



**INSTITUTO FEDERAL**  
Rio Grande do Sul

---

# Métricas

Qualidade de Software  
5º Semestre

Tecnologia em Análise e Desenvolvimento de Sistemas  
Revisão Prof Vinicius Machado  
Material Prof. Cibele da Rosa Christ Sinoti



# Conceito

---

- Métrica é a medição de um atributo (produto, processo ou recursos).
- Medições se dedicam a obter um ou mais valores numéricos para um atributo de qualidade
  - Comparando os números, é possível inferir sobre a qualidade de um produto.
- Exemplos:
  - Número de Linhas de código;
  - Número de defeitos encontrados por fase de desenvolvimento;
  - Tempo para a realização de uma tarefa;
  - ...



# Por que medir?

---

- Avaliações de qualidade podem ser demoradas;
- Medição permite empregar avaliações automatizadas;
- Muitas empresas não utilizam medições sistemáticas para avaliar qualidade, pois:
  - Os processos não são maduros o suficiente;
  - Não há métricas padronizadas;
  - As ferramentas de medição oferecem um apoio limitado.



# O que medir?

---

- **Atributos Internos** são mais facilmente quantificáveis e medidos diretamente;
  - Linhas de código, número de pontos de decisão, ...
- **Atributos Externos** são mais difíceis de serem quantificados, pois só são observados quando o software é executado
  - Quantidade de falhas observadas pelos usuários, dificuldade de navegação entre telas, tempo gasto para procurar uma informação, ...
- **Atributos de Qualidade** – são atributos externos que definem qualidade (segurança, confiabilidade, portabilidade, usabilidade, ...)



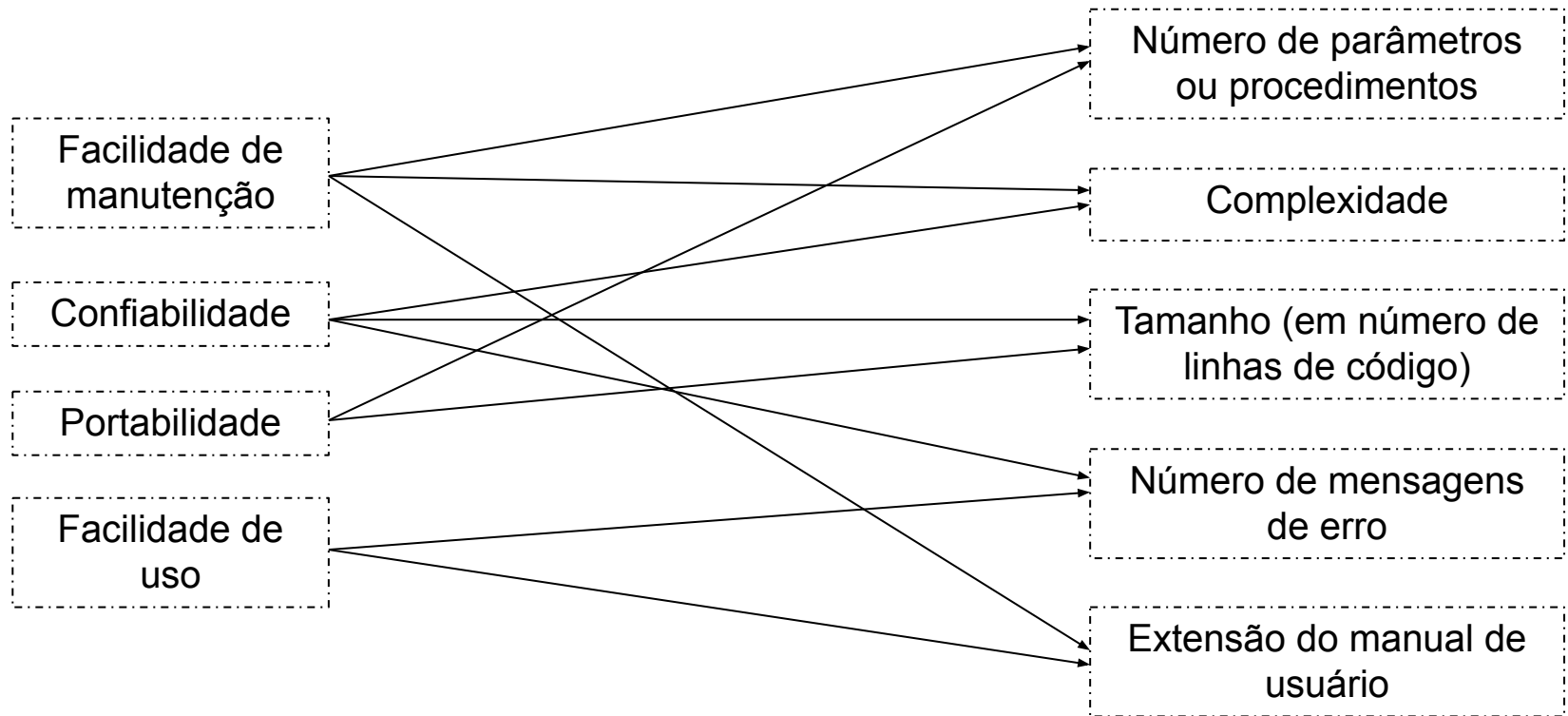
# Problema

---

- Geralmente, é impossível medir um atributo de qualidade diretamente;
  - Facilidade de manutenção;
  - Facilidade de uso;
  - Confiabilidade;
  - ...
- Deveria haver um relacionamento claro e válido entre atributos de qualidade e atributos internos (linhas de código, número de defeitos, ...)



# Modelo de Qualidade



# Processo de medição

---

- Deve fazer parte do processo de controle de qualidade (utilizam dados históricos de projetos anteriores);
- Atividades:
  1. Escolher medições a serem realizadas;
  2. Selecionar componentes a serem avaliados;
  3. Medir características dos componentes;
  4. Identificar medições anômalas;
  5. Analisar componentes anômalos.



# 1. Escolher medições a serem realizadas

- Uma das abordagens para escolher as medições é o **GQM** (*Goal-Question-Metric*);
- GQM é uma forma de sistematizar o processo, evitando medições que possuem pouco significado ou utilidade;
- Consiste em conduzir a análise a partir da meta organizacional, onde questões são formuladas para atender um objetivo e métricas são escolhidas para responderem às questões.





# GQM

---

- **Goal** – objetivo
  - Definem o que a organização quer melhorar.
  - Ex.: produtividade
- **Question** – questões
  - Refinamento dos objetivos em áreas de incertezas.
  - Ex.: é possível aumentar o número de linhas de código produzidas?
- **Metric** – métricas
  - Medição necessária para responder à questão.
  - Ex.: linhas de código por desenvolvedor



## 2. Selecionar componentes a serem avaliados

---

- Pode não ser desejável (ou necessário) medir todo o sistema;
- É possível escolher um subconjunto representativo de componentes ou componentes particularmente críticos
- Exemplo:
  - Quais são os casos de uso principais?
  - Qual funcionalidade será mais usada?
  - Qual parte do sistema será mais “estressada” durante o uso?



# 3. Medir características dos componentes

---

- Componentes selecionados são medidos;
- Medidas são associadas aos atributos de qualidade;
- Podem ser usadas outras ferramentas para medir
  - Logs armazenados;
  - Ferramentas de testes;
  - ...



## 4. Identificar medições anômalas

---

- Comparar as medições obtidas com medições anteriores → utilizar dados históricos em busca de valores incomuns
  - Valores muito baixos ou muito altos para cada métrica;
  - Identificar **outliers** → dados que se diferenciam drasticamente de todos os outros. É um valor que foge da normalidade e que pode (e provavelmente irá) causar anomalias nos resultados obtidos.



## 5. Analisar componentes anômalos

---

- Componentes que estão gerando valores anômalos devem ser examinados – essa inspeção irá decidir se existe ou não problema no componente;
- Importante! Valores incomuns para componentes não necessariamente significam que a qualidade seja baixa.



# Exemplo

- **Objetivo 1:** Melhorar a precisão das estimativas de projeto
- **Questões:**
  1. Qual a precisão da estimativa de cronograma?
- **Métricas:**
  - 1a. Precisão total do cronograma = 
$$\frac{\text{tempo real de todo o projeto}}{\text{tempo estimado do projeto}}$$
  - 1b. Precisão do cronog. por macro-atividade = 
$$\frac{\text{tempo real por macro-atividade}}{\text{tempo estimado para macro-ativ.}}$$



# Exemplo

- **Objetivo 2:** Aumentar a qualidade dos produtos liberados para uso
- **Questões:**
  1. Qual qualidade dos produtos antes da sua liberação para uso?
  2. Qual a qualidade dos produtos após sua liberação para uso?
- **Métricas:**
  - 1a. Densidade de defeitos =  $\frac{\text{número de erros e modificações}}{\text{tamanho do sistema}^*}$
  - 2a. Deterioração do software =  $\frac{\text{esforço}^{**} \text{ após liberação para uso}}{\text{esforço antes da liberação}}$

*\* número de linhas de código, excluindo linhas em branco e comentários;*

*\*\* somatório do total de horas empregadas por todos os membros da equipe na execução de uma tarefa.*

# Exemplo

- **Objetivo 3:** Diminuir o custo final dos projetos
- **Questões:**
  1. Qual é o esforço total dos projetos?
  2. Qual o percentual de retrabalho em relação ao esforço total?
- **Métricas:**
  - 1a. Esforço total do projeto;
  - 1b. Esforço por macro-atividade do projeto
  - 2a. Percentual de retrabalho =  $\frac{\text{esforço em retrabalho} * 100}{\text{esforço total do projeto}}$
  - 2b. Percentual retrab. Macro-atividade =  $\frac{\text{esforço em ret. Macro-ativ} * 100}{\text{esforço total na macro-atividade}}$