

Características Subjetivas e Métodos Objetivos para Avaliação do Código-Fonte

7 de junho de 2010

- **Alunos:**
Lucianna Thomaz Almeida
João Machini de Miranda
- **Supervisor:**
Prof. Dr. Fabio Kon
- **Colaboradores:**
Paulo Meirelles
Vinicius Kiwi Daros

1 Tema

Baseado nas ideias e conceitos propostos por renomados especialistas em desenvolvimento de software orientado a objetos quanto ao código fonte, este trabalho tem o objetivo de estabelecer uma relação direta entre esses aspectos subjetivos estudados e seu mapeamento para métricas de código fonte.

Para cada conceito levantado, tentaremos apresentar detalhadamente uma ou mais métricas para servirem como indicativo. Por fim, o trabalho apresentará ferramentas para auxiliar na extração desses valores e que possibilitarão um acompanhamento do código ao longo do desenvolvimento do software de forma automatizada.

2 Resumo da monografia

Ao produzir software, os desenvolvedores não podem ficar satisfeitos com um código que simplesmente faz o trabalho que deve ser feito. É preciso considerar que será necessário manter a aplicação, fazer mudanças à medida em que os requisitos se alterarem e que outros terão que usar e aprimorar o código.

Diante disso, este trabalho aborda ideias e conceitos elaborados por especialistas no desenvolvimento de software orientado a objetos, buscando um maior entendimento de boas soluções, práticas e cuidados quanto ao código-fonte. Segundo Robert Martin, um programador deveria sempre fazer com que seu código seja uma composição de instruções e abstrações que possam ser facilmente entendidas, uma vez que gastamos a maior parte do tempo lendo-o para incluir funcionalidades e corrigir falhas.

Os seguintes livros são base para os nossos estudos: *Clean Code* de Robert Martin, *Code Complete* de Steve McConnell, *Implementation Patterns* de Kent Beck e *Beautiful Code* que contém opiniões de muitos desenvolvedores compiladas por Andy Oram e Greg Wilson. Todos possuem diversos aspectos que ajudam a identificar o que é um bom código como escolhas de bons nomes, falta de duplicações, organização e simplicidade. Esta monografia conterà uma reunião

de todos esses, explicitando sua relevância e aplicação. Adicionalmente, conterà uma visão geral sistematizada de entrevistas com o Prof. Fabio Kon, Prof. Alan Durhan e com Joseph Yoder quanto aos aspectos que julgam importantes.

Além de conter o levantamento teórico acima citado, esse trabalho também apresentará um mapeamento entre os conceitos encontrados na literatura em uma avaliação objetiva através de métricas de código-fonte. A ideia é possibilitar aos programadores fazer uma avaliação do software usando dados que espelham o que os especialistas recomendam, tornando possível um acompanhamento do projeto.

A grande vantagem de usar métricas de código-fonte é que elas podem ser bons indicativos para os critérios subjetivos, além de serem extraídas computacionalmente com ferramentas capazes de interpretar o código-fonte e calculá-las. Diante disso, apresentaremos o Projeto Mezero que contém ferramentas com intuito de automatizar a extração dessas métricas.

3 Objetivos

O principal objetivo deste trabalho é auxiliar os programadores a melhorarem a qualidade de seus códigos-fonte através de um mapeamento objetivo do conhecimento de grandes especialistas em desenvolvimento de software. Para tanto, agruparemos as principais ideias e conceitos da literatura relacionada e proporemos um conjunto de métricas para facilitar uma avaliação mais profunda.

A proposta surgiu da nossa própria necessidade de entender o que é um bom código orientado a objetos. Gostaríamos de desenvolver um trabalho para nos aprofundarmos no tema, além de possivelmente auxiliar outras pessoas a pensarem nos seus códigos. Com um mapeamento das ideias subjetivas para medidas objetivas, podemos facilitar o processo de avaliação e dar uma base de comparação, tanto entre projetos e suas versões, quanto com os conceitos propostos.

Uma vez com um modelo teórico, gostaríamos de obter as métricas de maneira automatizada. Sendo assim, também temos como objetivo desenvolver e apresentar ferramentas que podem calcular tais valores e que possibilitem um acompanhamento do projeto.

4 Atividades já realizadas

Participamos de um grupo de pesquisa orientado pelo Prof. Fabio Kon, no qual colaboramos com o desenvolvimento de ferramentas capazes de calcular métricas de código-fonte. Já estamos trabalhando em cima de uma ferramenta livre, chamada Analizo, desde o segundo semestre de 2009, que, atualmente, analisa projetos nas linguagens Java, C e C++. Além disso, estamos desenvolvendo uma aplicação Web na Disciplina de Laboratório de Programação Extrema para possibilitar o acompanhamento online das métricas de um projeto. Essa aplicação chama Mezero e conta com a colaboração dos doutorandos Paulo Meirelles do IME e Antonio Terceiro da UFBA.

Dito isso, já temos algum conhecimento da área em que vamos trabalhar, pois já tivemos contato com diversas métricas e alguns artigos relacionados.

5 Cronograma

6 Estrutura esperada da monografia

1. Introdução

Nesta seção discutiremos por que temos que nos preocupar com o código-fonte de nossas aplicações. Usaremos argumentos bastante conhecidos pela Engenharia de Software e abordaremos aspectos negativos da falta de preocupação com o código. Além disso, falaremos um pouco sobre a bibliografia usada e os aspectos que compõe o trabalho.

Atividade / Mês	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov
Desenvolvimento de Ferramentas do Mezuro	X	X	X	X	X				
Estudo da Literatura		X		X	X	X			
Elaboração do Mapeamento					X	X	X		
Seminário sobre Beleza de Código						X			
Entrevistas						X	X		
Monografia						X	X	X	X
Apresentação									X

2. Critérios Subjetivos

Apresentaremos uma visão geral dos principais conceitos e idéias da literatura feita por grandes especialistas na área. Faremos um apanhado geral dos pontos em que os autores concordam, mostrando como podemos melhorar nosso código e porque determinados aspectos devem ser evitados.

Nesta sessão, também apresentaremos uma visão geral das entrevistas que realizaremos.

3. Métricas Objetivas

Apresentaremos uma visão geral sobre métricas de código-fonte e como podem servir como indicativos para identificar atributos importantes do código-fonte. A sessão também conterá uma discussão sobre como podemos mapear os conceitos da seção anterior em métricas objetivas. A ideia é apresentar quais métricas podem ser utilizadas, como obtê-las e como cobrem os aspectos subjetivos. Ao final, mostraremos um pouco sobre a opinião dos entrevistados sobre desse mapeamento.

4. Projeto Mezuro: ferramentas para acompanhamento de métricas de código-fonte

Conterá uma apresentação das ferramentas desenvolvidas no contexto do Projeto Mezuro capazes de calcular as métricas e prover o acompanhamento de um projeto. Além disso, mostraremos como as ferramentas podem ser utilizadas para automatizar o mapeamento proposto.

5. Conclusão

6. Bibliografia