

Processos e Qualidade de Software

Medições e Métricas de Software



PUC Minas

Instituto de Ciências Exatas
e Informática

Prof. Lesandro Ponciano

Departamento de Engenharia de Software
e Sistemas de Informação (DES)

Objetivos da Aula

- Apresentar os conceitos de métrica e de medição
- Contextualizar métricas e medição no ambiente de software
 - Discutir a importância e o uso
- Discutir Modelos de Qualidade

Mensuração

- Mensuração em software
 - Processo de definir, coletar, analisar e agir sobre medidas que possam melhorar a qualidade dos produtos e do processo
- Visa à obtenção de dados quantitativos
 - Suporte à tomada de decisões
- Permite
 - Identificar características do produto
 - Controlar a evolução
 - Identificar divergências (real vs. estimado)

Métricas

- Uma métrica é um padrão de medida pelo qual elementos (produtos, processos, etc.) podem ser avaliados
- Uma métrica de software é qualquer medição que se refere ao software
 - Número de linhas de código (*Lines of code*, LOC)
 - Número de defeitos relatados
 - Número de classes
 - Etc.

Características de uma Boa Métrica

■ Linearidade

- Valor deveria ser linearmente proporcional à performance sendo medida
- Se valor da métrica muda por uma certa razão, o desempenho está mudando na mesma razão
- Ex.: Ao comprar um computador com 2 vezes o desempenho, a expectativa é que o tempo de execução de uma tarefa seja a metade

■ Repetibilidade

- O mesmo valor (estatisticamente falando) deve resultar ao repetir um ensaio

Características de uma Boa Métrica

■ **Confiabilidade**

- Se uma métrica indica que A é mais rápido que B para um certo contexto, então:
 - Qualquer execução dentro desse contexto deve resultar em A mais rápido que B

■ **Facilidade de medição**

- Se a métrica não for fácil de medir, é improvável que seja usada
- A dificuldade também pode afetar a precisão e pode causar erros

Características de uma Boa Métrica

■ **Consistência**

- A unidade da métrica e sua definição precisa são as mesmas em sistemas diferentes e em configurações diferentes de um mesmo sistema
- Se não houver consistência, não se podem realizar comparações de sistemas

■ **Independência**

- Métricas devem ser definidas de forma independente de influência/pressões de fornecedores para que não haja interferência ou métricas enviesadas para um certo fornecedor, sistema, etc

Exemplos de Métricas

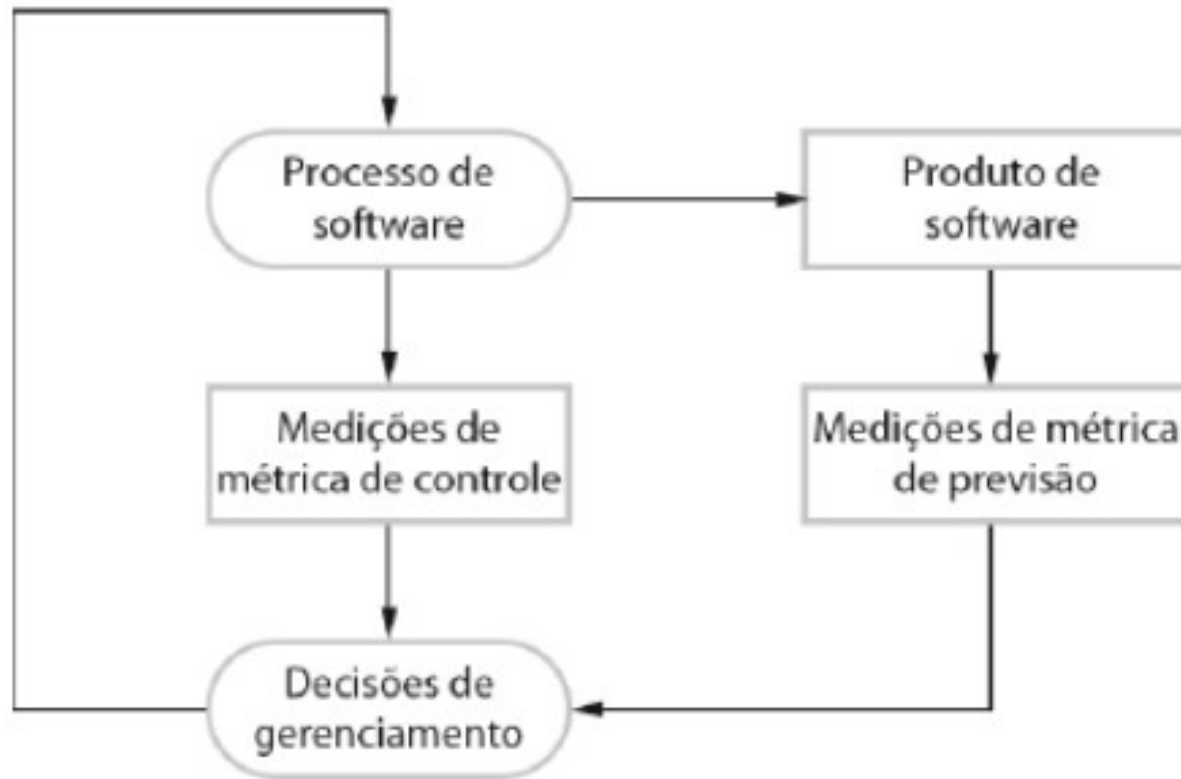
Entidade	Atributo	Métrica
Projeto completo	Duração	Meses entre início e fim
Projeto completo	Duração	Dias entre início e fim
Código	Tamanho	Número de linhas de código (LOC, KLOC)
Código	Tamanho	Número de comandos executáveis
Processo de teste de integração	Duração	Horas entre início e fim
Processo de teste de integração	Taxa de descoberta	Número de defeitos descobertos por KLOC
Testador	Eficiência	Número de defeitos descobertos por KLOC
Código	Qualidade	Número de defeitos descobertos por KLOC
Código	Confiabilidade	MTTF em horas de CPU
Código	Confiabilidade	Taxa de ocorrência de falhas em horas de CPU

“Mean Time To Failure” (MTTF)

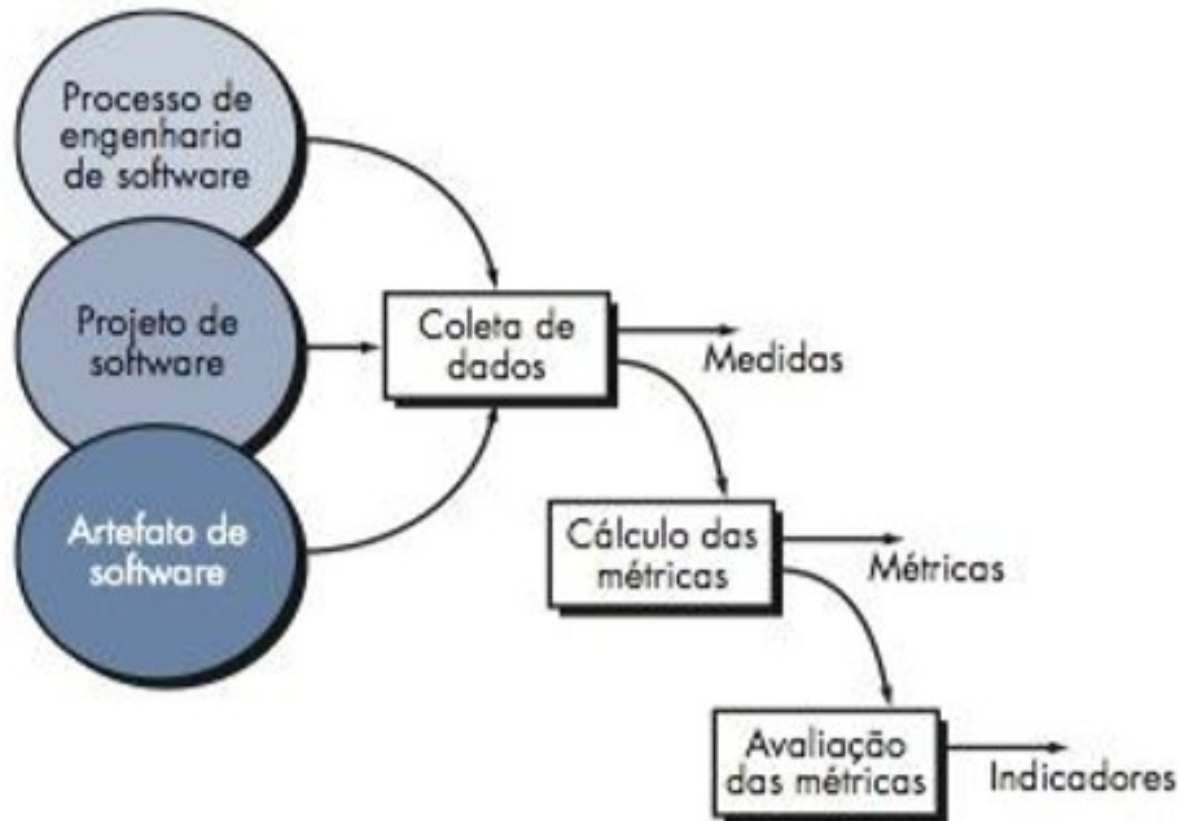
Métricas de Previsão e de Controle

- Métrica de **previsão**
 - Também são chamadas de métricas de produto
 - Geralmente associadas ao software em si
 - Visam prever características do software
 - Subsidiem na estimativa de esforço
 - Ex.: número de atributos, número de operadores, número de classes
- Métrica de **controle**
 - Geralmente associadas com os processos de software
 - Suportam o processo de gerenciamento
 - Subsidiem mudanças nos processos
 - Ex. esforço médio, tempo necessário para reparar defeitos

Medições de Previsão e de Controle



Medidas, Métricas e Indicadores



Medições

- Medições se dedicam a obter um ou mais valores numéricos para um atributo
 - Comparando os números, é possível tirar conclusões sobre o elemento que está sendo medido
- Medições de software podem ser usadas de duas maneiras
 - Avaliar a qualidade do software e fazer previsões gerais sobre ele (exemplo: número de defeitos)
 - Para identificar partes (ou módulos) problemáticas

Atributos Internos e Externos

- Medições do produto
- Atributos internos
 - São mais facilmente quantificáveis
 - São medidos diretamente
 - Ex.: Linhas de código, número de pontos de decisão
- Atributos externos
 - São mais difíceis de serem quantificados
 - São observados quando o software é executado
 - Ex.: quantidade de falhas observadas pelos usuários, dificuldade de navegação entre telas, tempo gasto para procurar uma informação no banco de dados

Atributos de Qualidade

- Em muitos dos casos a ênfase está em atributos de qualidade
- São atributos externos que definem qualidade
 - Segurança
 - Proteção
 - Confiabilidade
 - Facilidade de recuperação
 - Portabilidade
 - Facilidade de aprendizado
 - ...

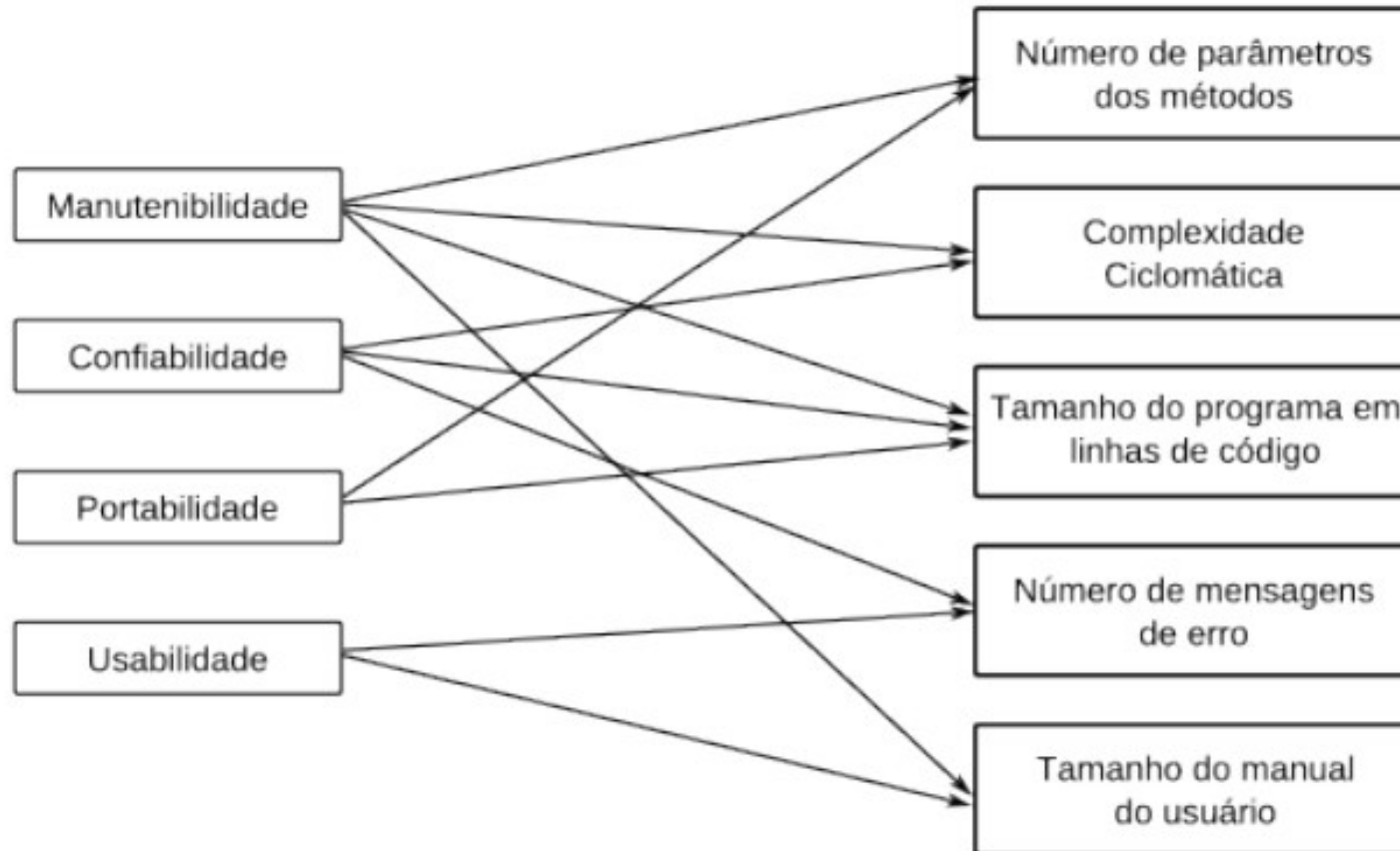
Problema Típico em Medições

- Geralmente é impossível medir um atributo de qualidade diretamente
 - Facilidade de manutenção
 - Facilidade de uso
 - Confiabilidade
- Atributos de qualidade são fatores externos ao software
- Métricas geralmente medem fatores internos

Modelos de Qualidade

- Modelo
 - Como atributos de qualidade são difíceis de se medir diretamente, utiliza-se uma ou mais métricas que são atributos internos
- Permitem estimar atributos de qualidade por meio de atributos internos
- Modelos de qualidade
 - Relacionam atributos internos com atributos de qualidade
 - Idealmente, deveria haver um relacionamento claro e válido entre atributos de qualidade e atributos internos

Exemplo de Modelo



Validade dos Modelos

- Três condições devem ser verificadas em modelos de qualidade
 - 1) O atributo interno deve ser precisamente medido
 - 2) Deve haver relacionamentos entre o que podemos medir e o que queremos saber
 - 3) Os relacionamentos são compreendidos e validados

Processo de Medição

- O processo de medição deve fazer parte do processo de controle da qualidade
 - Utilizam dados históricos de projetos anteriores
- As atividades do processo
 - 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

- Uma abordagem para escolher as medições é o GQM (*Goal-Question-Metric*)
 - Objetivos definem o que a organização quer melhorar (Ex.: produtividade)
 - Questões são refinamento dos objetivos em áreas de incertezas (Ex.: linhas de código produzidas podem ser aumentadas?)
 - Métricas são medições necessárias para responder as questões (Ex.: LOC por desenvolvedor)
- Questões são formuladas para atender um objetivo
- Métricas são escolhidas para responderem as questões

2 - Selecionar Componentes

- Pode não ser necessário (ou desejável) medir todo o sistema
- Estratégias de escolha
 - Escolher um subconjunto que seja representativo de todos os componentes
 - Escolher os componentes particularmente críticos na aplicação
- Exemplo de perguntas a se fazer
 - Quais são os casos de uso principais?
 - Qual funcionalidade será mais usada?
 - Qual parte do sistema será mais “estressada” durante o uso?

3 – Medir

- Os componentes selecionados são medidos
- As medidas são associadas aos atributos de qualidade
 - Geralmente envolve uma representação dos componentes
- Ferramentas de medição podem estar incorporadas a outras ferramentas (ou ambientes) de desenvolvimento
 - Ex.: logs armazenados pelo sistema, monitoramento realizado por sistemas externos, ferramentas de implementação de testes

4 – Identificar Medições Anômalas

- Uma vez feita as medições, é preciso compará-las a medições anteriores
 - Dados históricos são utilizados
- A análise deve procurar valores incomuns
 - Ou seja, valores muito altos ou muito baixos para cada métrica
 - São *outliers*?
 - Usar técnicas estatísticas para lidar com tais valores

5 – Identificar Componentes Anômalos

- Componentes que estão gerando valores anômalos
- Se um componente tem valores anômalos, este deve ser examinado
 - A inspeção é responsável por decidir se existe (ou não) problema no componente
- Um valor incomum para um componente não necessariamente significa que o componente tenha baixa qualidade
 - É preciso investigar cada caso
 - Ex.: muitas implementações são heurísticas que podem gerar valores estranhos em algumas situações

Análise de Dados

- Nem sempre é óbvio o que os dados significam
- Entender uma grande quantidade de números é muito difícil
- Estatísticos devem ser consultados, se estiverem disponíveis
- A análise de dados deve levar em conta as circunstâncias locais

Uso no Mercado de Software

- Muitas empresas ainda não usam medições sistemáticas para avaliar a qualidade
- Algumas razões
 - Os processos das empresas não são maduros o suficiente
 - A ausência de métricas padronizadas
 - Limitado apoio de ferramentas de medição

Atividade de Fixação

- 1) Defina atributos internos e atributos externos de software
- 2) Defina modelo de qualidade
- 3) Apresente um modelo de qualidade para
 - Segurança
 - Adequação funcional

Referências

- Ian Sommerville. Engenharia de Software, 9ª Edição. Pearson Education, 201. (Cap. 24: Gerenciamento de Qualidade)
- PRESSMAN, Roger. Engenharia de software. 8. Porto Alegre ISBN 9788580555349. (Capítulo 25: Métricas de Processo e Projeto)
- Eduardo Figueiredo, “Medição de Software”
http://homepages.dcc.ufmg.br/~figueiredo/disciplinas/aulas/intro-medicao_v01.pdf