

# Kalibro Metrics: a multi-repository and multi-language source code analysis tool

Carlos Morais<sup>1</sup>, Paulo Meirelles<sup>1</sup>, Vinícius Daros<sup>1</sup>, Fabio Kon<sup>1</sup>

<sup>1</sup>Department of Computer Science  
Institute of Mathematics and Statistics  
University of São Paulo, Brazil  
{morais, paulormm, vkdaros, fabio.kon}@ime.usp.br

Carlos Santos Jr.<sup>1,2</sup>

<sup>2</sup>Horizon Institute  
University of Nottingham, United Kingdom  
carlos.denner@nottingham.ac.uk

## Resumo—

## Index Terms—

### I. INTRODUÇÃO

A qualidade interna é um dos fatores de sucesso de projetos de software, pois corresponde a aspectos primordiais do software tais como manutenibilidade e segurança. Softwares com boa qualidade interna proporcionam maior produtividade uma vez que possibilitam a criação de mais testes automatizados, são mais compreensíveis, reduzem o risco de bugs e facilitam as modificações e evoluções no código. Além disso, um estudo do ICAT/NIST de 2005 já apontava que 80% das vulnerabilidades exploráveis estão ligadas a má codificação.

Portanto, o Engenheiro de Software é um dos responsáveis por esse sucesso, uma vez que deve reunir um conjunto de habilidades e conhecimentos que o permitam aplicar práticas, técnicas e ferramentas para a criação de softwares seguros e com bom design. Nesse sentido, a medição pode ser utilizada como um processo de apoio ao acompanhamento da segurança e qualidade, através do estabelecimento de metas e indicadores que indiquem oportunidades de melhorias observáveis do produto. Em um cenário otimista, os próprios Engenheiros de Software podem adotar como prática a medição do código-fonte para auxiliar as tomadas de decisões, ou até mesmo para avaliação do código inserido ou da aplicação de refatorações. Entretanto, uma grande quantidade de métricas, coletas manuais e poucos recursos de visualização são fatores que acabam por desmotivar o uso dessas para o monitoramento do código. Além disso, a compreensão do significado de valores obtidos através de métricas não é uma tarefa trivial, demandando um grande esforço de interpretação necessárias para a tomada de decisão efetiva sobre o projeto de software.

Portanto, neste artigo iremos mostrar o trabalho de conclusão de curso de Engenharia de Software que está sendo desenvolvido que explora a utilização de métricas para o monitoramento de código-fonte para compreender e estabelecer possíveis relações existentes entre as mesmas no que diz respeito a vulnerabilidades e qualidade de software. No trabalho, partir do estabelecimento de cenários, esperamos identificar oportunidades de utilização de métricas na melhoria contínua do desenvolvimento e, conseqüentemente, na qualidade interna

do produto. Com o objetivo de facilitar a interpretação e evitar possíveis equívocos, que são baseados em análises errôneas sobre métricas isoladas, correlações inexistentes ou até mesmo a escolha de métricas inadequadas cujo problemas são discutidos em [1], estes cenários serão compostos a partir da análise de relação e correlação entre métricas. Tal correlação buscará evidenciar boas e más características de bom design de um projeto que impactam em vulnerabilidades no sistema. Para auxiliar no monitoramento e na tomada de decisão, será explorado o uso de plataforma de monitoramento de código-fonte Mezuro e um ambiente de DWing. Dessa forma, será observado o uso dessas duas soluções e suas contribuições para a melhoria do desenvolvimento e qualidade do produto. Também mostraremos a metodologia que utilizamos para a criação deste trabalho, assim como os aspectos que são importantes para nossa formação ao abordar este assunto. (MELHORAR ESSE FINAL.. talvez encaixar em outro lugar...)

O restante deste artigo está organizado com as seguintes seções: a Seção II descreve como foi nossa organização e metodologia utilizada durante a execução do trabalho; a Seção III apresenta os estudos e resultados obtidos até o momento no trabalho; e a Seção IV conclui este artigo e mostra os trabalhos futuros.

### II. METODOLOGIA

### III. ESTUDOS E RESULTADOS

### IV. CONSIDERAÇÕES FINAIS

### REFERÊNCIAS

- [1] S. R. Chidamber and C. F. Keremer, "A metrics suite for object oriented design," in *IEEE Transactions on Software Engineering*, vol. 20, 1994, pp. 476–493.