

Qualidade de Software

Qualidade de Produto

Engenharia de Software II

Profa. Andréa Sabedra Bordin

Roteiro

- Qualidade de Software
- Qualidade do Produto x Qualidade do Processo
- **Qualidade do Produto**
- **Medição da Qualidade do Produto**

Introdução

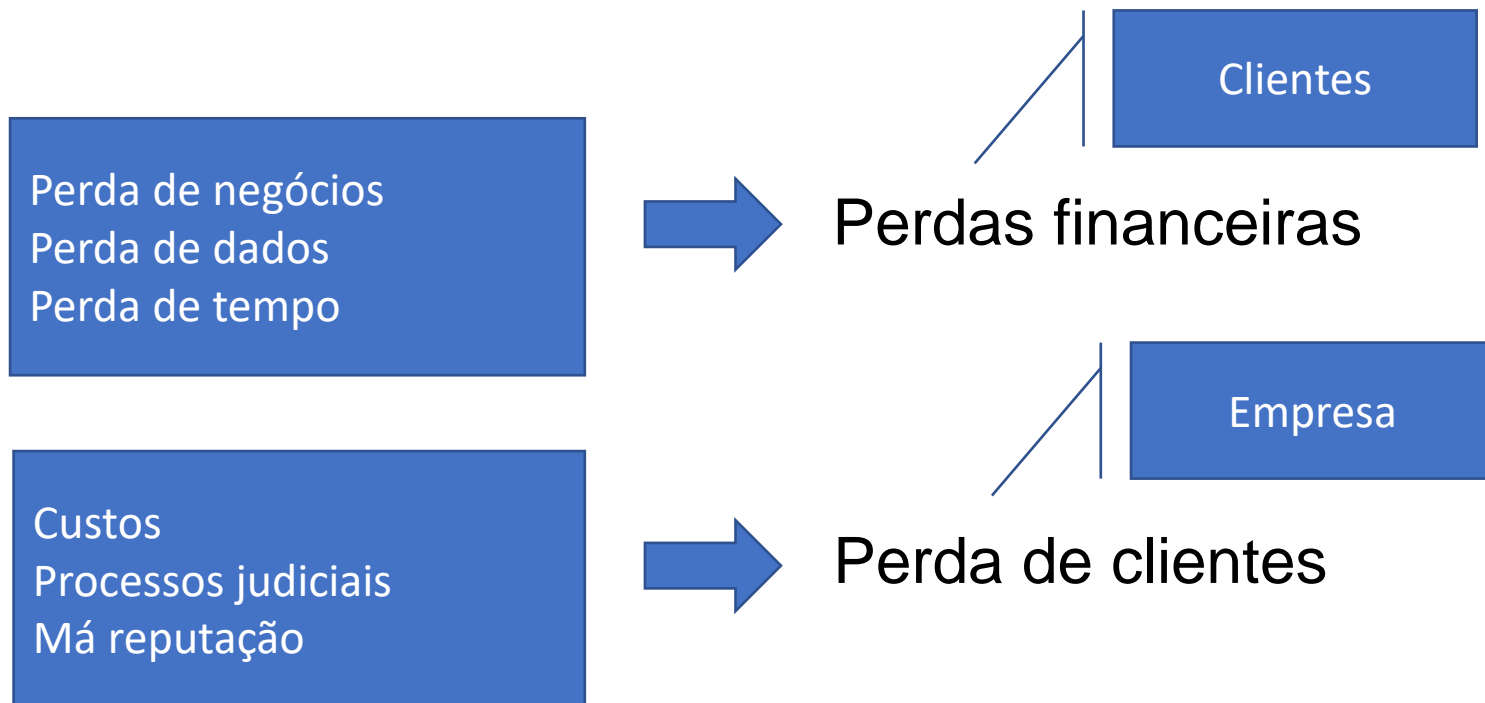
Caso de Falha em Sistema



- Em 1996, o veículo espacial Ariane 5 saiu do curso e explodiu segundos após o seu lançamento.
- Levou uma década de desenvolvimento e custou 7 bilhões de dólares.
- Os **testes insuficientes em componentes reutilizados** do veículo Ariane 4 foram a causa do acidente.
- Motivo da falha: Erro de software no cálculo da velocidade horizontal do foguete.
- A variável que armazenava este valor tinha 64 bits (*floating point*) e foi erroneamente modificada para 16 bits (*signed integer*). O valor era maior que 32.767 (maior inteiro), gerando uma falha de conversão!!

Introdução

- Problemas decorrentes da baixa qualidade de software



Introdução

- Uma das principais **metas** da Engenharia de Software é a produção de **software de qualidade**.
- Empresas que **desenvolvem** software de qualidade são mais **competitivas**;
- Empresas que **utilizam** software de alta qualidade podem, em geral, oferecer um melhor serviço a um preço mais **competitivo**.

Introdução

- **Qualidade está fortemente relacionada à conformidade com os requisitos.**

- Observado x especificado.
 - Pode haver problemas na observação.
 - Pode haver problemas na especificação.

- **Qualidade diz respeito à satisfação do cliente.**

- Requisitos são especificados por pessoas e com o objetivo de satisfazer outras pessoas.
 - Depende das escolhas feitas pelo cliente.
 - Pode haver problemas na especificação.

“the degree to which a software product meets established requirements; however, quality depends upon the degree to which those established requirements accurately represent stakeholder needs, wants, and expectations”

IEEE Std. P730-2013 Draft Standard for Software Quality Assurance Processes.

Introdução

- **Qualidade de Software** se refere:
 - a **características** desejáveis de produtos de software,
 - Aos **processos, ferramentas e técnicas** usadas para alcançar essas características.

A qualidade de um produto de software não se atinge de forma espontânea.

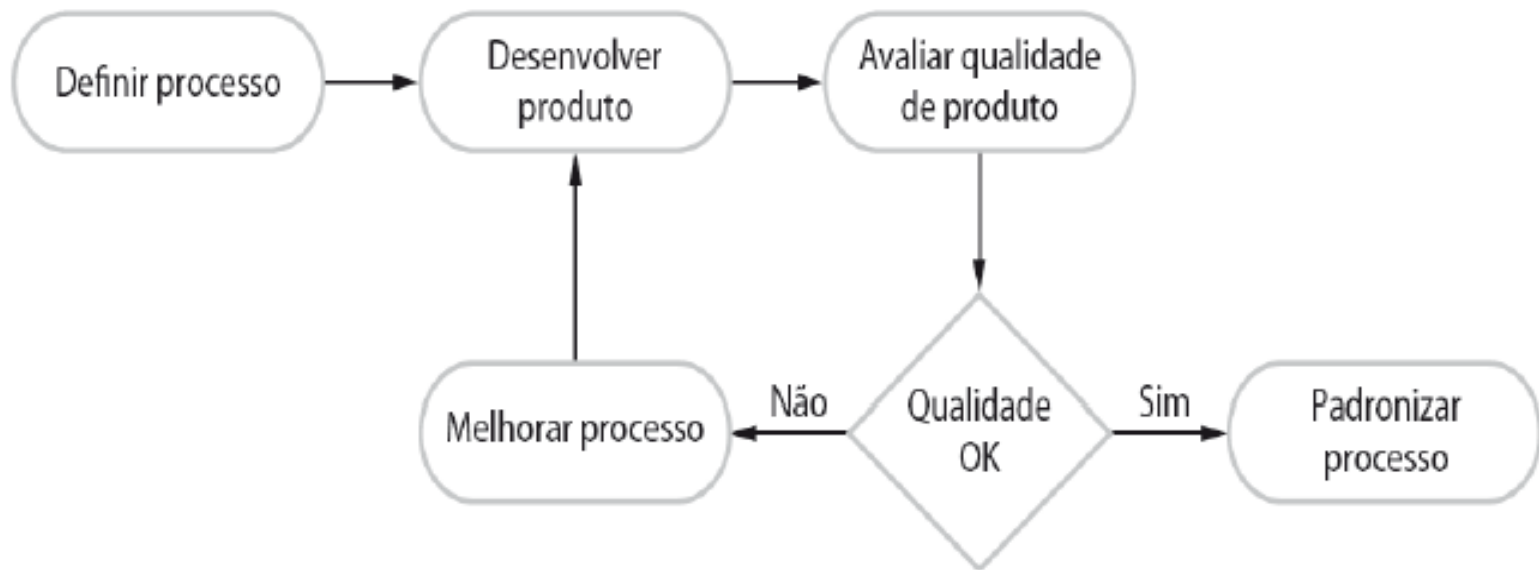
Ela tem de ser construída ao longo do processo de software.

Qualidade: Características do Produto

- Quais **características** um terminado produto de software deve apresentar para considerarmos que o mesmo tem qualidade?
 - adequação funcional, desempenho, confiabilidade, usabilidade, portabilidade e manutenibilidade, são algumas que estão diretamente relacionados à qualidade.
 - O grau em que cada uma dessas características precisa ser atendida pode variar de sistema para sistema.

Qualidade: Processo de desenvolvimento

- Como o processo de desenvolvimento influencia na qualidade do software?
 - A **qualidade de um produto** desenvolvido é influenciada pela **qualidade do processo** de produção.
 - Origem nos sistemas de manufatura.



Qualidade de produto e de processo

- Na Engenharia de Software a relação entre a **qualidade do processo** e a **qualidade do produto** é mais complexa.
 - ✓ A aplicação das habilidades individuais e experiência é particularmente importante no desenvolvimento de software (processo criativo).
 - ✓ Fatores externos, tais como a pressão para liberar um *release* pode prejudicar a qualidade de produto.

Para se obter um **produto de software de qualidade** é preciso verificar:

- As **características do produto (qualidade do produto)**
- O seu **processo de desenvolvimento (qualidade do processo)**.

Qualidade de Produto de Software

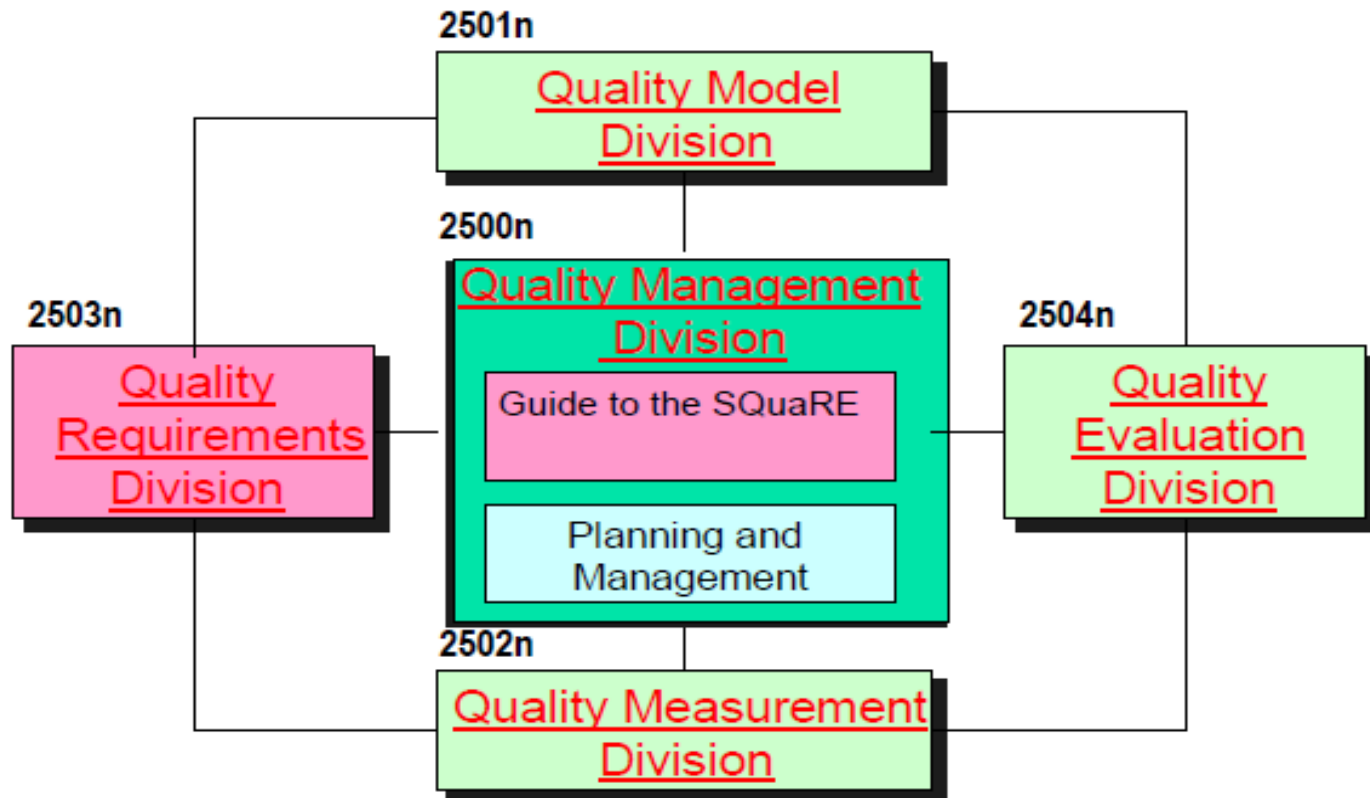
- **Como atingir a qualidade num produto de software ?**
 - A determinação do conjunto de características que atende às necessidades de seus usuários.
 - A forma de avaliar se tais características foram alcançadas num grau que satisfaça seus usuários.
- **Diversas iniciativas surgiram para buscar as respostas a essa questão.**
 - Série de normas **ISO/IEC 25000 (SQuaRE)** tratam definições relacionadas a qualidade de produto de software.

Qualidade de Produto de Software

- *Software product Quality Requirements and Evaluation (SQuaRE)*
 - Requisitos e avaliação de qualidade de produto de software.
 - Conjunto de normas que descrevem um modelo de qualidade, um processo de avaliação, uma diretriz para requisitos de qualidade e um conjunto de medidas de qualidade.



Qualidade de Produto de Software



Arquitetura SQuaRE

Qualidade de Produto de Software

ABNT NBR ISO/IEC 25000:2008 Guia do SQuaRE

Fornece uma visão geral do conteúdo do SQuaRE, de seus modelos de referência e definições, bem como o relacionamento entre os documentos, permitindo aos usuários do Guia um bom entendimento dessa série de normas, associado aos respectivo propósito de uso.

ABNT NBR ISO/IEC 25010:2008 Modelo de Qualidade

ABNT NBR ISO/IEC 25020:2009 Guia e modelo de referência para medição

Seleção e construção de medidas de qualidade de produto de software.

ABNT NBR ISO/IEC 25030:2008 Requisitos de qualidade

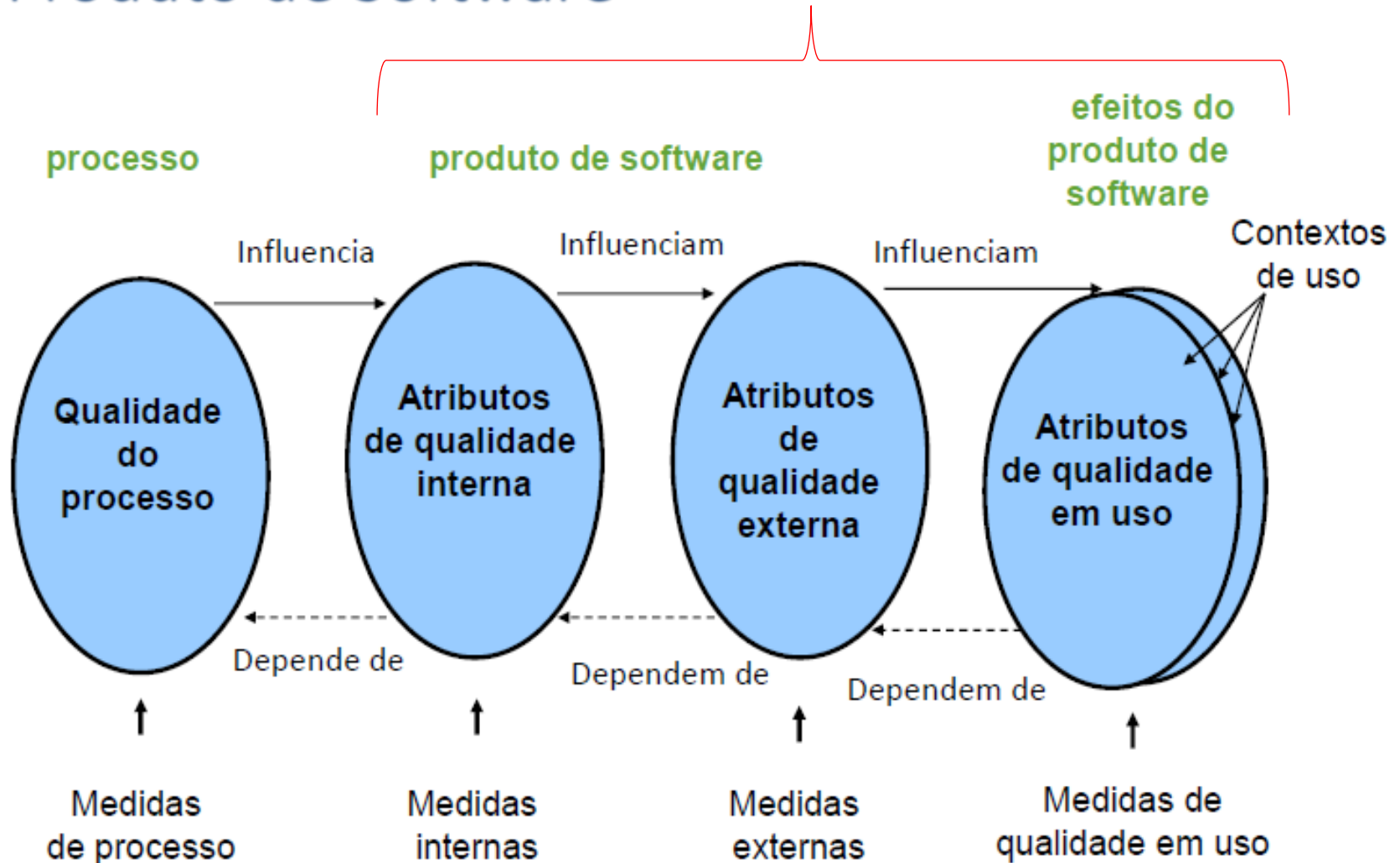
Fornece os requisitos e recomendações para especificação de requisitos de qualidade de produto de software.

ISO/IEC 25010 – Modelo de Qualidade de Produto de Software

Norma ISO/IEC 25010:2011

- - Define atributos de qualidade internos, externos e de uso do sistema.
- **Atributos internos:** aspectos que podem ser avaliados pela equipe de desenvolvimento.
 - Ex.: capacidade de manutenção.
- **Atributos externos:** aspectos que podem ser avaliados pela equipe de desenvolvimento sob o ponto de vista do usuário.
 - Ex. Eficiência.
- **Atributos do uso do sistema:** aspectos de qualidade do software em seu ambiente final, que só podem ser medidos pelo usuários finais.
 - Ex. Satisfação dos usuários.

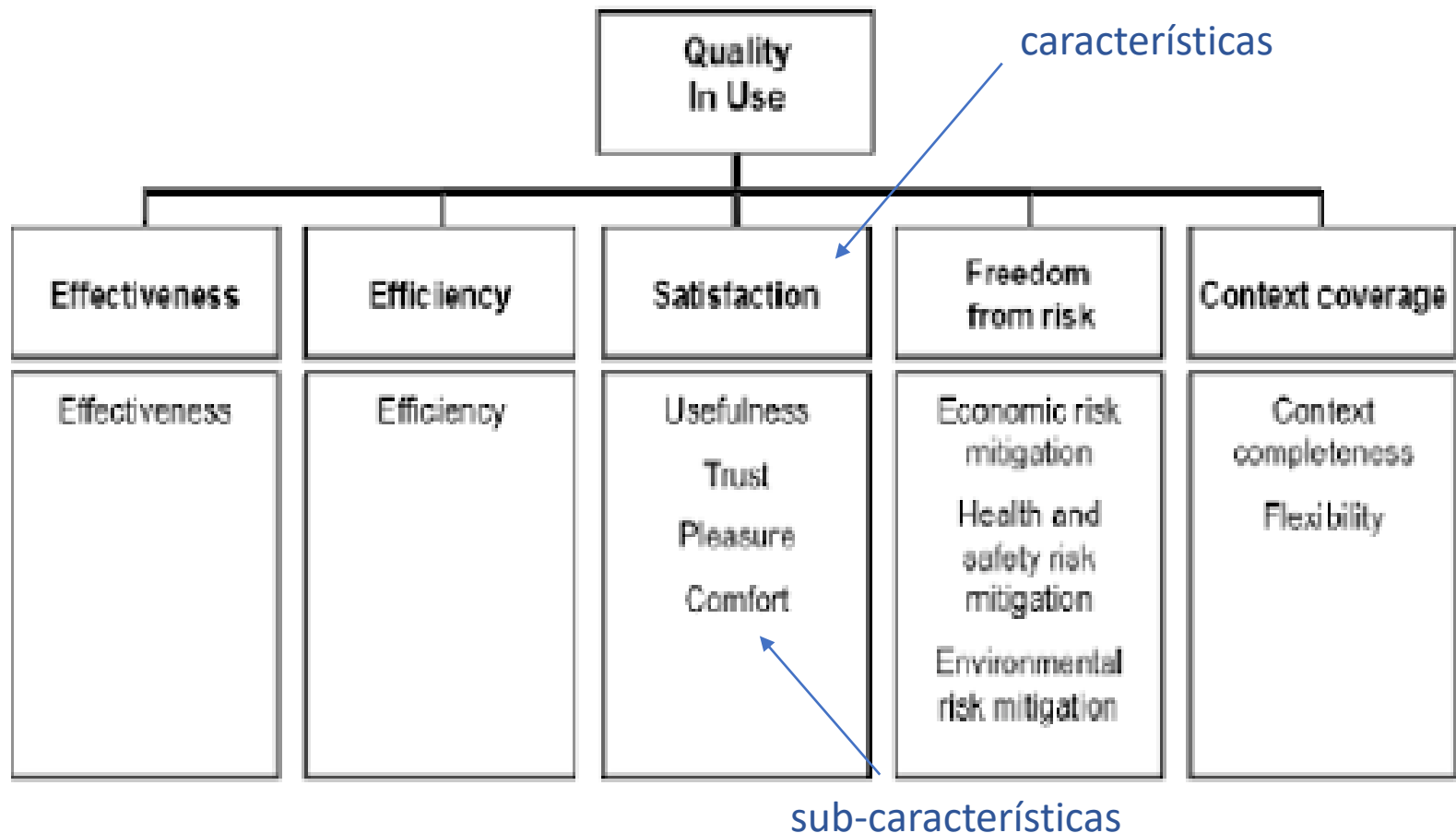
ISO/IEC 25010 – Modelo de Qualidade de Produto de Software



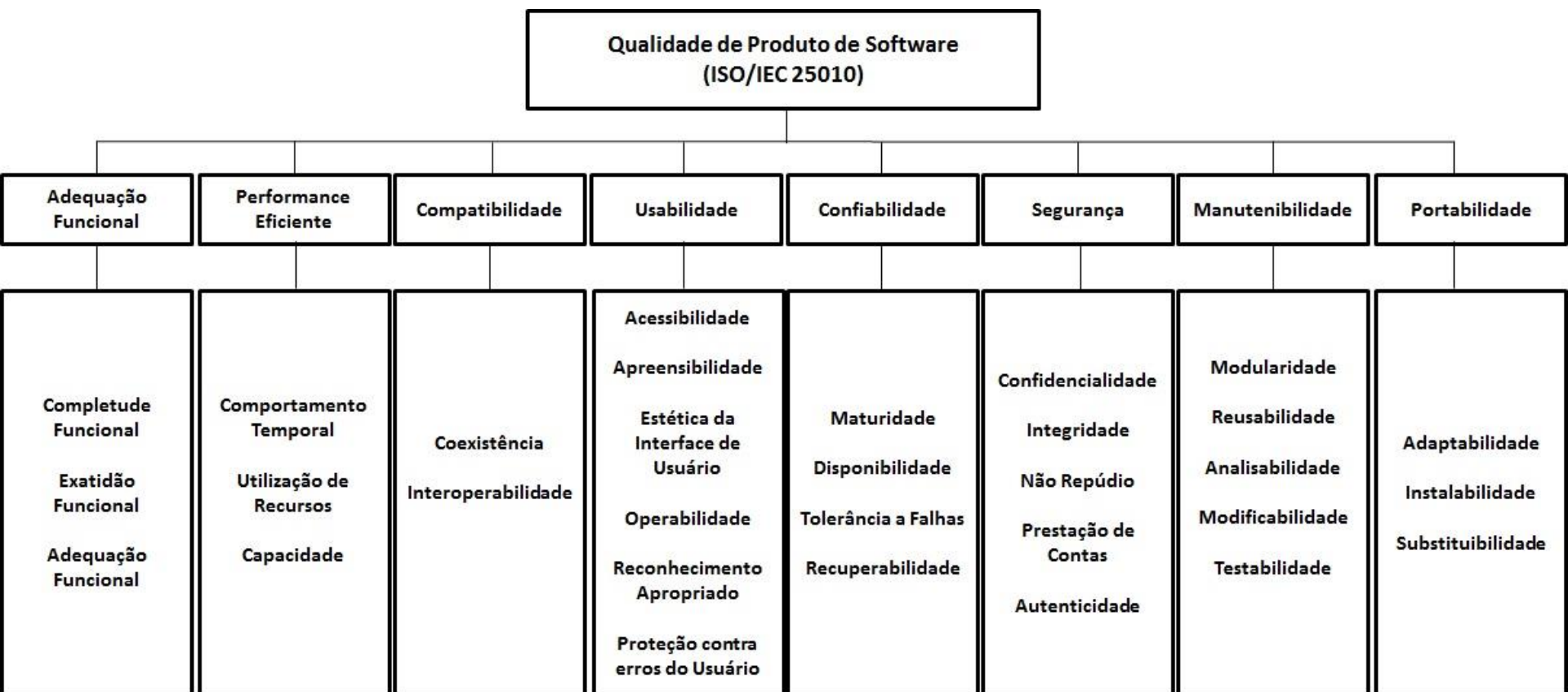
ISO/IEC 25010 – Modelo de Qualidade de Produto de Software

- Define **dois modelos de qualidade** que fornecem uma terminologia consistente para especificar, medir e avaliar a qualidade de produtos de software.
- **modelo de qualidade em uso:** composto de **cinco características** relacionadas a saída da interação quando um produto é usado em um contexto particular de uso.
- **modelo de qualidade de produto:** composto de **oito características** relacionadas à propriedades estáticas e dinâmicas do software.

ISO/IEC 25010 – Modelo de Qualidade de Produto de Software (Qualidade em Uso)



ISO/IEC 25010 – Modelo de Qualidade de Produto de Software (Qualidade de Produto)



ISO/IEC 25010 – Modelo de Qualidade de Produto de Software

- Pode não ser possível otimizar um sistema para todos esses atributos
 - Por exemplo, melhorar a robustez poderá levar à perda de desempenho.
- Portanto, o **plano de qualidade** deve definir os atributos de qualidade mais importantes para o software que está sendo desenvolvido.
- O plano também deve incluir uma definição **do processo de avaliação/medição de qualidade**
 - Uma forma acordada de avaliar se as características de qualidade estão presentes no produto.

Medição de Software

- A **medição** de software está preocupada com a derivação de um valor numérico para um atributo de um produto de software ou processo.
 - Permite comparações entre si e com os padrões que se aplicam a organização.
 - Permite tirar conclusões sobre a **qualidade** do software e também auxilia o **planejamento**.
- Embora algumas empresas introduzam programas de medição, a maioria das organizações ainda não fazem uso sistemático de medição de software.
- Usando medição de software um sistema pode ser avaliado usando uma variedade de métricas, e a partir dessa medição, deduzir um valor para o sistema.

Medição de Software

São quatro os principais papéis da Medições de Software:

Entender: Métricas ajudam a entender o comportamento e funcionamento de processos, produtos e serviços de software.

Avaliar: Métricas podem ser utilizadas para tomar decisões e determinar o estabelecimento de padrões, metas e critérios de aceitação.

Controlar: Métricas podem ser utilizadas para controlar processos, produtos e serviços de software.

Prever: Métricas podem ser utilizadas para prever valores de atributos.

Medição de Software

Métrica, Medida e Medição

Exemplo:

Deseja-se saber se uma pessoa está com seu peso ideal ou não.

Para tal, duas **medidas** são importantes: altura (H) e peso (P).

Ao tomar essas medidas, está-se efetuando uma **medição**.

A **métrica** “índice de massa corporal (IMC)” é calculada segundo a seguinte fórmula: $IMC = P / H^2$.

Medição de Software

Métricas de Software

- Qualquer **tipo de medida** que se relaciona com um processo ou produto de software.
 - ✓ As linhas de código em um programa (Lines of Code – LOC).
 - ✓ O índice Fog – comprimento médio de palavras e sentenças em documentos – Mede a legibilidade.
 - ✓ O número de pessoas-dia necessárias para desenvolver um componente.
- Permitem que o **produto de software** e o **processo de software** sejam quantificados.
- Podem ser usados para prever os atributos de produto ou para controlar o processo de software.

Medição da Qualidade do Produto de Software

Exemplo: Ferramenta CASE de modelagem UML (Jude)

- Características e sub-características:
 - Adequação Funcional
 - Completude Funcional
 - Usabilidade
 - Facilidade para o aprendizado
 - Facilidade para a operação
 - Facilidade de compreender
 - Eficiência
 - Em termos de tempo
 - Em relação à utilização de recursos
 - Portabilidade

Como medir?

Medição da Qualidade do Produto de Software

Exemplo: Ferramenta CASE de modelagem UML (Jude)

- Adequação Funcional
 - Completude Funcional
 - **Cobertura da funcionalidade implementada (CFI)**
 - Quão correta está a implementação funcional?
 - $CFI = 1 - NFNI / NFE$
 - NFNI: número de funções ausentes (não implementadas) ou incorretamente implementadas detectado na avaliação
 - NFE: número de funções descritas na especificação de requisitos
 - Quanto mais CFI estiver perto de 1.0, melhor.
 - $CFI < 0.8$, descartar a possibilidade de adoção do produto.

Medição da Qualidade do Produto de Software

Exemplo: Ferramenta CASE de modelagem UML (Jude)

- Adequação Funcional
 - Completude Funcional
 - **Cobertura das funcionalidades implementadas (CFI)**
 - Quão correta está a implementação funcional?
 - $CFI = 1 - NFNI / NFE$
 - NFNI: número de funções ausentes (não implementadas) ou incorretamente implementadas detectado na avaliação
 - NFE: número de funções descritas na especificação de requisitos
 - Quanto mais CFI estiver perto de 1.0, melhor.
 - $CFI < 0.8$, descartar a possibilidade de adoção do produto.

Adequação Funcional → Completude

Medida Externa

Nome da medida: *completude das funções*

Propósito: determinar a taxa de implementação das funções
definidas na especificação de requisitos

Fórmula: $1 - (\# \text{ funções faltantes} / \# \text{ funções especificadas})$

Interpretação: $0 \leq x \leq 1$; quanto mais próximo de 1, melhor

Entradas: Especificação de requisitos, relatórios de avaliação

Confiabilidade → Tolerância a falhas

Medida Externa

Nome da medida: *capacidade de evitar falhas*

Propósito: determinar controle de ocorrência de falhas

Fórmula: $(\# \text{ falhas evitadas} / \# \text{ casos de teste})$

Interpretação: $0 \leq x \leq 1$; quanto mais próximo de 1, melhor

Entradas: relatórios de teste e de operação

Usabilidade → Capacidade de ser aprendido (*Learnability*)

Medida Interna

Nome da medida: *completude do sistema de ajuda e documentação*

Propósito: determinar proporção de funções documentadas ao usuário

Fórmula: $(\# \text{ funções documentadas} / \# \text{ funções})$

Interpretação: $0 \leq x \leq 1$; quanto mais próximo de 1, melhor

Entradas: Especificação de requisitos, projeto, relatórios de revisão

usabilidade → Capacidade de ser aprendido (*Learnability*)

Medida Externa

Nome da medida: *eficácia de sistema de ajuda e documentação*

Propósito: determinar proporção de documentação efetiva ao usuário

Fórmula: $(\# \text{ funções executadas com sucesso (*1)} / \# \text{ funções})$

Interpretação: $0 \leq x \leq 1$; quanto mais próximo de 1, melhor

Entradas: relatório de operação

*1 após consulta do usuário ao sistema de ajuda

Eficácia

Medida Qualidade em uso

Nome da medida: *tarefas completadas*

Propósito: determinar proporção de tarefas completadas

Fórmula: $(\# \text{ tarefas completadas} / \# \text{ tarefas tentadas})$

Interpretação: $0 \leq x \leq 1$; quanto mais próximo de 1, melhor

Entradas: relatório de operação, registro de histórico de uso.

ISO/IEC 25020:2009 – Guia e Modelo de Referência para Medição

ISO/IEC 25021:2009 – Elementos de Medida de Qualidade

ISO/IEC 25022:2009 – Medição da qualidade interna

ISO/IEC 25023:2009 - Medição da qualidade externa

ISO/IEC 25024:2009 - Medição da qualidade em uso

Referências

- Engenharia de Software – Notas de Aulas. Prof. Ricardo Falbo. UFES.
- Waslawick, Raul. Engenharia de Software: Conceitos e práticas. Ed. Campus. 2013.