

Qualidade de Software

Simone Senger Souza
srocio@icmc.usp.br

ICMC/USP

Atividades do Processo de Desenvolvimento



Atividades Guarda-Chuva

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

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

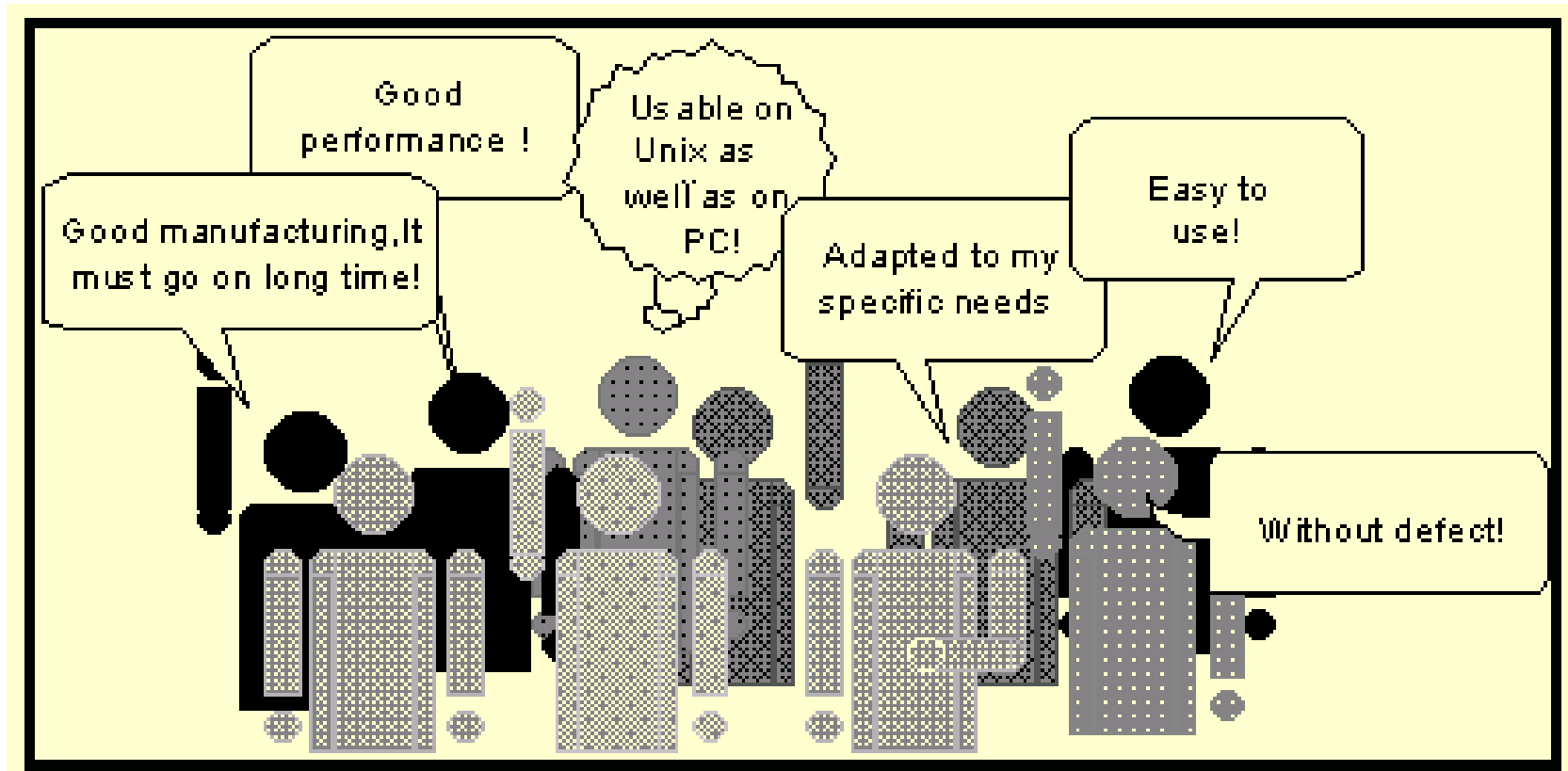
3. Manutenção

Qualidade de Software

- O que é qualidade?
- Como medir?



Qualidade de Software



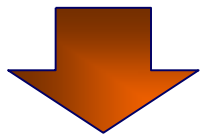
A Qualidade depende do Tipo de Aplicação

Sistema de Missão Crítica

Software Embarcado

EXEMPLO

Qualidade Importante



Fazer aquilo que eu quero



Comportar-se com precisão



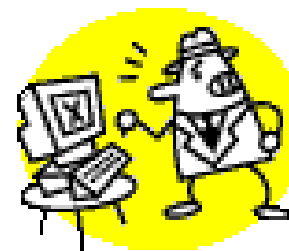
Ser fácil de usar



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

“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- Existe um conjunto de **requisitos implícitos** que freqüentemente 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

3 - 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

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



desenvolvedor

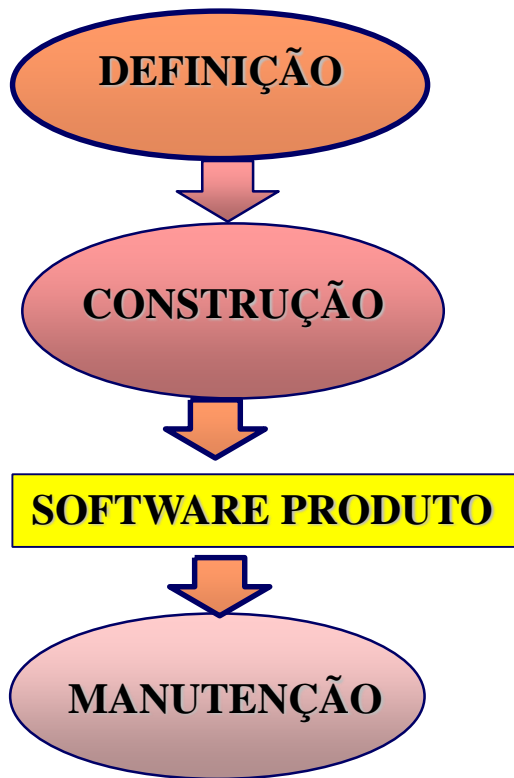
A qualidade fica mais voltada às características internas do software: legibilidade, testabilidade, eficiência.



gerente

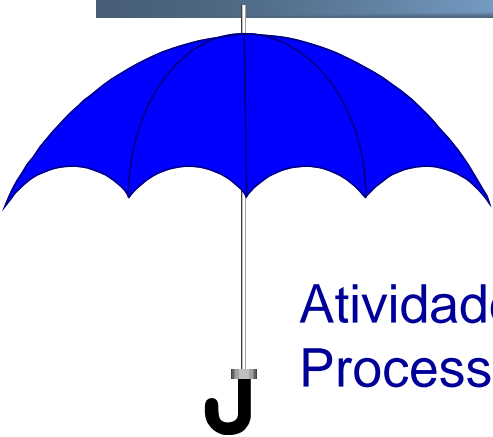
A qualidade do produto não pode ser desvinculada dos interesses da organização: custos e prazos.

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 de Software



Atividades Guarda-Chuva do
Processo de desenvolvimento:

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



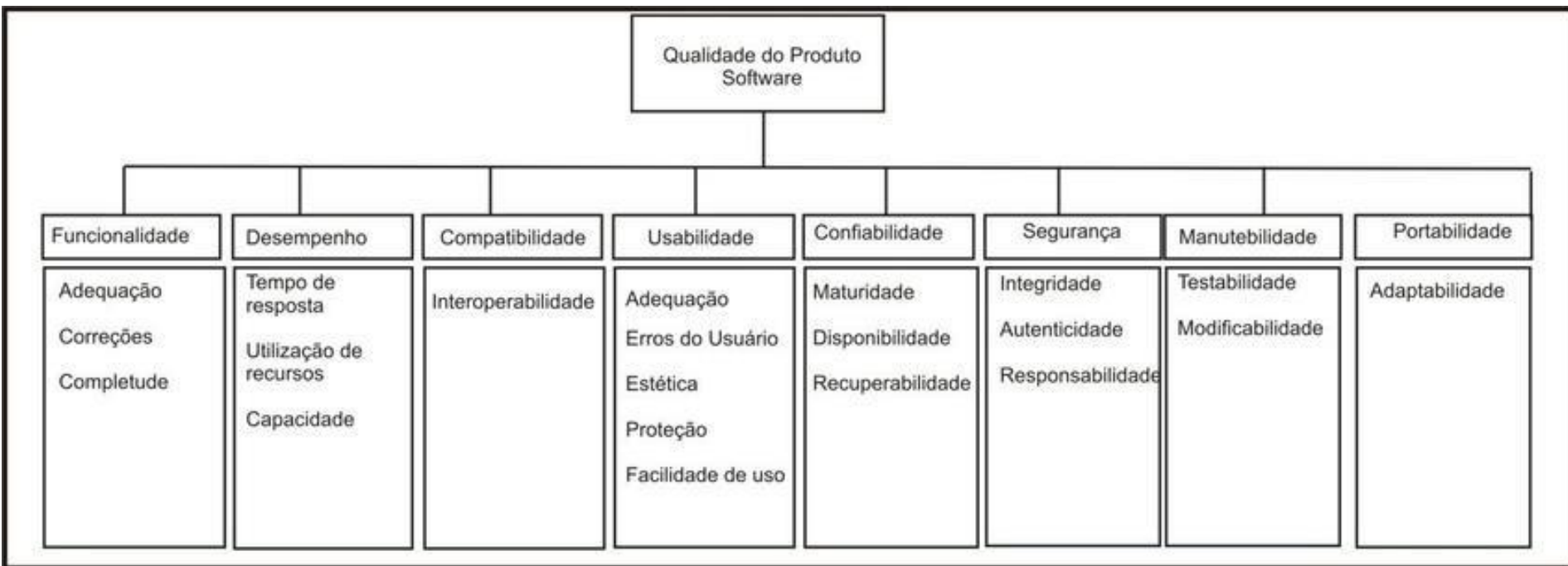
Métodos técnicos
Revisões técnicas formais
Teste de software
Aplicação de padrões
Controle de mudanças
Medição
Manutenção de registros

Avaliação da Qualidade

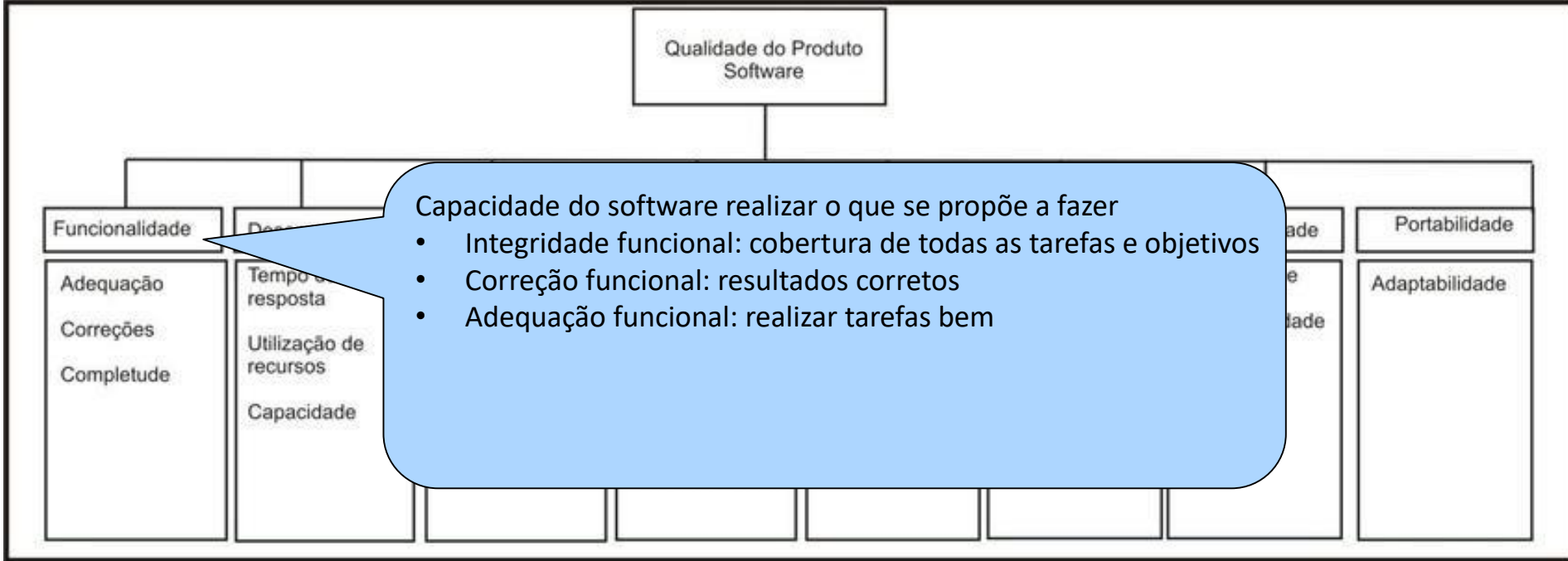
■ Algumas Normas

- [ISO 25010 – Qualidade de produto de software](#)
 - ISO 9126 – Qualidade de produto de software (old)
- 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

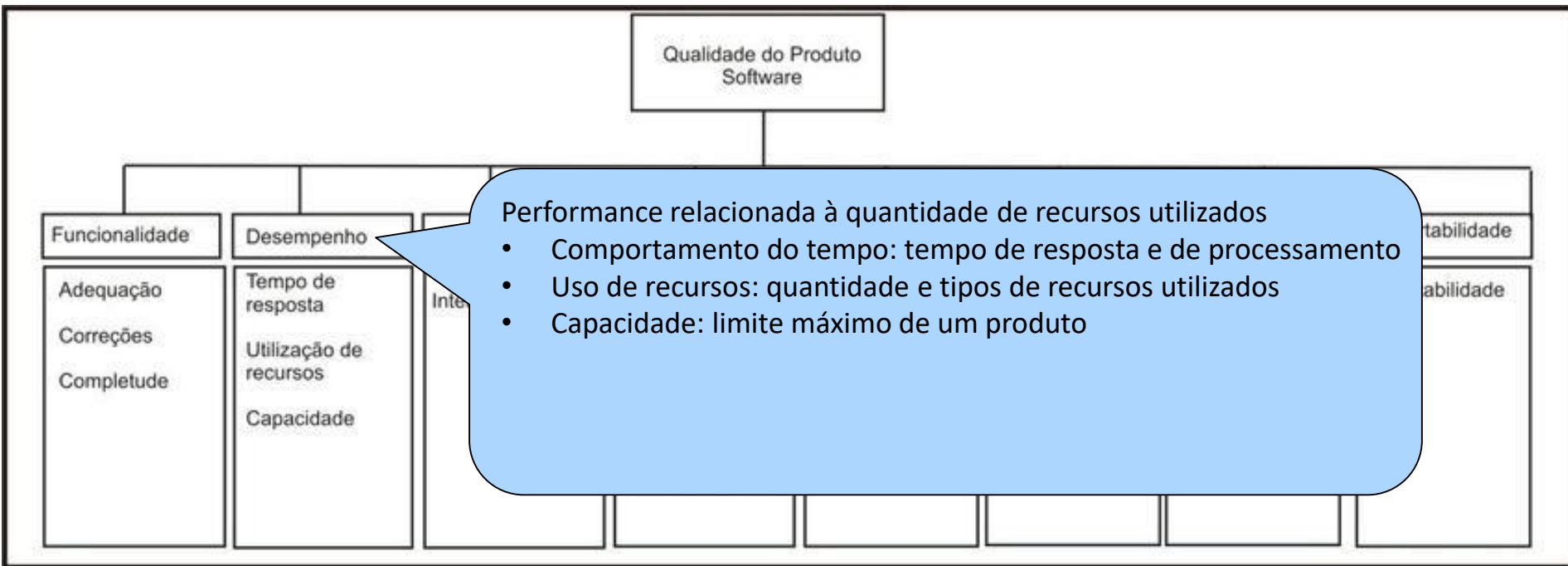
Norma ISO/IEC 25010



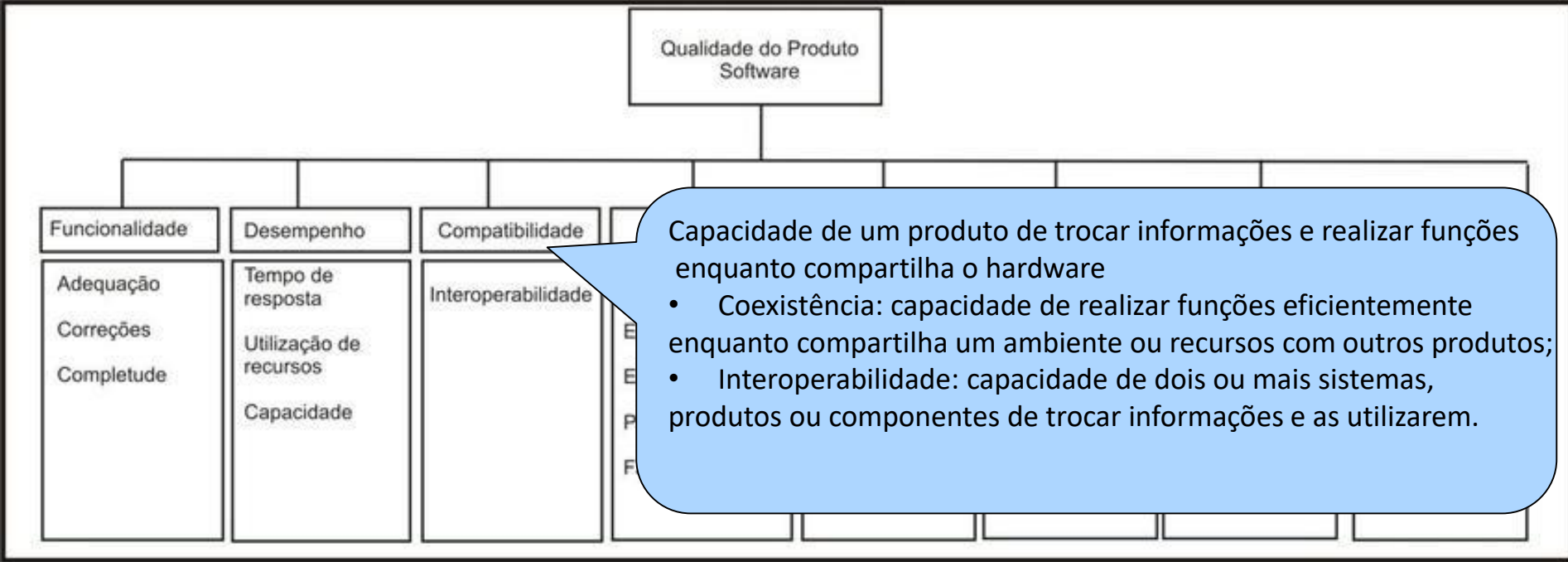
Norma ISO/IEC 25010



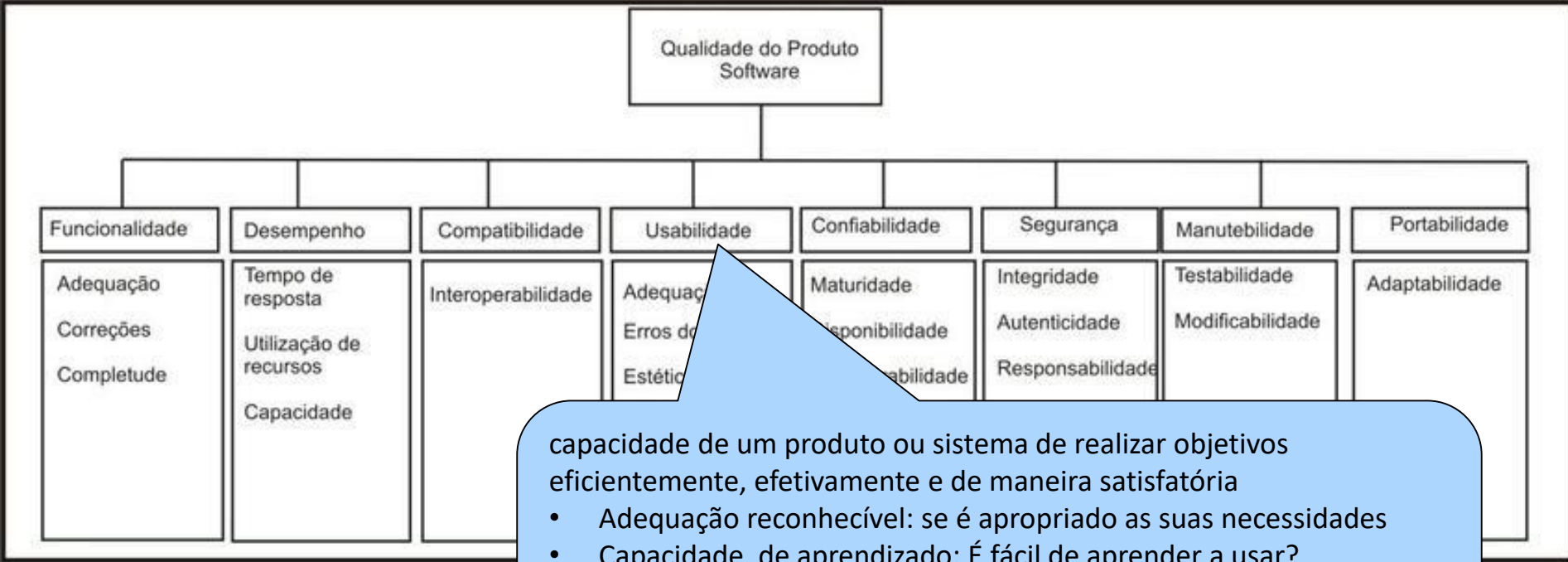
Norma ISO/IEC 25010



Norma ISO/IEC 25010



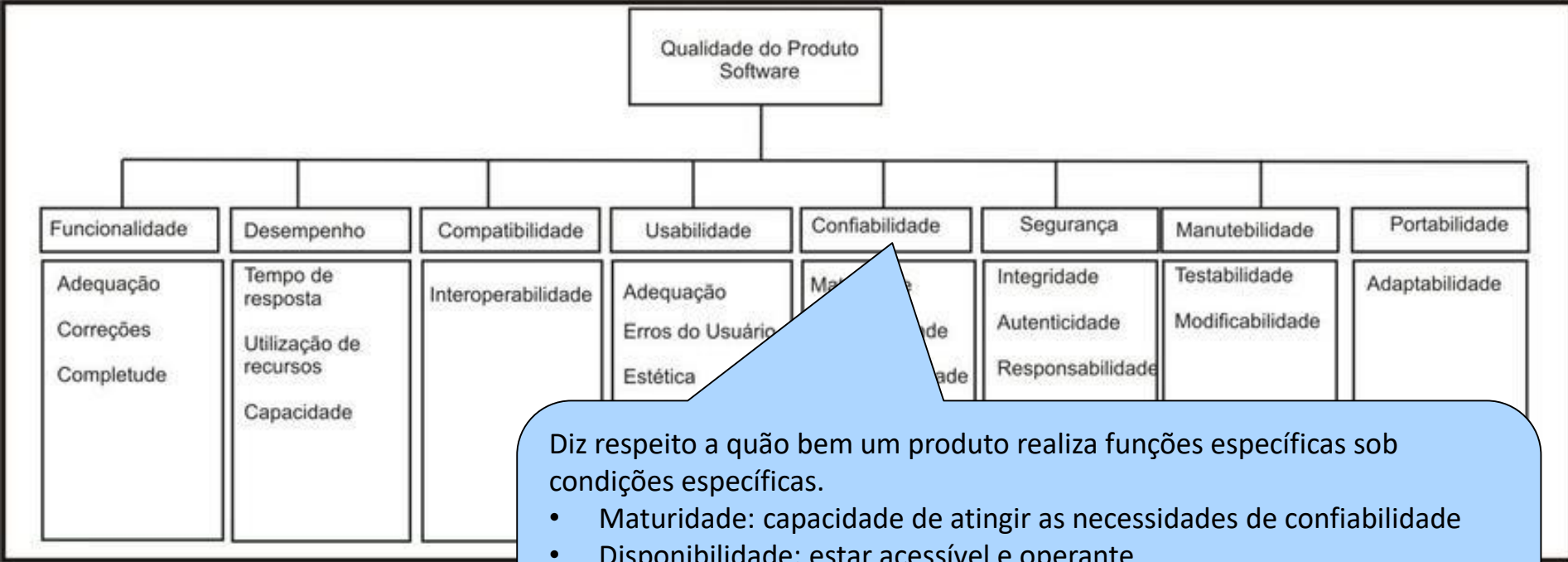
Norma ISO/IEC 25010



capacidade de um produto ou sistema de realizar objetivos eficientemente, efetivamente e de maneira satisfatória

- Adequação reconhecível: se é apropriado as suas necessidades
- Capacidade de aprendizado: É fácil de aprender a usar?
- Operabilidade: É fácil de usar ou controlar?
- Proteção a erros do usuário
- Estética: agradabilidade da interface
- Acessibilidade: ser utilizado por pessoas de diferentes capacidades

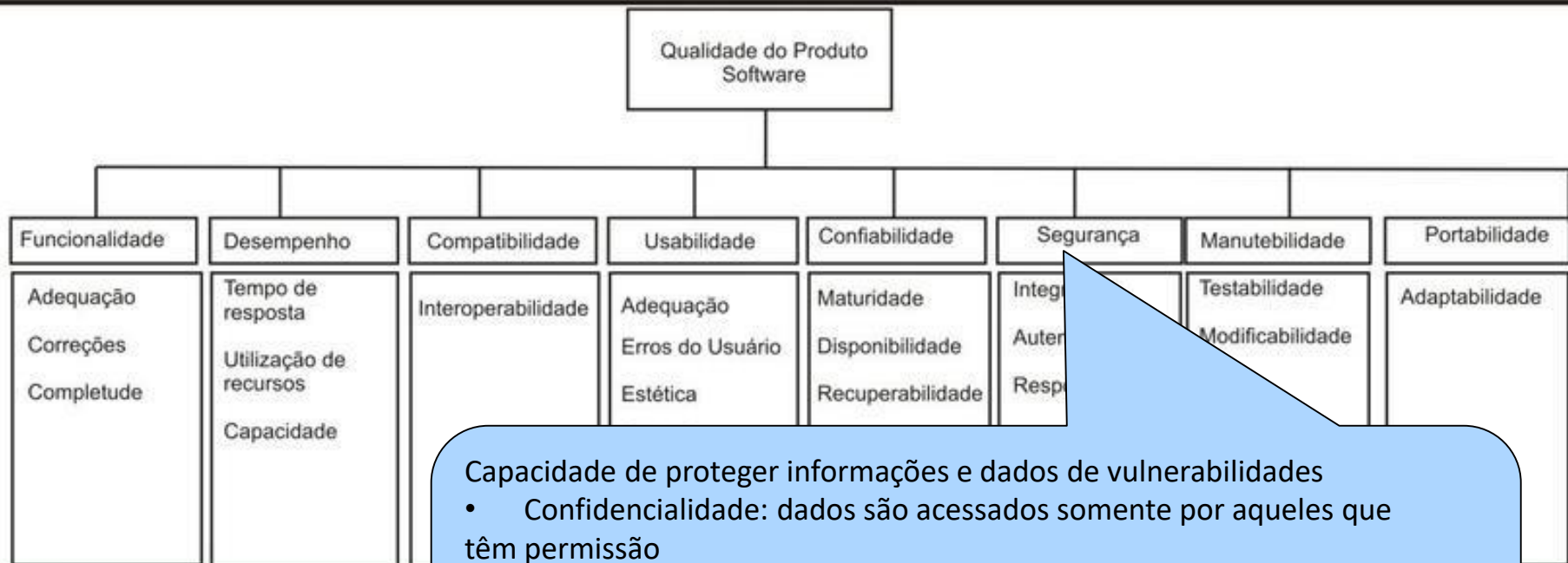
Norma ISO/IEC 25010



Diz respeito a quão bem um produto realiza funções específicas sob condições específicas.

- Maturidade: capacidade de atingir as necessidades de confiabilidade
- Disponibilidade: estar acessível e operante
- Tolerância a falhas: capacidade de continuar operando apesar de falhas de hardware ou software
- Recuperabilidade: capacidade de se recuperar em falhas e interrupções

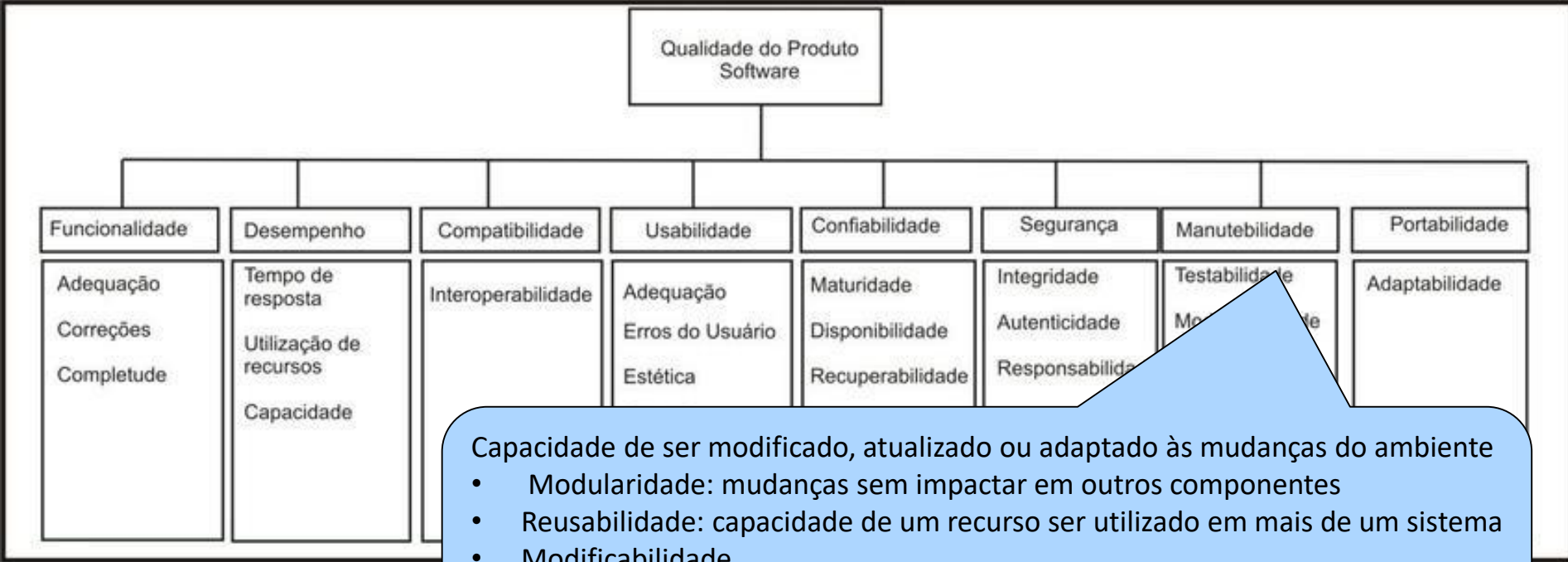
Norma ISO/IEC 25010



Capacidade de proteger informações e dados de vulnerabilidades

- Confidencialidade: dados são acessados somente por aqueles que têm permissão
- Integridade: previne acesso de não autorizados
- Ausência de repúdio: capacidade de provar que ações ou eventos aconteceram
- Rastreabilidade: ações de usuários não autorizados devem ser ligados a eles
- Autenticidade: identidade de um sujeito ou recurso pode ser provada

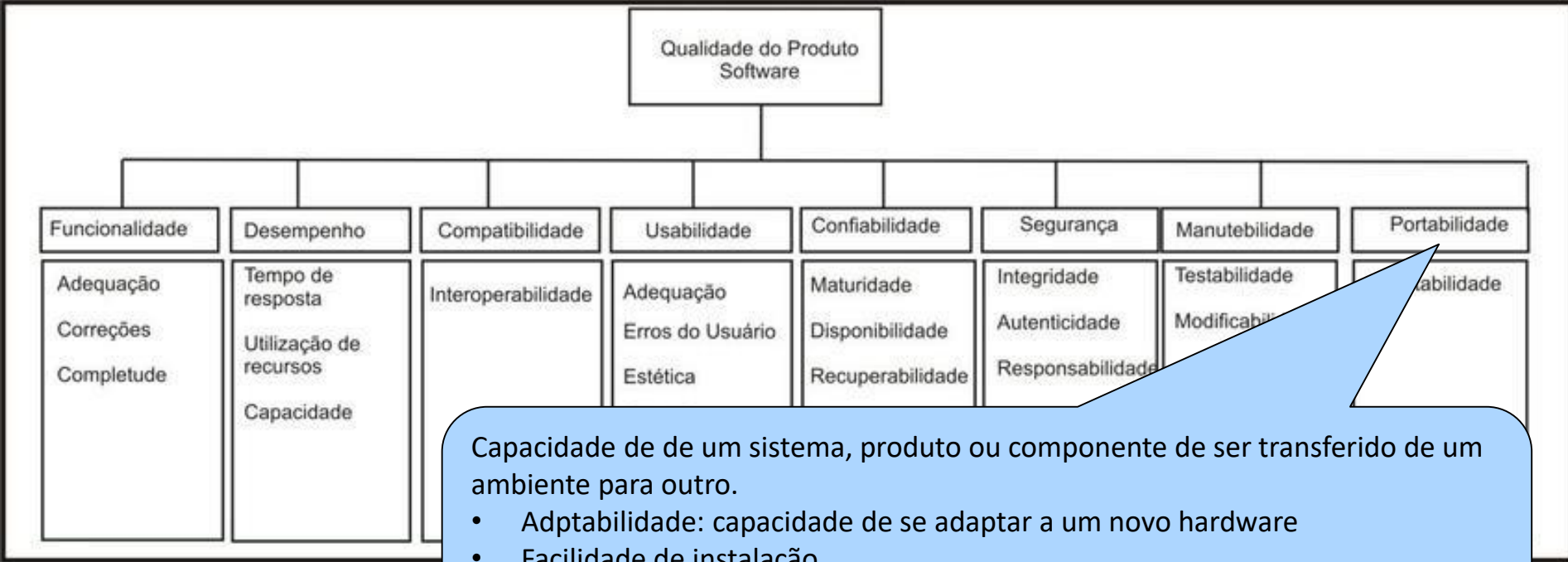
Norma ISO/IEC 25010



Capacidade de ser modificado, atualizado ou adaptado às mudanças do ambiente

- Modularidade: mudanças sem impactar em outros componentes
- Reusabilidade: capacidade de um recurso ser utilizado em mais de um sistema
- Modificabilidade
- Testabilidade

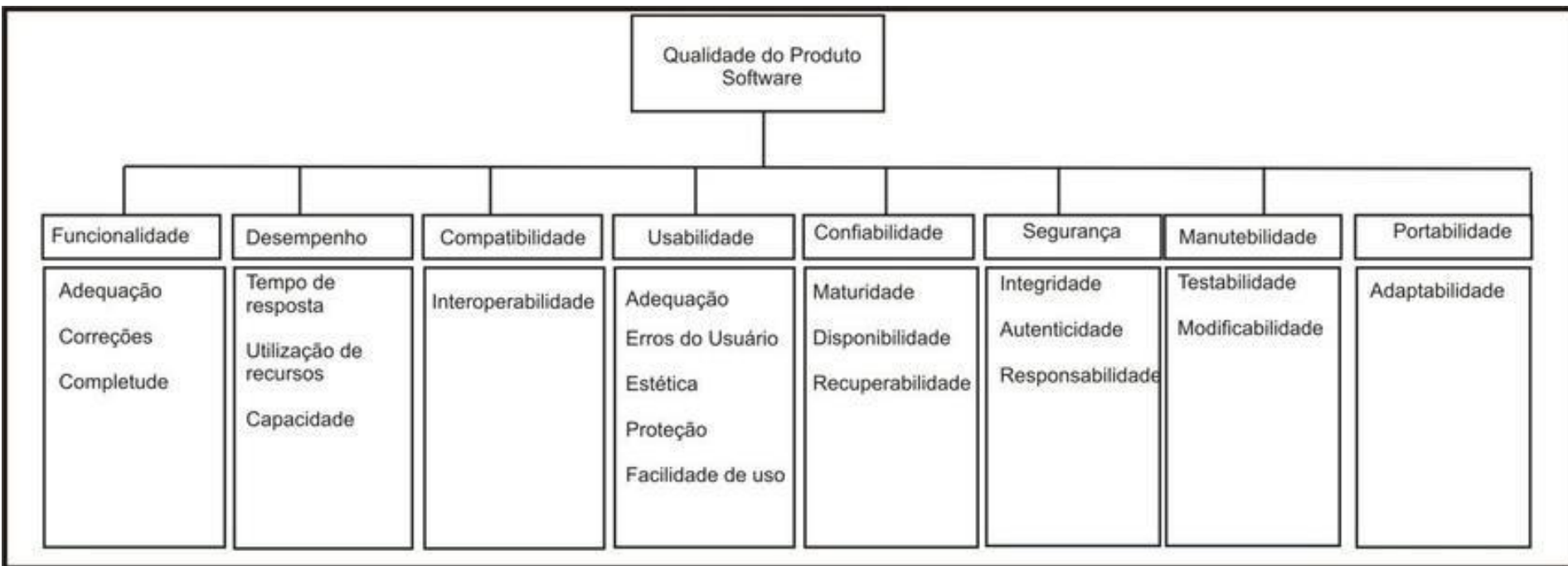
Norma ISO/IEC 25010



Capacidade de de um sistema, produto ou componente de ser transferido de um ambiente para outro.

- Adptabilidade: capacidade de se adaptar a um novo hardware
- Facilidade de instalação
- Capacidade de substituição: de ser trocado por outro produto similar

Norma ISO/IEC 25010



Métricas de Qualidade

- Como medir a qualidade de um software?
 - Funcionalidade?
 - Confiabilidade?
 - Usabilidade?
 - Eficiência?
 - Manutenibilidade?
 - Portabilidade?



Métricas de Qualidade

- Como medir a qualidade de um software?
 - Métricas dinâmicas
 - Durante teste ou uso do sistema
 - Durante depuração
 - Métricas estáticas
 - Informações coletadas do código



Métricas de Qualidade



■ Exemplos de métricas estáticas:

- Fan-in/fan-out
 - Fan-in: número de módulos que módulo x;
 - Fan-out: número de módulos chamados por módulo x;
- Tamanho do código (LoC)
- Quantidade de identificadores
- Complexidade ciclomática

OO possui métricas específicas!

Exemplo – Complexidade Ciclomática

```
void bolha(int a[], int size) {
/*1*/   int i, j, aux;
/*2*/   for (i = 0; i < size; i++)
/*3*/       for (j = size - 1; j > i; j--)
/*4*/           if (a[j - 1] > a[j])
/*5*/               {
/*6*/                   aux = a[j - 1];
/*7*/                   a[j - 1] = a[j];
/*8*/                   a[j] = aux;
/*9*/               }
/*10*/ }
```

A complexidade ciclomática define o número de **caminhos independentes** de um programa.

Exemplo – Complexidade Ciclomática

```
void bolha(int a[], int size) {
/*1*/   int i, j, aux;
/*2*/   for (i = 0; i < size; i++)
/*3*/       for (j = size - 1; j > i; j--)
/*4*/           if (a[j - 1] > a[j])
/*5*/               {
/*6*/                   aux = a[j - 1];
/*7*/                   a[j - 1] = a[j];
/*8*/                   a[j] = aux;
/*9*/               }
/*10*/ }
```

Complexidade ciclomática =

num de decisões binárias + 1

Exemplo – Complexidade Ciclomática

```
void bolha(int a[], int size) {
/*1*/   int i, j, aux;
/*2*/   for (i = 0; i < size; i++)
/*3*/       for (j = size - 1; j > i; j--)
/*4*/           if (a[j - 1] > a[j])
/*5*/               {
/*6*/                   aux = a[j - 1];
/*7*/                   a[j - 1] = a[j];
/*8*/                   a[j] = aux;
/*9*/               }
/*10*/ }
```

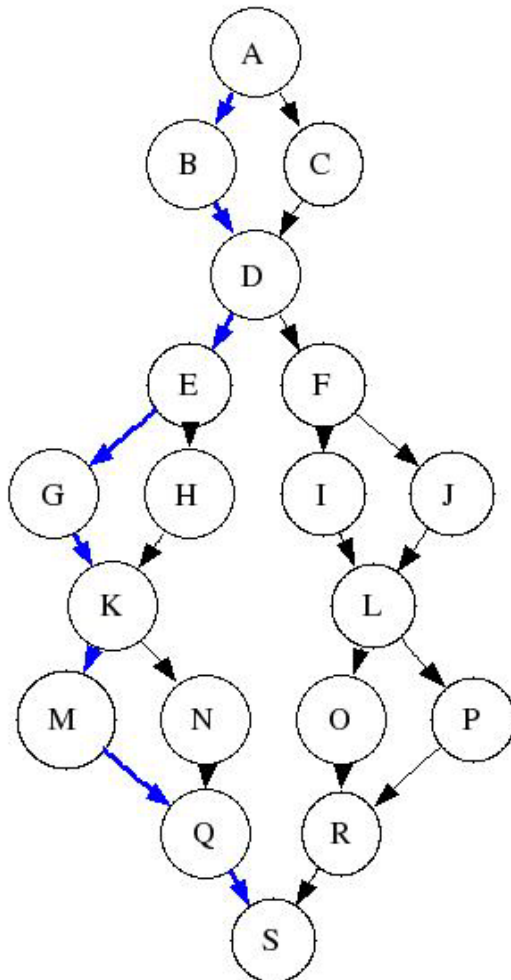
Complexidade ciclomática = 3 + 1

Métricas de Qualidade



- Complexidade ciclomática:
 - Auxílio no teste do software (confiabilidade e funcionalidade)
 - Auxílio na manutenibilidade:
 - De 0 a 5: Seu código está, provavelmente, ok;
 - Entre 6 e 10: Pense em maneiras de simplificar o código;
 - Maior que 10: Código precisa ser dividido

Métricas de Qualidade



Exemplo de grafo
de programa com
Complexidade
ciclomática = 7

Métricas de Qualidade



- A maioria das métricas são obtidas da validação/verificação do software
 - Teste de software
 - Revisão (Inspeção)
 - ...

Resumindo

- Qualidade é objetivo da engenharia de software
- Diferentes características de qualidade
- Importante medir a qualidade (métricas)
 - Ex: Em OO refatoração busca melhorar o código (analisando code smells)
 - Métricas podem ser derivadas de boas ferramentas de acompanhamento de projetos
 - SonarQube tool – code quality and security

Tarefa

1. Analise quais características de qualidade são importantes para o projeto da disciplina
2. Para cada uma, identifique como serão inseridas atividades que garantam que essas características sejam atendidas

Cada time faz para o seu projeto e inclui na documentação do projeto

Qualidade de Software

Simone Senger Souza
srocio@icmc.usp.br

ICMC/USP