo seu profissionalismo e destreza no relacionamento com seus orientados. Agradeço pela relação de confiança e amizade que estabelecemos. Agradeço por ter me concedido liberdade para crescer, mas ao mesmo tempo não ter deixado de se fazer presente nas minhas principais experiências vivenciadas durante este mestrado.

Agradeço também à CAPPES, à PUC-Rio e à Fundação Padre Leonel Franca pelo auxílio financeiro ao longo desses anos.

Resumo

Silva, L. M.; Lucena, C. J. P.; Garcia, A. F.. **Uma Abordagem Sensível à História para Detecção de Anomalias de Código.**

Rio de Janeiro, 2010. 120p. Dissertação de Mestrado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A modularização do código pode influenciar diretamente a manutenibilidade de sistemas de software. Por isso, pesquisadores têm proposto mecanismos que visam contribuir com a identificação de potenciais anomalias de modularidade no código. Nesse contexto, um mecanismo baseado em métricas que vêm sendo bastante divulgado são as estratégias de detecção. Comumente, elas são compostas por métricas que consideram apenas propriedades de versões isoladas dos sistemas. Entretanto, estudos recentes relatam que tal abordagem têm se apresentado contraproducente. Nossa pesquisa está relacionada à investigação dos possíveis benefícios de se considerar informações sobre a evolução do código na detecção de anomalias. Nesse contexto, este trabalho propõe um conjunto de métricas e de estratégias de detecção que consideram propriedades históricas do código em evolução. Além disso, uma ferramenta de medição e avaliação para dar suporte à abordagem também foi desenvolvida. Essa ferramenta permite a especificação declarativa de diferentes estratégias de detecção através de uma linguagem específica de domínio. Tal fato atende a necessidades particulares de desenvolvedores na configuração de estratégias e disponibiliza a pesquisadores um ambiente propício à experimentação de estratégias de detecção. Ainda como contribuição desta pesquisa, é apresentada uma avaliação das estratégias em termos de precisão e revocação em dois sistemas de domínio diferentes. Resultados de estratégias convencionais e sensíveis à história são comparados em detecções de anomalias clássicas, tais como *God Class*, *Divergent Change* e *Shotgun Surgery* em um total de 16 versões desses sistemas. Como resultado desse estudo observou-se que a utilização de informações relacionadas à evolução do código pode trazer importantes contribuições à detecção de anomalias de código.

Palavras-chave

Manutenção de Sistemas; Anomalias de Modularidade; Estratégias de Detecção; Métricas; Métricas Sensíveis à História; Engenharia de Software Empírica;

Abstract

Silva, L. M.; Lucena, C. J. P.; Garcia, A. F.. **A History Sensitive Approach to Code Anomaly Detection.** Rio de Janeiro, 2010. 120p. MsC Dissertation — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The modularization of the code can directly influence the software maintainability. Therefore, researchers have proposed mechanisms to contribute to the identification of potential modularity anomalies in source code. In this context, a mechanism based on metrics that have been widespread are the detection strategies. Commonly, they are based on metrics that consider only properties of isolated versions of the systems. However, recent studies have reported that these strategies have been considered counter-productive. Our research is related to the investigation of the possible benefits of considering information about the code evolution to detect anomalies. In this context, this paper proposes a set of metrics and detection strategies that consider historic properties of the code evolution. Furthermore, a measuring and assessing tool to support the approach was also developed. This tool allows the declarative specification of different detection strategies through a domain-specific language. This fact meets the particular needs of developers in setting strategies and dispose to researchers an opportune environment to detection strategies experimentations. Also as a contribution of this research is presented an evaluation of strategies in terms of precision and recall in two systems of different domains. Results of conventional strategies and of history-sensitive strategies are compared in detections of classical modularity flaws, such as God Class, Divergent Change and Shotgun Surgery in a total of 16 versions of these systems. Results of this study showed that the use of information related to the code evolution can provide important contributions to detect design flaws in code.

Keywords

Software Maintenance; Code Smells; Detection Strategies; Metrics; History-Sensitive Metrics; Empirical Software Engineering;

Sumário

1	Introdução	13
1.1	Contextualização	13
1.2	O Problema	14
1.3	Limitações de Trabalhos Relacionados	16
1.4	Visão Geral da Solução	17
1.5	Estrutura do Texto	20
2	Fundamentação	22
2.1	Manutenção e Evolução de Sistemas	22
2.2	Refatoração e Anomalias de Modularidade de Código	25
2.3	Avaliação Sensível à História	26
3	Estado da Arte e Trabalhos Relacionados	27
3.1	Medição de Código	27
3.2	Estratégias de Detecção	30
3.3	Ferramentas de Suporte a Estratégias de Detecção	37
3.4	Uma técnica chamada Detecção Experta	42
3.5	Detecção Baseada em Visualização de Software	43
4	Métricas Sensíveis à História	47
4.1	Métricas Independentes	47
4.2	Métricas Dependentes	48
5	Estratégias de Detecção Sensíveis à História	59
5.1	A Questão dos Valores Limites	60
5.2	Detecções Simples	61
5.3	Detecções de Anomalias de Modularidade	63
6	Hist-Inspect: A Ferramenta de Medição e Avaliação	70
6.1	Levantamento das Necessidades	70
6.2	Descrição das Funcionalidades	71
6.3	Visão Geral da Arquitetura e Funcionamento	76
7	Avaliação	79
7.1	Procedimentos	80
7.2	Resultados e Discussões	87
8	Considerações Finais	101
8.1	Contribuições	102
8.2	Trabalhos Futuros	104
	Referências Bibliográficas	107
A	Catálogo de Métricas	114

Lista de figuras

3.1	Processos de filtragem e composição pertencentes a estratégias de detecção (Figura adaptada de (Marinescu 2004)).	32
3.2	Exemplo de <i>God Class</i> : classe <i>ImageAccessor</i> . Os valore de métricas apresentados à direita foram obtidos com base na utilização da ferramenta <i>Together</i> (<i>Together 2009</i>).	35
3.3	Notação para estratégia de detecção segundo (Lanza et al., 2006)	37
3.4	Ferramenta <i>Together</i> : suporte a estratégias de detecção	39
3.5	<i>inCode</i> : detecção de anomalias e sugestões de refatoração	40
3.6	<i>iPlasma</i> e <i>inFusion</i> : tela de configuração de estratégias	40
3.7	Cartão de regras da abordagem DETEX: especificação de anomalias em alto nível por meio de linguagem específica de domínio (DSL)	43
3.8	Visão de dependências entre os módulos – ferramenta <i>Source Miner</i>	45
3.9	Força de acoplamento entre módulos – ferramenta <i>Source Miner</i>	45
3.10	Visão Polimétrica – ferramenta <i>Source Miner</i>	46
5.1	Estratégia de detecção convencional para <i>God Class</i>	64
5.2	Estratégia sensível à história para <i>God Class</i>	65
5.3	Estratégia de detecção convencional para <i>Shotgun Surgery</i>	66
5.4	Estratégia sensível à história para <i>Shotgun Surgery</i>	67
5.5	Estratégia de detecção sensível à história para <i>Divergent Change</i>	68
6.1	Gráficos de evolução através da biblioteca JFreeChart	72
6.2	Apresentação de métricas sensíveis à história	73
6.3	Relatório HTML das anomalias detectadas	74
6.4	Exemplos de especificação de estratégias e respectivas anomalias	75
6.5	Funcionamento e elementos principais da arquitetura da Hist-Inspect	77

Lista de tabelas

3.1	Ferramentas de suporte a métricas de código	31
3.2	Comparação entre ferramentas de suporte a estratégias de detecção	42
4.1	Exemplo de dados para medição sensível à sistória	47
5.1	Exemplos de estratégias sensíveis à história com uma única métrica: estratégias simples	62
7.1	Principais características das aplicações do estudo	85
7.2	Detecção sensível à história no Mobile Media	88
7.3	Detecção convencional vs. sensível à história de GC no Mobile Media	90
7.4	Detecção convencional vs. sensível à história de SS no Mobile Media	92
7.5	Detecção baseada em múltiplas perspectivas visuais vs. sensível à história de GC no Mobile Media	93
7.6	Detecção baseada em múltiplas perspectivas visuais vs. sensível à história de DC no Mobile Media	94
7.7	Detecção convencional vs. sensível à história de GC no Health Watcher	96
7.8	Detecção sensível à história de DC no Health Watcher	96
B.1	Oráculos para o Mobile Media. Anomalias de classes: <i>God Class</i> <i>(GC)</i> , <i>Divergent Change</i> (DC) e <i>Shotgun Surgery</i> (SS). Células em tons de cinza indicam que o módulo não existe nas versões especificadas ou que existe com um outro nome (refatoração “rename”).	119
B.2	Oráculo para o Mobile Media. Anomalia de método: <i>Feature Envy</i> (FE). Células em tons de cinza indicam que o módulo não existe nas versões especificadas ou que existe com um outro nome (refatoração “rename”).	119
B.3	Oráculos para o Health Watcher. Anomalias de classes: <i>God Class</i> <i>(GC)</i> e <i>Divergent Change</i> (DC). Células em tons de cinza indicam que o módulo não existe nas versões especificadas ou que existe com um outro nome (refatoração “rename”).	120