UNIVÀS
UNIVERSIDADE DO VALE DO SAPUCAÍ

Gestão da Qualidade de Software

Introdução às Métricas de Software: Produto, Processo e Projeto

Prof. Flávio Belizário da Silva Mota
Universidade do Vale do Sapucaí – UNIVAS
Sistemas de Informação

Métricas de software

- Métricas de software são instrumentos que permitem quantificar atributos de produtos, processos e projetos de software.
- São fundamentais para monitorar, controlar e melhorar a qualidade, fornecendo dados objetivos para tomada de decisão.
- A IEEE define métricas como “uma medida **quantitativa** do grau em que um sistema, componente ou processo possui um determinado atributo”.
- A qualidade pode ser subjetiva:
  - “A interface deve ser muito agradável para o usuário”
    - O que é muito ou pouco?
    - O que é ser agradável?

Por que medir?

- Medir é essencial para **gerenciar**. Sem métricas, a avaliação da qualidade torna-se **subjetiva e imprecisa**.
- Motivos para medir produto, processo e projeto
  - Para avaliar a qualidade do software
  - Para descobrir e corrigir problemas potenciais
  - Para ajudar a estimar esforço
  - Para controlar o andamento do projeto
  - Avaliar a evolução da qualidade ao longo do tempo

Métricas

- Uma métrica é a medição de um atributo (propriedades ou características) de uma determinada entidade
  - Tempo, em dias/horas/meses, para programação de um sistema
  - Custo, em R$, para a realização de uma tarefa
  - Grau de satisfação do cliente com um determinado software

Propriedades desejáveis de uma métrica

- Fácil de ser calculada, entendida e testada
- Passível de estudos estatísticos
- Expressa em alguma unidade (tempo, pessoas, $)
- Obtida o mais cedo possível no ciclo de vida do software
- Passível de automação
- Repetível e independente do observador
- Comprometida com uma estratégia de melhoria

Categorização das Métricas

- Métricas:
  - **Métricas de Produto** – Avaliam características do software.
  - **Métricas de Processo** – Avaliam como o software é produzido.
  - **Métricas de Projeto** – Avaliam o gerenciamento e execução do projeto.

Categorização das Métricas

- **Métricas diretas (fundamentais ou básicas)**:
  - Medida realizada em termos de atributos observados (usualmente determinada pela contagem)
  - Exemplos:
    - Custo ($)
    - Número de linhas de código (LoCs)
    - Número de páginas
    - Número de diagramas
    - Etc...

Categorização das Métricas

- **Métricas indiretas (derivadas)**:
  - Medidas obtidas a partir de outras métricas
  - Exemplos:
    - Produtividade (funcionalidades/tempo)
    - Qualidade (taxa % de retrabalho)
    - Custo
    - Complexidade
    - Facilidade de manutenção

Categorização das Métricas

- **Métricas orientadas a tamanho**:
  - São medidas diretas do tamanho dos artefatos de software associados ao processo por meio do qual o software é desenvolvido
  - Exemplos:
    - Custo da implementação
    - LoCs
    - Número de defeitos em uma especificação de requisitos

Categorização das Métricas

- **Métricas orientadas por função**:
  - Consiste em um método para medição do software do ponto de vista do usuário, determinando de forma consistente o tamanho e complexidade de um software
  - Exemplos:
    - SOD (Speed of Delivery) – medida do número de funções desenvolvidas/entregues em um determinado período de tempo (mês), utilizando a equipe disponível

Categorização das Métricas

- **Métricas de produtividade**:
  - Concentram-se na saída/resultado do processo de engenharia de software
  - Exemplo:
    - PDR (Project Delivery Rate) – horas necessárias para desenvolver parte do sistema (função)
- **Métricas de qualidade**:
  - Oferecem uma indicação de quanto o software se adapta às exigências do cliente
    - Ex.: defeitos por artefato

Categorização das Métricas

- **Métricas técnicas:**
  - Concentram-se nas características do software e não no processo por meio do qual o software foi desenvolvido
  - Exemplo:
    - complexidade lógica e grau de manutenibilidade (LoCs, número de filhos de uma classe)

Como escolher métricas

- Critérios importantes:
  - **Relevância**: está ligada aos objetivos do projeto?
  - **Mensurabilidade**: é possível coletar dados de forma consistente?
  - **Custo de coleta**: vale o esforço de medir?
  - **Ação**: a métrica fornece informações que levam a decisões concretas?

ATENÇÃO -> cuidado com métricas de vaidade (números que não indicam valor real, como “linhas de código escritas”)

Paradigma GQM (Goal-Question-Metric)

- O Paradigma GQM é uma abordagem estruturada para definição de métricas.
- Etapas:
  1. **Goal (Meta):** definir o objetivo de medição.
  2. **Question (Pergunta):** formular questões que, se respondidas, mostram se a meta está sendo atingida.
  3. **Metric (Métrica):** identificar dados que respondam às perguntas.

Benefício: garante que as métricas estejam alinhadas a objetivos reais e relevantes.

Exemplo aplicando GQM

- Meta (Goal): Reduzir a quantidade de bugs em produção
- Perguntas (Questions):
  - Quantos bugs são encontrados em produção por mês?
  - Qual é a gravidade média dos bugs reportados?
- Métricas (Metrics):
  - Número de bugs/mês
  - Percentual de bugs críticos
  - Tempo médio para corrigir

Mini caso prático – Aplicando GQM

- **Cenário:**
  A empresa *SoftMax* está desenvolvendo um aplicativo de delivery. Nos últimos dois meses:
  - Reclamações sobre lentidão no carregamento das telas aumentaram 40%.
  - Clientes relataram problemas na finalização de pedidos em horários de pico.
  - A equipe de desenvolvimento diz que “tudo está funcionando” nos testes internos.

Mini caso prático – Aplicando GQM

- **Meta (Goal):** Melhorar o desempenho e a confiabilidade do aplicativo em horários de pico.
- **Perguntas (Questions)**:
  - Qual é o tempo médio de carregamento das telas em horários de pico?
  - Quantos pedidos falham durante o processo de finalização?
  - Qual a taxa de abandono de carrinho nesses horários?
- **Métricas (Metrics):**
  - Tempo médio de carregamento (ms) medido em logs.
  - Percentual de pedidos com erro de finalização.
  - Percentual de carrinhos abandonados.