

# Qualidade de Software

## Visão Geral e Qualidade de Produto



**Profa Rosana Braga - 1/2017**

**Material elaborado por docentes do grupo de Engenharia de  
Software do ICMC/USP**

**-Atualizado pela Profa. Simone Souza em 2016**

# Qualidade de Software

---

---

- O que é qualidade de software?



# Necessidade de Qualidade

---

---

- Na década de 80, o fator qualidade emergiu como uma necessidade básica na luta pelo mercado cada vez mais

*“Não basta vender barato, as novas regras de mercado são orientadas à produção de bens e serviços **com qualidade**, prazo de entrega determinado, atendimento correto, além de um baixo custo” (Werneck 1994).*

# Definição de Qualidade

---

---

- Visão Popular
  - Algo abstrato
  - Perfeição
  - Luxo e questão de gosto
  
- Visão Profissional
  - Conformidade aos requisitos
  - Adequação ao uso

# Visão de Qualidade de Software

---

---

- Defeito zero
- Grande número de funções
- Codificação elegante
- Alto desempenho
- Baixo custo de desenvolvimento
- Desenvolvimento rápido
- Facilidade para o usuário

*(voltaremos a isso já...)*

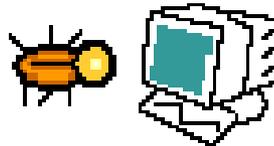
# Bugs históricos

---

---

- Piores bugs da história do software:

- <http://www.wired.com/software/coolapps/news/2005/11/69355?currentPage=all>



# Bugs históricos

---

---

## ■ Mariner I – 1962

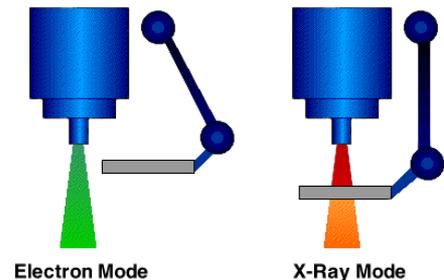
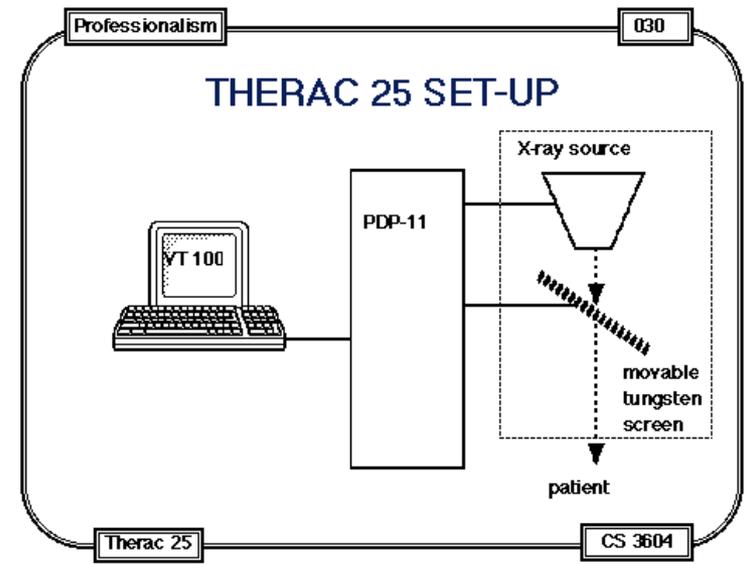
- Missão observar planeta Vênus
- Fórmula matemática foi equivocadamente transcrita para o computador
- Desviou do curso e foi destruído 4 min após lançamento
- Prejuízo: US\$ 18,5 mi



# Bugs históricos

## ■ Therac-25 – 1985/1987

- Dispositivo de terapia por radiação sobre células cancerosas
- Libera doses letais de radiação em vários consultórios médicos
- Condição de disputa no SO
- 5 mortes, várias pessoas feridas



# Bugs históricos

## ■ **Míssil Patriot – 1991**

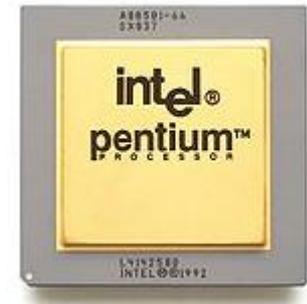
- Míssil de defesa
- Dhahran, Arábia Saudita
- Erro de software no relógio do míssil: a cada 100 horas o relógio interno do sistema desviava um terço de segundo
- Recomendação era reiniciar o sistema de tempos em tempos
- Resultado: 600 metros de erro na distância em uma interceptação
- 28 soldados americanos mortos



# Bugs históricos

## ■ Divisão de pontos flutuantes nos processadores Pentium da Intel – 1993

- Erro em divisões dentro de uma faixa de números (erro ~0,006% no arredondamento)
- 3 a 5 milhões de peças com defeito
- *Recall* para todos que quiseram trocar
- Custou à Intel US\$ 475 milhões



$$\frac{4195835}{3145727} = 1.333820449136241002$$

$$\frac{4195835}{3145727} = 1.333739068902037589$$

# Bugs históricos

## ■ Ariane 5 voo 501 – 1996

- Levou uma década de desenvolvimento e custou 7 bilhões de dólares.
- Foguete com código reutilizado do Ariane 4 (outro hardware);
- *Overflow* de inteiro: conversão de *float* de 64-bits para inteiro 16-bits com sinal;
- O processador primário do foguete sobrecarrega os motores que se desintegraram em 40 segundos;
- Não tripulado (sem vítimas); prejuízo de US\$ 370 milhões



# Bugs históricos

## ■ Bug do milênio (Y2K) – 2000

- Datas com apenas 2 dígitos para o ano
- Uma das maiores histerias da história
- Ao virar o ano 2000, a preocupação era que contasse como 1900
- Entre US\$ 300 e US\$ 500 bi no mundo todo



# Bugs históricos

## ■ Toyota Prius – 2010

- Problema no software do sistema ABS de freios – acelerador fica preso, dificultando desaceleração
- Recall de 400.000 veículos
- ~ US\$ 2 bilhões de prejuízo, desvalorização de 15% nas ações



# Bugs históricos

## ■ Play Station Network - 2011

- Invasão do sistema
- Dados privados e de cartão de crédito de ~70 mi de pessoas foram roubados



PLAYSTATION®  
Network



26/04/2011 18h03 - Atualizado em 26/04/2011 19h59

### Dados pessoais de usuários da PSN foram roubados, admite Sony

Empresa não descarta roubo de informações de cartões de crédito.  
PlayStation Network está fora do ar há seis dias.

# Bugs Recentes

---

---

## ■ Malas retidas - ano bissexto – 2016

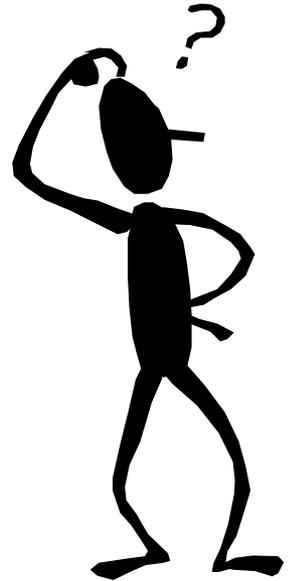
- Aeroporto de Düsseldorf – Alemanha
- Sistema de transporte de bagagem (2015) custou 70 milhões de euros não reconheceu o 29 de fevereiro.
- Esteiras pararam – despacho manual de bagagens
  - transportadas entre 25 mil e 50 mil malas por dia!
  - 1.200 malas acabaram não sendo despachadas
- *“A empresa responsável pelo software fez a reprogramação necessária, incluindo o dia 29 de fevereiro de 2016 no sistema. Daqui a quatro anos, este procedimento será novamente necessário, afirmou o porta-voz”*

# Bugs históricos

---

---

- Por que essas falhas ocorrem?
- Poderiam ser evitadas?
- De quem é a culpa?



# Atividades do Processo de Desenvolvimento



## Uma Visão Genérica: 3 Fases

### 1. Definição - “o que”

- Engenharia do Sistema
- Planejamento do Projeto
- Engenharia de Requisitos

### 2. Desenvolvimento - “como”

- Projeto
- Geração do Código
- Teste

### 3. Manutenção

## Atividades Guarda-Chuva

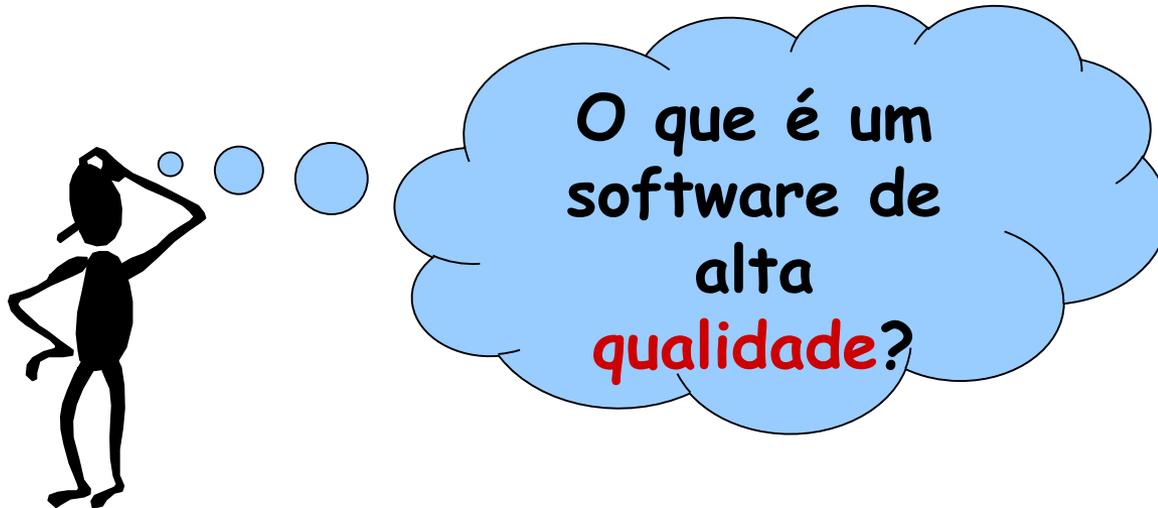
- Controle de Projeto
- Revisões Técnicas Formais
- **Garantia de Qualidade**
- Gerenciamento de Configuração
- Gestão de Reutilização
- Medição
- Gestão de Risco

# Engenharia de Software

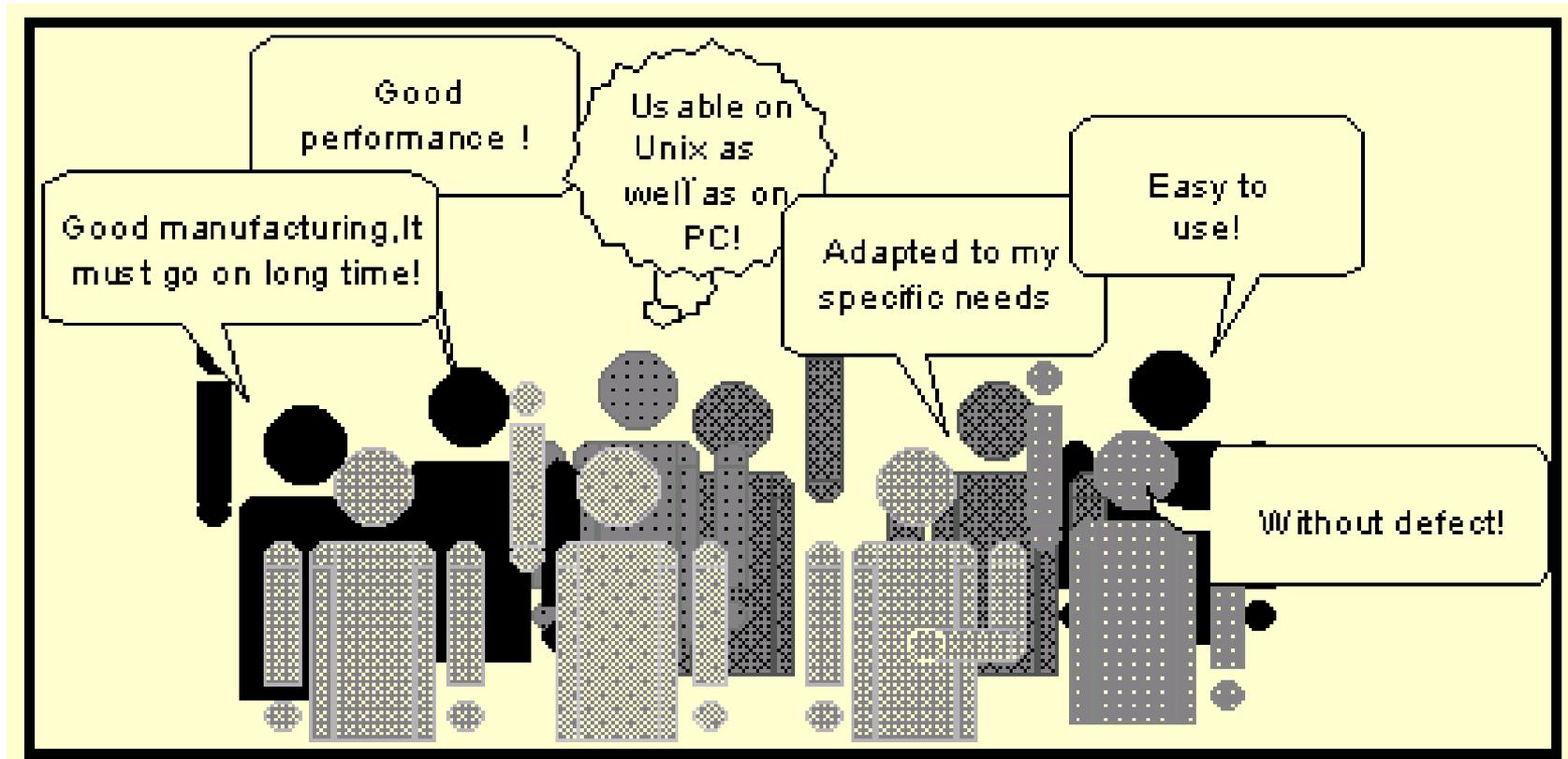
---

---

- A **Engenharia de Software** é uma disciplina que aplica os princípios de engenharia com o objetivo de produzir software de **alta qualidade** a **baixo custo**.



# Qualidade de Software



# A Qualidade depende do Tipo de Aplicação

Sistema de Missão Crítica

Software Embarcado

## EXEMPLO

*Qualidade Importante*



Fazer aquilo que eu quero



Comportar-se com precisão



Ser fácil de usar



Rodar bem no hardware



Fácil de alterar



# A Qualidade depende do Tipo de Aplicação

Software para Folha de Pagamento

Software Interativo com o usuário

## EXEMPLO

*Qualidade Importante*



Fazer aquilo que eu quero



Se comportar com precisão



Ser fácil de usar



Rodar bem no hardware



Fácil de alterar



# Qualidade de Software

---

---

- Qualidade é um termo que pode ter diferentes interpretações.



# Qualidade de Software

---

---

“A qualidade de um projeto engloba o grau de **atendimento às funções e características especificadas** no modelo de requisitos”

[Pressman,2011]

satisfação do usuário = produto compatível + boa  
qualidade + entrega no prazo + entrega dentro do  
orçamento

# Aspectos Importantes da Definição de Qualidade

---

---

1- Os requisitos de software são a base a partir da qual a qualidade é medida.

A falta de conformidade aos requisitos significa falta de qualidade.

# Aspectos Importantes da Definição de Qualidade

---

---

2- Padrões especificados definem um conjunto de critérios de desenvolvimento que orientam a maneira segundo a qual o software passa pelo trabalho de engenharia.

Se os critérios não forem seguidos, o resultado seguramente será a falta de qualidade.

# Aspectos Importantes da Definição de Qualidade

---

---

3- Existe um conjunto de **requisitos implícitos** que frequentemente não são mencionados na especificação. Por exemplo, o desejo de uma boa **manutenibilidade**.

Se o software atende aos requisitos explícitos, mas falha nos requisitos implícitos, a qualidade é suspeita.

# Aspectos Importantes da Definição de Qualidade

---

---

- Existe, ainda, uma visão de qualidade de software do ponto de vista gerencial.
  - O software é considerado de qualidade desde que possa ser desenvolvido dentro do prazo e do orçamento especificados.

# A Qualidade depende do Ponto de Vista



**usuário**



**desenvolvedor**



**gerente**

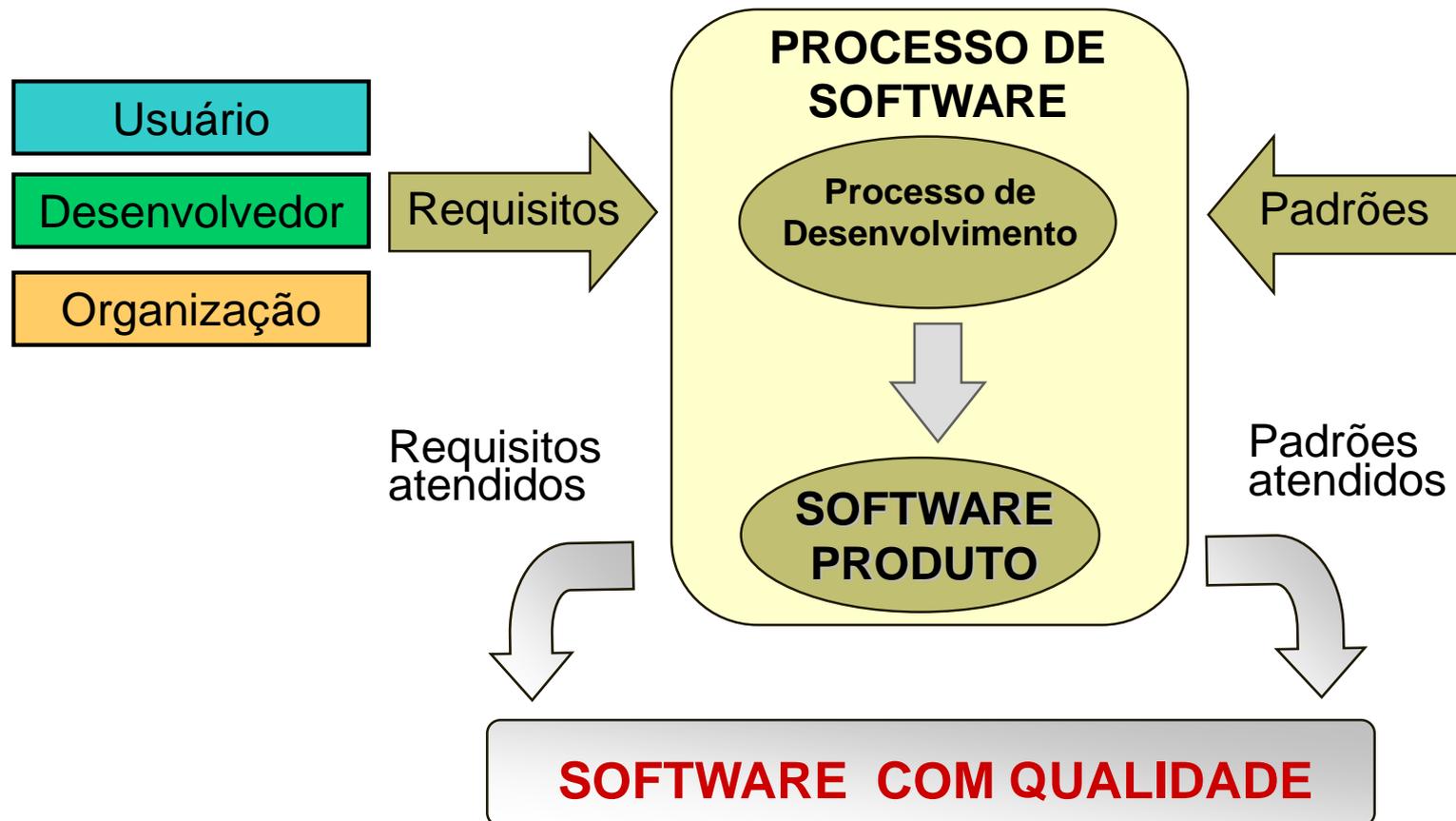
O interesse fica concentrado principalmente no uso do software: facilidade de uso, requisitos atendidos.

A qualidade fica mais voltada às características internas do software: legibilidade, testabilidade, eficiência.

A qualidade do produto não pode ser desvinculada dos interesses da organização: custos e prazos.

# Requisitos de Software

## Base da Qualidade



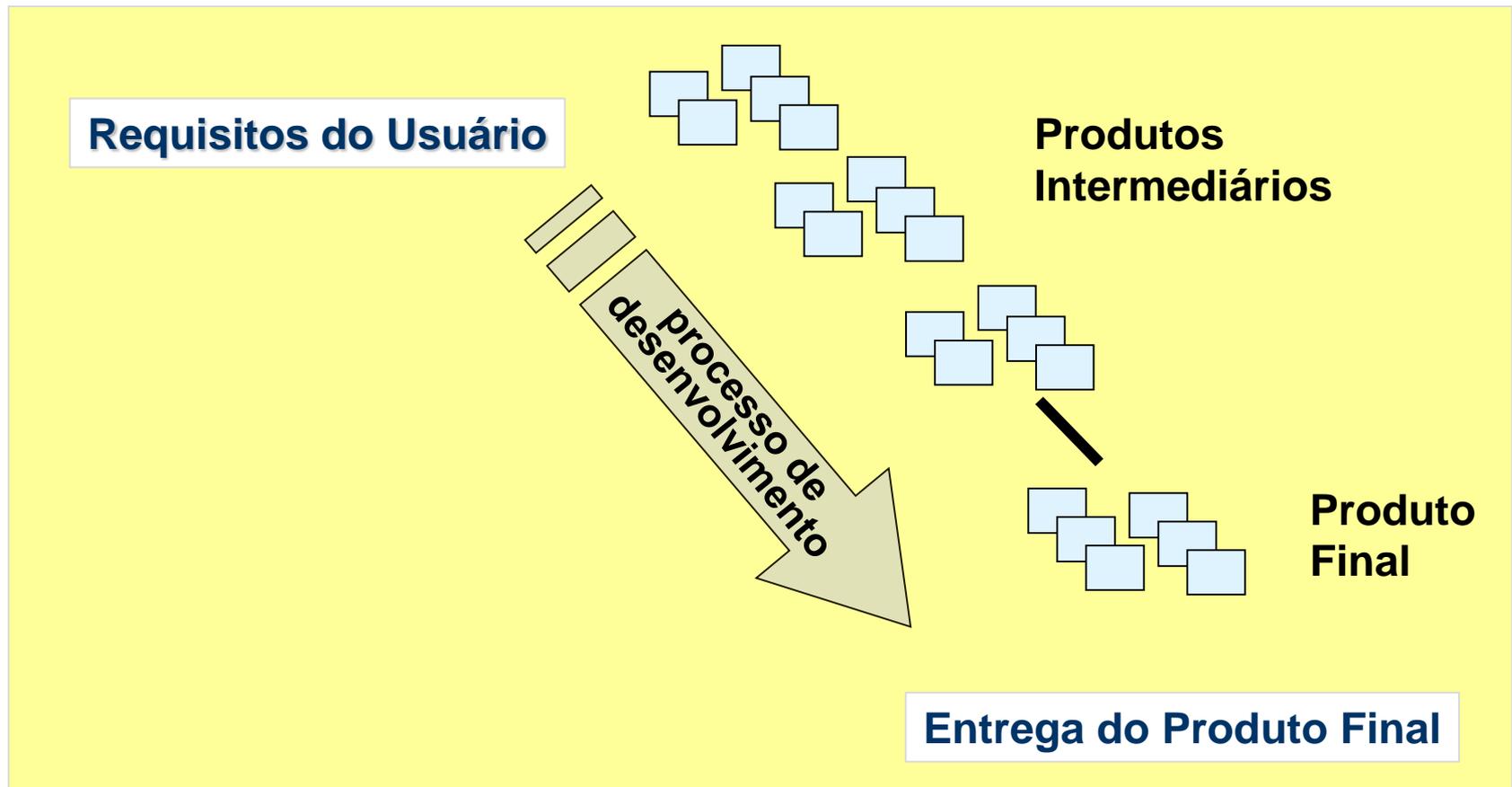
# Incorporação da Qualidade

---

---

- Raramente a qualidade pode ser incorporada ao produto final após o **término** do processo de desenvolvimento.
  - Não pode ser imposta depois que o produto estiver finalizado.

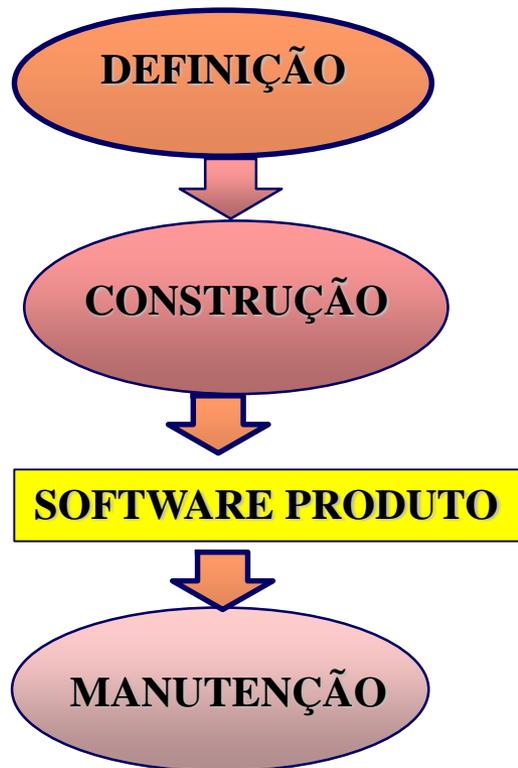
# Incorporação da Qualidade



# Qualidade de Software

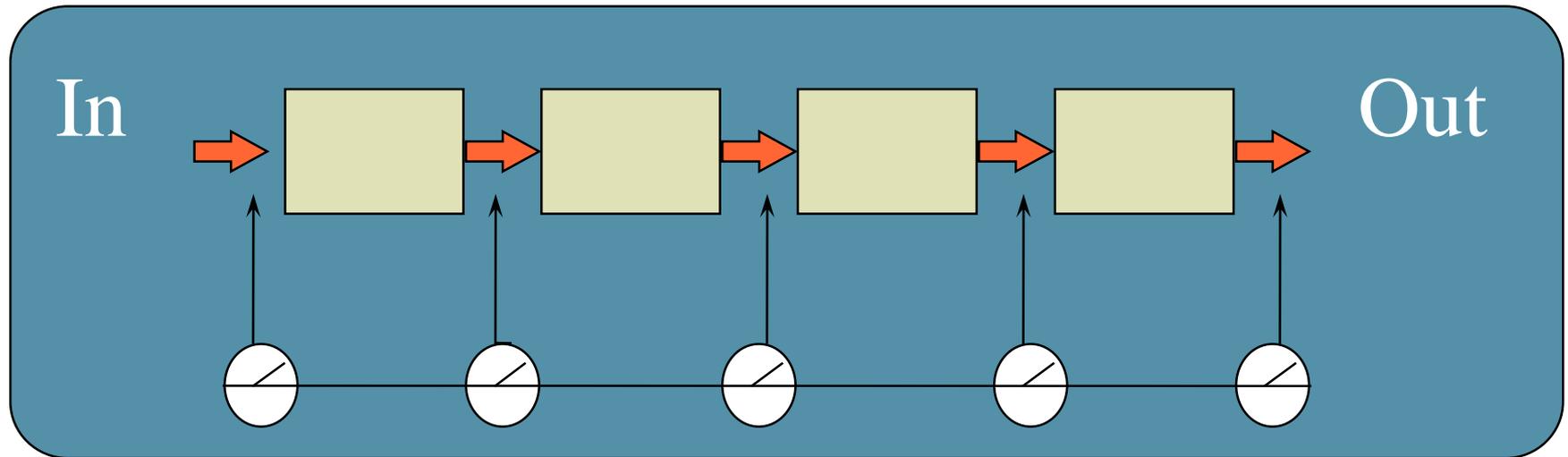
---

---

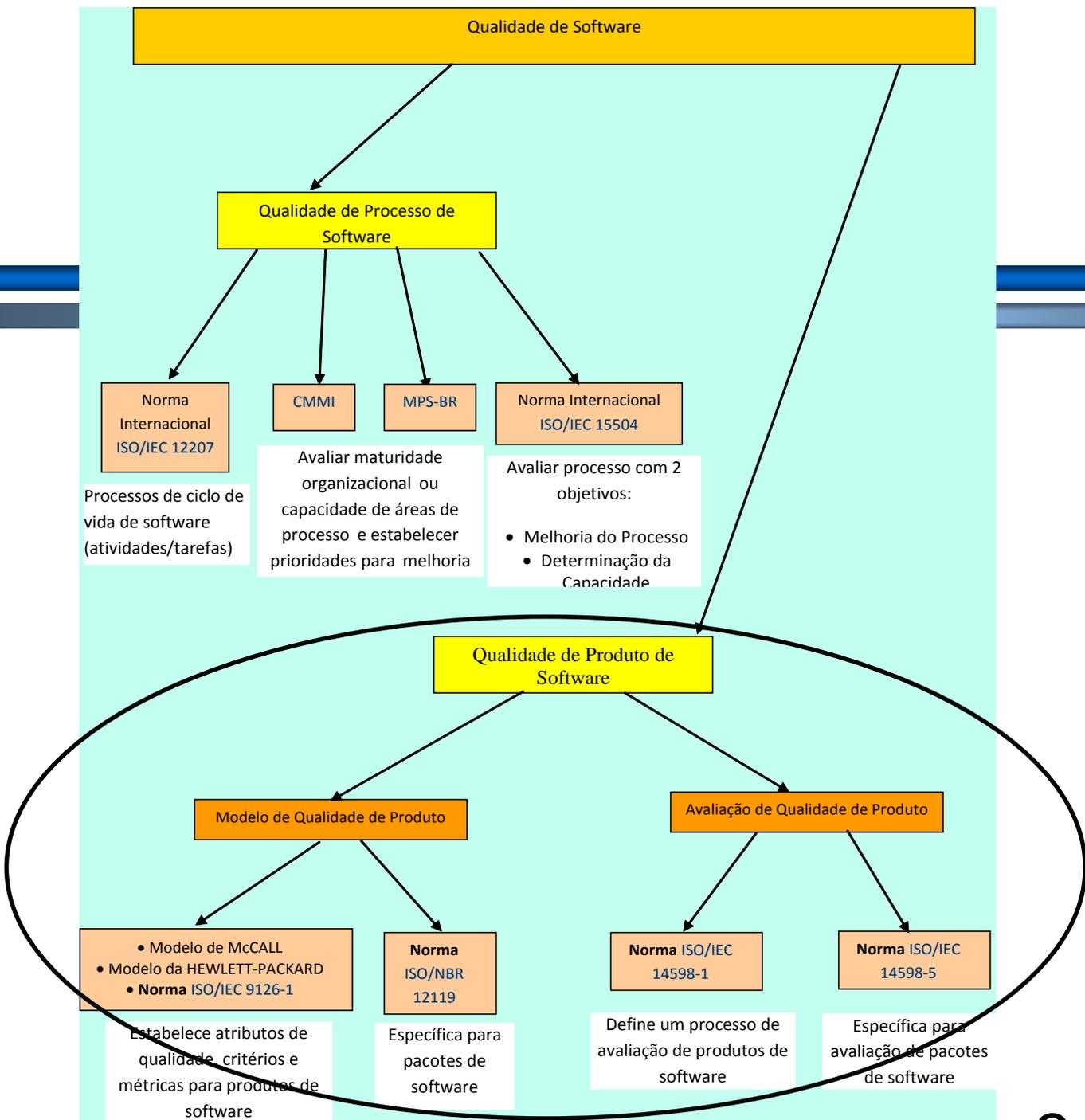


- A qualidade não pode ser incorporada ao produto depois de pronto.
- Para que a qualidade possa ser efetivamente incorporada ao **produto**, ela deve ser um objetivo constante do **processo de desenvolvimento**.

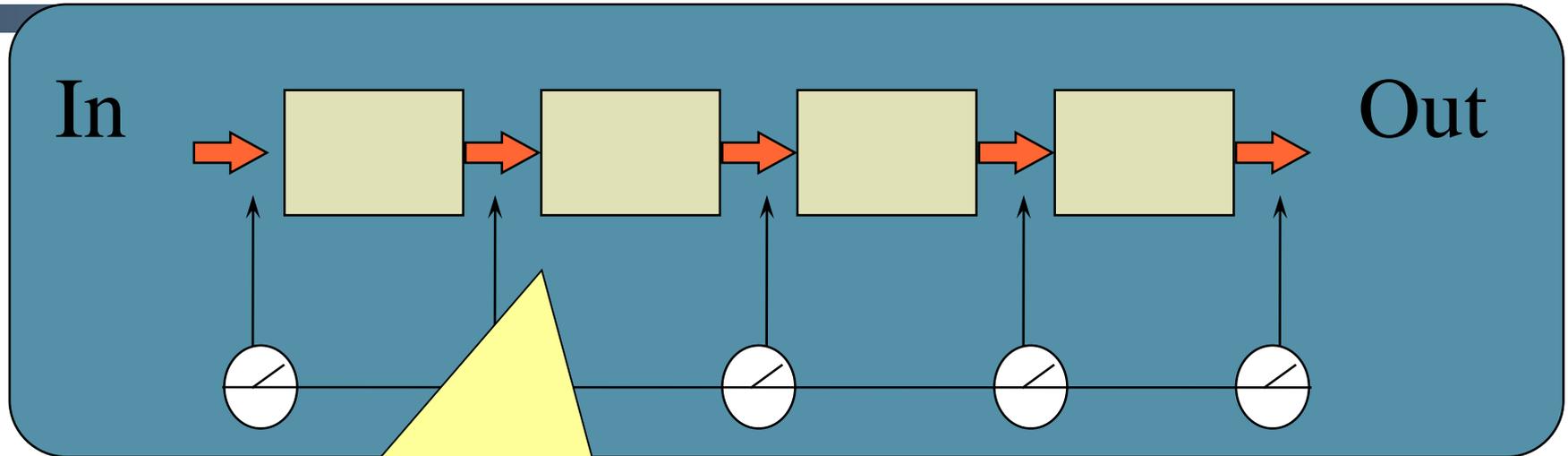
# Garantia de Qualidade



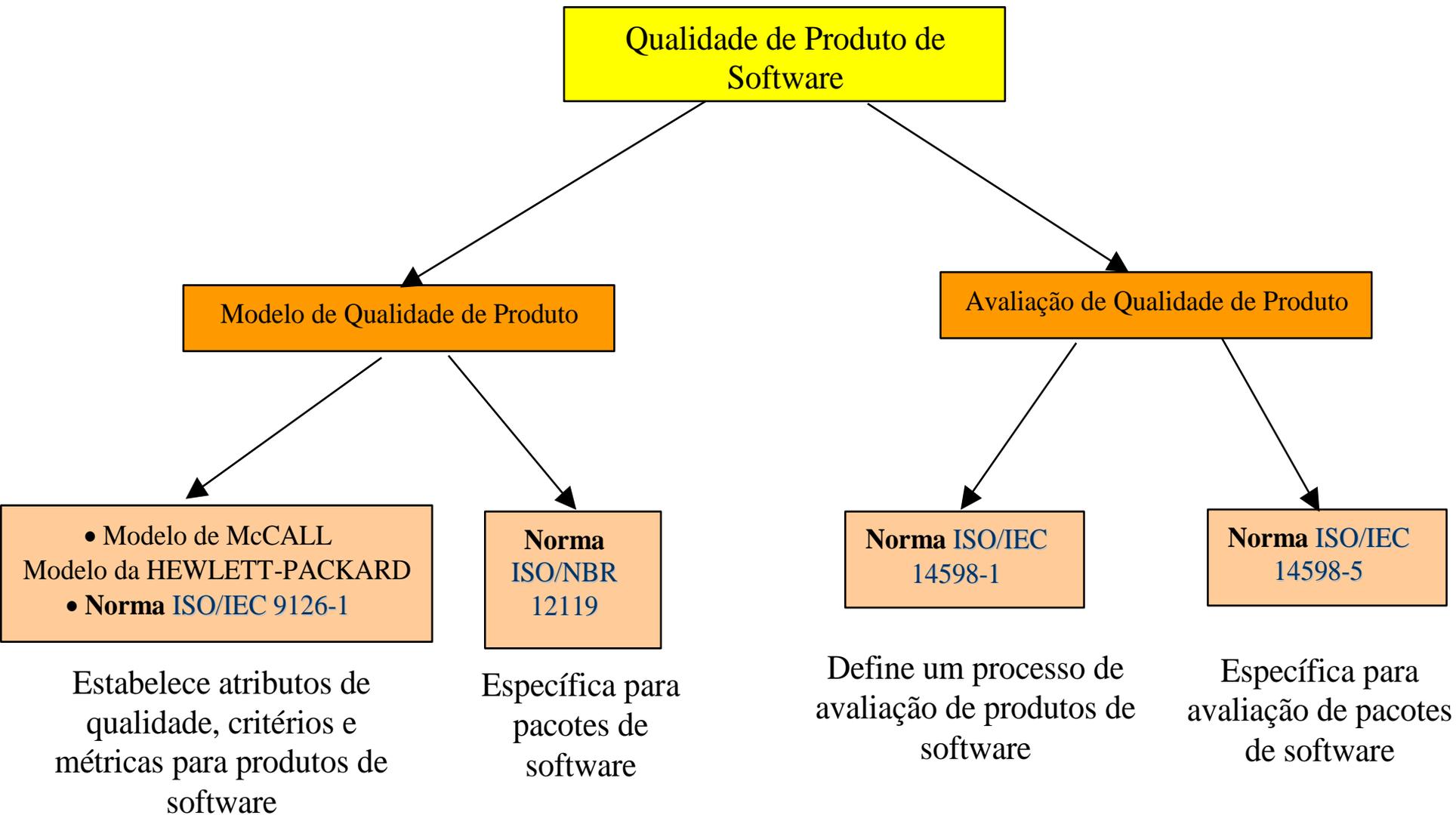
A **Garantia da Qualidade de Software (SQA)** promove visibilidade sobre o desempenho do **processo** utilizado pelo projeto assim como sobre a qualidade dos **produtos** gerados.



# Garantia de Qualidade



Como avaliar a qualidade dos  
**produtos** ???



# Garantia de Qualidade

---

---

## ■ Algumas Normas

- ISO 9126 – Qualidade de produto de software
- ISO 12207 – Qualidade do processo de software
- ISO 27000 – Segurança da informação
- IEEE 829 – Documentação de testes
- IEEE 1028 – Revisão de software
- IEEE 1044 – Classificação de incidentes

# Norma ISO/IEC 9126

---

---

É uma referência mundial para qualidade de software

*ISO: The International Standardization Organization,*

fundada em 1947, coordena o trabalho de 127 países membros para promover a padronização de normas técnicas em âmbito mundial

*IEC: The International Electrotechnical Commission,*

fundada em 1906, conta com mais de 50 países e publica normas internacionais relacionadas com eletricidade, eletrônica e áreas relacionadas

# Norma ISO/IEC 9126

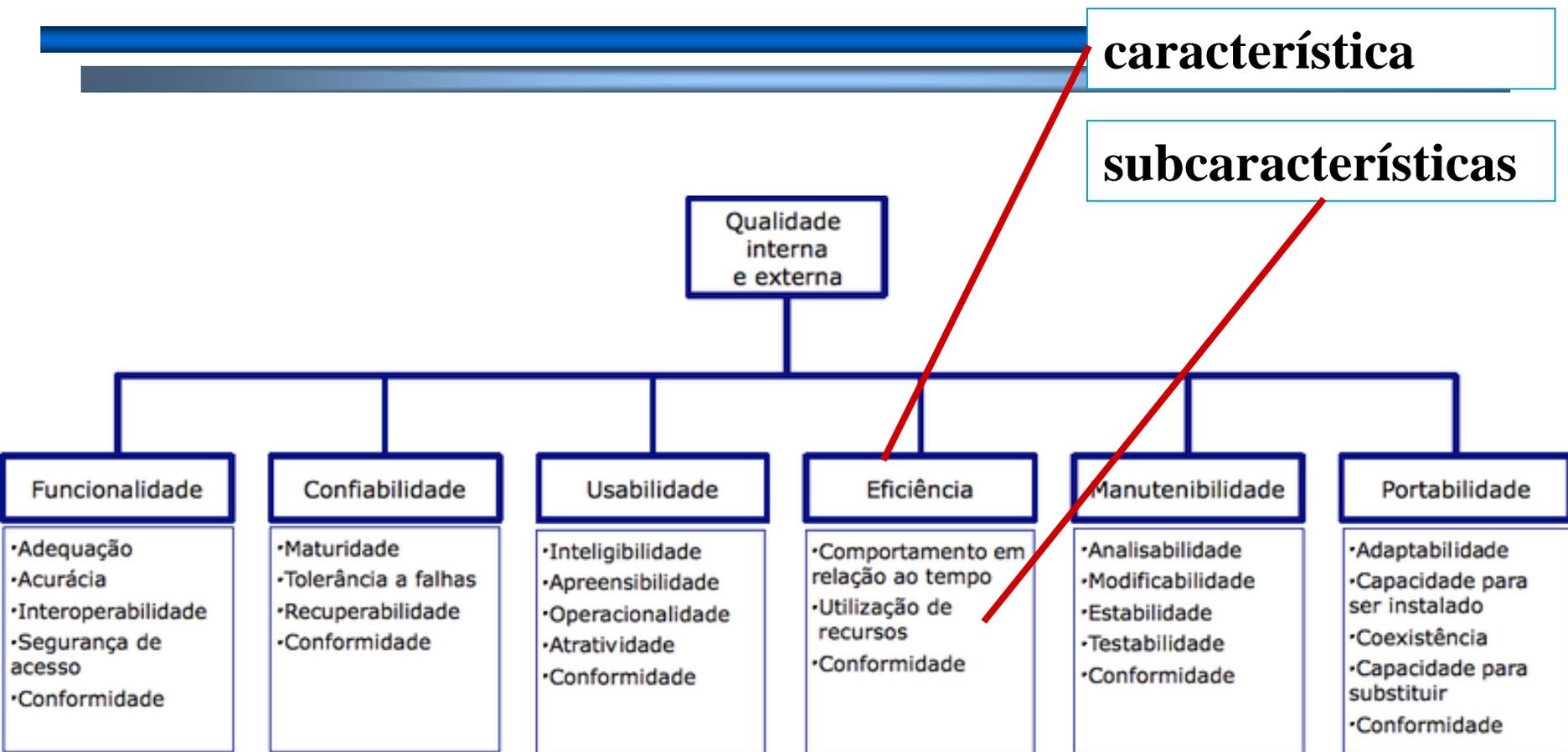
---

---

- Baseada em três níveis:
  - Características, Sub-características e Métricas.
    - Cada **característica** é refinada em um conjunto de **sub-características** e cada sub-característica é avaliada por um conjunto de **métricas**.

**NOTA IMPORTANTE:** a ISO/IEC 9126 foi substituída em 2011 pela ISO/IEC 25010

# Norma ISO/IEC 9126



# Norma ISO/IEC 9126

## **FUNCIONALIDADE - Satisfaz as necessidades implícitas e explícitas do usuário?**

### **SUBCARACTERÍSTICA**

- Adequação
- Acurácia
- Interoperabilidade
- Conformidade
- Segurança de Acesso

### **PERGUNTA-CHAVE**

- É adequado as necessidades do usuário?
- Faz o que foi proposto de forma correta?
- É capaz de interagir com os sistemas especificados?
- Está de acordo com as normas, leis, etc. relacionadas à funcionalidade?
- Evita acesso não autorizado a programas e dados?

# Norma ISO/IEC 9126

**CONFIABILIDADE - o software, durante um período de tempo, funciona de acordo com as condições pré-estabelecidas?**

## **SUBCARACTERÍSTICA**

- **Maturidade**
- **Tolerância a Falhas**
- **Recuperabilidade**
- **Conformidade**

## **PERGUNTA-CHAVE**

- Com que frequência apresenta falhas?**
- Ocorrendo falhas, como ele reage?**
- É capaz de recuperar dados após uma falha?**
- Está de acordo com as padrões, normas, etc. relacionadas à confiabilidade?**

# Norma ISO/IEC 9126

## USABILIDADE – O software é fácil de usar?

### SUBCARACTERÍSTICA

- Intelegibilidade
- Apreensibilidade
- Operacionalidade
- Atratividade
- Conformidade

### PERGUNTA-CHAVE

- É fácil entender os conceitos utilizados?
- É fácil aprender a usar?
- É fácil operar e controlar?
- É atrativo ao usuário?
- Está de acordo com as padrões, normas, etc. relacionadas à usabilidade?

# Norma ISO/IEC 9126

## EFICIÊNCIA – O software não desperdiça recursos?

### SUBCARACTERÍSTICA

- Comportamento em Relação ao Tempo
- Comportamento em Relação aos Recursos
- Conformidade

### PERGUNTA-CHAVE

- Qual é o tempo de resposta e de processamento?
- Quanto recurso usa? Durante quanto tempo?
- Está de acordo com as normas, leis, etc. relacionadas à eficiência?

# Norma ISO/IEC 9126

## MANUTENIBILIDADE – O software é fácil de alterar?

### SUBCARACTERÍSTICA

### PERGUNTA-CHAVE

- **Analisabilidade**      **É fácil encontrar uma falha, quando ocorre?**
- **Modificabilidade**      **É fácil modificar e remover defeitos?**
- **Estabilidade**      **Existe risco de efeitos inesperados quando se faz alterações?**
- **Testabilidade**      **É fácil testar o software modificado?**
- **Conformidade**      **Está de acordo com as normas, leis, etc.? relacionadas à manutenibilidade?**

# Norma ISO/IEC 9126

## PORTABILIDADE - É fácil de usar em outro ambiente?

### SUBCARACTERÍSTICA

- Adaptabilidade
- Capacidade para ser instalado
- Capacidade para substituir
- Conformidade
- Co-existência

### PERGUNTA-CHAVE

- É fácil adaptar a ambientes diferentes?
- É fácil instalar?
- É fácil usar para substituir outro?
- Está de acordo com as normas, leis, etc. relacionadas à portabilidade?
- Pode coexistir com outros produtos independentes compartilhando recursos?

# Métricas de Qualidade

---

---

- Como medir a qualidade de um software?
  - Métricas dinâmicas
    - Durante teste ou uso do sistema
  - Métricas estáticas
    - Tamanho do código, tamanho de identificadores

# Métricas de Qualidade

---

---

- Exemplos de métricas
  - Fan-in/fan-out
  - Comprimento de código
  - Complexidade ciclomática
  - Comprimento de identificadores
  - Profundidade de aninhamento condicional
  - Índice Fog

# Outras normas de qualidade

---

---

- **ISO/IEC 12119**, estabelece os requisitos de qualidade para pacotes de software e instruções para teste, considerando esses requisitos
- **ISO/IEC 14598-5**, define um processo de avaliação da qualidade de produto de software

# Exercício:

---

---

- Escolha um produto de software que você usa e faça uma análise de sua qualidade, considerando cada característica da ISO/IEC 9126.
- Identifique quais requisitos de qualidade que você consideraria caso fosse especificar a aquisição/construção desse produto.

