

# Qualidade de Software: Visão Geral

Engenharia de Software 1

Aula 05

# Qualidade de Software

---

- ✓ Existem muitas **definições de qualidade** de software propostas na literatura, sob diferentes pontos de vista
- ✓ Qualidade é um termo que pode ter **diferentes interpretações**

# Qualidade de Software

---

- ✓ **Definição Peters (2002):** “Qualidade de software é avaliada em termos de atributos de alto nível chamados **fatores**, que são medidos em relação a atributos de baixo nível chamados de **critérios**” .
- ✓ **Definição Sanders (1994):** “Um produto de software apresenta qualidade dependendo do grau de satisfação das necessidades dos clientes sob todos os aspectos do produto” .

# Qualidade de Software

---

## *Definição Pressman:*

- ✓ *“Qualidade de software é a **conformidade** a requisitos funcionais e de desempenho que foram explicitamente declarados, a padrões de desenvolvimento claramente documentados, e a características implícitas que são esperadas de todo software desenvolvido por profissionais”.*

# Qualidade de Software

---

## *Definição ISO9126 (1994):*

- ✓ *“Qualidade é a **totalidade** de características e critérios de um produto ou serviço que exercem suas habilidades para satisfazer às necessidades declaradas ou envolvidas “.*

# Aspectos Importantes das Definições de Qualidade

---

As definições enfatizam **três** aspectos importantes:

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

A falta de conformidade com os requisitos significa falta de qualidade;

# Aspectos Importantes das Definições 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 quase que seguramente será a falta de qualidade.

# Aspectos Importantes das Definições de Qualidade

---

3- Existe um conjunto de *requisitos implícitos* que freqüentemente não são mencionados na especificação.

Por exemplo, o desejo de uma boa **Integridade** no acesso ao Sistema.

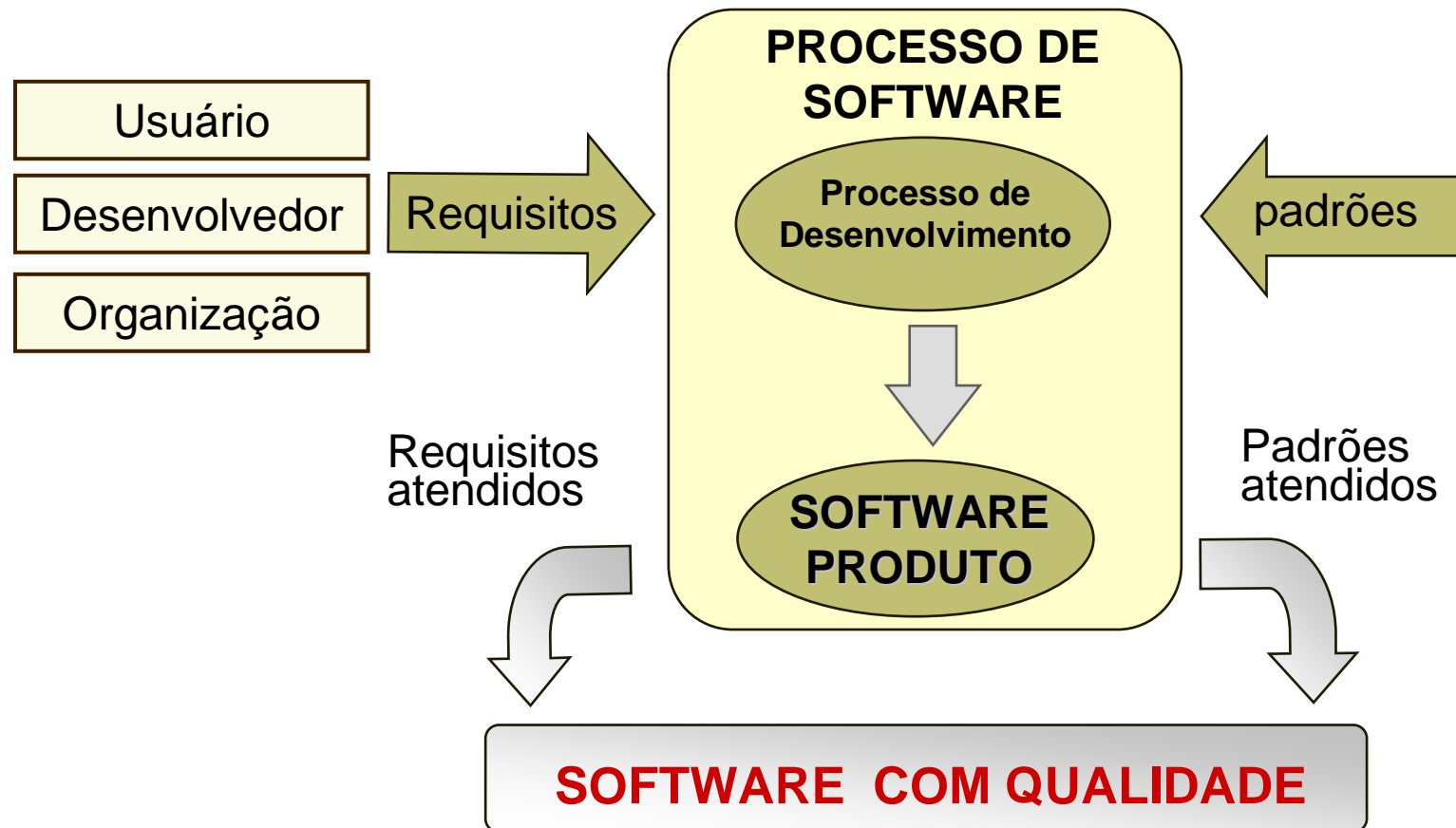


# Qualidade

---

- **Garantia de qualidade:** consiste nas funções gerenciais de auditar e relatar.
- **Controle de qualidade:** envolve a série de inspeções, revisões e testes usados ao longo do processo de software para garantir que cada produto de trabalho satisfaça os requisitos estabelecidos para ele.

# Requisitos de Software - Base da Qualidade



# Incorporação da Qualidade

---

- Raramente qualidade pode ser incorporada ao **produto final**, após o processo de desenvolvimento ter terminado.
- Dos requisitos do usuário à entrega do produto final, existe um **processo de desenvolvimento** que é complexo e frequentemente envolve uma série de **etapas** que podem comprometer a **qualidade do produto final**. Dependem dos produtos intermediários na construção do Software.

# Incorporação da Qualidade: Qualidade do Produto Final

---

- Cada produto **intermediário** tem certos atributos de qualidade que afetam a qualidade do **produto intermediário** da **próxima etapa** e assim, afetam a **qualidade do produto final**.

# Qualidade: Objetivo do Processo de Desenvolvimento

---

- A qualidade do produto de software é um objetivo do **processo de desenvolvimento**.
- Assim, ao desenvolver-se um produto, deve-se ter previamente estabelecidas, como perspectiva, as **características de qualidade** que se deseja alcançar.

# Atividades de Padronização

---

- Qualidade do Produto
  - ❖ ISO/IEC 9126 (AVALIAÇÃO)
- Qualidade do Processo
  - ❖ CMM/CMMI
  - ❖ ISO/IEC 12207
  - ❖ ISO/IEC 15504 SPICE

# ISO/IEC 9126

---

- Representa a atual padronização mundial para a qualidade de software e do produto.
- **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.

# ISO/IEC 9126

---

➤ As características são:

❖ **Funcionalidade** - Satisfaz as necessidades?

❖ **Confiabilidade** - É imune a falhas?

❖ **Usabilidade** - É fácil de usar?

❖ **Eficiência** - É rápido e “enxuto” ?

❖ **Manutenibilidade** - É fácil de modificar?

❖ **Portabilidade** - É fácil de usar em outro ambiente?



# ISO/IEC 9126

---

***Funcionalidade***: “Conjunto de atributos que evidenciam a existência de um conjunto de funções e suas propriedades especificadas. As funções são as que satisfazem as necessidades explícitas e implícitas”.

***Usabilidade***: “Conjunto de atributos que evidenciam o esforço necessário para utilizar o software, bem como o julgamento individual desse uso, por um conjunto explícito ou implícito de usuários”.

# ISO/IEC 9126

---

***Confiabilidade***: “Conjunto de atributos que evidenciam a capacidade do software de manter seu nível de desempenho sob condições estabelecidas durante um período de tempo estabelecido”.

***Eficiência***: “Conjunto de atributos que evidenciam o relacionamento entre o nível de desempenho do software e a quantidade de recursos usados, sob condições estabelecidas”.

# ISO/IEC 9126

---

***Manutenibilidade***: “Conjunto de atributos que evidenciam o esforço necessário para fazer modificações especificadas no software”.

***Portabilidade***: “Conjunto de atributos que evidenciam a capacidade do software de ser transferido de um ambiente para outro” .

merecedor  
de crédito

## Outros fatores de Qualidade

---

- ***Fidedignidade***: Até que ponto o software satisfaz seus requisitos – critérios de avaliação: rastreabilidade, completude, consistência.
- ***Testabilidade***: Esforço necessário para garantir que o software desempenhe as funções a que se destina – critério: Simplicidade, modularidade, instrumentação, capacidade de auto-descrição.
- ***Integridade***: Extensão do controle de modificações ou acessos acidentais – controle de acesso e auditoria de acesso.

# ISO/IEC 12207

---

- A **ISO/IEC 12207** é a norma **ISO/IEC** que define **processo de desenvolvimento de software**. A norma internacional *ISO/IEC 12207* tem como objetivo principal estabelecer uma estrutura comum para os processos de ciclo de vida de **software** visando ajudar as **organizações** a compreenderem todos os **componentes** presentes na aquisição e fornecimento de software e, assim, conseguirem firmar **contratos** e executarem projetos de forma **mais eficaz**.
- Um **processo** é uma seqüência de passos realizados para um determinado propósito [*IEEE, 1990*]; o **processo de software** envolve *métodos, técnicas, ferramentas e pessoas*. Um processo pode ser descrito de duas formas: por **propósito ou resultado** e **por atividade**.

# Descrição por Propósito ou Resultado

---

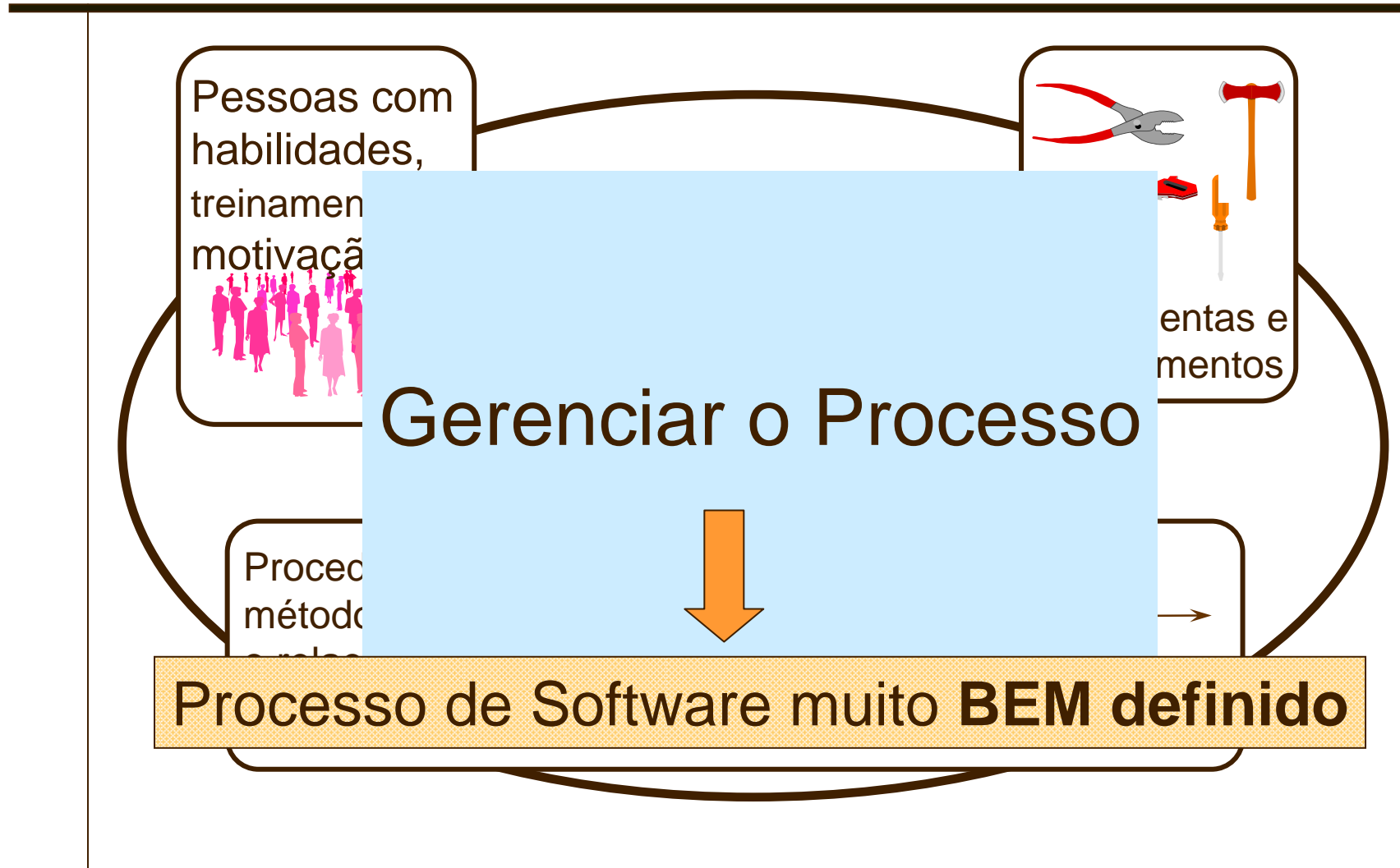
- É utilizada quando não há necessidade de detalhar o processo, apenas indicar o objetivo e o resultado. Essa abordagem poderá ser utilizada na avaliação do processo em relação aos modelos de maturidade de software como, por exemplo, o modelo **CMMI** e o modelo da **ISO/IEC 15504 (SPICE)**.

# Descrição por Atividade

---

- É a abordagem mais **conhecida** e **intuitiva**. Nela são descritas as atividades com as **inter-relações** e o **algoritmo de execução** de cada atividade. As atividades devem atingir o propósito do processo.

# Processo de Software





# ISO/IEC 12207

---

- Para isso deve adotar as **premissas**:
  - ❖ Que **procedimentos e métodos** serão usados para a execução das atividades;
  - ❖ Que **ferramentas e equipamentos** suportarão a realização das atividades, de forma a simplificar e *automatizar o trabalho*;
  - ❖ Qual o **perfil adequado** de quem irá executar as atividades e qual o **treinamento requerido** nos *procedimentos, métodos, ferramentas* para que se possam realizar as atividades de forma adequada;
  - ❖ Quais as **métricas de processo** que poderão ser empregadas para que a execução do processo possa ter a *qualidade avaliada*.

# Processo de Software

---

## Definição

- **processo** - uma seqüência de passos realizados para um determinado propósito (IEEE).
- **processo de software** - conjunto de atividades, métodos, práticas e tecnologias que as pessoas utilizam para desenvolver e manter software e produtos relacionados (CMM).

# Processo de Software

---

## Definição

- **processo de software:** Consiste em uma série de atividades, práticas, eventos, ferramentas e métodos que garantem, técnica e administrativamente que o software pode ser desenvolvido com qualidade e de maneira organizada, disciplinada e previsível.

# Modelos de Processo de Software

---

- Uma das **maiores dificuldades** encontradas pelas empresas de software é o **gerenciamento** de seus **processos de software**.



**Modelos de Processo de Software**

# Modelos de Processo de Software

---

- Um **modelo de processo** procura descrever formalmente e de maneira organizada todas as atividades que devem ser seguidas para a obtenção segura de um produto de software.
- É importante escolher um **modelo apropriado** às metas da organização e saber em que **grau** esse modelo será implementado.
- Modelos *CMM*, *SPICE* e *Bootstrap* são **padrões** relacionados a processo de software.

# Um Processo de Software Imaturo

---

- **Ad hoc**; processo improvisado por profissionais e gerências.
- **Não é** rigorosamente **seguido** e o cumprimento não é **controlado**.
- Altamente **dependente dos profissionais** atuais.
- **Baixa visão** do progresso e da qualidade.

# Um Processo de Software Imaturo

---

- A **qualidade** do produto pode ficar **comprometida** para que prazos sejam cumpridos.
- **Arriscado** do ponto de vista do uso de **nova tecnologia** e novos padrões.
- Custos de **manutenção** excessivos.
- Qualidade **difícil de prever**.

# Um Processo de Software Maduro

---

- **Coerente** com as linhas de ação, o trabalho é efetivamente concluído.
- Definido, documentado e melhorando constantemente:
  - compreendido
  - utilizado
  - vivo e ativo



# Um Processo de Software Maduro

---

- Com o **apoio** visível da alta administração e outras gerências.
- **Bem controlado** - fidelidade ao processo é objeto de auditoria e de controle.
- São utilizadas medições do produto e do processo.
- Uso disciplinado da tecnologia.

# Maturidade de Processo de Software

---

- A **qualidade do processo de software** pode ser analisada por intermédio do nível de maturidade do processo .
- A **maturidade dos processos de software** de uma organização influencia a sua capacidade de atingir metas de custo, qualidade e cronograma.

# Valor da Melhoria de Processo

---

- A **melhoria do processo** de software oferece um retorno no investimento que pode ser medido - **quando é medido**.
- **Benefícios adicionais** são intangíveis e não podem ser quantificados facilmente.
- O **CMM** é uma ferramenta útil para orientação no processo de melhoria. Porém muito complexa para ser implementada.

# Capability Maturity Model (CMM)

---

- Modelo de Maturidade de Processo de Software que auxilia as empresas a melhorar seus **processos** de software.
- 1986 - Software Engineering Institute (SEI) por solicitação do *DoD* que necessitava de um método de avaliar a capacidade de seus fornecedores de software.
- 1991 - 1ª versão do CMM (Versão 1.0)
- 1993 - Versão 1.1 do CMM (atualmente usada)

# Capability Maturity Model (CMM)

---

- O primeiro ponto importante a se observar no modelo CMM:
  - ❖ É um modelo que se refere ao processo pelo qual o produto é construído.

# CMMI (Capability Maturity Model Integration)

---

- O **CMMI** é um modelo de referência que contém práticas (*Genéricas ou Específicas*) necessárias à maturidade em disciplinas específicas (*Engenharia de Sistemas (SE), Engenharia de Software (SE), Desenvolvimento de Processo e Produto Integrado (IPPD), Supplier Sourcing (SS)*). Desenvolvido pelo SEI (*Instituto de Engenharia de Software*).
- O **CMMI** é uma evolução do **CMM** e procura estabelecer um modelo único para o processo de melhoria corporativo, integrando diferentes modelos e disciplinas.

# CMMI

- *Até o ano de 2002, os EUA tinham realizado 1,5 mil avaliações de CMMI, a Índia feito 153, o Reino Unido 103 e o Brasil apenas 15. Em 2004 a TATA Consultancy Services alcançou o nível 5 em todas as unidades da empresa, tendo sido avaliada inclusive a unidade brasileira (a primeira empresa presente no Brasil a receber o nível máximo na avaliação). Outras importantes empresas presentes no Brasil, alcançaram boas qualificações na avaliação, dentre elas destacam-se:*
  - *IBM Brazil - nível 5 2005*
  - *EDS Brasil - nível 4 2005*
  - *Unysis Brazil - nível 3 - 2005*
  - *Politec - nível 5 - 2006*

- 
- *Entre abril de 2002 e junho de 2006 foram conduzidas 1581 avaliações em 1377 organizações. Segue abaixo o resultado obtido pelas empresas na avaliação (resultados encaminhados para o SEI até 30/06/2006):*
    - *18,2%: nível 5 (Optmizing);*
    - *4,4%: nível 4 (Quantitatively Managed);*
    - *33,8%: nível 3 (Defined);*
    - *33,3%: nível 2 (Managed);*
    - *1,9%: nível 1 (Initial);*
    - *8,4%: sem qualificação (Not Given).*



# CMMI

---

- Supõe-se que uma organização que possui um processo maduro tem **maiores probabilidades** de produzir consistentemente bons produtos do que uma outra cujo processo é imaturo e caótico.

# CMMI

---

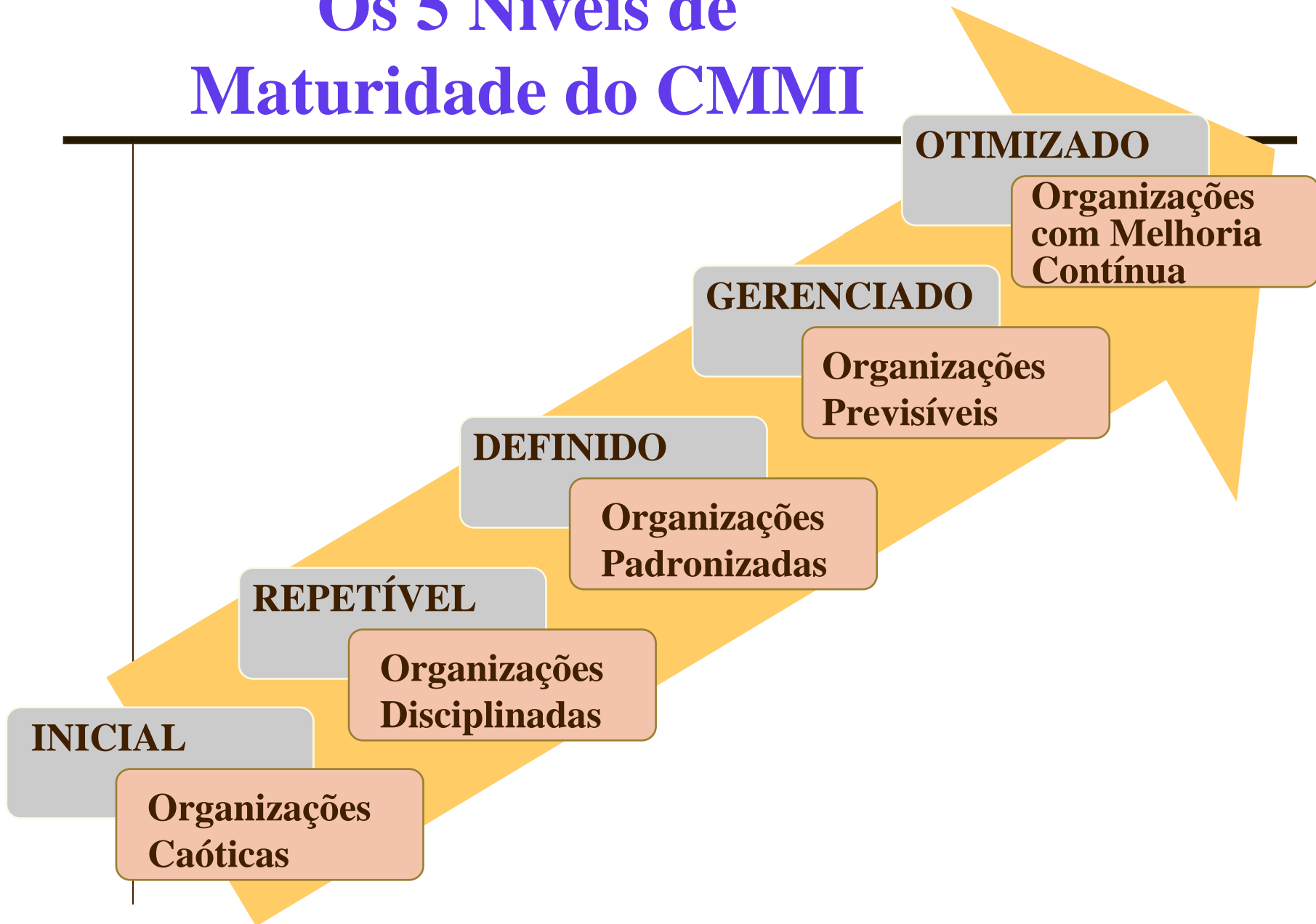
- O CMMI é um modelo de **referência** para a qualidade do processo de produção de software.
- Por meio de um **processo formal de avaliação**, uma organização é classificada em um “**nível de maturidade**” que varia de um a cinco.

# CMMI

---

- O nível de maturidade indica em que medida os processos daquela organização são maduros.
- Quanto maior o nível de maturidade, melhores e mais maduros são os processos.

# Os 5 Níveis de Maturidade do CMMI



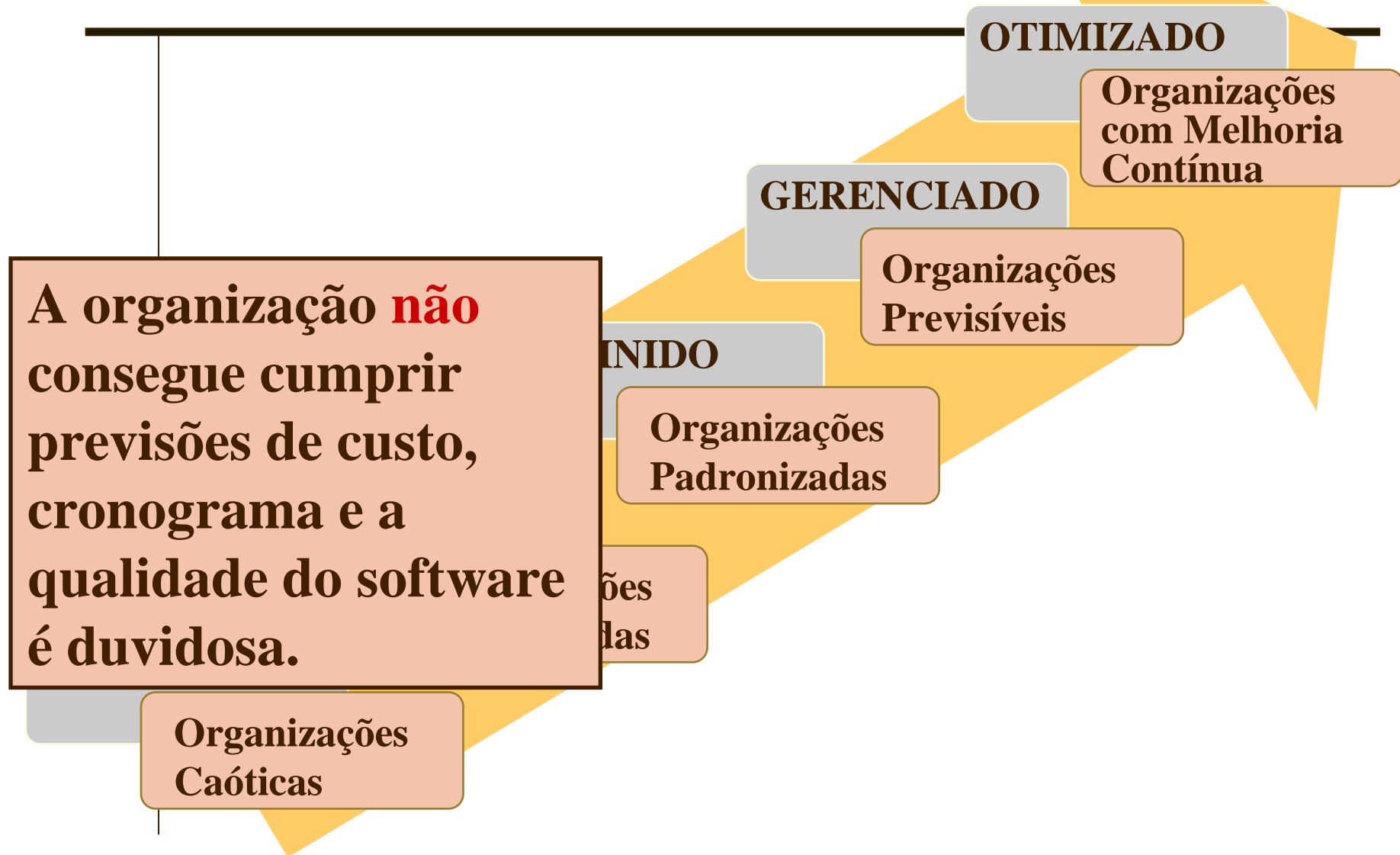
# Os 5 Níveis de Maturidade do CMMI

- Os 5 Níveis do CMMI são organizados de modo a estabelecer as prioridades na condução de programas de melhoria do processo de software.
- Cada Nível é considerado como pré-requisito para o nível seguinte.
- Não é possível o “salto” de níveis.
- Cada nível se apóia nas competências que a organização desenvolveu no nível imediatamente inferior.

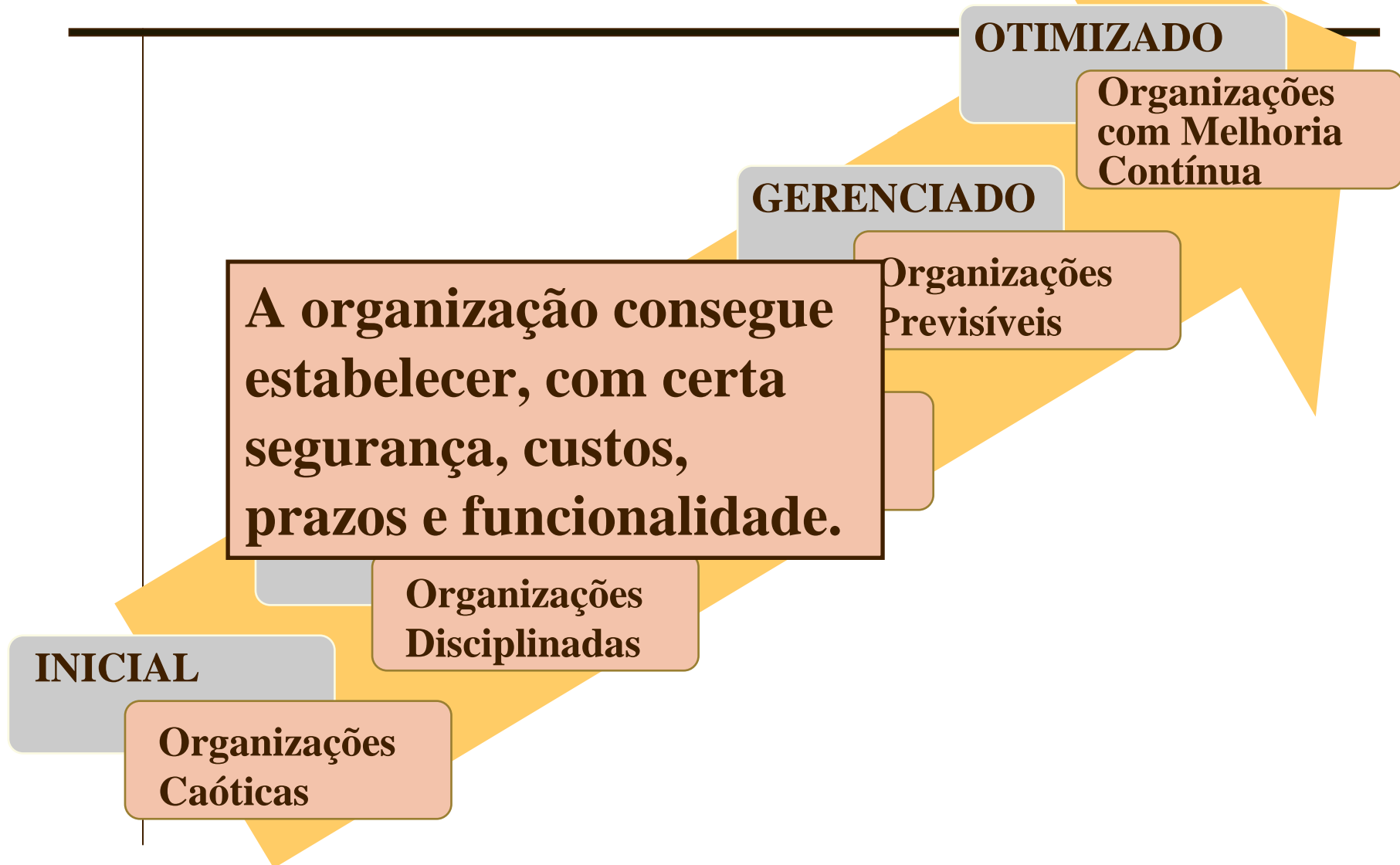
INICIAL

Organizações  
Caóticas

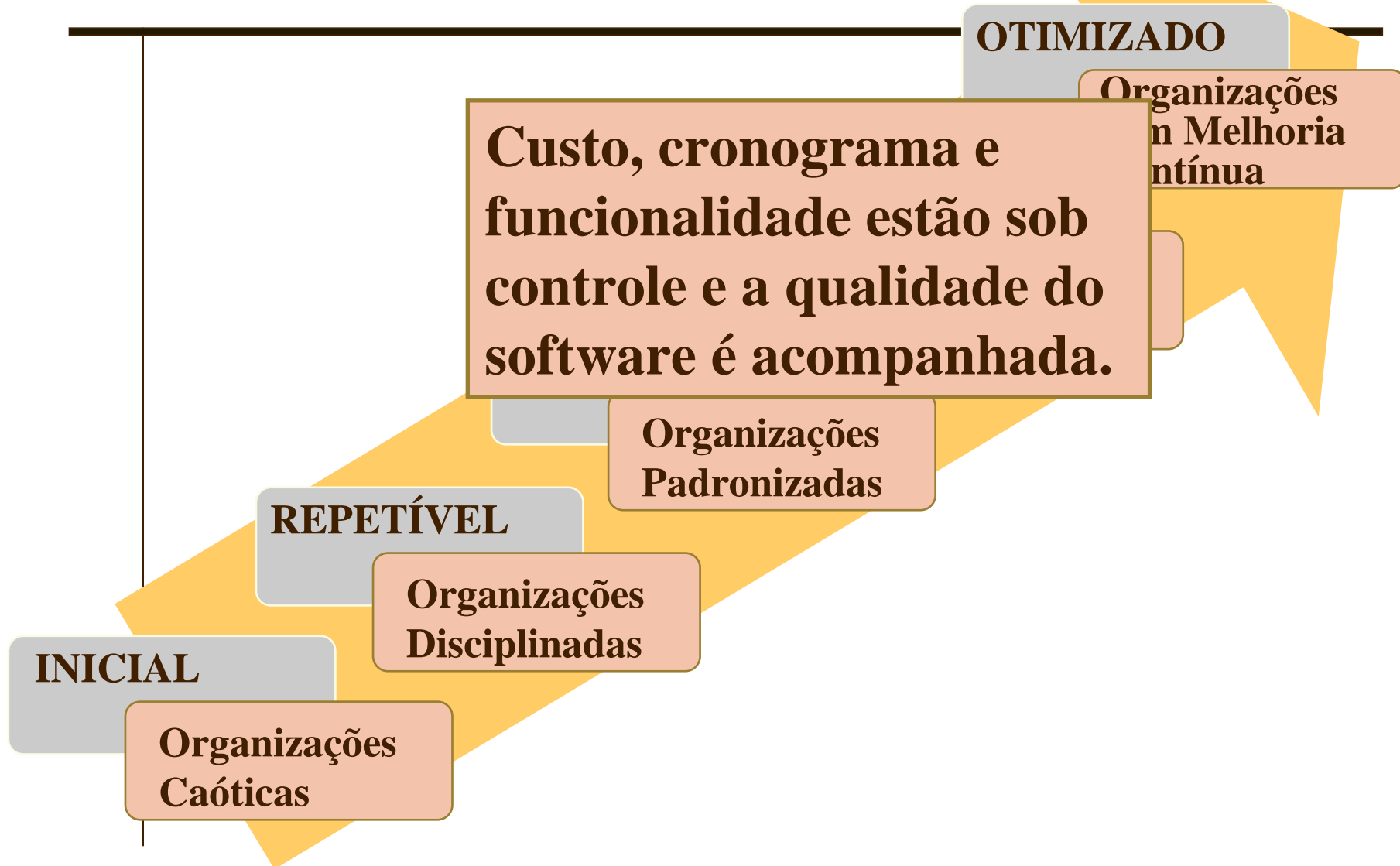
# Nível 1 de Maturidade do CMMI



# Nível 2 de Maturidade do CMMI

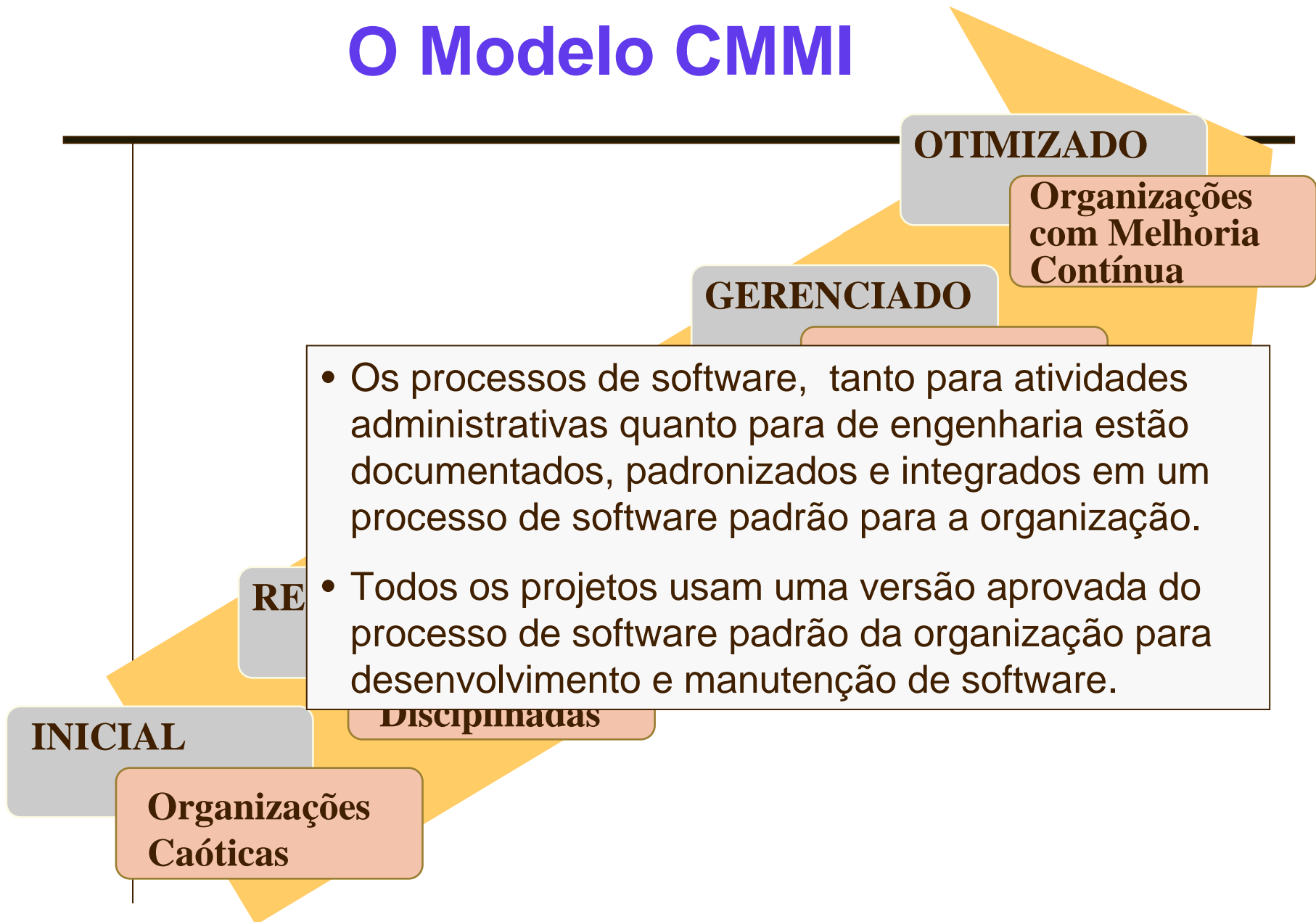


# Nível 3 de Maturidade do CMMI

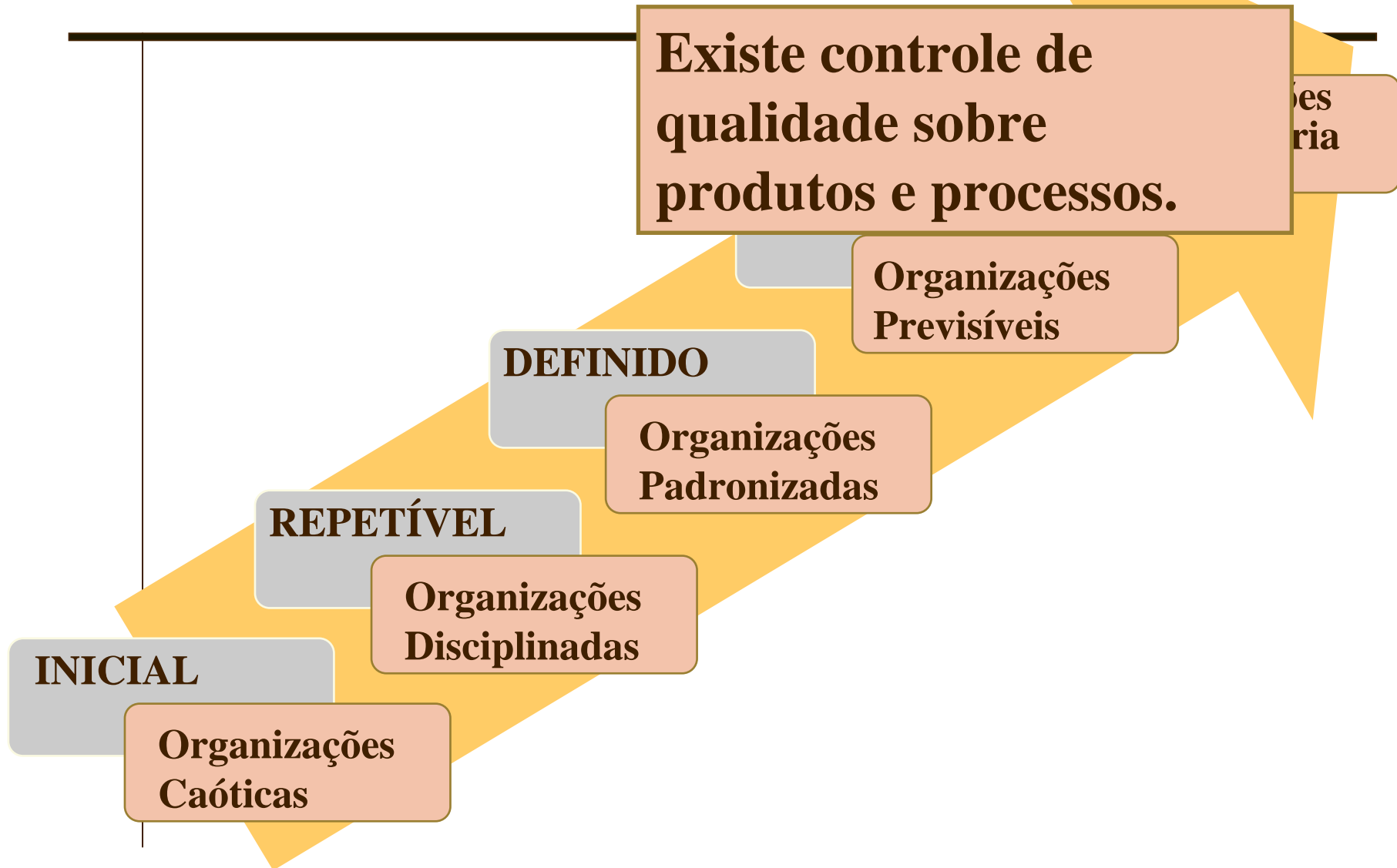




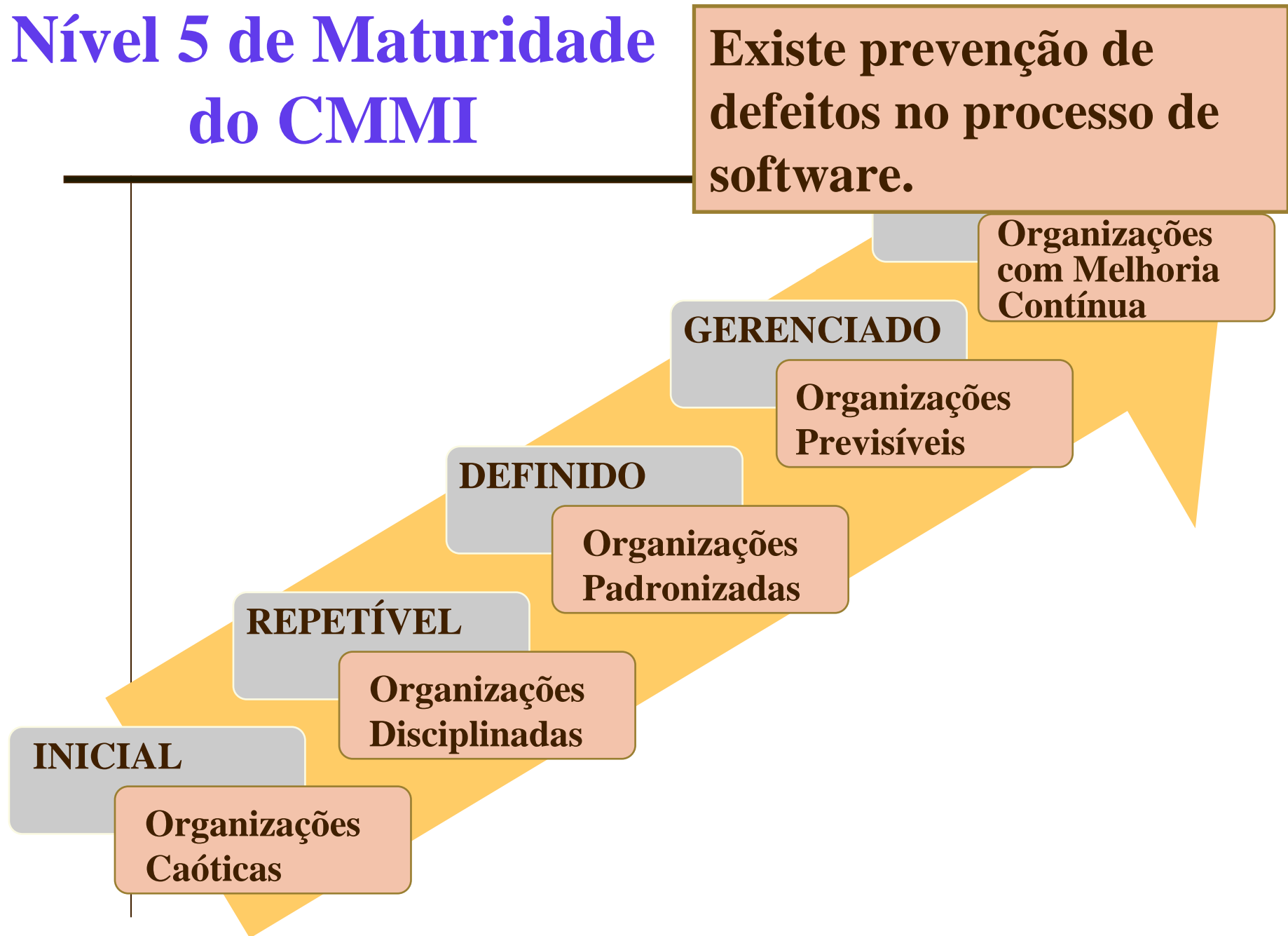
# O Modelo CMMI



# Nível 4 de Maturidade do CMMI



# Nível 5 de Maturidade do CMMI



# Melhoria de Processo de Software - MPS

---

- **Melhoria do Processo de Software Brasileiro**, é simultaneamente um movimento para a melhoria e um modelo de qualidade de processo voltada para a realidade do mercado de pequenas e médias empresas de desenvolvimento de software no Brasil.
- Ele é baseado no **CMMI**, nas normas **ISO/IEC 12207** e **ISO/IEC 15504** e na realidade do mercado brasileiro.
- No Brasil, uma das principais vantagens do modelo é seu custo reduzido de certificação em relação as normas estrangeiras, sendo ideal para micro, pequenas e médias empresas.
- Um dos objetivos do projeto é replicar o modelo na América Latina, incluindo o Chile, Argentina, Costa Rica, Peru e Uruguai.

<http://pt.wikipedia.org/wiki/MPS.BR>

# ISO/IEC 15504

---

- A **ISO/IEC 15504**, também conhecida como **SPICE**, é a norma **ISO/IEC** que define **processo de desenvolvimento de software**. Ela é uma evolução da **ISO/IEC 12207** mas possui níveis de capacidade para cada processo assim como o **CMMI**.
- 1993: SPICE (*Software Process Improvement and Capability dEtermination*)
- **Objetivo**: gerar **normas** para **avaliação** de processos de software, visando a **melhoria** contínua do processo e a determinação da sua capacidade.

# Benefícios do SPICE

---

➤ Para **avaliadores**:

- ❖ uma **estrutura** que define todos os aspectos para **conduzir** avaliações.

# Benefícios do SPICE

---

## ➤ Para **fornecedores**:

- ❖ determinar a **capacidade atual** e potencial de seus processos de software.
- ❖ efetuar a **melhoria** ou uma estrutura que define um roteiro para a melhoria do processo de software.

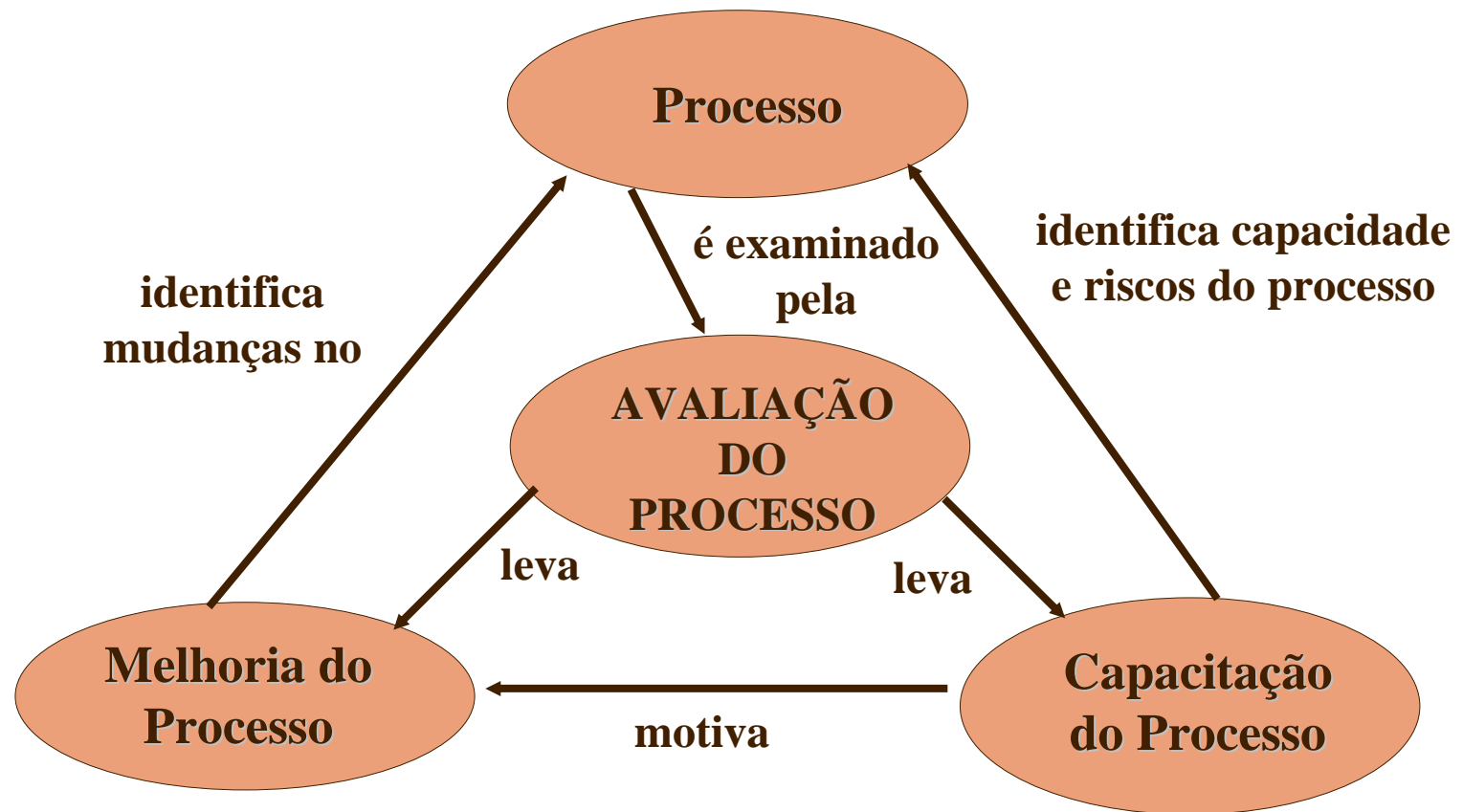
# Benefícios do SPICE

---

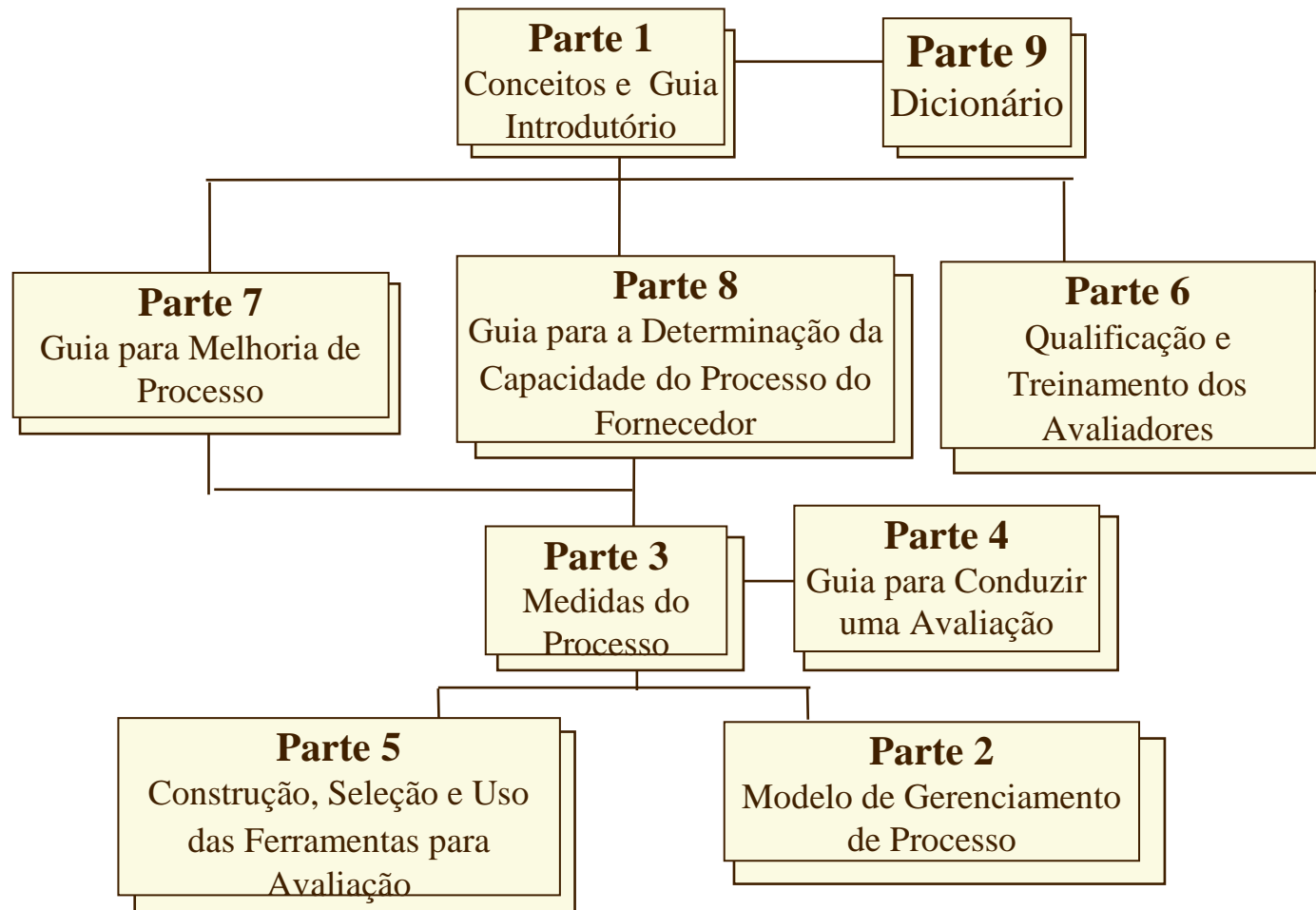
- Para os **compradores**:
  - ❖ determinar a **capacidade atual** e potencial de um processo de software do fornecedor.



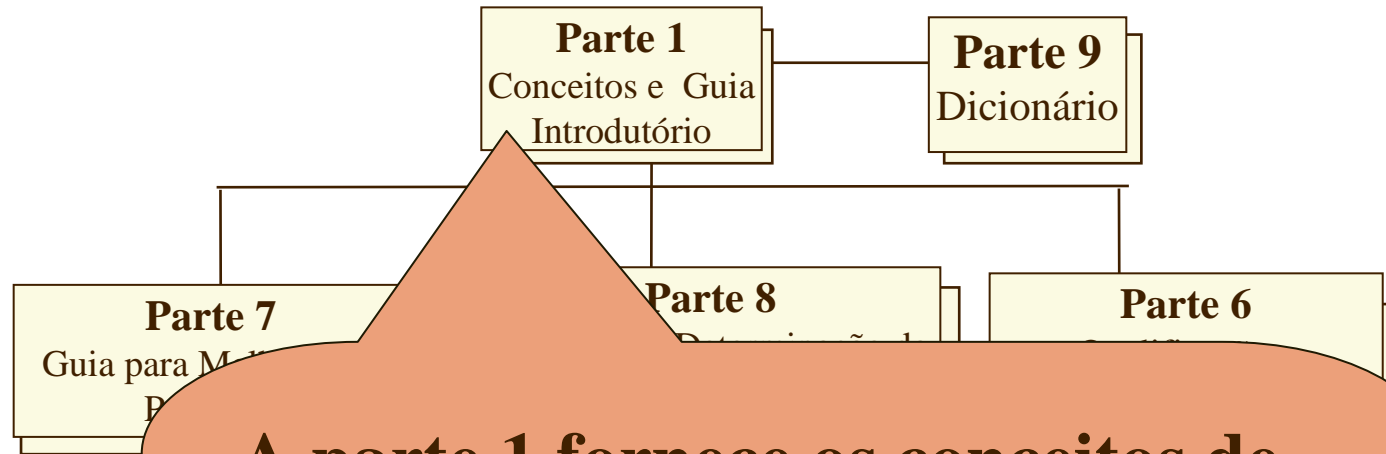
# Processo de Avaliação



# Componentes do Modelo SPICE



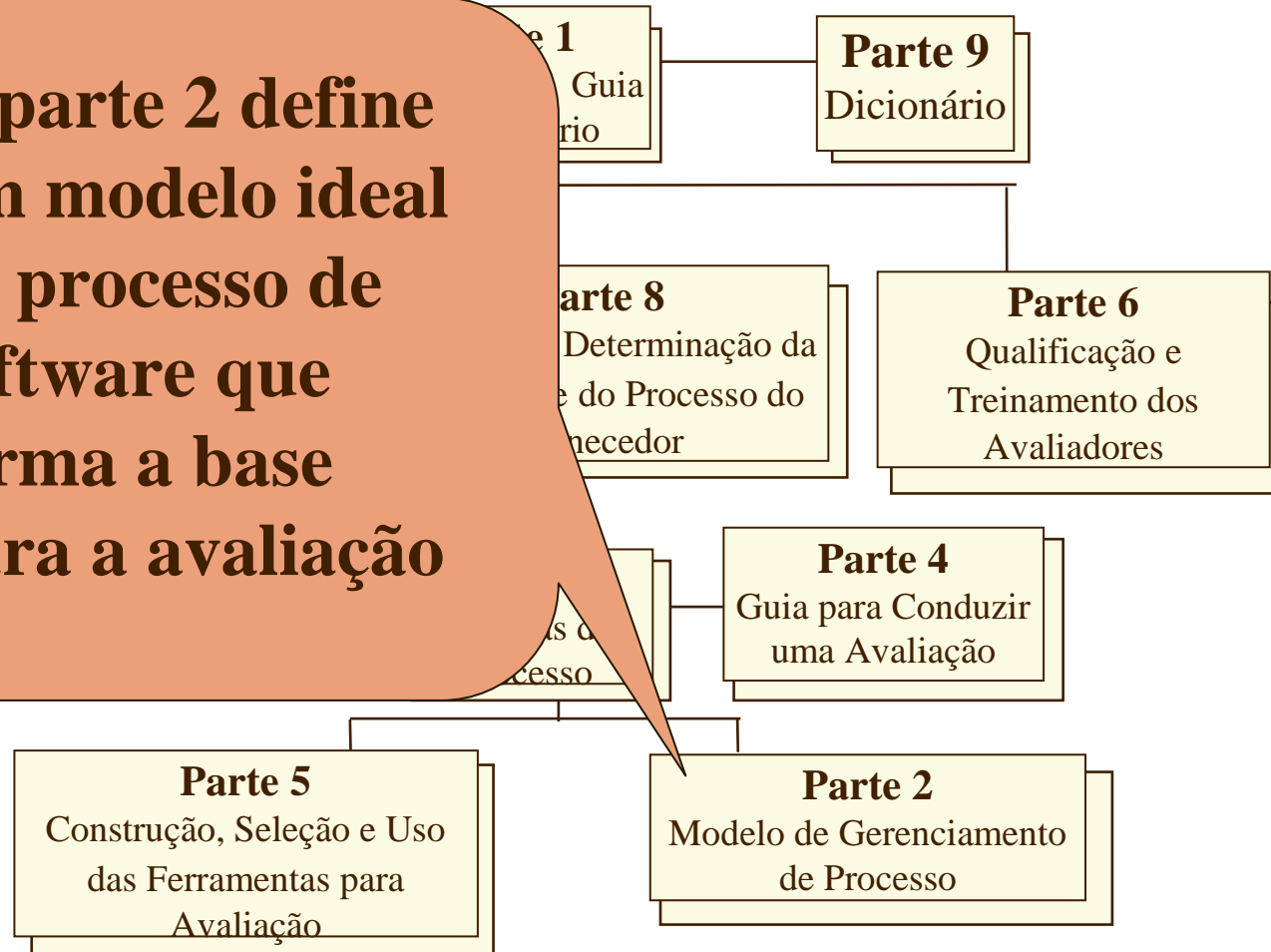
# Componentes do Modelo SPICE



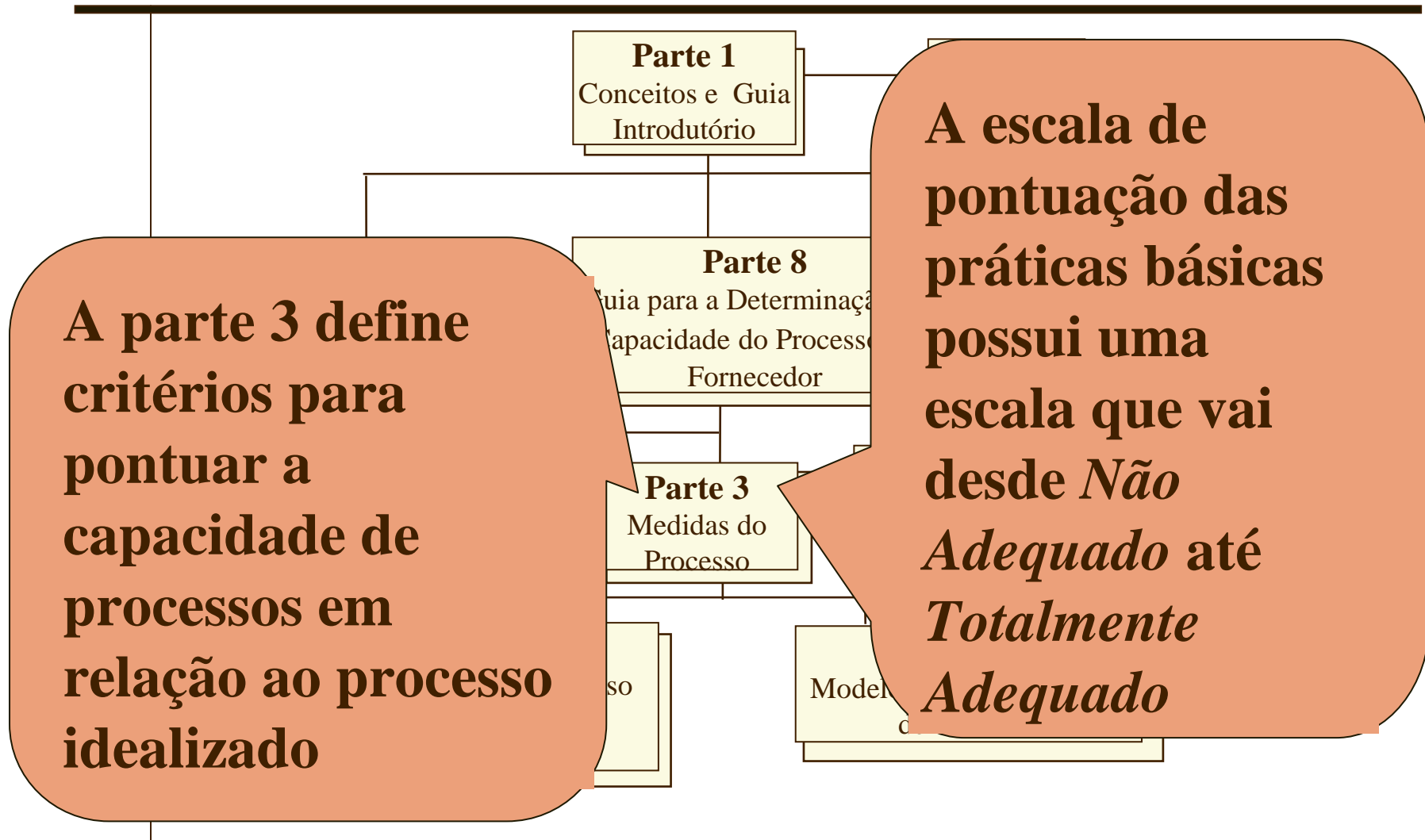
**A parte 1 fornece os conceitos de avaliação de processo de software e o uso desses conceitos no contexto de melhoria de processo e determinação da capacidade do processo.**

# Componentes do Modelo SPICE

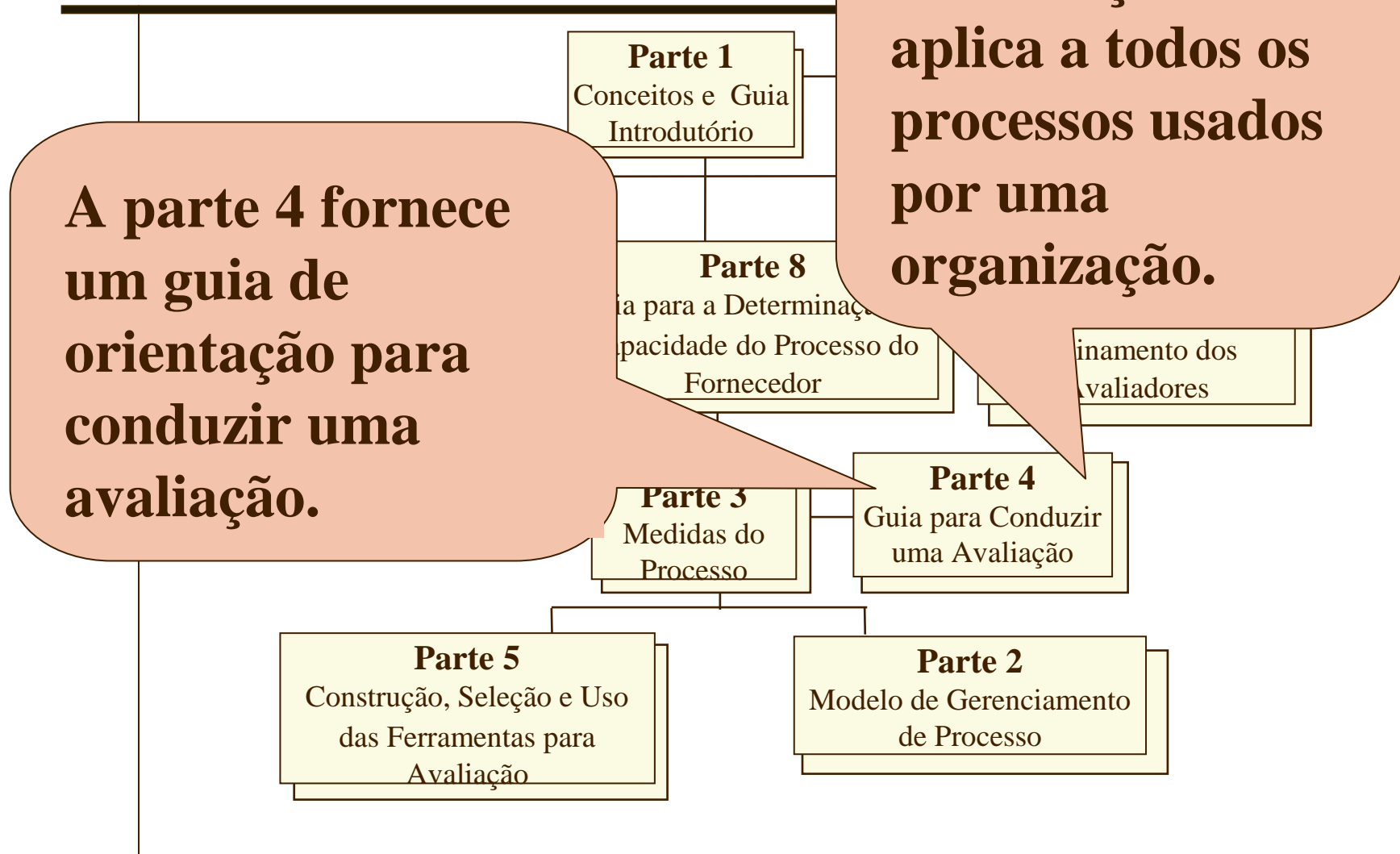
A parte 2 define um modelo ideal de processo de software que forma a base para a avaliação



# Componentes do Modelo SPICE

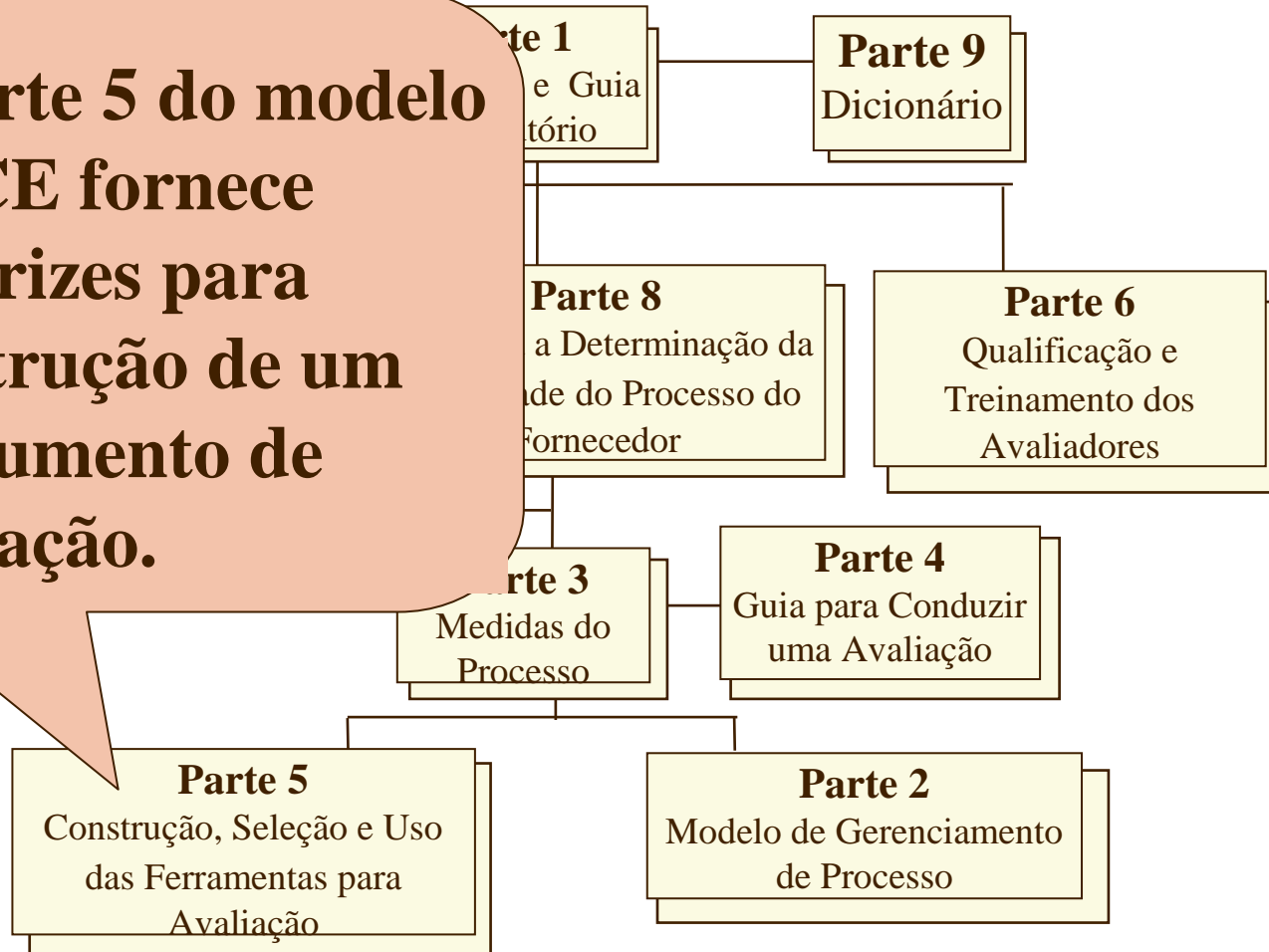


# Componentes do Modelo SPICE



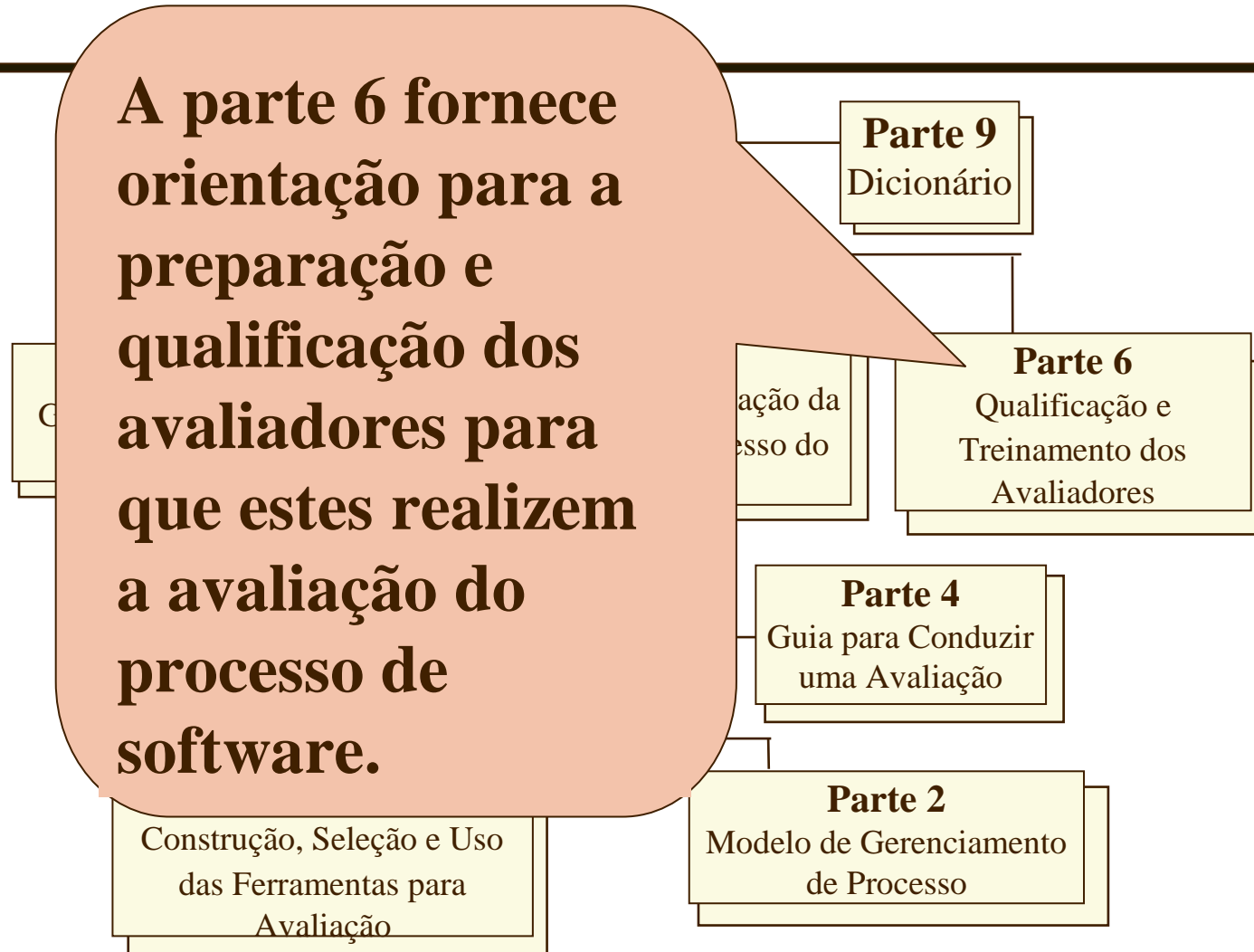
# Componentes do Modelo SPICE

A parte 5 do modelo SPICE fornece diretrizes para construção de um instrumento de avaliação.



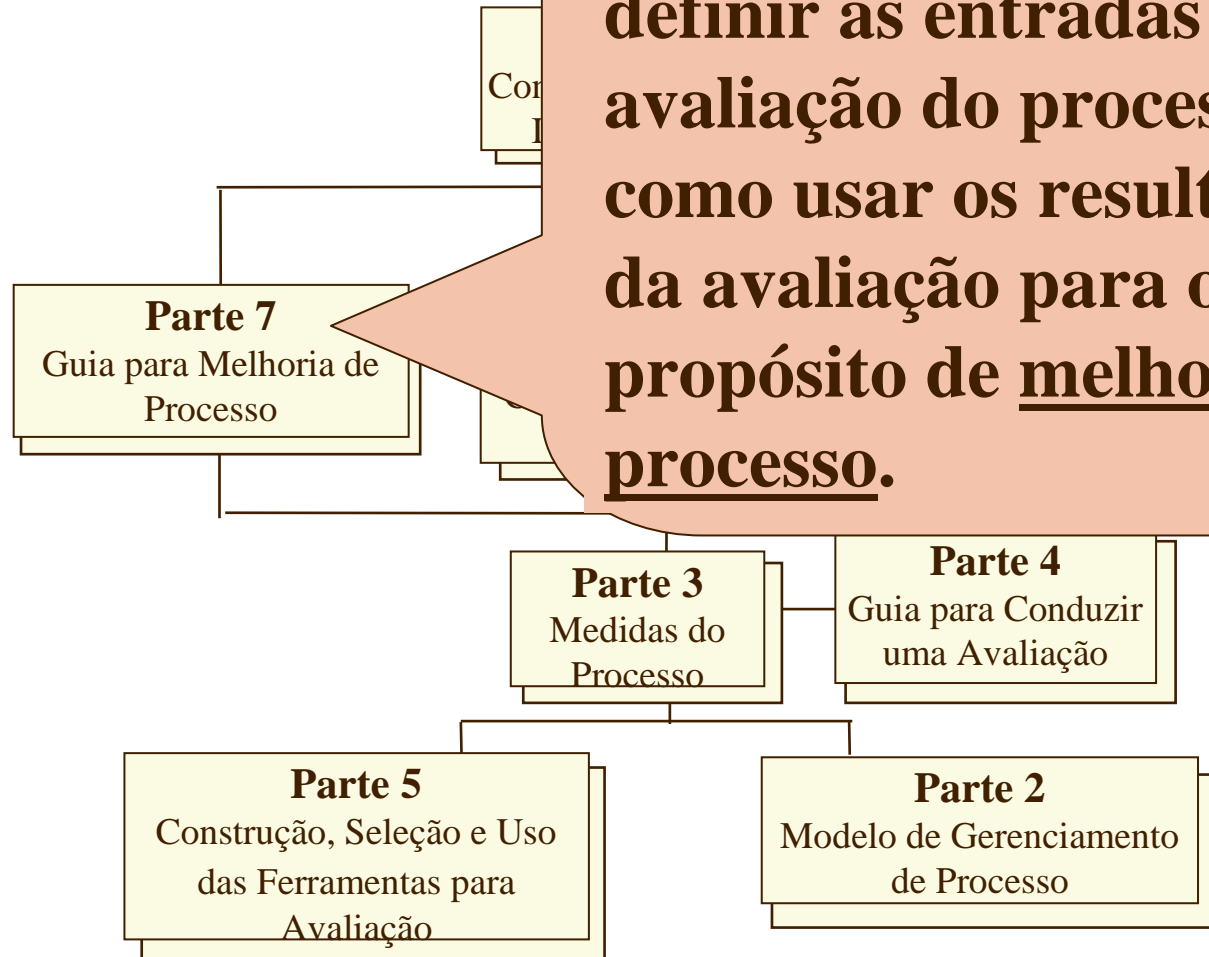
# Componentes do Modelo SPICE

A parte 6 fornece orientação para a preparação e qualificação dos avaliadores para que estes realizem a avaliação do processo de software.



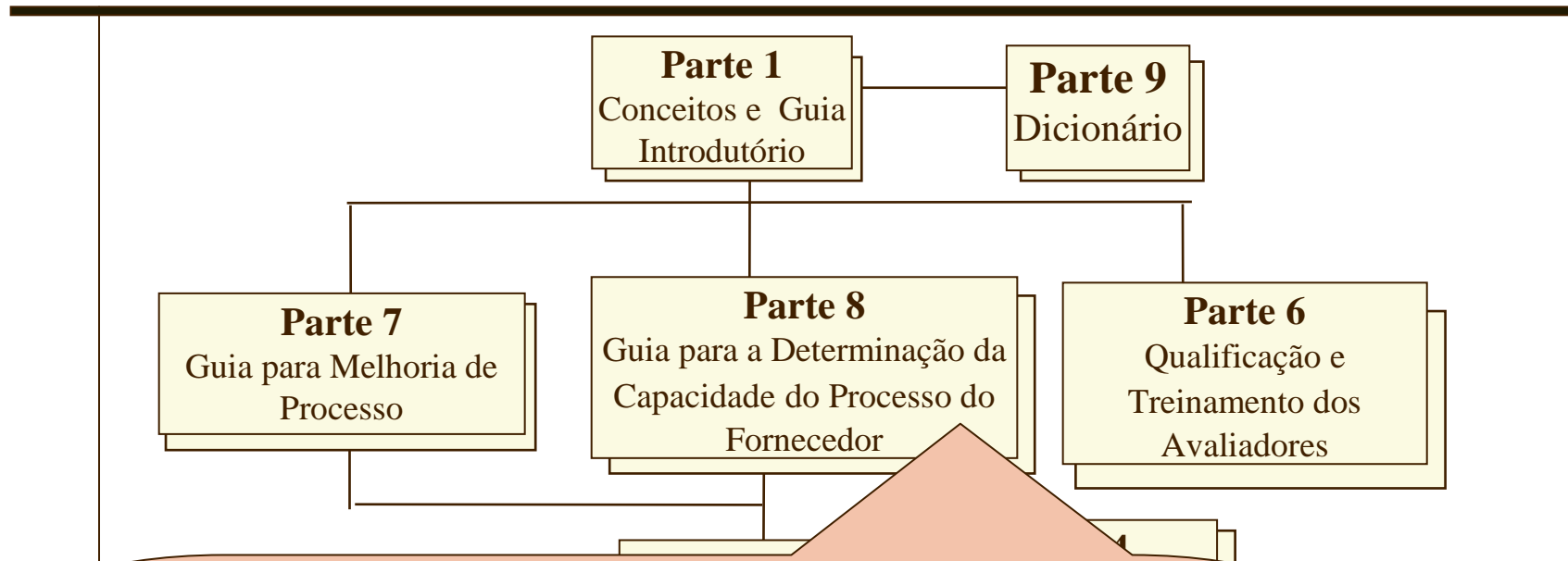


# Componentes



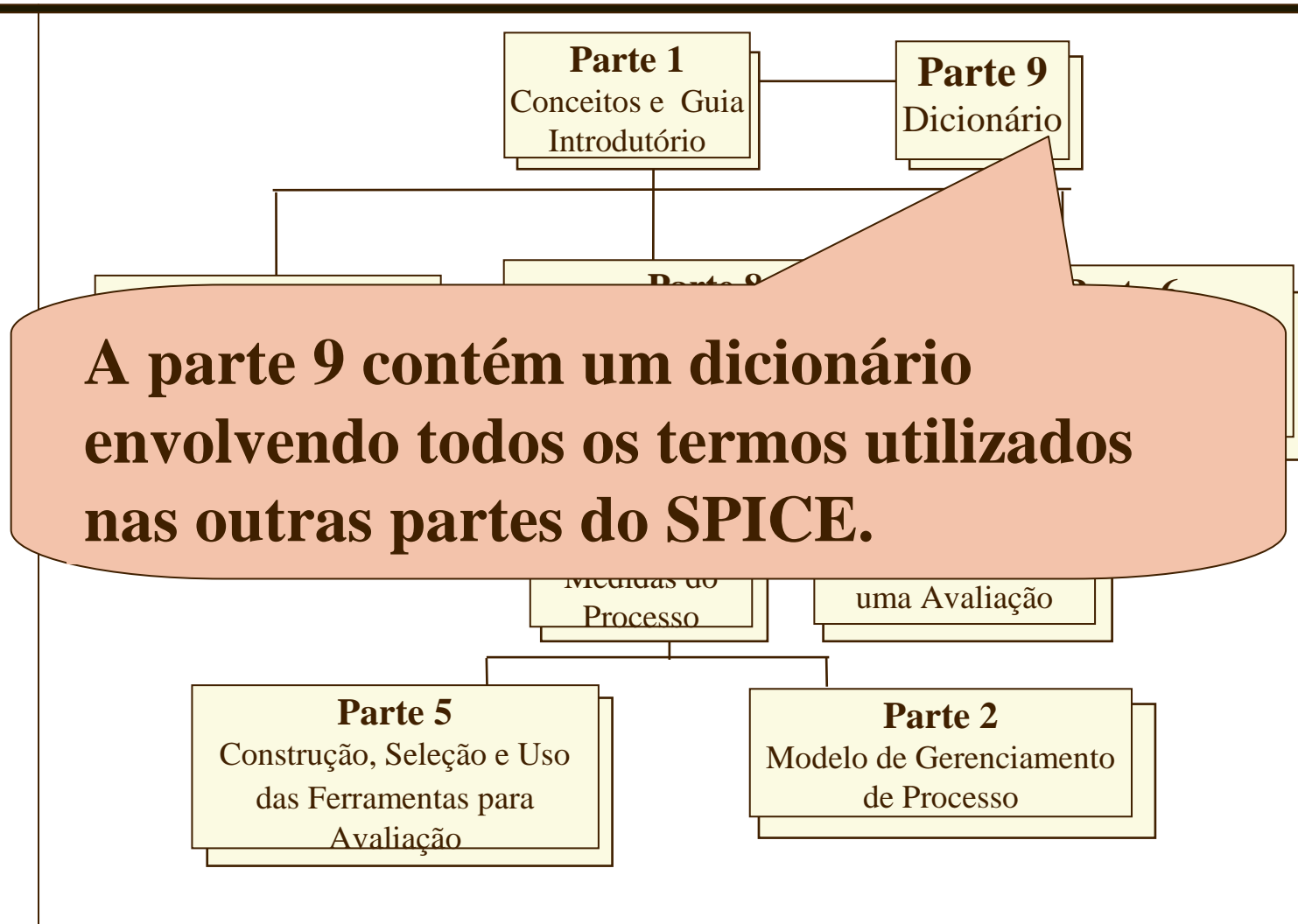
A parte 7 descreve como definir as entradas para a avaliação do processo e como usar os resultados da avaliação para o propósito de melhoria de processo.

# Componentes do Modelo SPICE



**A parte 8 descreve como definir entradas para a avaliação do processo e como usar os resultados da avaliação para o propósito de determinar a capacidade do processo.**

# Componentes do Modelo SPICE



# Conclusão

---

- **Importância** da Qualidade de Software e de Produto para uma Empresa vai além da maturidade e responsabilidade em manter tal maturidade;
- **Valorização** do produto e software é um investimento no conforto do cliente e da empresa.
- **Buscar a qualidade** em um processo de desenvolvimento é a principal atividade dos desenvolvedores e gerentes.