Métricas de Software

[3]

Metodologias de desenvolvimento de software

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas
Prof. Felippe Scheidt – IFPR – Campus Foz do Iguaçu
2024

Tópicos

- O que é Métrica de Software?
- Exemplos de métricas
  - SLOC
  - Cobertura
  - Complexidade

Qualidade de código

ES compreende o conjunto de etapas que envolvem: ferramentas, métodos e processos com objetivo de melhorar e garantir a qualidade.

Qualidade do código

- Como avaliar a qualidade do código?
- Uso de métricas

Não se consegue controlar o que não se consegue medir

Métrica de Software

- Métrica é um padrão de medida de determinada propriedade calculada a partir do código de um software.
- Métricas são essenciais para:
  - Planejar o orçamento do projeto
  - Estimar o custo
  - Garantir qualidade
  - Realizar testes
  - Otimização e debug de código
  - Atribuição de tarefas

Métricas

- O gerente de projeto precisa estimar o custo, o tempo e esforço exigidos para o desenvolvimento dos projetos de software.
- Como podemos estimar, ou medir software?
- Métricas de software permitem mensurar em termos quantitativos diferentes aspectos e características do código.

Métricas

- A medição é algo comum no mundo da engenharia.
- Métricas de softwares possibilitam realizar uma das atividades mais fundamentais do gerenciamento de projetos que é o **planejamento**.
- Precisamos de uma estimativa do esforço, uma forma de prever realisticamente a quantidade de recursos que serão usados, como por exemplo:
  - Quantidade de Horas-por-desenvolvedor
  - Dinheiro

✅ SLOC

- **Linhas de código fonte** é uma medida de software usada para medir o tamanho de um software, através da contagem do **número de linhas** do código fonte.
- Experimentos mostram uma correlação de SLOC com o tempo necessário de desenvolvimento.
- Assim, SLOC é usada para prever a quantidade de **esforço** que será necessário para desenvolver um programa, bem como a estimativa de produtividade

Exemplo

Linhas em branco são ignoradas.

BASIC	C

`PRINT "hello, world"`	`#include <stdio.h>`
	
	`int main() {`
	`    printf("hello, world\n");`
	`}`

Linhas de código: 1	Linhas de código: 4

SLOC - Sistema operacional

SLOC do Sistema operacional Windows:

Year	Operating system	SLOC (million)
1993	Windows NT 3.1	4–5
1994	Windows NT 3.5	7–8
1996	Windows NT 4.0	11–12
2000	Windows 2000	more than 29
2001	Windows XP	45
2003	Windows Server 2003	50

Debian

Um estudo realizado com o Debian, contendo LOC de 55M, estima que seu desenvolvido necessitaria de 14 mil pessoas/ano, e custaria $1.9 bilhões

Year	Operating system	SLOC (million)
2000	Debian 2.2	55–59
2002	Debian 3.0	104
2005	Debian 3.1	215
2007	Debian 4.0	283
2009	Debian 5.0	324
2012	Debian 7.0	419

Limitações

- SLOC não faz sentido quando compara-se código escrito em **diferentes** linguagens.
- Entretanto **funcionalidade** é menos correlacionada com SLOC pois um desenvolvedor experiente pode escrever uma função em poucas linhas de código enquanto que outro desenvolvedor pode escrever a mesma funcionalidade em um código muito mais extenso.

✅ Cobertura de código

- Cobertura de código (**code coverage**) representa o percentual do código fonte que é testado durante a execução de uma suíte de teste.
- Um programa com alta cobertura de código indica que há uma menor chance existir bugs em relação a um programa com baixa cobertura.
- O cálculo da cobertura pode ser realizado considerando o código do ponto de vista do:
  - Percentual de funções executadas
  - Percentual de declarações executadas

✅ Complexidade ciclomática

- Complexidade condicional
- Indica a complexidade de um programa
- Mede a quantidade total de caminhos independentes existentes no código fonte do programa.
- Complexidade ciclomática pode ser aplicada a funções, módulos, métodos ou classes.

Exemplo

- Se um código não possui controle de fluxo (IF-ELSE) a complexidade do código é igual a 1, pois há somente um caminho de execução possível.
- Se existe uma condição IF, então a complexidade é 2.
- Um IF que contém outro IF, apresenta complexidade 3.
- Regra geral é, se a complexidade de um código é maior que 10, pode-se dizer que o custo de manter o código é alto, e geralmente é um bom candidato para refatoração.