PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO

Isela Macía Bertrán

Avaliação da Qualidade de Software com Base em Modelos UML

Dissertação de Mestrado

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

Orientadores: Prof. Arndt von Staa
Cláudio Nogueira Sant'Anna

Rio de Janeiro

Março de 2009

Isela Macía Bertrán

Avaliação da Qualidade de Software com Base em Modelos UML

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
Departamento de Informática - PUC-Rio

Prof. Carlos José Pereira de Lucena
Departamento de Informática - PUC-Rio

Prof.ª Simone Diniz Junqueira Barbosa
Departamento de Informática - PUC-Rio

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

Rio de Janeiro, 25 de Março de 2009

Isela Macía Bertrán

Graduou-se em Ciência da Computação na Universidade de Havana (UH) em 2006. Atuou como desenvolvedor de software até 2007.

Ficha Catalográfica

Macía Bertrán, Isela

Avaliação da qualidade de software com base em modelos UML / Isela Macía Bertrán ; orientadores: Arndt Von Staa, Cláudio Nogueira Sant'Anna. – 2009.

131 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. Design de software orientado a objetos. 3. Diagramas de classe. 4. Métricas de software. 5. Estratégias de detecção de bad smells. 6. Modelos da qualidade. 7. Engenharia de software experimental. I. Staa, Arndt von. II. Sant'Anna, Cláudio Nogueira. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. IV. Título.

CDD: 004

Aos meus pais

Agradecimentos

Meus agradecimentos mais especiais vão para as pessoas que julgo mais responsáveis por tudo que alcancei até aqui: meus maravilhosos pais. Muito obrigada pelo amor, confiança, educação e uma vida inteira de dedicação que, sem dúvida, tem sido essenciais na minha formação e luta pelos meus objetivos. Ao resto de minha família, pelo carinho, apoio e momentos de alegria sempre importantes, ainda mais na distância: à minha irmã, avô, tias, tios e primos.

Ilimitados agradecimentos a Julio pela maneira única que me ama e compreende dia a dia. Por sua paciência e incentivo em meus momentos de dúvidas e temores. Sou muito grata e orgulhosa de tê-lo ao meu lado.

Agradeço especialmente ao meu orientador, professor Arndt von Staa pois, sem seu apoio, este trabalho não teria sido possível. Ele me deu liberdade para escolher o tema deste trabalho, indicando-me sempre o caminho correto. Agradeço por ter-me mostrado o mundo das métricas e da qualidade de software.

Ao meu co-orientador, Cláudio Sant'Anna, pela orientação, dicas, parceria e dedicação. Você foi fundamental para o meu trabalho, sempre fornecendo comentários e sugestões para a melhoria da qualidade do conteúdo e da redação de minha dissertação.

Aos colegas do LES e do departamento de informática, pelos momentos compartilhados e aprendizado dividido. Agradeço especialmente a Andrew por ter-me facilitado parte dos sistemas analisados nesta dissertação e a Camila por sua ajuda na confecção de minhas primeiras apresentações.

Aos amigos Francisco, Frank, Ilaria, Evelin e Dunieskys pelo companheirismo e momentos que passamos juntos.

A Merlin, Yuanersi, Yahima, Ivan, grandes amigos de longa data, e cuja amizade sempre se mantive forte apesar da distância.

Agradeço, finalmente, a CAPES pelo apoio financeiro.

Resumo

Macía, Isela. **Avaliação da Qualidade de Software com Base em Modelos UML**. Rio de Janeiro, 2009. 131p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Um dos objetivos da engenharia de software é a construção de software com um nível de qualidade elevado com o menor custo e no menor tempo possível. Nesse contexto, muitas técnicas para o controle da qualidade de *design* de software têm sido definidas. Além disso, mecanismos baseados em métricas para a detecção de problemas também têm sido definidos. A maioria dessas técnicas e mecanismos foca a análise do código fonte. Porém, para reduzir retrabalho inútil, é importante utilizar técnicas de análise da qualidade capazes de detectar problemas de *design* já desde os modelos dos sistemas. Esta dissertação propõe: (i) um conjunto de estratégias de detecção para identificar, em modelos UML, problemas de *design* específicos e recorrentes na literatura: *Long Parameter List*, *God Class*, *Data Class*, *Shotgun Surgery*, *Misplaced Class* e *God Package*, e (ii) a utilização do modelo da qualidade QMOOD para avaliar *design* de software a partir de seus diagramas de classes. Para automatizar a aplicação destes mecanismos foi implementada uma ferramenta: a QCDTool. Os mecanismos desenvolvidos foram avaliados no contexto de dois estudos experimentais. O primeiro estudo avaliou a acurácia, precisão e *recall* das estratégias de detecção propostas. Esse estudo mostrou os benefícios e desvantagens da aplicação, em modelos, das estratégias de detecção propostas. O segundo estudo avaliou a utilidade da aplicação do modelo da qualidade QMOOD em diagramas UML. Esse estudo mostrou que foi possível identificar, em diagramas de classes, variações das propriedades de *design*, e, conseqüentemente, dos atributos da qualidade nos sistemas analisados.

Palavras-chave

Design de software orientado a objetos, diagramas de classe, métricas de software, estratégias de detecção de *bad smells*, modelos da qualidade, engenharia de software experimental.

Abstract

Macía, Isela. **Evaluation of Software Quality Based on UML Models**. Rio de Janeiro, 2009. 131p. MSc. Dissertation - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

One of the goals of software engineering is the development of high quality software at a small cost an in a short period of time. In this context, several techniques have been defined for controlling the quality of software designs. Furthermore, many metrics-based mechanisms have been defined for detecting software design flaws. Most of these mechanisms and techniques focus on analyzing the source code. However, in order to reduce unnecessary rework it is important to use quality analysis techniques that allow the detection of design flaws earlier in the development cycle. We believe that these techniques should analyze design flaws starting from software models. This dissertation proposes: (i) a set of strategies to detect, in UML models, specific and recurrent design problems: Long Parameter List, God Class, Data Class, Shotgun Surgery, Misplaced Class and God Package; (ii) and the use of QMOOD quality model to analyze class diagrams. To automate the application of these mechanisms we implemented a tool: the QCDTool. The detection strategies and QMOOD model were evaluated in the context of two experimental studies. The first study analyzed the accuracy, precision and recall of the proposed detection strategies. The second study analyzed the utility of use QMOOD quality model in the class diagrams. The results of the first study have shown the benefits and drawbacks of the application in class diagrams of some of the proposed detection strategies. The second study shows that it was possible to identify, based on class diagrams, variations of the design properties and consequently, of the quality attributes in the analyzed systems.

Keywords

Object oriented software design, class diagrams, software metrics, bad smells detection strategies, quality models, experimental software engineering.

Sumário

Lista de figuras

Lista de tabelas

Lista de Acrônimos e Abreviações

ATFD - Access To Foreign Data.

CA - Changing Attributes.

CAM - Coesion Among Methods.

CC - Client Class.

CL - Class Locality.

CM - Changing Methods.

DD - Depency Dispersion.

DTD - Document Type Definition.

GQM - Goal-Question-Metric.

GUI - Graphical User Interface.

MOF - MetaObject Facility.

NOAM - Number Of Access Methods.

NOCC - Number Of Client Class.

NOCP - Number Of Client Packages.

NOED - Number Of External Dependencies.

NOP - Number Of Parameters.

NOPA - Number Of Public Attributes.

OMG - Object Management Group.

OO - Orientado a Objetos.

PC - Package Cohesion.

PS - Package Size.

QMOOD - Quality Model Object Oriented Design.

SAX - Simple API XML.

TCC - Tight Class Cohesion.

UML - Unified Modeling Language.

WMC - Weighted Method per Class.

WOC - Weight Of Class.

XMI - XML Metadata Interchange.

XML - Extensible Markup Language.