

## Métricas de Produto

Eduardo Figueiredo

<http://www.dcc.ufmg.br/~figueiredo>

## Métricas de Produto

- Quantificam atributos internos do software
- Exemplos de atributos
  - Tamanho
  - Acoplamento entre componentes
  - Coesão de um componente, etc.

## Tipos de Métricas

- Métricas Dinâmicas
  - São coletadas por medições realizadas durante a execução do programa
- Métricas Estáticas
  - São coletadas por medições realizadas na documentação de projeto ou código fonte do programa

## Dinâmicas x Estática

- Métricas dinâmicas ajudam a avaliar atributos de qualidade como eficiência e confiabilidade
  - São medidas após o sistema ter sido implementado
- Métricas estáticas ajudam a avaliar atributos como complexidade e facilidade de manutenção
  - Podem ser medidas na fase de projeto

## Métricas Estáticas Tradicionais

## Algumas Métricas Estáticas

- Fan-in / Fan-out
- Tamanho do código
- Complexidade Ciclomática
- Tamanho do Vocabulário
- Profundidade de Aninhamento

## Fan-in e Fan-out

- Fan-in
  - Conta o número de funções que chamam uma determinada função
  - Valor alto significa grande impacto em mudanças (propagação)
- Fan-out
  - Conta o número de funções chamadas pela função
  - Valor alto significa grande complexidade da função

## Tamanho e Complexidade

- Tamanho
  - Tamanho tem se mostrado como as métricas mais confiáveis e úteis
  - Em geral, quanto maior, mais complexo e propenso a erros será o componente
- Complexidade Ciclomática
  - Mede a complexidade de controle do programa (*if*, *while*, *for*, etc.)
  - Está relacionada a facilidade de compreensão

## Vocabulário e Aninhamento

- Tamanho do Vocabulário
  - Conta a quantidade de identificadores (exemplo, nome de classes) do programa
  - Mais identificadores podem significar que eles são mais significativos
- Profundidade de Aninhamento
  - Conta estruturas internas como *if* e *while* aninhados
  - Estruturas aninhadas são mais difíceis de se compreender

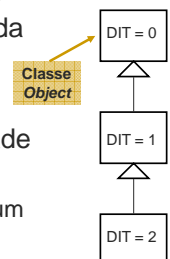
## Métricas para Programas Orientados a Objetos

## Métricas de Programas OO

- Métricas de Chidamber-Kemerer (CK)
  - Métodos Ponderados por Classes (WMC)
  - Profundidade da Herança (DIT)
  - Número de Filhos (NOC)
  - Acoplamento entre Objetos (CBO)
  - Falta de Coesão em Métodos (LCOM)
- Número de Operações Sobreescritas

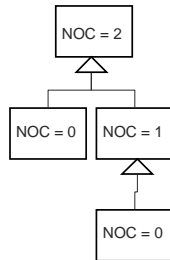
## Profundidade de Herança (DIT)

- Representam o número de níveis que uma classe herda métodos e atributos
- Quanto maior a profundidade
  - Mais complexo o projeto
  - Mais difícil de se entender um módulo



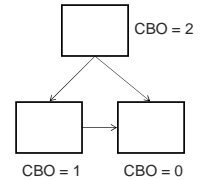
## Número de Filhos (NOC)

- Conta o número de subclasses diretas
  - Mede a largura da hierarquia de uma classe
- Valor alto, pode indicar maior reuso



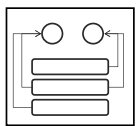
## Acoplamento entre Objetos (CBO)

- Semelhante a Fan-out
  - Conta classes chamadas por uma classe
- Quanto mais acoplado uma classe
  - Mais difícil de entender e de manter

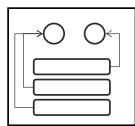


## Falta de Coesão (LCOM)

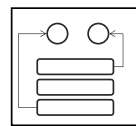
- Mede o quanto os métodos de uma classe acessam atributos em comum
  - Mais atributos em comum, maior coesão, menor perda de coesão (LCOM)



LCOM = 0 (1 - 2)



LCOM = 1 (2 - 1)



LCOM = 3 (3 - 0)

## Métricas para Métodos

- Métodos Ponderados por Classes (WMC)
  - Atribui pesos aos métodos de uma classe
  - Uma forma é “pesar” por linhas de código
  - Valores altos indicam complexidade
- Número de Operações Sobrescritas
  - Conta as operações de uma classe que são sobrescritas por subclasses
  - Valores altos indicam problema na hierarquia de herança

## Bibliografia da Aula

- Ian Sommerville. Engenharia de Software, 9ª Edição. Pearson Education, 2011.
  - Cap. 24 Gerenciamento de Qualidade (Seção 24.4)