

# Garantia de Qualidade

**UNIP - Araraquara**

**Curso: Ciências da Computação**

**Disciplina: Engenharia de Software**

**Profº: João Paulo Moreira dos Santos**



# QUALIDADE DE SOFTWARE

- O que é **Engenharia de Software**?
- Área da computação voltada à especificação, desenvolvimento e criação de sistemas de software, com a aplicação e práticas de gerência de projetos e outras disciplinas, visando a organização, produtividade e qualidade.



# QUALIDADE DE SOFTWARE

- O que é **Qualidade de Software**?
- Um produto que apresenta um grau de satisfação das necessidades dos clientes sob todos os aspectos.



# QUALIDADE DE SOFTWARE

- Crescente preocupação com a qualidade de software.
  - Dependência das organizações.
- Qualidade do produto resulta do processo empregado durante seu desenvolvimento.
  - Estabelecer previamente as características de qualidade desejável.



# QUALIDADE DE SOFTWARE

- Em que momento temos que incorporar a qualidade?
  - Durante todo o processo de desenvolvimento.
- É possível incorporar ao produto final, após o processo de desenvolvimento ter terminado?
  - Não.
- Existe uma maneira de medir a qualidade de software?
  - Técnicas, métricas e padrões de qualidade.



# QUALIDADE DE SOFTWARE

- **Qualidade do Produto.**
  - **ISSO/IEC 9126**
- **Qualidade do Processo.**
  - **CMM/CMMI**



# QUALIDADE DO PRODUTO

- Qualidade do Produto - ISO/IEC 9126:
  - Padronização mundial.
  - Baseada em três níveis.
    - Características, sub-características e métricas.

# QUALIDADE DO PRODUTO

- Qualidade do Produto - ISSO/IEC 9126:
  - **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?





# QUALIDADE DO PROCESSO

- O que é **processo de software**?
  - Conjunto de atividades e diretrizes referentes ao desenvolvimento de sistemas computacionais.
  - Modelos de Processo de Software.



# QUALIDADE DO PROCESSO

- Existe um processo ideal que seja utilizado para qualquer projeto e organização?
  - Não. As organizações desenvolvem suas próprias abordagens.
  - É importante escolher um modelo apropriado às metas da organização.



# QUALIDADE DO PROCESSO

- O que um processo imaturo pode causar?
  - Prazos não cumpridos;
  - Custos de manutenção excessivos;
  - Qualidade difícil de prever;
  - Impossibilidade de inserir novas tecnologias.



# QUALIDADE DO PROCESSO

- O que um processo *Maturo* pode causar?
  - Prazos cumpridos;
  - Baixo custos de manutenção;
  - Qualidade do produto;
  - Retorno financeiro.



# QUALIDADE DO PROCESSO

- 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.



# QUALIDADE DO PROCESSO

## ► Capability Maturity Model (CMM)

- Modelo de maturidade de processo de software.
- Auxilia as empresas a melhorar seus processos de software.
- Criado em 1986 - Software Engineering Institute (SEI).
- Solicitado pelo Departamento de Defesa dos EUA.
- Necessidade de um método para avaliar a capacidade de seus fornecedores de software.

# QUALIDADE DO PROCESSO

- É um modelo que se refere ao processo pelo qual o produto é construído.
- Seu objetivo é atingir um alto nível de qualidade de produto.
- Não deve ser entendido como sendo uma metodologia.
- Descreve os principais elementos de um processo de desenvolvimento de software.
- Orienta as organizações sobre como ganhar controle do processo de software.



# QUALIDADE DO PROCESSO

- **Capability Maturity Model Integration (CMMI)**
  - Evolução do CMM.
  - Procura estabelecer um modelo único para o processo.
  - Integra diferentes modelos.





# QUALIDADE DO PROCESSO

- **Capability Maturity Model Integration (CMMI)**
  - Supõe que uma organização que possui um processo maduro tem maiores probabilidades de produzir bons produtos do que uma outra cujo processo é imaturo e caótico.
  - Modelo de referência para a qualidade do processo de software.
  - Classifica a organização em um “Nível de maturidade” por meio de uma avaliação formal.



# QUALIDADE DO PROCESSO

- **Capability Maturity Model Integration (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.

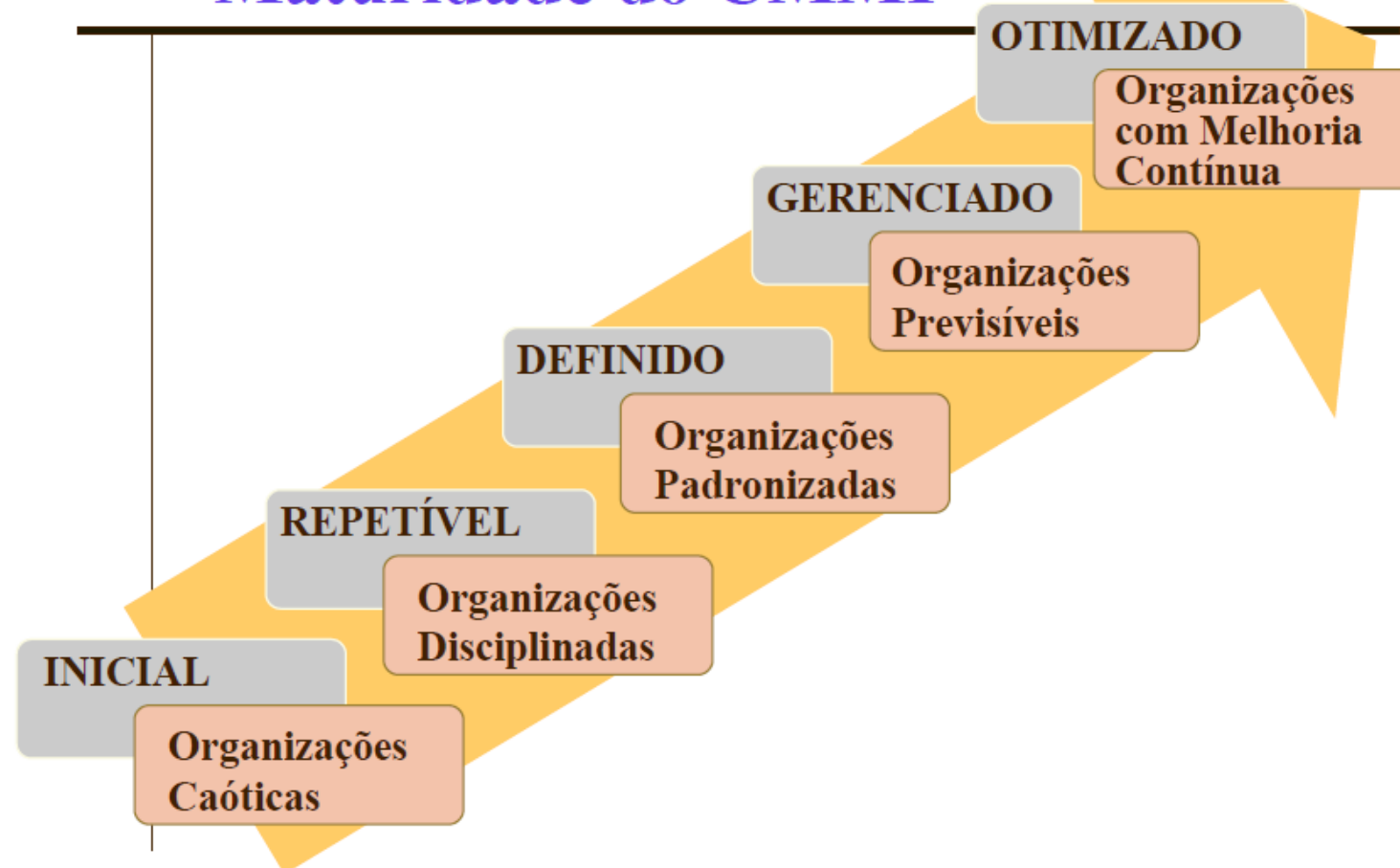


# QUALIDADE DO PROCESSO

- **Capability Maturity Model Integration (CMMI)**
- Os 5 níveis de maturidade do CMMI

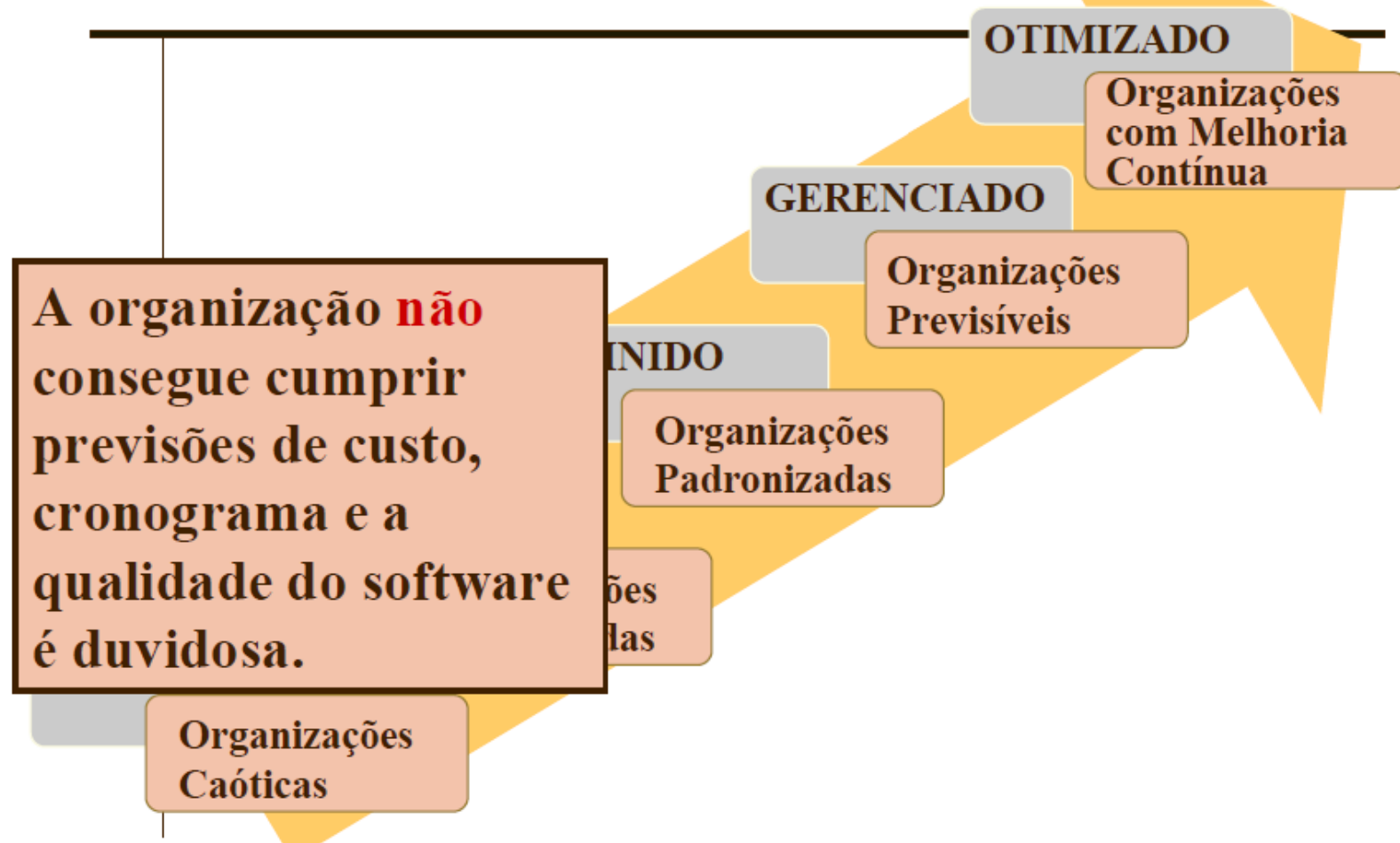
# QUALIDADE DO PROCESSO

## Os 5 Níveis de Maturidade do CMMI



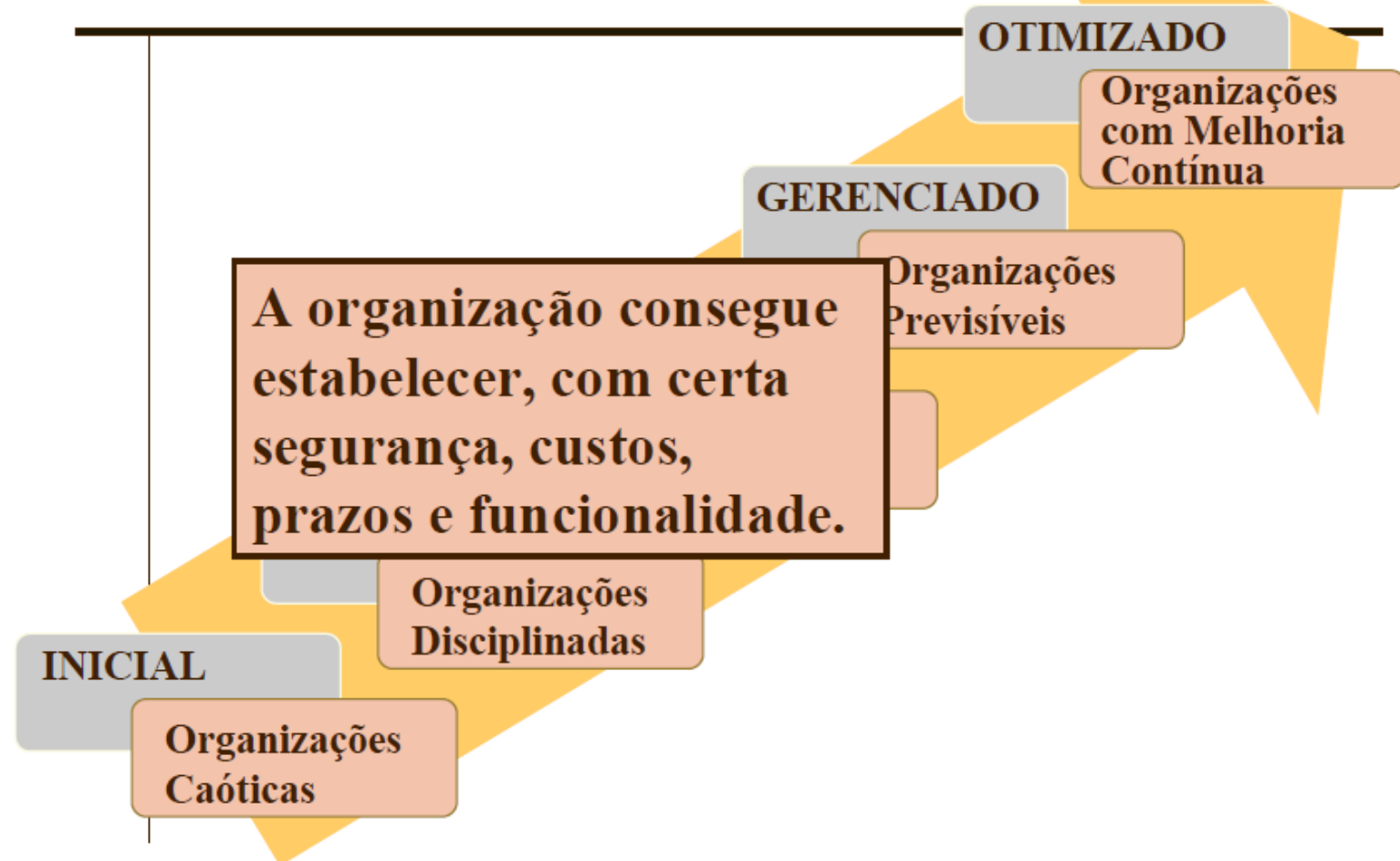
# QUALIDADE DO PROCESSO

## Nível 1 de Maturidade do CMMI



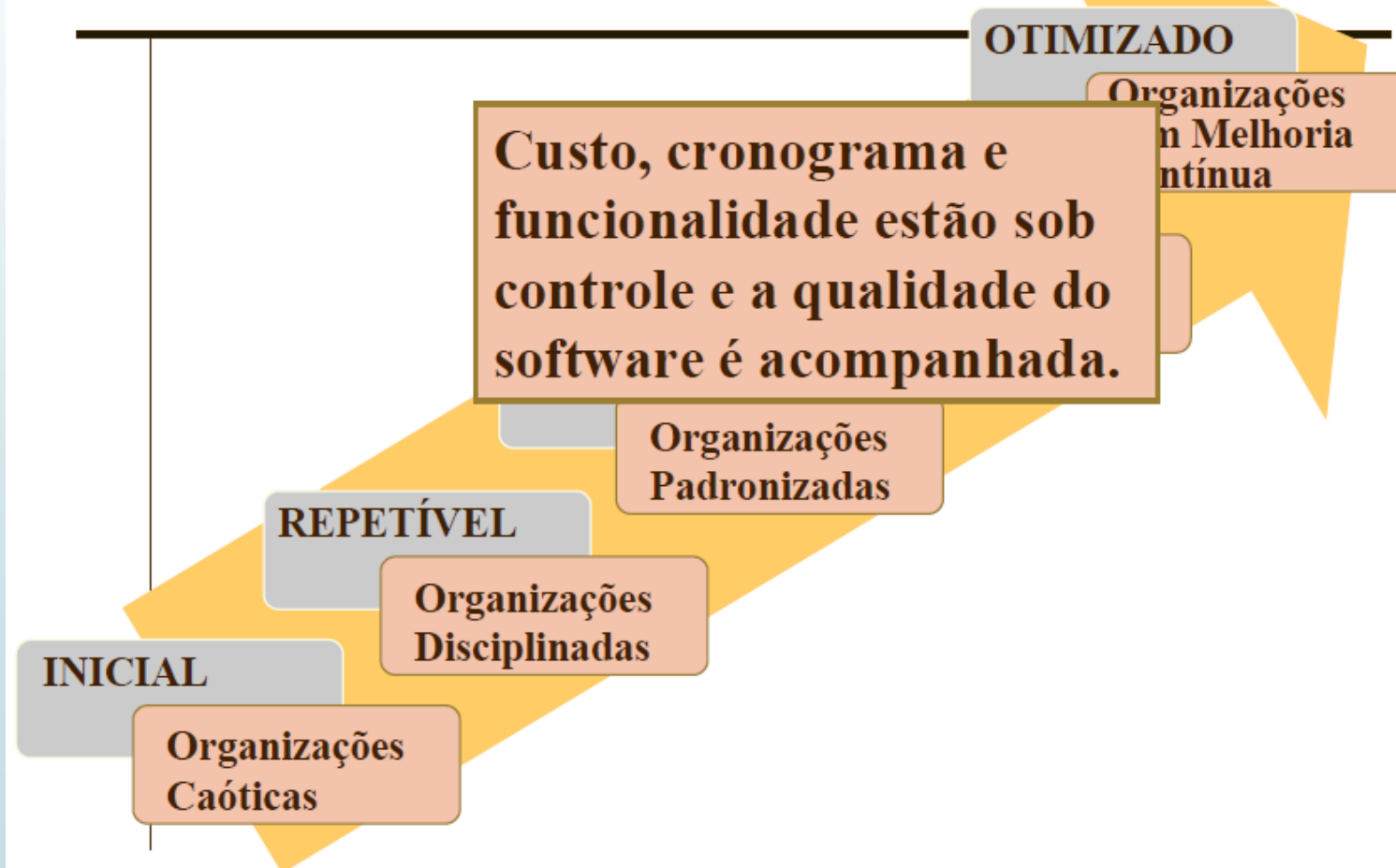
# QUALIDADE DO PROCESSO

## Nível 2 de Maturidade do CMMI



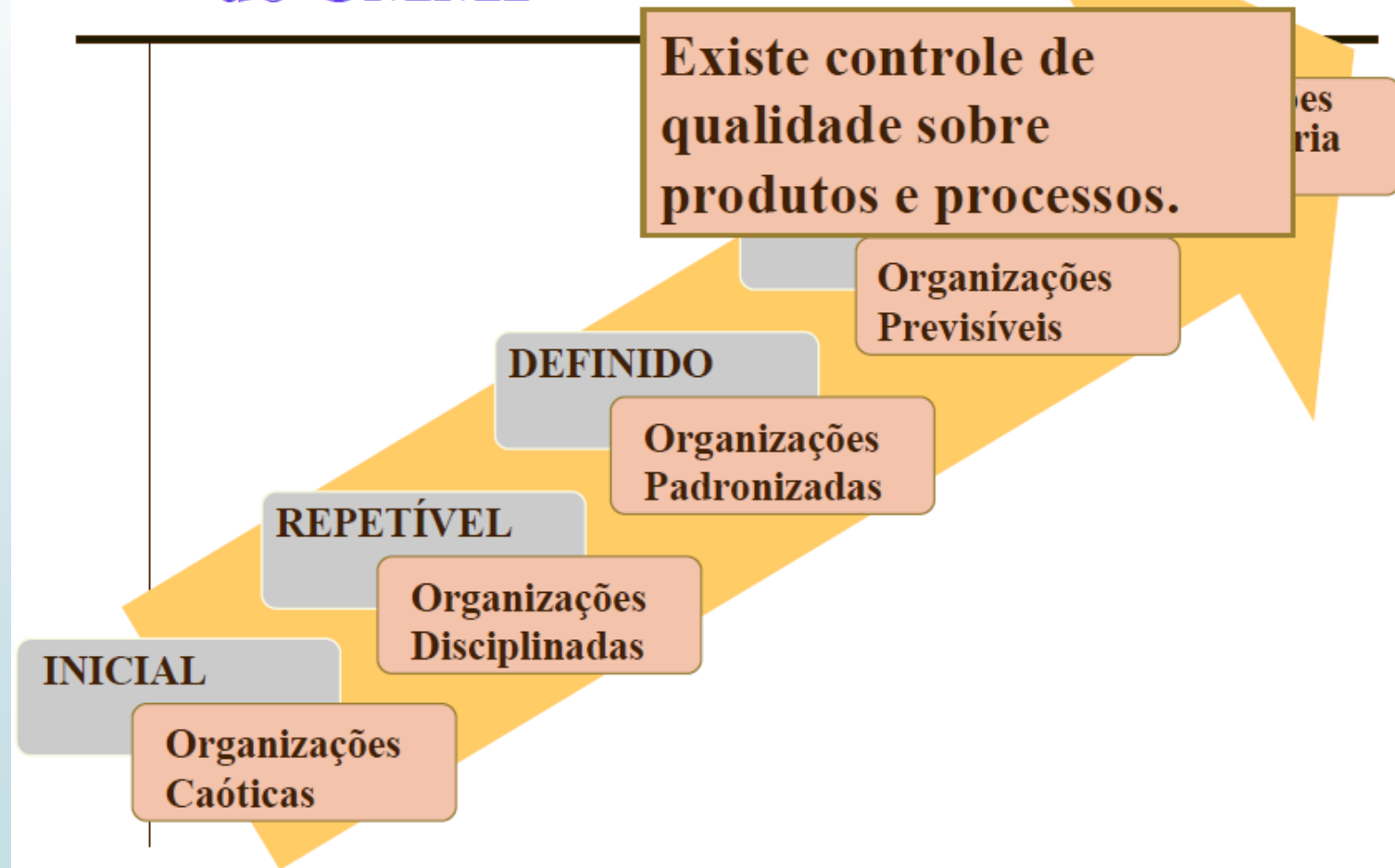
# QUALIDADE DO PROCESSO

## Nível 3 de Maturidade do CMMI



# QUALIDADE DO PROCESSO

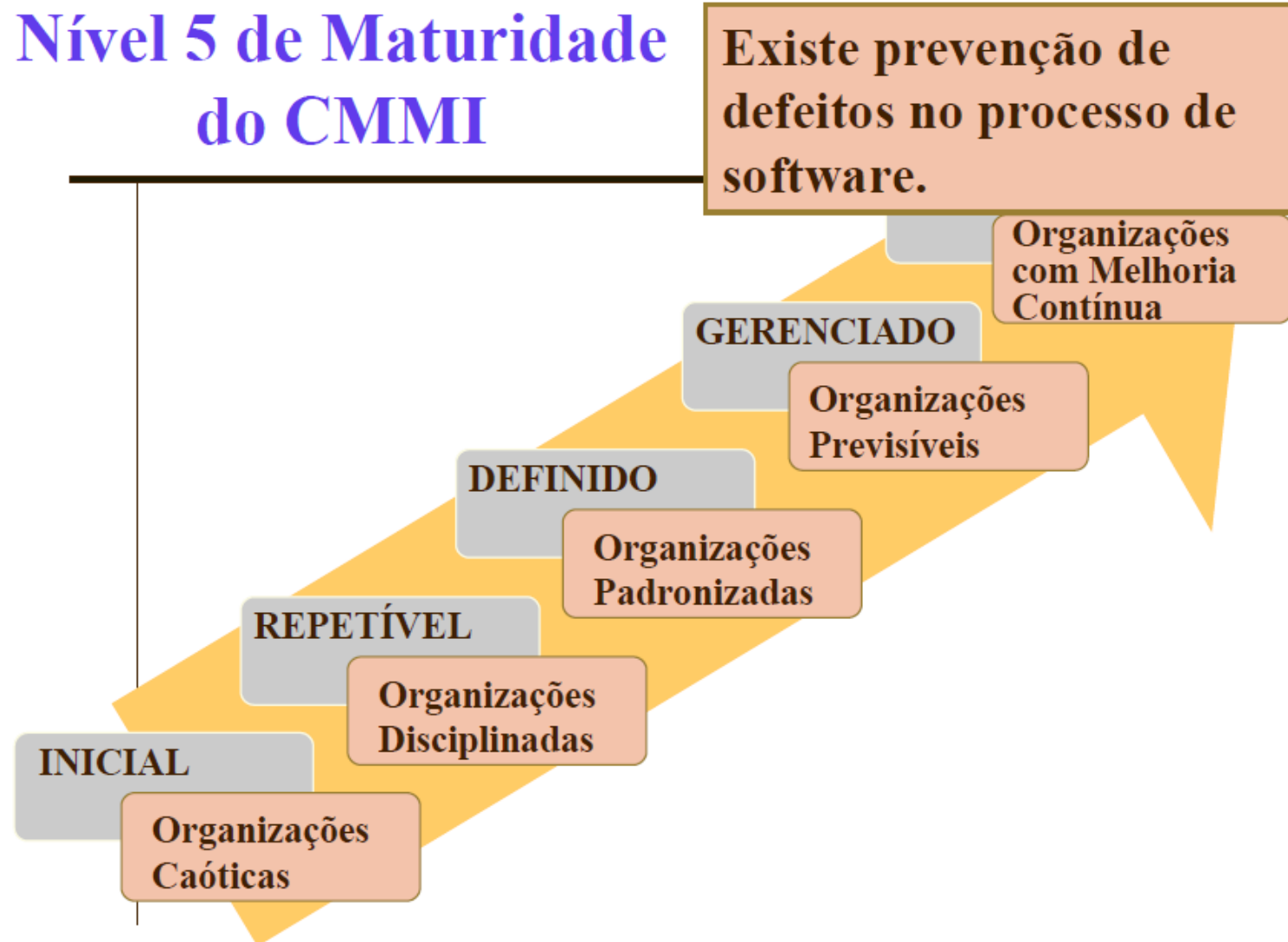
## Nível 4 de Maturidade do CMMI





# QUALIDADE DO PROCESSO

## Nível 5 de Maturidade do CMMI





# QUALIDADE DO PROCESSO

## ► Melhoria do Processo de Software Brasileiro - MPS

- Modelo de qualidade de processo voltada para a realidade do mercado de pequenas e médias empresas no Brasil.
- Baseado no CMMI e na realidade do mercado Brasileiro.
- Custo reduzido em relação às normas estrangeiras, ideal para micro, pequenas e médias empresas.
- Desejo de expansão para américa latina.

# Software

- Por que devemos nos preocupar com a **qualidade** do software?



# Qualidade de software

- Os requisitos de software são a base da qual a qualidade é medida.
  - A falta de conformidade aos requisitos significa falta de qualidade.
- 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.

# Qualidade de software

- Existe um conjunto de **requisitos implícitos** que frequentemente não são mencionados na especificação.
  - Por exemplo, o desejo de uma boa manutenibilidade e a facilidade de uso.
- Se o software atende aos requisitos **explícitos**, mas falha nos requisitos **implícitos**, a qualidade é suspeita.
- 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.

# Qualidade de software

## ► Visões de qualidade de software.



**Usuário**



**Desenvolvedor**



**Organização**

**Facilidade de uso, desempenho,  
confiabilidade dos resultados,  
preços do software, etc.**

**Taxa de defeitos, facilidade de  
manutenção e conformidade em relação  
aos requisitos dos usuários, etc.**

**Cumprimento de prazo, boa previsão de  
custo, boa produtividade.**

# Garantia de qualidade

- **SQA – Software Quality Assurance**
- Conjunto de atividades técnicas aplicadas durante todo o processo de desenvolvimento.
- Garantir que tanto o processo de desenvolvimento quanto o produto de software atinjam os níveis de qualidade especificados.

# Garantia de qualidade

- **SQA – Software Quality Assurance**
- **Atividades de SQA:**
  - Aplicação de métodos técnicos.
    - Ajudar o analista a conseguir uma especificação de qualidade.
    - Ajudar o projetista a desenvolver um projeto de qualidade.
  - Realização de revisões.
    - Avaliar a qualidade da especificação, do projeto, do código, ...
  - Atividades de teste de software.
    - Ajudar a garantir que a detecção de erros seja efetiva.
  - Aplicação de padrões e procedimentos formais.
    - Garantir que estes sejam seguidos durante o desenvolvimento.



# Garantia de qualidade

- **SQA – Software Quality Assurance**

- **Atividades de SQA:**

- Processo de controle de mudanças.

- Atividade associada ao gerenciamento de configuração de software.

- Mecanismos de medição.

- Apoio no acompanhamento da qualidade de software.

- Avaliar o impacto de mudanças metodológicas e procedimentais.

- Anotação e manutenção de registros.

- Procedimentos para coleta e disseminação de informações de garantia de qualidade.

# Atividades de V&V

- Dentre as atividades de SQA, estão as atividades de verificação e validação de software.
- O objetivo é minimizar a ocorrência de erros e riscos associados.
  - Detectar a presença de erros nos produtos de software.

# Atividades de V&V

## ► **Verificação:**

- Assegurar consistência, completude e corretude do produto em cada fase e entre fases consecutivas do ciclo de vida.

**Estamos construindo corretamente o produto?**

- Assegurar que o produto, ou uma determinada função do mesmo, esteja sendo implementado corretamente.
  - Verifica-se inclusive se os métodos e processos de desenvolvimento foram adequadamente aplicados.

# Atividades de V&V

## ► **Validação:**

- Assegurar que o produto sendo desenvolvido corresponde ao produto correto, conforme os requisitos do usuário.

**Estamos construindo o produto certo?**

# Atividades de V&V

- V&V abrangem um amplo conjunto de atividades de SQA:
  - Revisões técnicas formais;
  - Auditoria de qualidade e configuração;
  - Simulação;
  - Estudo de viabilidade;
  - Revisão da documentação;
  - Revisão da base de dados; e
  - Testes.

# Atividades de V&V

- V&V abrangem um amplo conjunto de atividades de SQA:
  - Revisões técnicas formais;
  - Auditoria de qualidade e configuração;
  - Simulação;
  - Estudo de viabilidade;
  - Revisão da documentação;
  - Revisão da base de dados; e
  - **Testes.**
- V&V envolvem atividades de **análise estática** e de **análise dinâmica.**

# Atividades de V&V

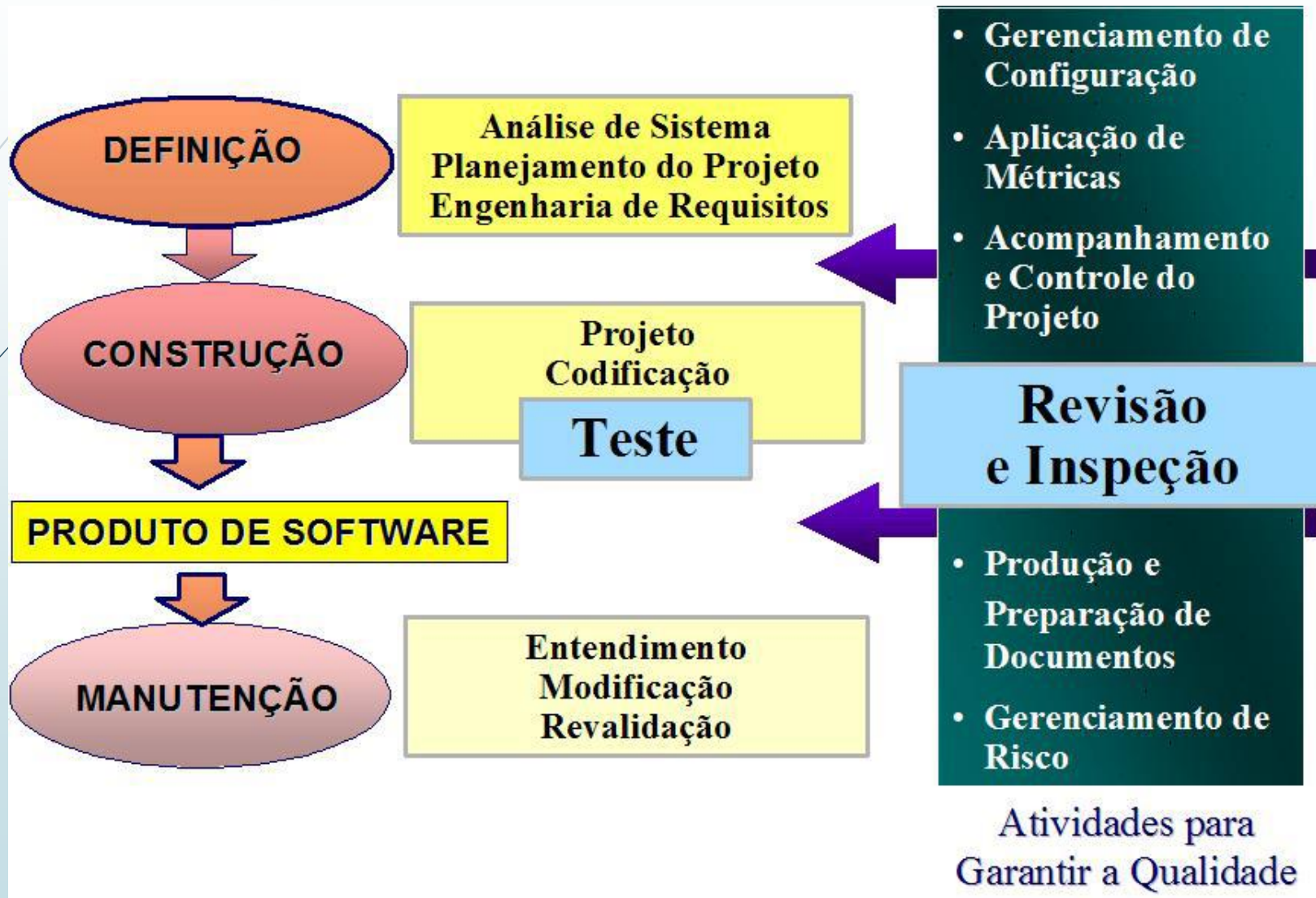
- **Análise Estática:**
- Não requerem a execução propriamente dita do produto.
- Podem ser aplicadas em qualquer produto intermediário do processo de desenvolvimento.
  - Documento de requisitos, diagramas de projeto, código-fonte, planos de teste, ...
- As revisões são o exemplo mais clássico de análise estática.
  - Inspeção;
  - Walkthrough;
  - Peer Review.

# Atividades de V&V

- **Análise Dinâmica:**
- Requerem a execução do produto.
  - Código ou quaisquer outras representações executáveis do sistema.
- Exemplos de atividades que constituem uma análise dinâmica do produto:
  - Teste de software;
  - Simulação.



# SQA no processo de desenvolvimento



# SQA no processo de desenvolvimento

## ➤ Aspectos positivos:

- O software terá menos defeitos latentes.
- Maior confiabilidade resultará e maior satisfação do cliente.
- O custo do ciclo de vida global do software pode ser reduzido.
- Os custos de manutenção podem ser reduzidos.

## ➤ Aspectos Negativos:

- Difícil de ser instituída em pequenas empresas.
- Representa uma mudança cultural.
  - Mudança nunca é fácil.