

Engenharia de Software II

Aula 12

<http://www.ic.uff.br/~bianca/engsoft2/>

Ementa

- Processos de desenvolvimento de software (Caps. 2, 3 e 4 do Pressman)
- Estratégias e técnicas de teste de software (Caps. 13 e 14 do Pressman)
- **Métricas para software (Cap. 15)**
- Gestão de projetos de software: conceitos, métricas, estimativas, cronogramação, gestão de risco, gestão de qualidade e gestão de modificações
- Reengenharia e engenharia reversa

Medição

- Um elemento-chave de qualquer processo de engenharia é a **medição**.
 - A medição é o processo pelo qual **números** são associados a propriedades de objetos do mundo real.
 - Serve para entender melhor as **características** dos objetos.
 - Serve para avaliar e monitorar a **qualidade** dos produtos.
- Ao contrário de outras engenharias, a engenharia de software não está ancorada nas leis quantitativas da física.
 - Medidas como massa, voltagem, temperatura, pressão não são apropriadas para software.
 - Isso quer dizer que software é não-mensurável?
 - Não: apenas outros critérios devem ser estabelecidos.

Métricas de Software

- As métricas de software servem para avaliar a **qualidade do software** à medida que ele está sendo construído.
- Essas medidas de atributos do software dão uma indicação em **tempo real** da eficácia de:
 - Modelos de análise
 - Modelos de projeto
 - Código
 - Casos de teste
 - Qualidade global

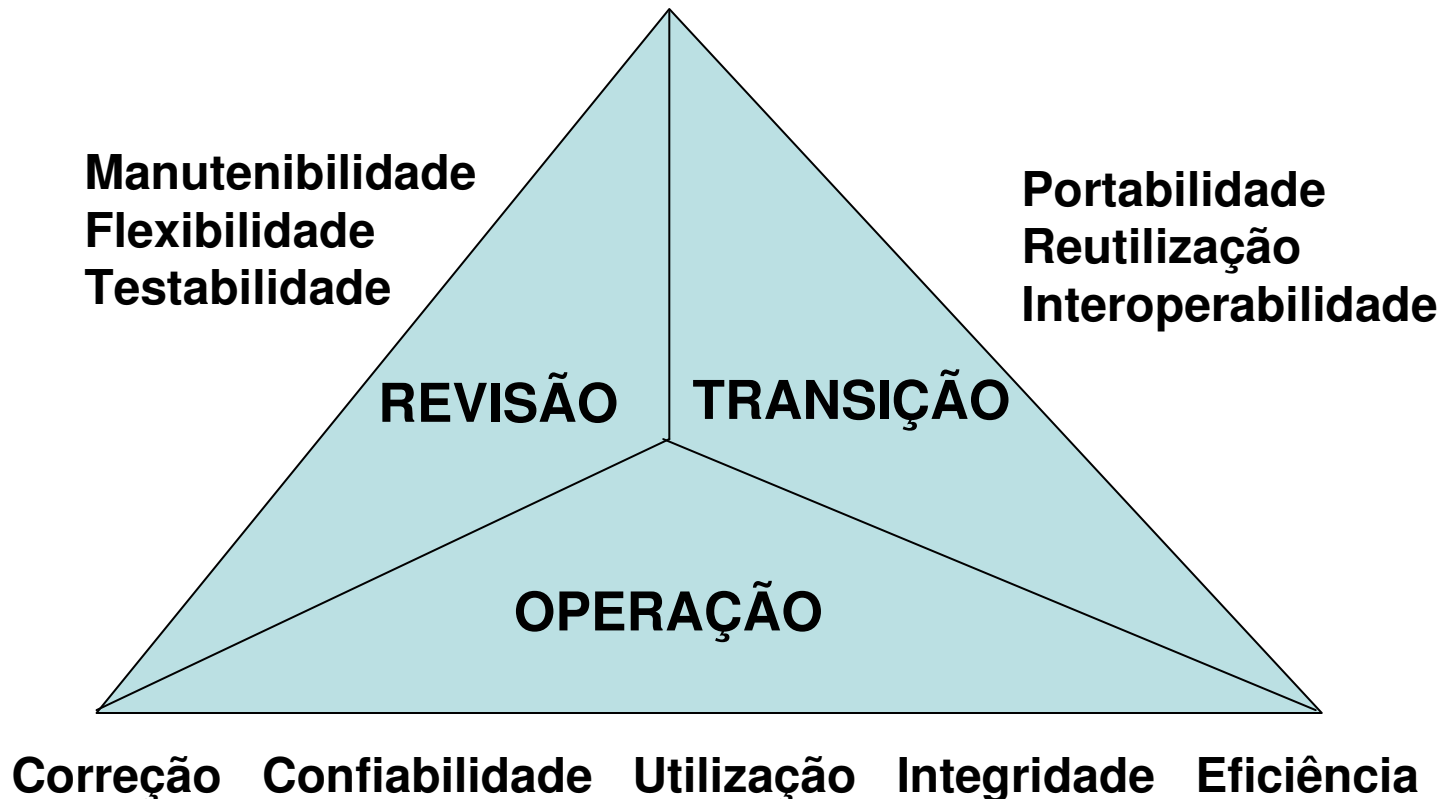
Como se define qualidade?

- Num sentido amplo, qualidade de software é dada por:
 1. Satisfação de **requisitos** funcionais e de desempenho.
 2. **Normas** de desenvolvimento que guiam o modo pelo qual o software é construído.
 3. Características **implícitas** que são esperadas de todo software desenvolvido profissionalmente.
 - Exemplo: facilidade de uso.

Fatores de qualidade

- Podem ser caracterizados em dois grupos amplos:
 1. Fatores que podem ser medidos diretamente.
 - Exemplo: Pontos por função.
 2. Fatores que podem ser medidos apenas indiretamente.
 - Exemplo: Usabilidade ou manutenibilidade
- Em cada caso deve haver medições.
 - Devemos comparar as medidas a algum valor e chegar a uma indicação da qualidade.

Fatores de Qualidade de McCall



- É difícil e em alguns casos impossível desenvolver medidas diretas desses fatores.
- Outras medidas quantitativas devem ser feitas e relacionadas a eles.

Fatores de Qualidade ISO 9126

- A norma ISO 9126 foi desenvolvida em uma tentativa de identificar atributos de qualidade de software.
- A norma identifica seis atributos-base:
 1. Funcionalidade
 2. Confiabilidade
 3. Usabilidade
 4. Eficiência
 5. Manutenibilidade
 6. Portabilidade
- Também não se prestam a medidas diretas.
 - Fornecem uma base para medidas indiretas.
 - Fornecem uma lista de verificação para avaliar a qualidade do software.

Transição para Visão Quantitativa

- Os fatores de qualidade McCall e ISO 9126 só podem ser julgados diretamente de forma **subjetiva**.
 - O julgamento precisa ser feito por **especialistas** que analisam o software qualitativamente.
- Foram desenvolvidas métricas de software que fornecem uma **avaliação qualitativa**.
 - Essas medidas não medem a qualidade diretamente, mas alguma **manifestação** da qualidade.
 - O fator chave é a **relação** entre a variável que é medida e a qualidade do software.

Vocabulário Básico

- **Medida:** fornece uma indicação quantitativa da extensão, quantidade, dimensão, capacidade ou tamanho de algum atributo.
- **Medição:** é o ato de determinar uma ou mais medidas.
- **Métrica:** relaciona as medidas individuais de algum modo.
 - Exemplo: Número médio de erros encontrados por teste de unidade.
- **Indicador:** é uma métrica ou conjunto de métricas que diz algo significativo sobre o processo de software ou o software em si.

Processo de Medição

- Um processo de medição pode ser caracterizado por cinco atividades:
 - Formulação
 - É a derivação de medidas e métricas adequadas.
 - Coleta
 - É o mecanismo usado para acumular os dados para derivar as métricas formuladas.
 - Análise
 - Cálculo de métricas e aplicação das ferramentas matemáticas.
 - Interpretação
 - Avaliação das métricas em um esforço de entender os e relacionar resultados.
 - Realimentação
 - Recomendações derivadas das métricas transmitidas à equipe.

Princípios de Medição

- Uma métrica de software deve ter as seguintes características:
 - O intervalo de valores deve fazer sentido.
 - Exemplo: 0 é o valor mínimo, 1 é o valor máximo e 0.5 é o valor médio.
 - O valor da métrica deve aumentar ou diminuir do mesmo modo que a característica do software que ela mede aumenta ou diminui com acontecimentos desejáveis ou indesejáveis.
 - A métrica deve ser validada e útil em uma grande variedade de contextos.

Diretrizes para Coleta e Análise

- Sempre que possível, a coleta de dados e a análise devem ser **automatizadas**.
- Técnicas **estatísticas** válidas devem ser usadas para relacionar os atributos internos do produto e as características externas de qualidade.
 - Exemplo: se o nível de complexidade arquitetural está correlacionado com o número de defeitos relacionados no uso.
- Diretrizes e recomendações **interpretativas** devem ser estabelecidas para cada métrica.

Paradigma GQM

- GQM = *Goal/Question/Metric* = Objetivo/Questão/Métrica
- É uma técnica para identificação de métricas significativas.
- Enfatiza a necessidade de:
 1. Estabelecer um **objetivo** explícito de medição.
 - Quase sempre entender, controlar ou aperfeiçoar.
 2. Definir um conjunto de **questões** que precisam ser respondidas a fim de alcançar o objetivo.
 3. Identificar **métricas** que ajudam a responder a essas questões.

Gabarito de Definição de Objetivo GQM

- **Analise** {atributo do software} **com a finalidade de** {objetivo global} **com relação a** {característica específica} **do ponto de vista de** {pessoal} **no contexto de** {ambiente}.

Exemplo

- **Objetivo:**
 - **Analise** a arquitetura de software X **com a finalidade de** avaliar componentes arquiteturais **com relação a** tornar o software X mais extensível **do ponto de vista de** desenvolvedores do software **no contexto de** aperfeiçoamento do produto nos próximos três anos.
- **Questões:**
 - Os componentes arquiteturais são caracterizados de um modo que compartimentalize a função e dados relacionados?
 - A complexidade de cada componente está dentro dos limites para facilitar a modificação e a extensão?
- **Métricas:**
 - Métrica que forneça uma indicação de coesão de um componente.
 - Complexidade ciclomática e outras métricas de complexidade.

Atributos de Métricas de Software Efetivas

- Simples e computáveis
 - Deve ser relativamente fácil aprender como derivar a métrica e seu cálculo deve ser rápido.
- Empíricas e persuasivas
 - Deve satisfazer às noções intuitivas do engenheiro sobre o atributo do produto que está sendo considerado.
- Consistentes e objetivas
 - Os resultados devem poder ser reproduzidos dadas as mesmas condições.
- Consistentes no uso de unidades e dimensões
 - Deve usar combinações de unidades que sejam compreensíveis.
- Independentes da linguagem de programação
 - Devem ser baseadas nos modelos de análise, projeto e na estrutura do programa.
- Mecanismo efetivo para realimentação de alta qualidade
 - Deve levar a uma produto final de alta qualidade.

Métricas para o Modelo de Análise

- Funcionalidade entregue
 - Fornece uma medida indireta da funcionalidade que é entregue com o software.
- Tamanho do sistema
 - Mede o tamanho global do sistema definido em termos de informação disponível como parte do modelo de análise.
- Qualidade da especificação
 - Fornece uma indicação da especificidade e completeza de uma especificação de requisitos.

Métricas para o Modelo de Projeto

- Métrica arquitetural
 - Fornece uma indicação da qualidade do projeto arquitetural.
- Métrica no nível de componente
 - Mede a complexidade dos componentes de software e outras características que influem na qualidade.
- Métricas de projeto de interface
 - Focalizam principalmente a usabilidade.
- Métricas especializadas em projetos OO.
 - Medem características de classes e suas características de comunicação e colaboração.

Métricas para Código-Fonte

- Métricas de Halstead
 - Medem a complexidade das expressões.
- Métricas de complexidade
 - Medem a complexidade lógica do código-fonte.
- Métricas de comprimento
 - Fornecem uma indicação do tamanho do software.

Métricas de Teste

- Métricas de cobertura de comando e desvio.
 - Levam ao projeto de casos de teste que fornecem cobertura do programa.
- Métricas relacionadas a defeito.
 - Enfocam erros encontrados, ao invés dos testes em si.
- Efetividade de teste
 - Fornecem uma indicação da efetividade dos testes que são conduzidos.
- Métricas em processo
 - Métricas relacionadas a processo que podem ser determinadas à medida que o teste é conduzido.