Ferramenta de apoio à coleta de métricas em software orientado a objetos

Bruno Góis Borges

Prof. Everaldo Artur Grahl - Orientador

Roteiro da Apresentação

- Introdução
- Objetivos do trabalho
- Métricas Orientada a Objetos (MOO)
- Visual Métrica
- Resultados e discussão
- Conclusão
- Extensões

Introdução

- Evolução da Engenharia de Software buscando qualidade no desenvolvimento
- Surgimento da Orientação a Objetos (OO)
- Crescimento dos sistemas e aumento de sua complexidade

Introdução

- Surgimento das Métricas Orientadas a Objetos
- Falta de ferramentas de suporte a Métricas OO para C# e Java
- Visual Métrica

Objetivo do Trabalho

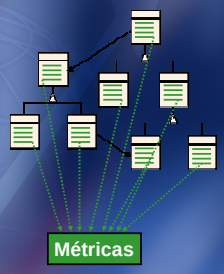
- **Desenvolver uma ferramenta capaz de efetuar a coleta de métricas OO em projetos escrito na linguagem C# e Java**
  - Realizar análises léxica e sintática dos programas C# e Java
  - Permitir que a ferramenta receba como entrada arquivos de projeto do C# e Java
  - Disponibilizar uma ferramenta para coletar métricas OO

MOO – Conceitos Básicos

"Para gerenciar produtividade e qualidade é necessário saber se ambas estão melhorando ou piorando. Isto implica a necessidade de métricas que indiquem as inclinações do desenvolvimento de sistema."

Arthur (1994, p. 25)

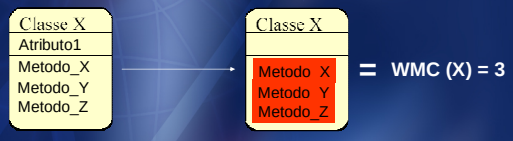
MOO – Métricas



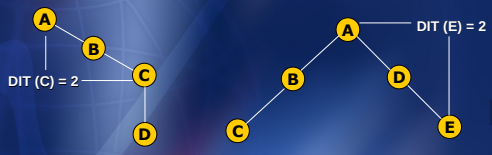
MOO – Chidamber e Kemerer

- Métodos ponderados por classe (ou WMC, *Weighted Methods per Class*)
- Profundidade da árvore de herança (ou DIT, *Depth of Inheritance Tree*)
- Número de filhos (ou NOC, *Number Of Children*)
- Falta de coesão em métodos (ou LCOM, *Lack of Cohesion in Methods*)
- Acoplamento entre classes de objetos (ou CBO, *Coupling Betwenn Object Classes*)
- Resposta para uma classe (ou RFC, *Response For a Class*)

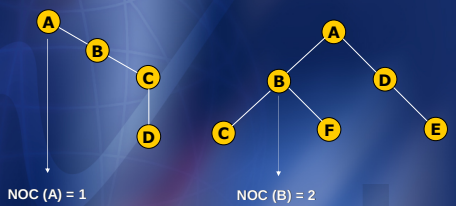
WMC – Métodos ponderados por classe



DIT – Profundidade da árvore de herança



NOC – Número de Filhos

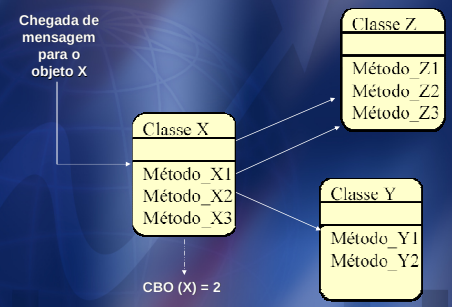


LCOM – Falta de Coesão em métodos

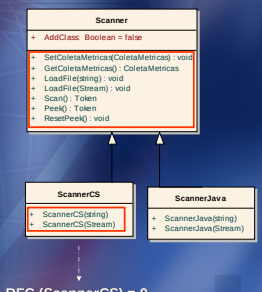
- Seja C uma classe com 4 métodos e com os seguintes conjuntos de atributos utilizados por método:
- – I1 = {a, b, x}
- – I2 = {a, e, f, g}
- – I3 = {c, x}
- – I4 = {v, u, w}

Para o cálculo de LCOM é necessário obter as intersecções entre os pares

CBO – Acoplamento entre Classes de Objetos



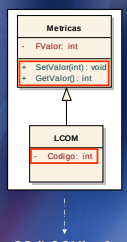
RFC – Resposta para uma Classe



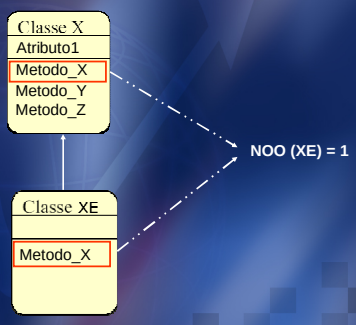
MOO – Lorenz e Kidd

- Tamanho da classe (CS, *Class Size*): CS = nº total de operações ou atributos da classe
- Número de operações redefinidas na subclasse (NOO, *Number of Operations Overriden*)
- Número de operações acrescentadas na subclasse (NOA, *Number of Operations Added*)

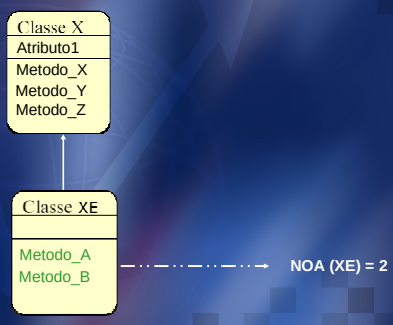
CS – Tamanho da Classe



NOO – Número de operações redefinidas

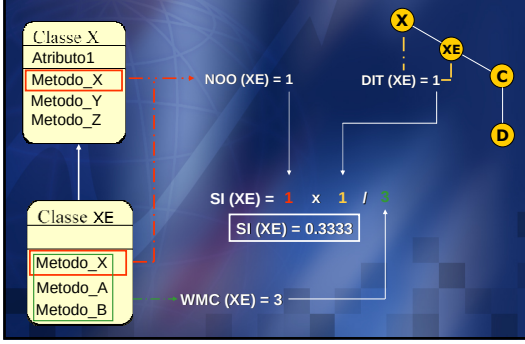


NOA – Número de operações adicionadas



SI – Índice de Especialização

SI = [NOO x DIT] / nº total de métodos da classe



Visual Métrica – Principais Requisitos

- Prover suporte a métricas OO propostas por CK e LK
- Programas para o cálculo das métricas devem ser escritos em C# e Java

Visual Métrica – Principais Requisitos

- **A ferramenta deve receber como entrada programas escritos em C# ou Java**
  - Projetos não pode conter erros
  - Projetos devem ser implementados utilizando orientação a objetos
  - Realizar análise léxica e sintática das classes do projeto

Visual Métrica – Principais Requisitos

- **Métricas calculadas pela ferramenta devem ser exibidas em forma de grade**
- **Disponibilizar um estudo das classes em forma de gráfico**

Especificação da Linguagem C# e Java

- **Técnicas e ferramentas utilizadas**
  - Definições regulares
  - Backus-Naur Form (BNF)
  - Coco/R for C#

Especificação da Linguagem C# e Java

- **Definições regulares**
  - C#

```
Primary
=
  ( Literal
  | "(" Expression ")"
  | ( "bool" | "byte"   | "char" | "decimal" | "double"
    | "float" | "int"   | "long" | "object" | "sbyte"
    | "short" | "string" | "uint" | "ulong"  | "ushort"
    ) "." ident [ IF (IsGeneric()) TypeArgumentList ]
  | ident { "!:" ident [ TypeArgumentList ] "." ident } [ IF (IsGeneric()) TypeArgumentList ]
  | "this"
  | "base" ( "." ident [ IF (IsGeneric()) TypeArgumentList ]
           | "[" Expression { "," Expression } "]"
           )
```

  - Java

```
Primary
= '(' Expression ')'
| "this" ArgumentsOpt
| "super" SuperSuffix
| Literal
| "new" Creator
| ident { IF(dotAndIdent()) '.' ident } [ IF(isIdentSuffix()) IdentifierSuffix ]
| BasicType BracketsOpt '.' "class"
| "void" '.' "class"
.
```

Especificação da Linguagem C# e Java

- **BNF**
  - C#

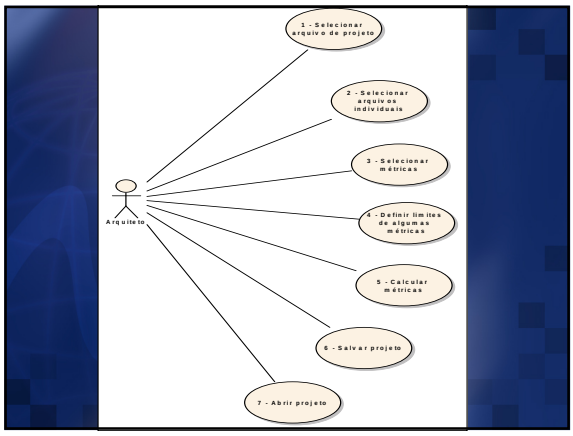
```
ClassBase
=
[ ":" ClassType { "," TypeName } ]
.

ClassBody
=
"{" { Attributes }                                    (. Modifiers m = new Modifiers(this); .)
      ModifierList<m>
      ClassMemberDeclaration<m>
    }
"}" .
```

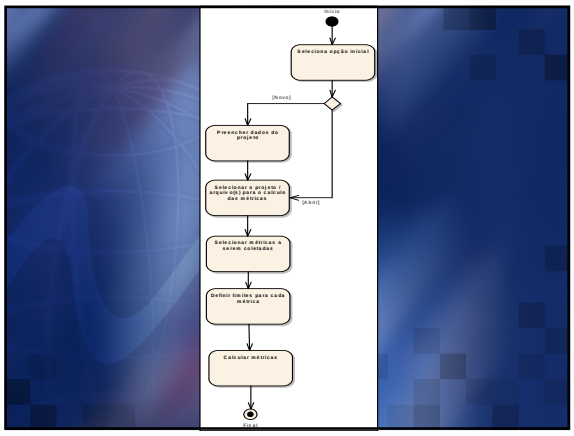
  - Java

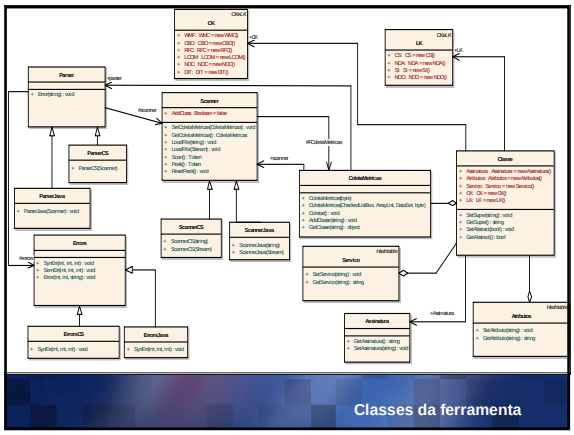
```
ClassDeclaration
= "class" ident ["extends" Type] ["implements" TypeList] ClassBody
.

ClassBody
= '{' {ClassBodyDeclaration} '}'
.

ClassBodyDeclaration
= ';'
| ["static"] ( Block
             | [ Modifier1 {Modifier} ] MemberDecl
             )
```

Especificação da Ferramenta

- **Técnicas e ferramentas utilizadas**
  - Unified Modeling Language (UML)
  - Enterprise Architec







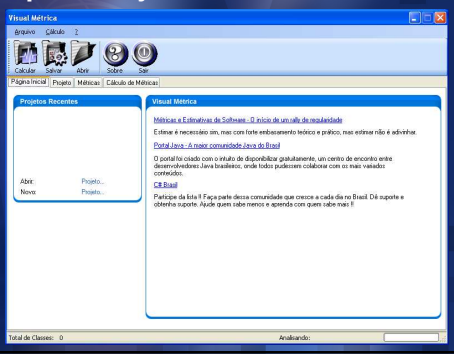
Classes da ferramenta

Visual Métrica - Implementação

- **Ferramentas utilizadas**
  - Linguagem C# .net Windows Forms – Ambiente Microsoft Visual C# 2005 Express Edition
  - Coco/R for C#

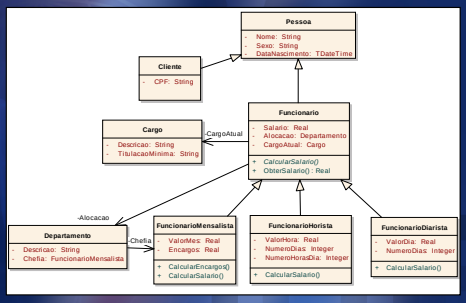
Visual Métrica - Operacionalidade

- Apresentação



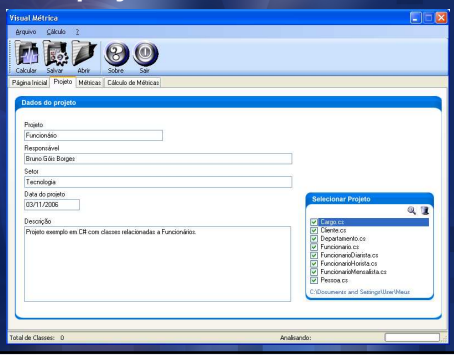
Visual Métrica - Operacionalidade

- Classes do projeto para cálculo



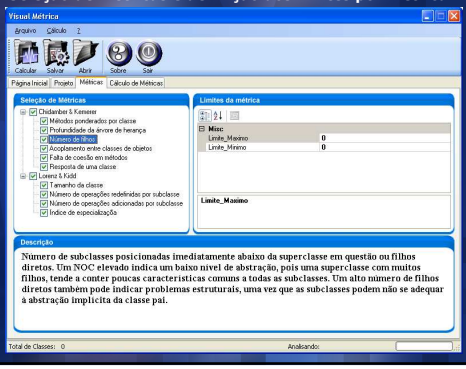
Visual Métrica - Operacionalidade

- Novo projeto



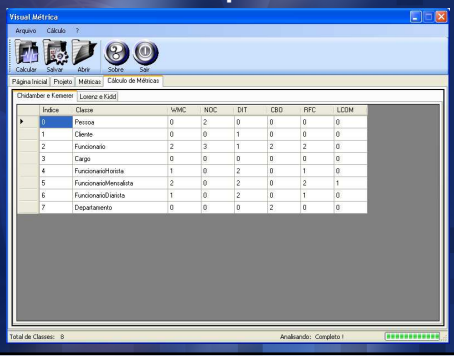
Visual Métrica - Operacionalidade

- Seleção de métricas e definição dos limites por métrica



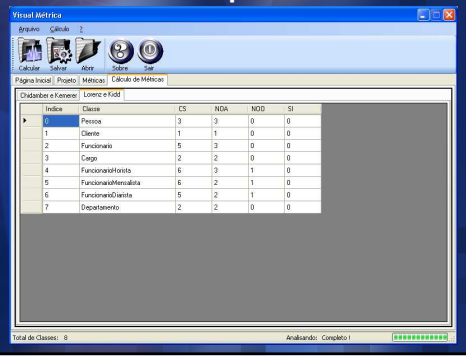
Visual Métrica - Operacionalidade

- Métricas calculas para CK



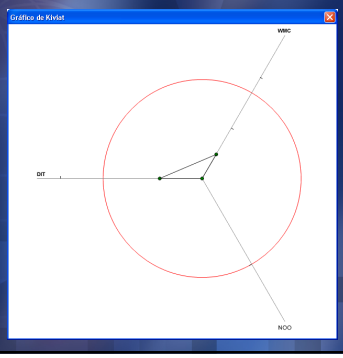
Visual Métrica - Operacionalidade

- Métricas calculas para LK



Visual Métrica - Operacionalidade

- **Gráfico de Kiviat – Classe "Funcionario"**



Resultado e Discussão

- **Apresentou bons resultados no sentido de prover suporte a métrics OO para C# e Java**
- **Limitações dos analisadores**
- **Facilidade e confiabilidade**

Conclusão

- Propicia um conhecimento sobre métricas OO
- Software mais planejados, organizados, controlados e direcionado
- Importância do surgimento de novas ferramentas de apoio a coleta de métricas OO
- Importância do Coco/R for C#
- Objetivos foram atingidos

Extensões

- Integrar as funcionalidades do Visual Métrica como *plug-in* para ferramentas CASE
- Adotar um número maior de métricas
- Ampliar a ferramenta para análise de outras linguagens OO