

CMM

Capability Maturity Model: Um Modelo para Melhoria do Processo (de Produção) de Software

Odisnei Galarraga
odisnei@atlas.unisinos.br

Objetivos

- z Explorar os conceitos básicos de qualidade de software
- z Ter uma visão geral do *Capability Maturity Model* (CMM) quanto a filosofia, estrutura e componentes do modelo
- z Rápido comparativo com normas ISO equivalentes/complementares

Tópicos

- z Origem do CMM
- z Conceitos básicos de Qualidade de Software
- z CMM - Estrutura e seus componentes
- z Modelo *IDEAL*
- z *Team Software Process (TSP) & Personal Software Process (PSP)*

Tópicos

- z Últimas mudanças (CMMs)
- z CMM x ISO
- z Futuro
- z Para mais informações...

Origem

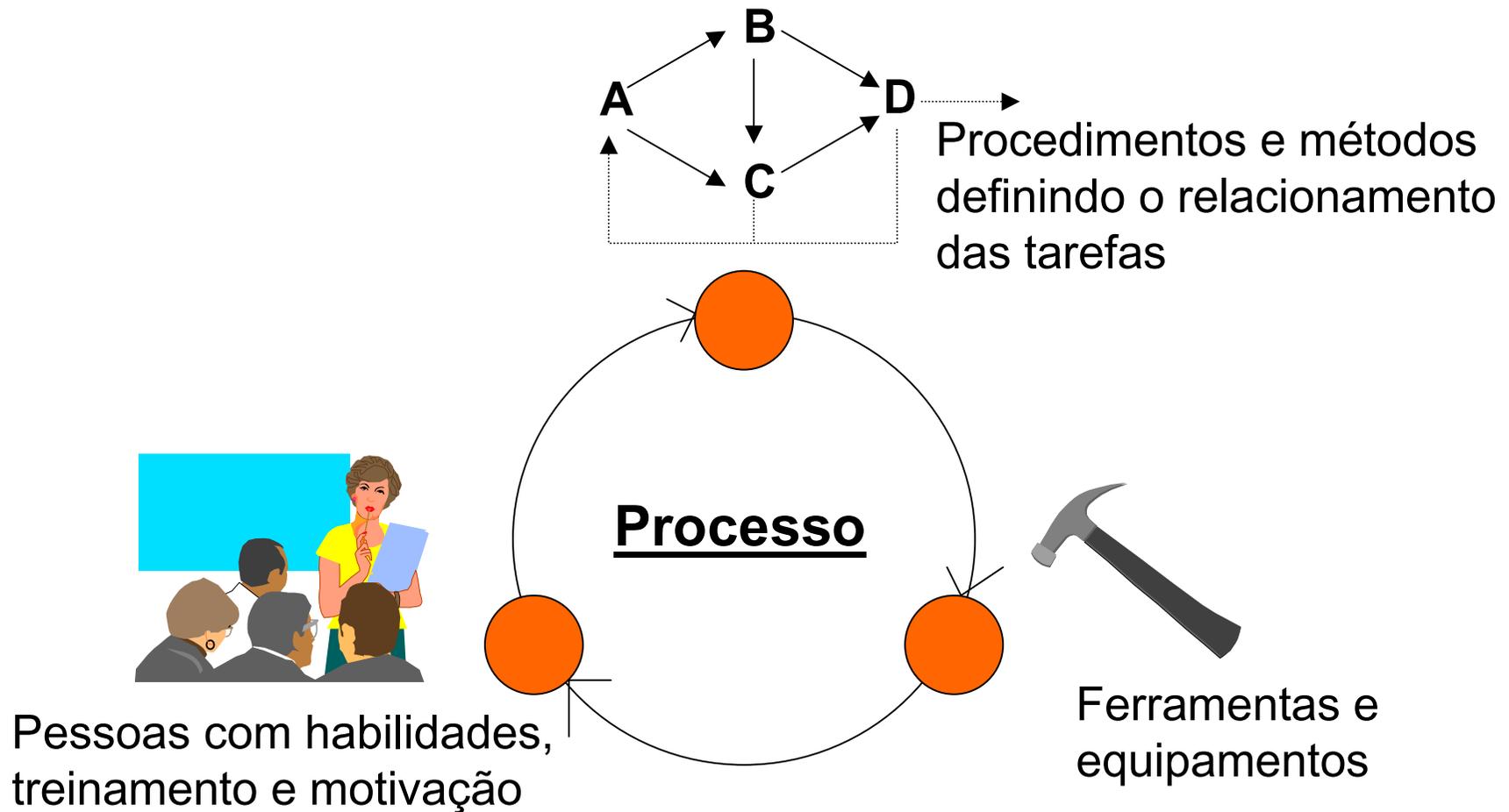
Software Engineering Institute (SEI)

Carnegie Mellon University Pittsburgh - PA

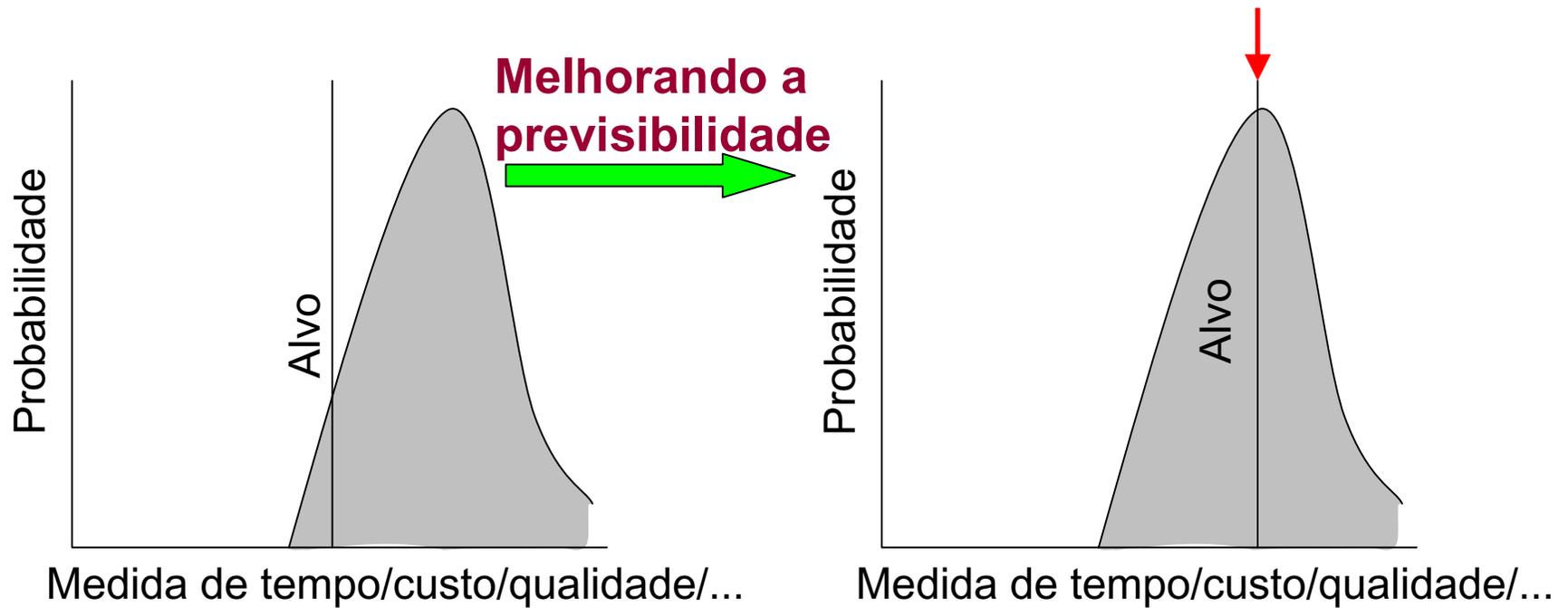
Baseado nas idéias de Watts S. Humphrey

MATURIDADE  **CAPACIDADE DO PROCESSO**

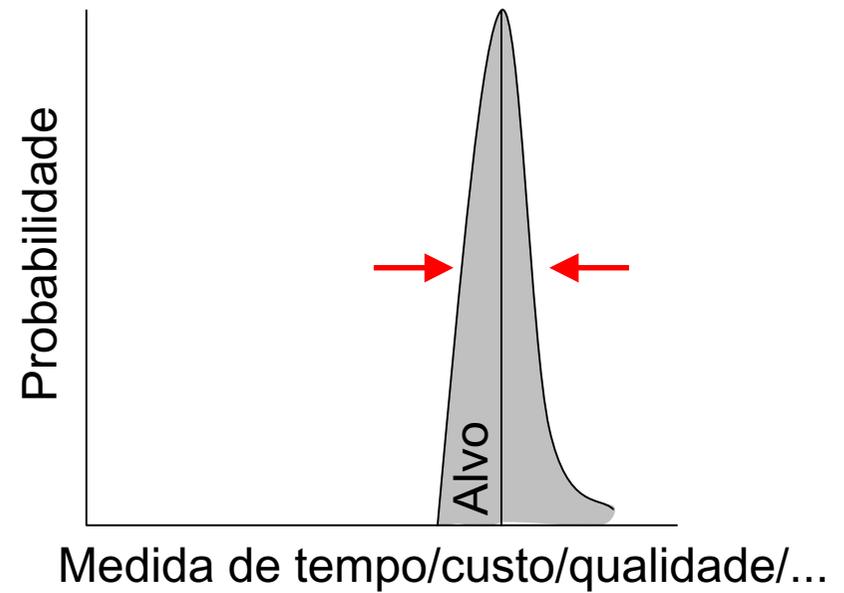
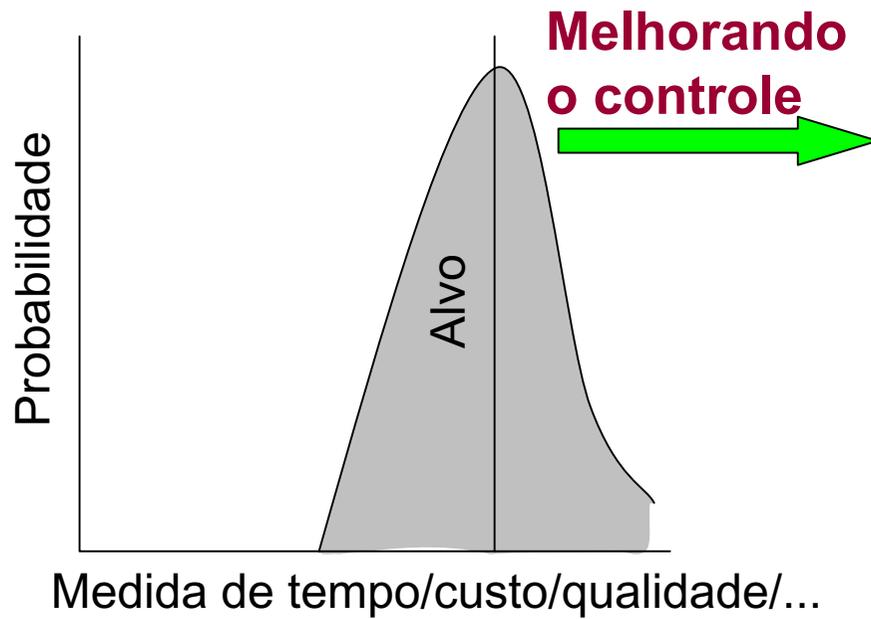
Juntando as partes



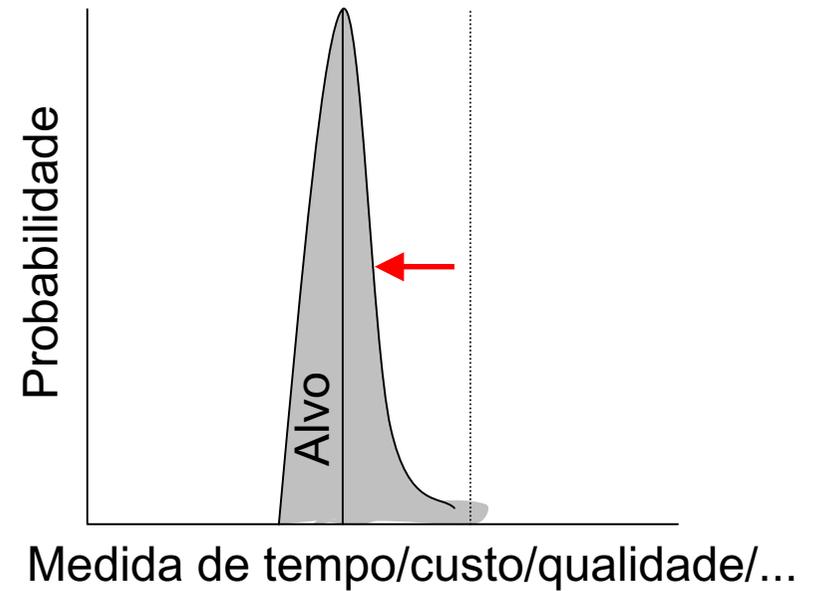
Previsibilidade



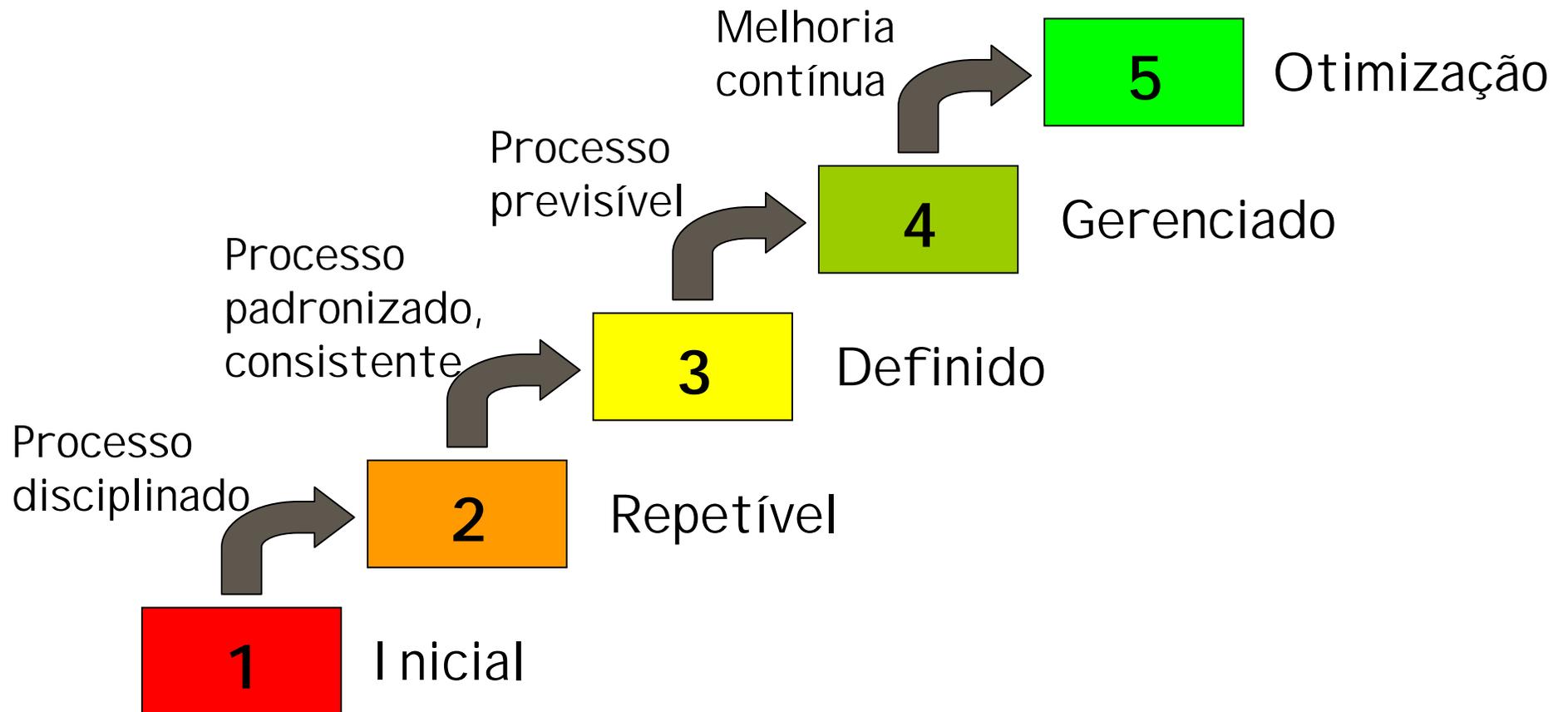
Controle



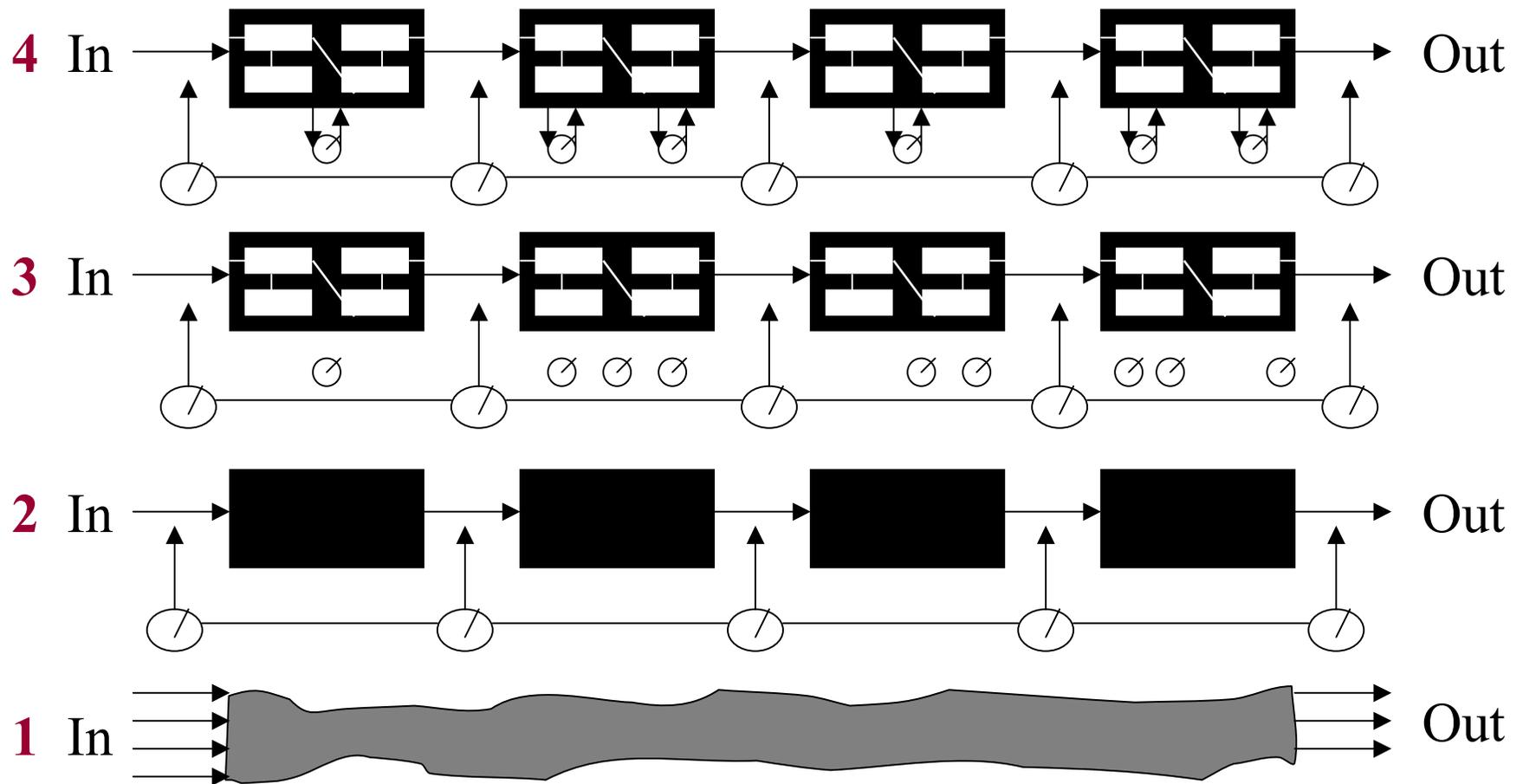
Efetividade



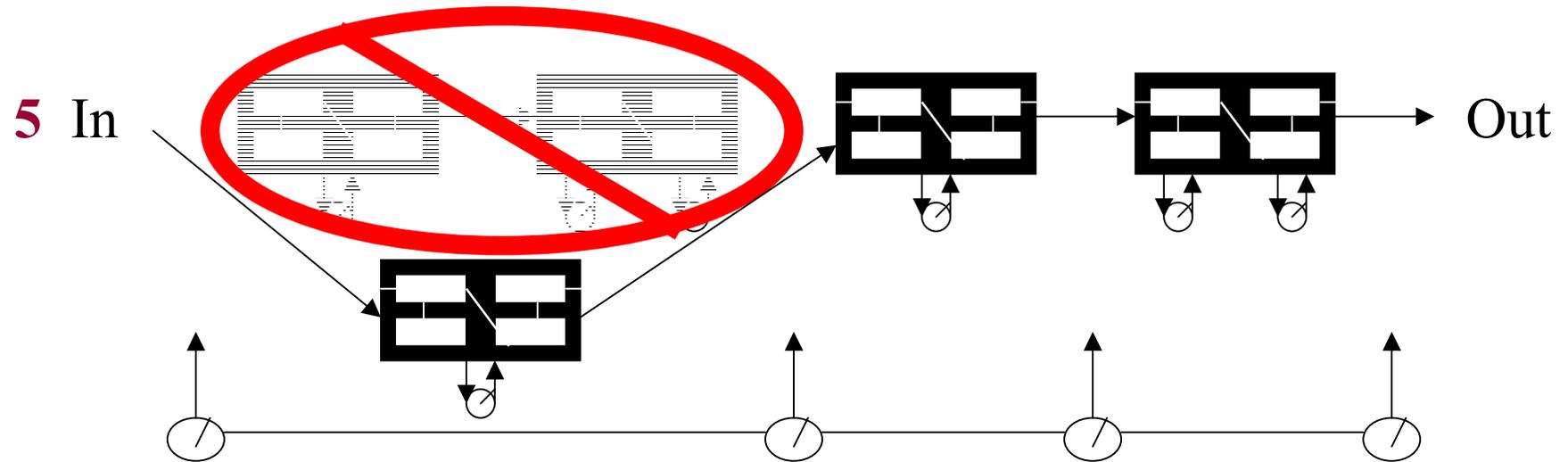
Níveis de maturidade



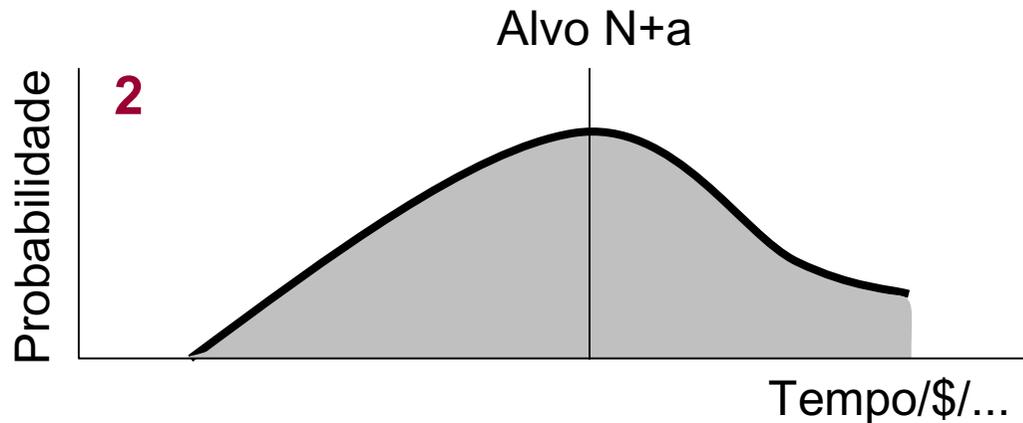
Visibilidade



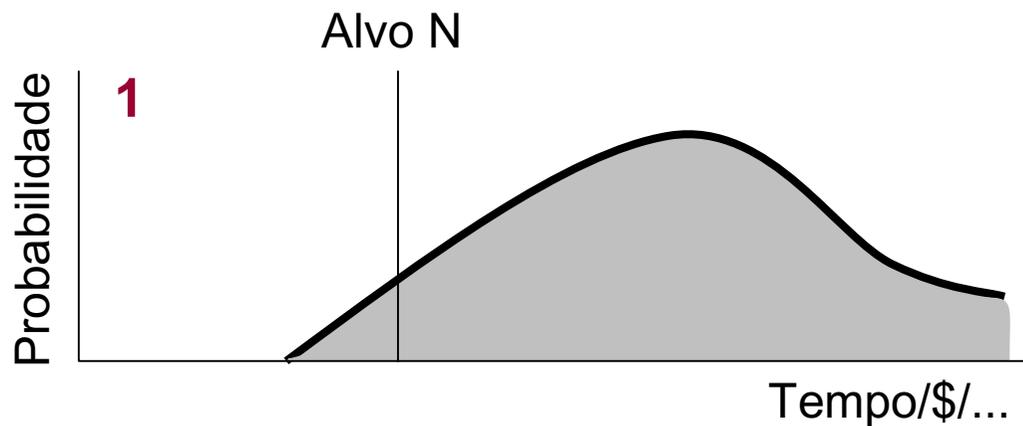
Visibilidade



Nível de maturidade = capacidade

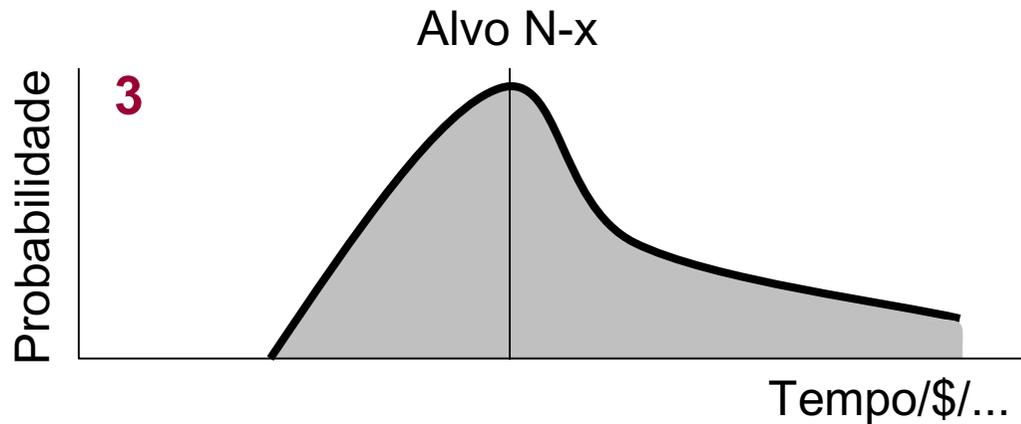


Os planos baseados na performance do passado são mais realistas em organizações Nível 2

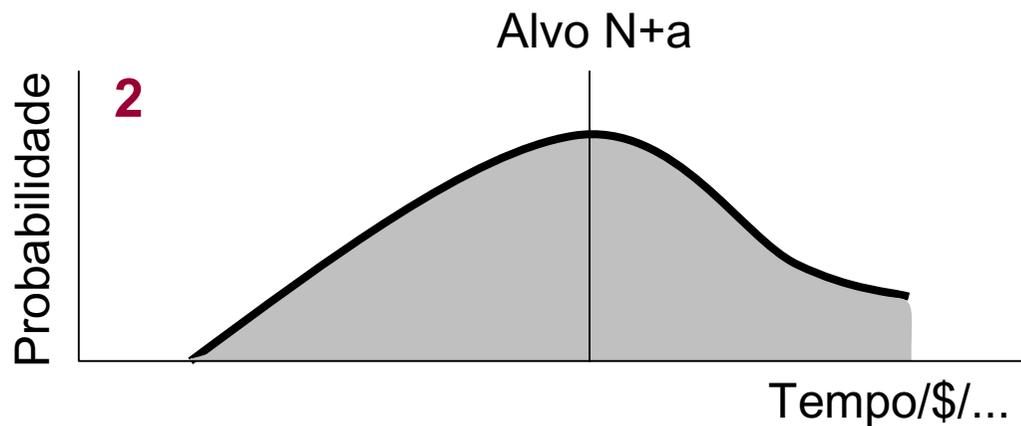


O Cronograma e a previsão de custos normalmente “estouram” em organizações Nível 1

Nível de maturidade = capacidade

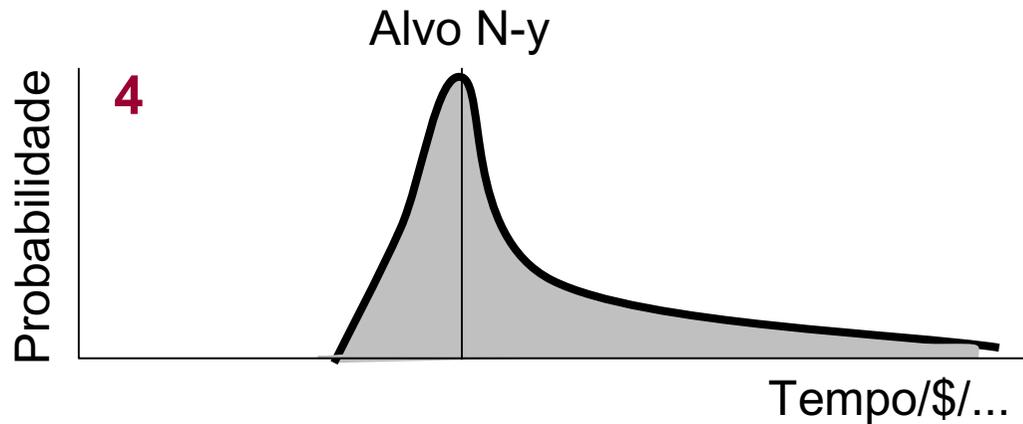


Com processos bem definidos, a performance aumenta em organizações Nível 3

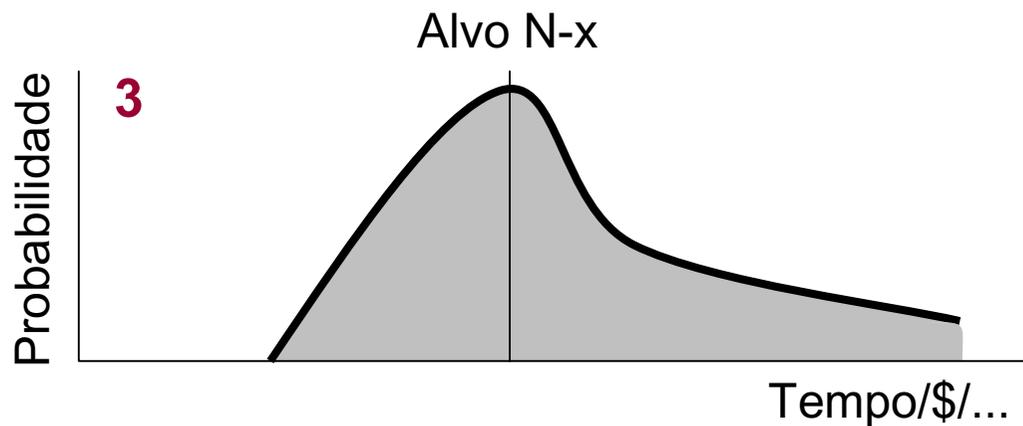


Os planos baseados na performance do passado são mais realistas em organizações Nível 2

Nível de maturidade = capacidade

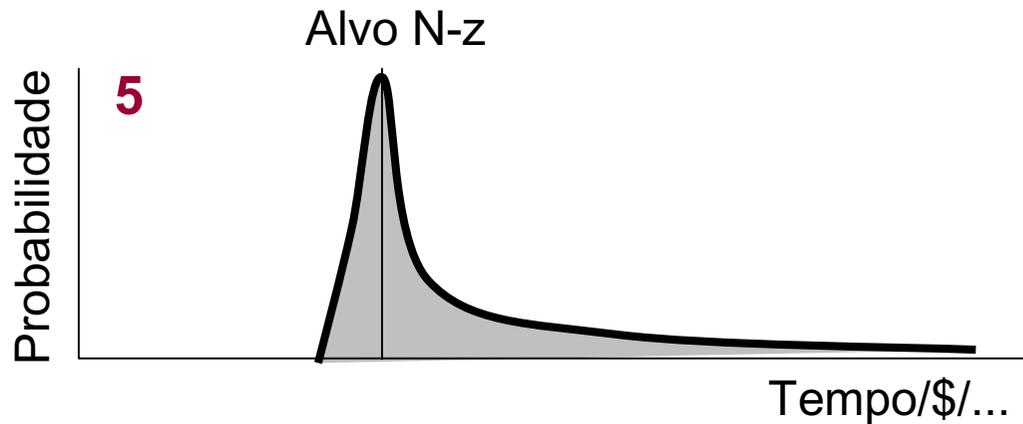


Com base na análise quantitativa do processo e do produto, a performance continua a evoluir em organizações Nível 4

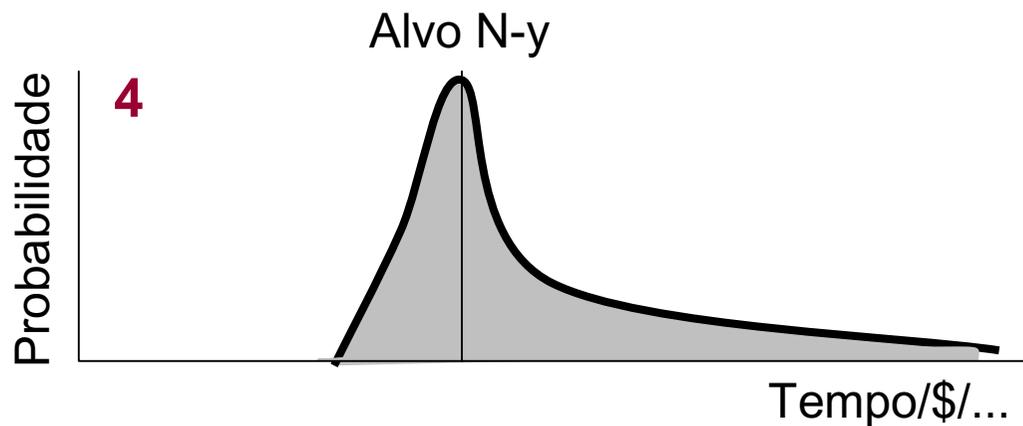


Com processos bem definidos, a performance aumenta em organizações Nível 3

Nível de maturidade = capacidade

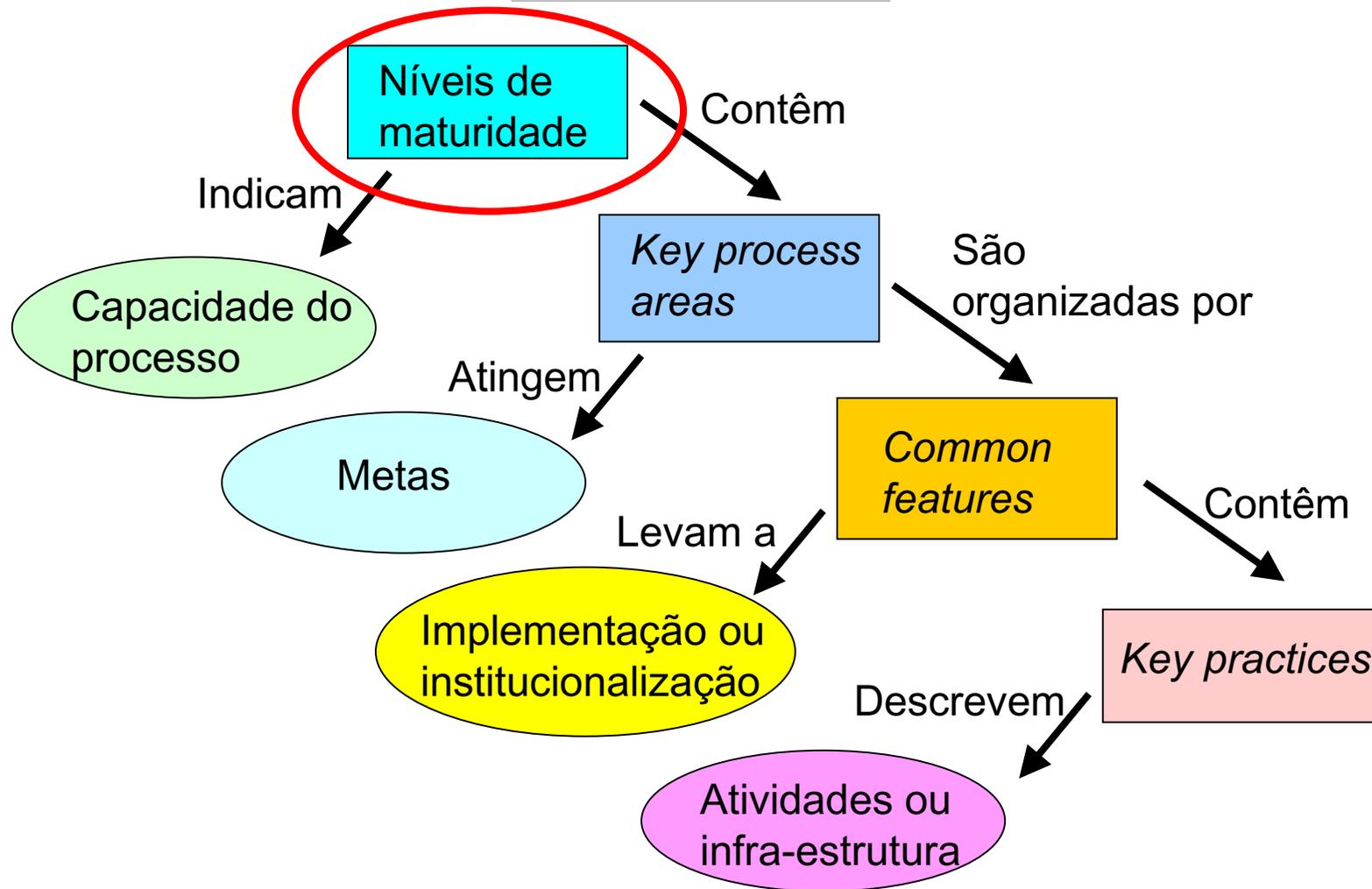


A performance evolui continuamente em organizações Nível 5

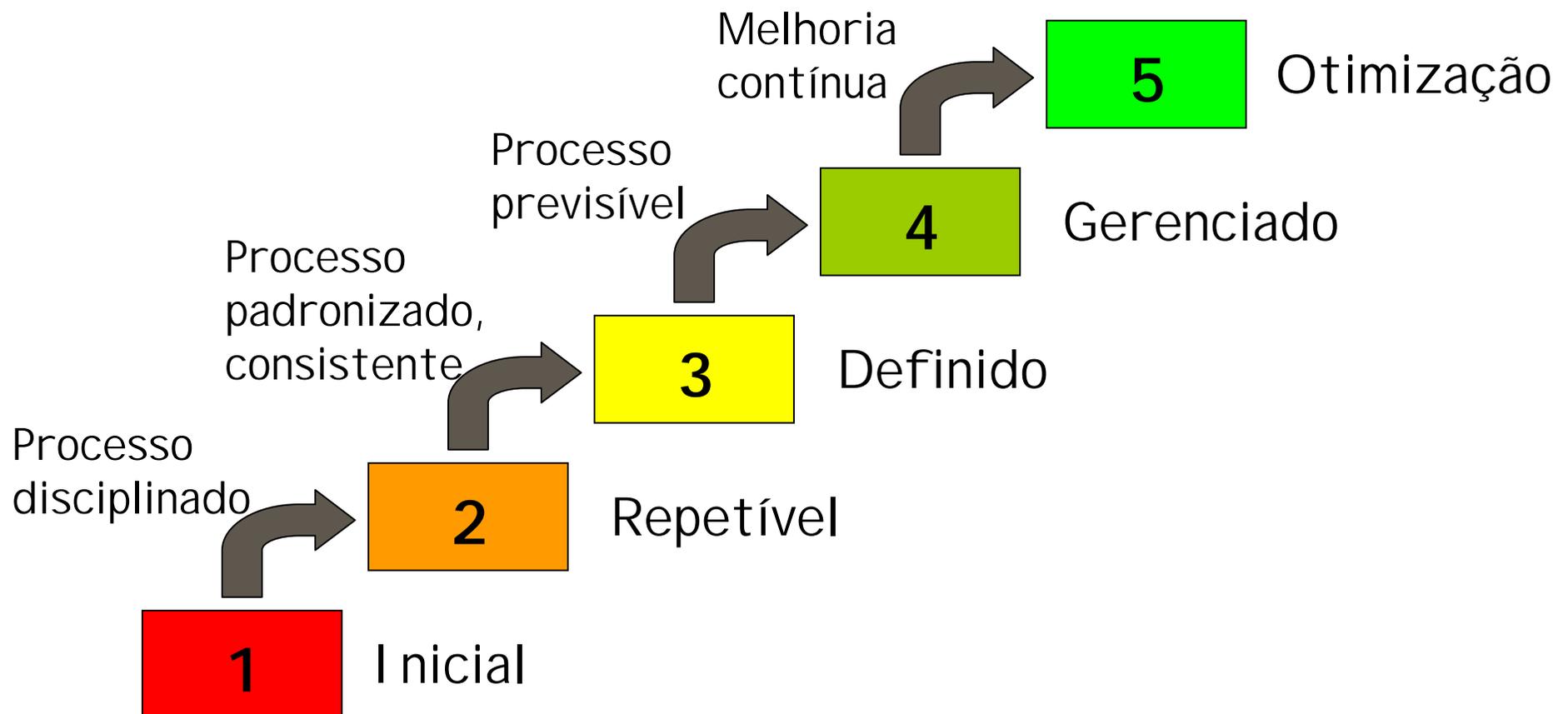


Com base na análise quantitativa do processo e do produto, a performance continua a evoluir em organizações Nível 4

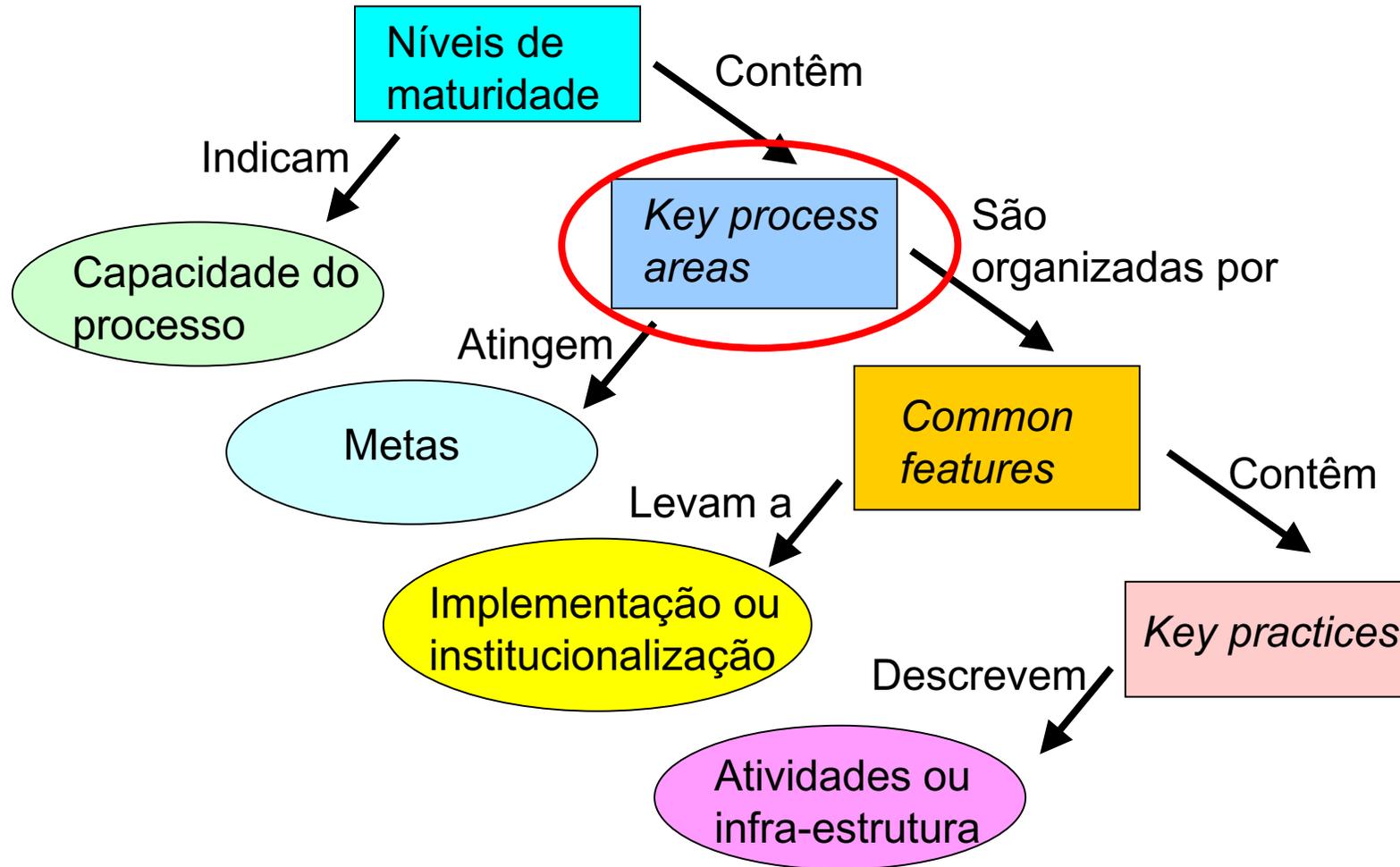
Estrutura



Níveis de maturidade (vistos anteriormente)



Estrutura



Key process areas

Repetível

2

Gerenciamento de requisitos
Planejamento de projeto de software
Acompanhamento e supervisão de projeto de software
Gerenciamento de subcontratação (terceiros)
Garantia da qualidade de software
Gerenciamento de configuração de software



1 Inicial

Key process areas

Definido

3

Foco nos processos da organização
Definição dos processos da organização
Programa de treinamento
Gerenciamento integrado de software
Engenharia de produto de software
Coordenação “interequipes”
Revisão por pares



2 Repetitivo

Key process areas

Gerenciado

4

Gerenciamento quantitativo dos processos
Gerenciamento da qualidade de software



3 Definido

Key process areas

Otimização

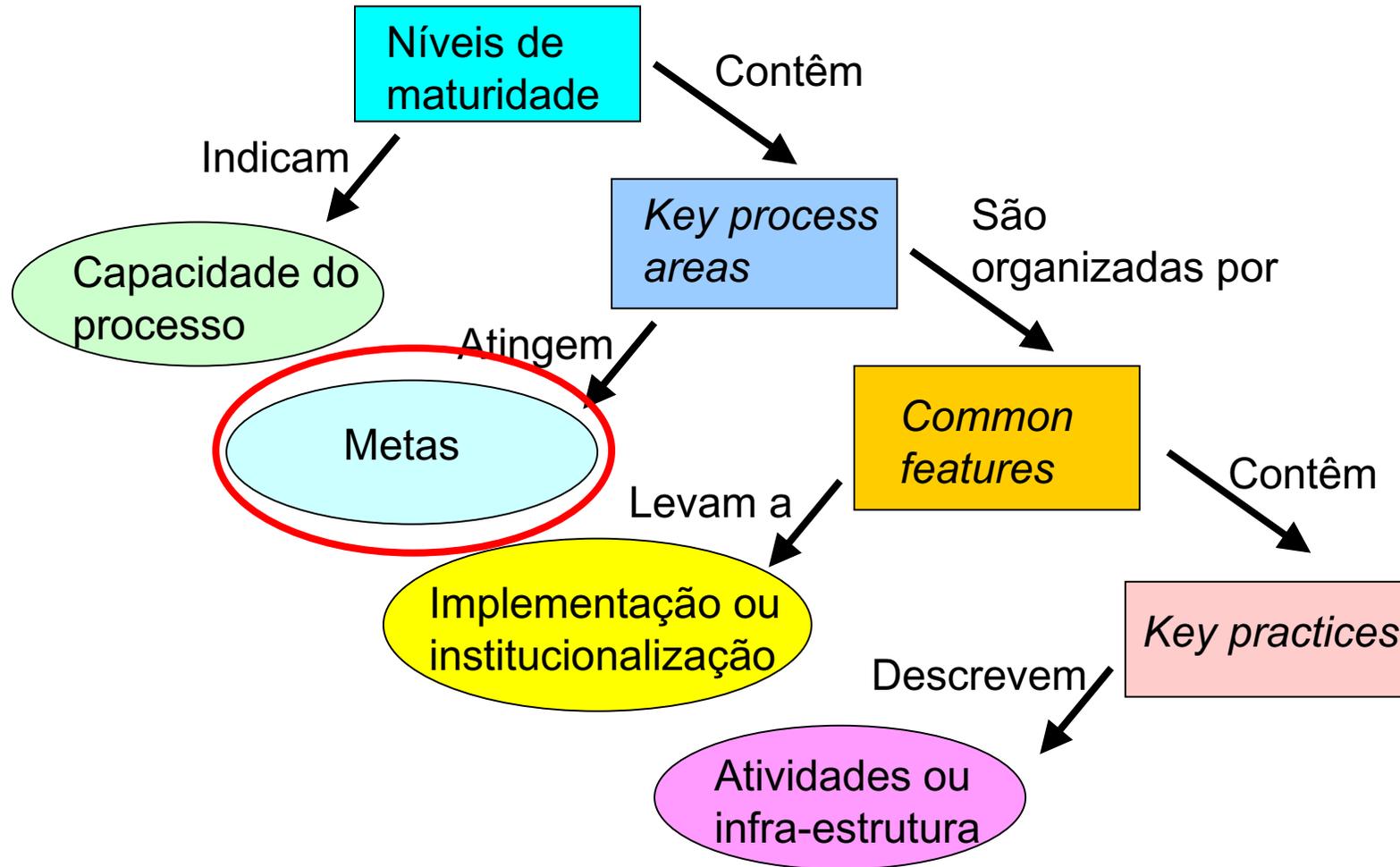
5

Prevenção de defeitos
Gerenciamento de mudanças tecnológicas
Gerenciamento de mudanças no processo



4 Gerenciado

Estrutura



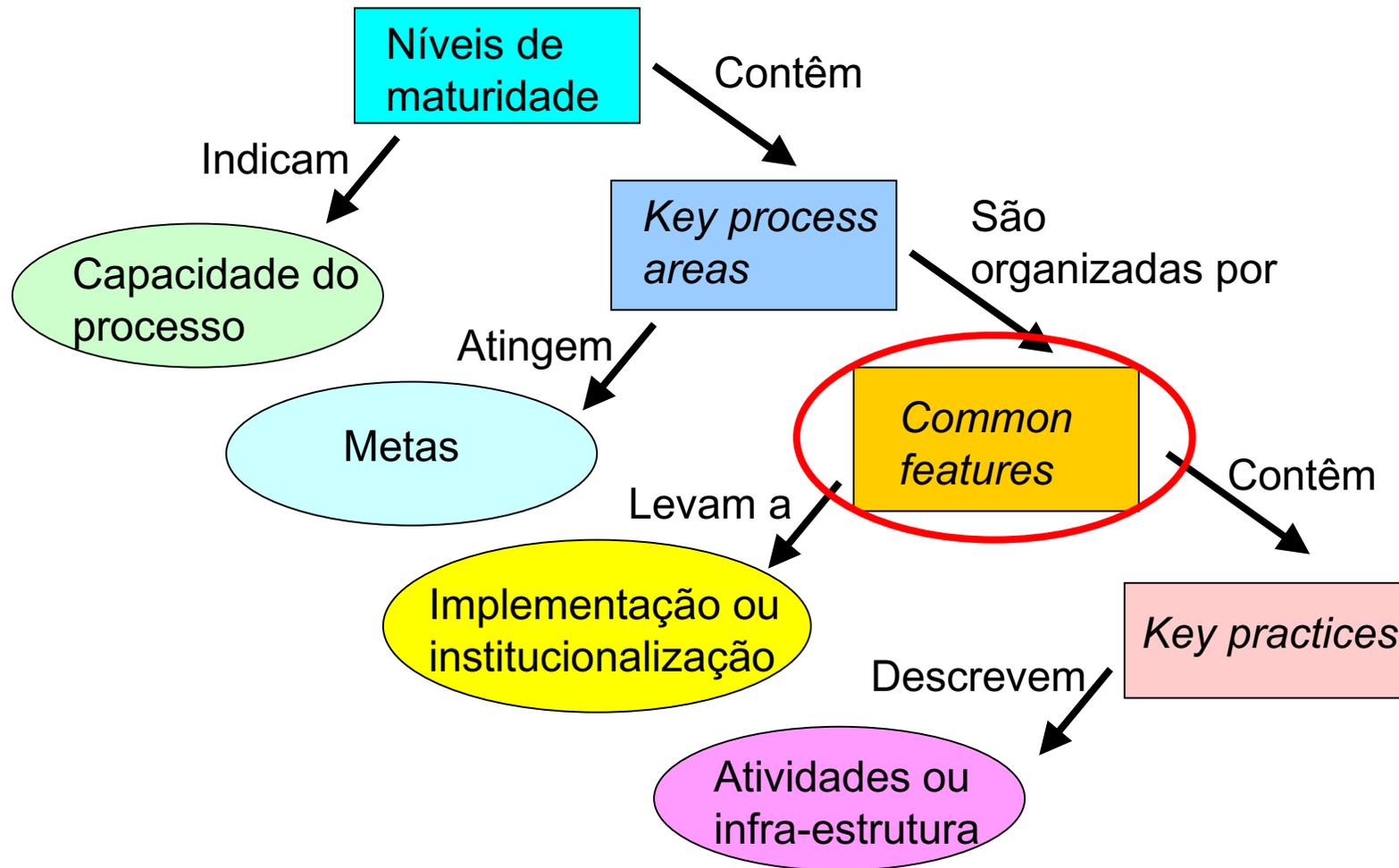
Metas: exemplo

z **Nível 2**

Key process area: Planejamento de projeto de software

- z **Meta 1:** Estimativas estão documentadas para uso no planejamento e acompanhamento do projeto de software
- z **Meta 2:** Atividades e compromissos do projeto de software estão planejados e documentados
- z **Meta 3:** Grupos e indivíduos envolvidos concordam com seus compromissos relativos ao projeto de software

Estrutura



Common features

- z Compromisso de Executar (*Commitment to Perform*)
- z Capacidade para Executar (*Ability to Perform*)
- z Atividades Executadas (*Activities Performed*)
- z Medição e Análise (*Measurement and Analysis*)
- z Verificação da Implementação (*Verifying Implementation*)

Common features

z Compromisso de Executar (*Commitment to Perform*)

Descreve as ações que a organização deve adotar para garantir que o processo está estabelecido e vai perdurar. Normalmente envolve políticas organizacionais e liderança.

Common features

- z Capacidade para Executar *(Ability to Perform)*

Descreve as pré-condições que devem existir no projeto ou organização para implementar o processo de software de forma competente. Normalmente envolve recursos, estruturas organizacionais e treinamento.

Common features

z Atividades Executadas (Activities Performed)

Descreve as atividades, papéis e procedimentos necessários para implementar a *key process area*. Normalmente envolve estabelecer planos e procedimentos, executar o trabalho, acompanhá-lo, e tomar ações corretivas quando necessário.

Common features

z Medição e Análise *(Measurement and Analysis)*

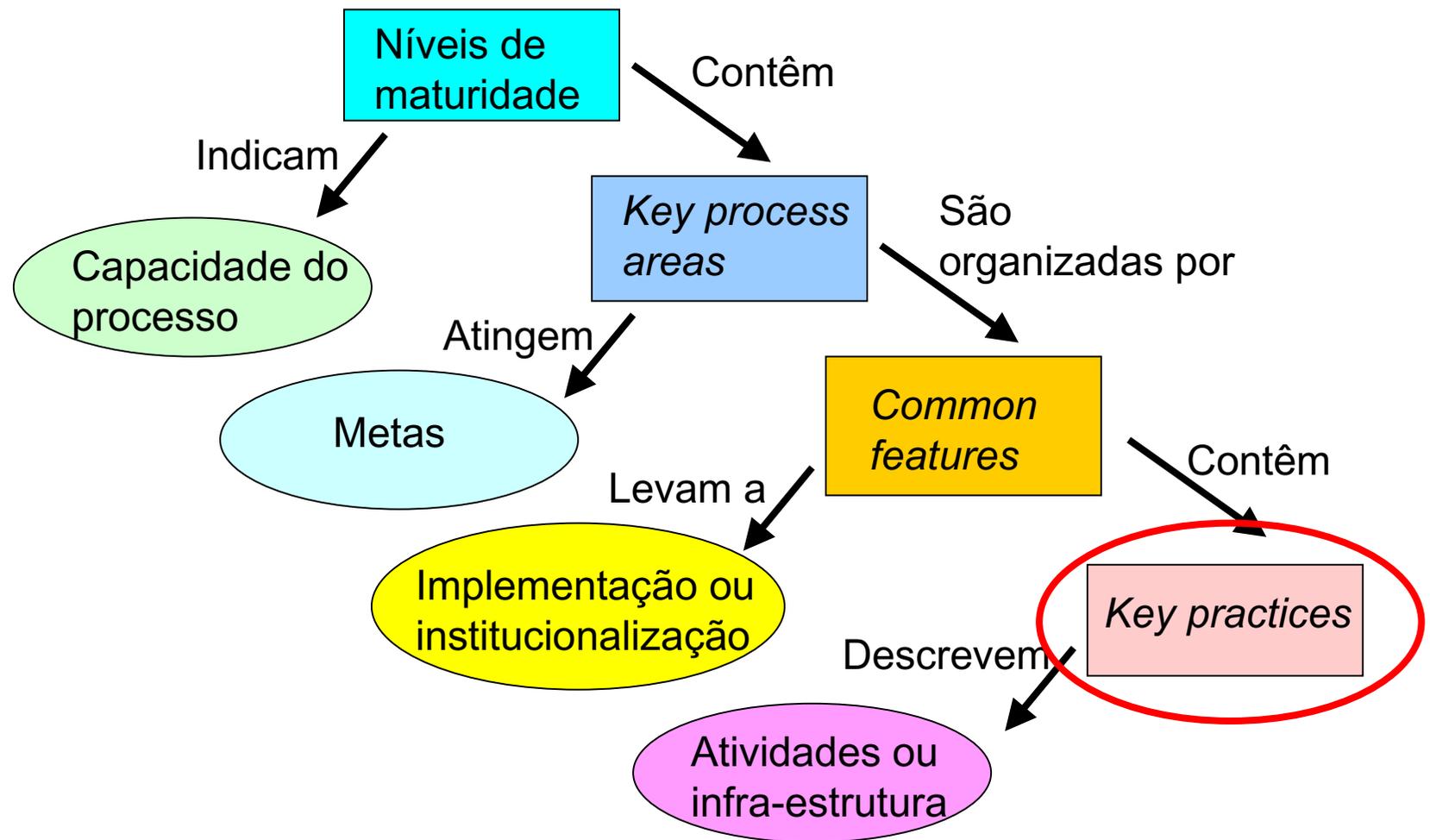
Descreve as práticas básicas de medição que são necessárias para determinar um status relativo ao processo. Essas medições são usadas para controlar e melhorar o processo. Normalmente inclui exemplos de medidas que poderiam ser obtidas.

Common features

z Verificação da Implementação (Verifying Implementation)

Descreve as etapas para assegurar que as atividades são executadas de acordo com o processo que foi estabelecido. Normalmente abrange revisões e auditorias pela gerência e garantia da qualidade de software.

Estrutura



Key practices: exemplo

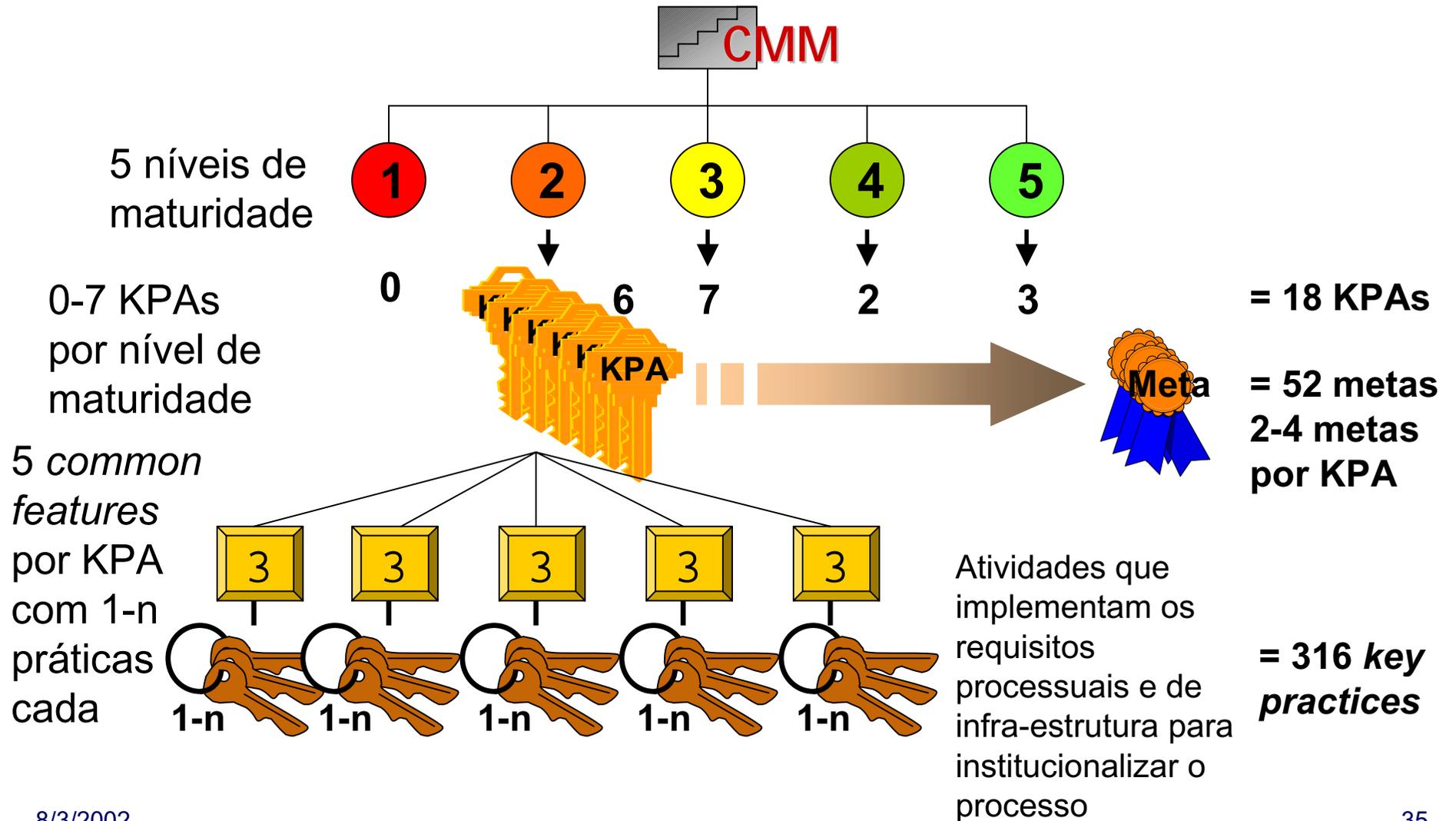
z **Nível 2**

Key process area: Acompanhamento e supervisão de projeto de software

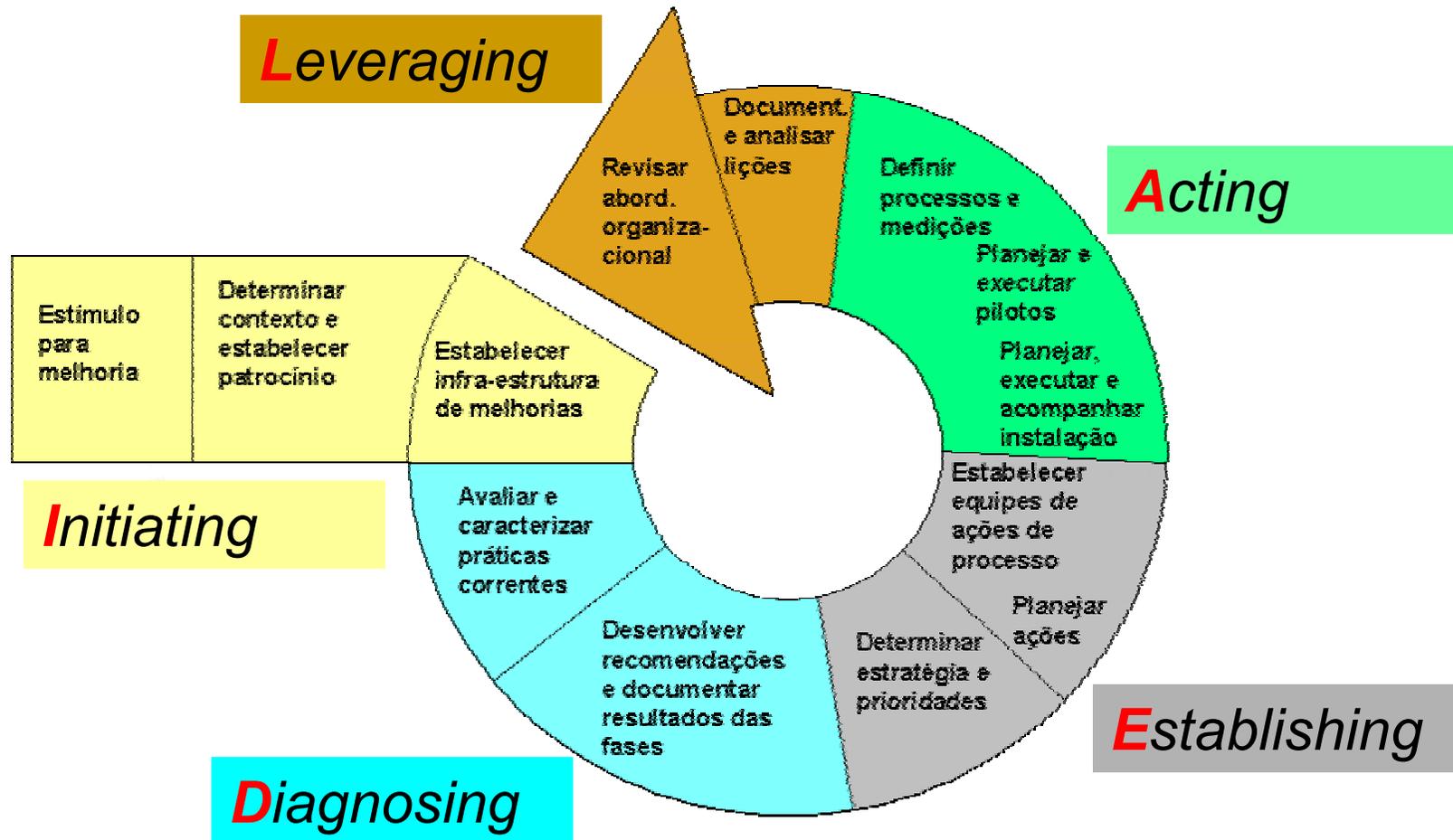
z **Common feature:** Medição e análise

Key practice: São feitas e usadas medições para determinar o status das atividades de acompanhamento e supervisão de software

Estrutura: visão geral



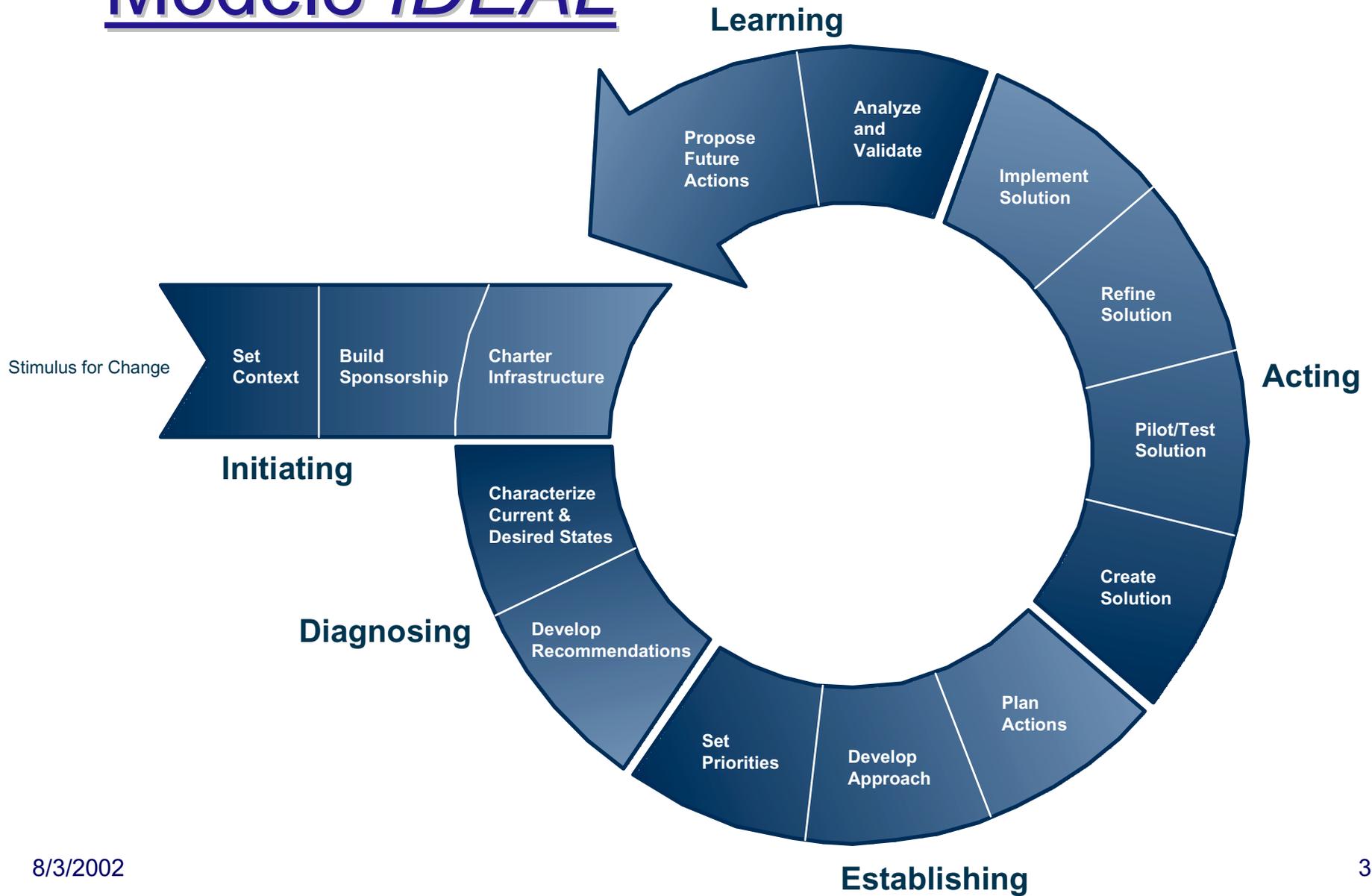
Abordagem IDEAL



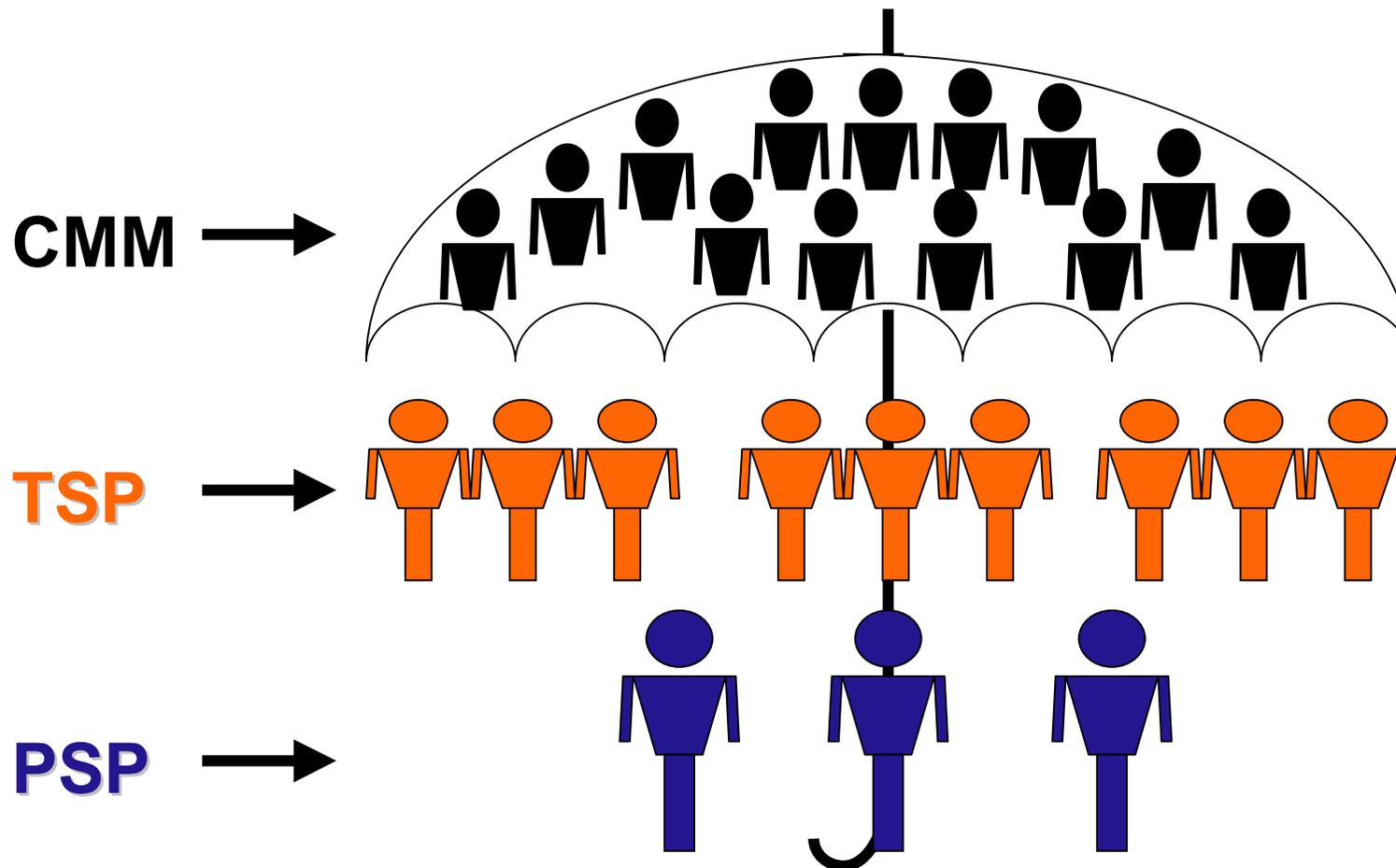
Motivos...

- z **Initiate** (Iniciar) “Uma jornada de 10.000 milhas começa com nada mais do que um simples passo.”
- z **Diagnose** (Diagnosticar) “Se você não sabe onde está, um mapa não vai ajudá-lo.”
- z **Establish** (Estabelecer) “Se você não sabe para onde está indo, qualquer caminho serve.”
- z **Act** (Agir) “Se você não confere a rota, não saberá quando estiver fora do curso.”
- z **Leverage** (Influenciar) “Se os pioneiros não relatam que o caminho é seguro, os colonizadores não vão segui-los.”

Modelo *IDEAL*



Team Software Process & Personal Software Process



Últimas mudanças

ATÉ AGORA:

CMM (versão 1.1)



EM ELABORAÇÃO:

SW-CMM (versão 2.0)

P-CMM

SA-CMM

SE-CMM

IPD-CMM

CMMI

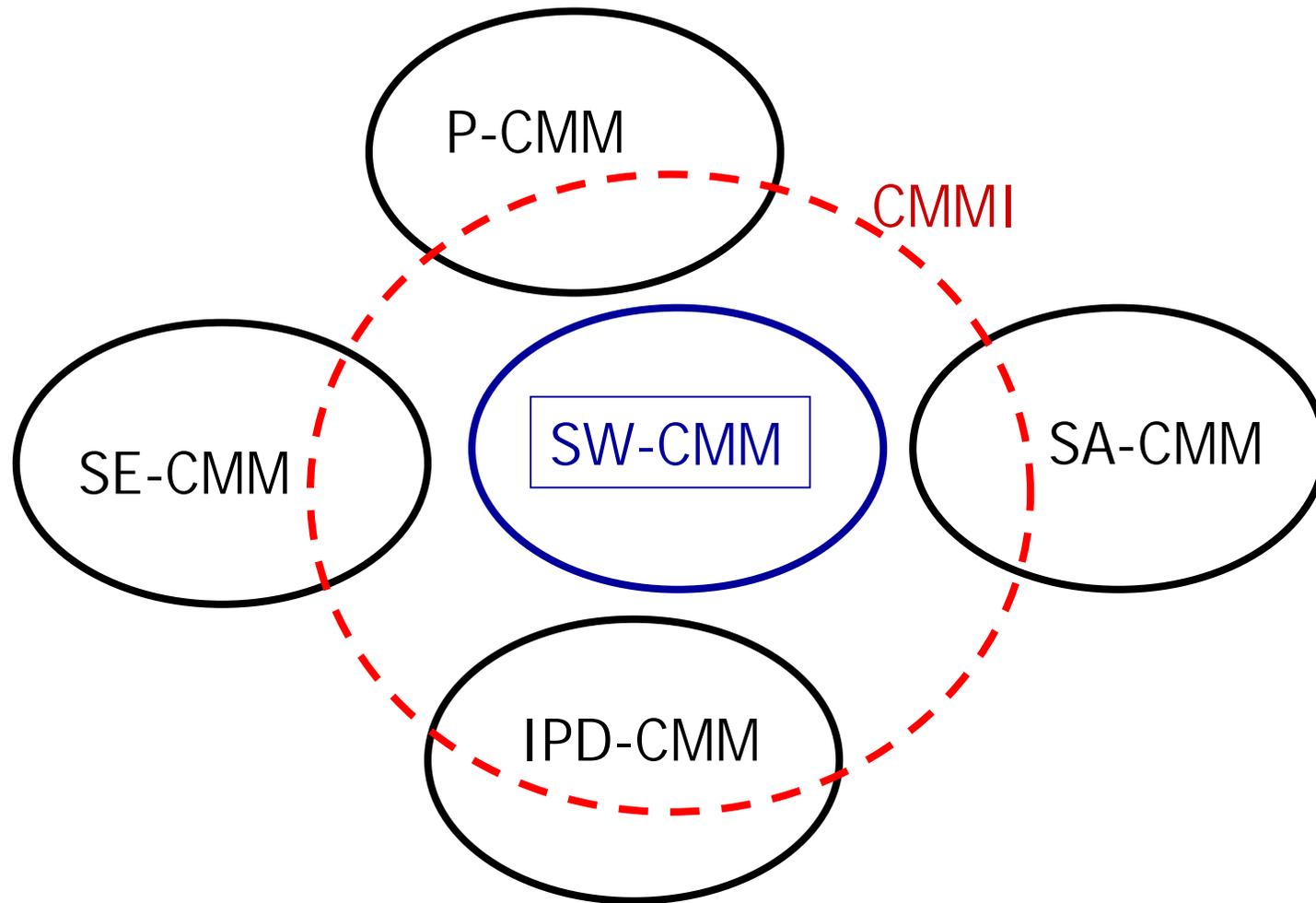
Os CMMs

- z CMMI: CMM Integration
- z SW-CMM: Capability Maturity Model for Software
- z P-CMM: People Capability Maturity Model
- z SA-CMM: Software Acquisition Capability Maturity Model

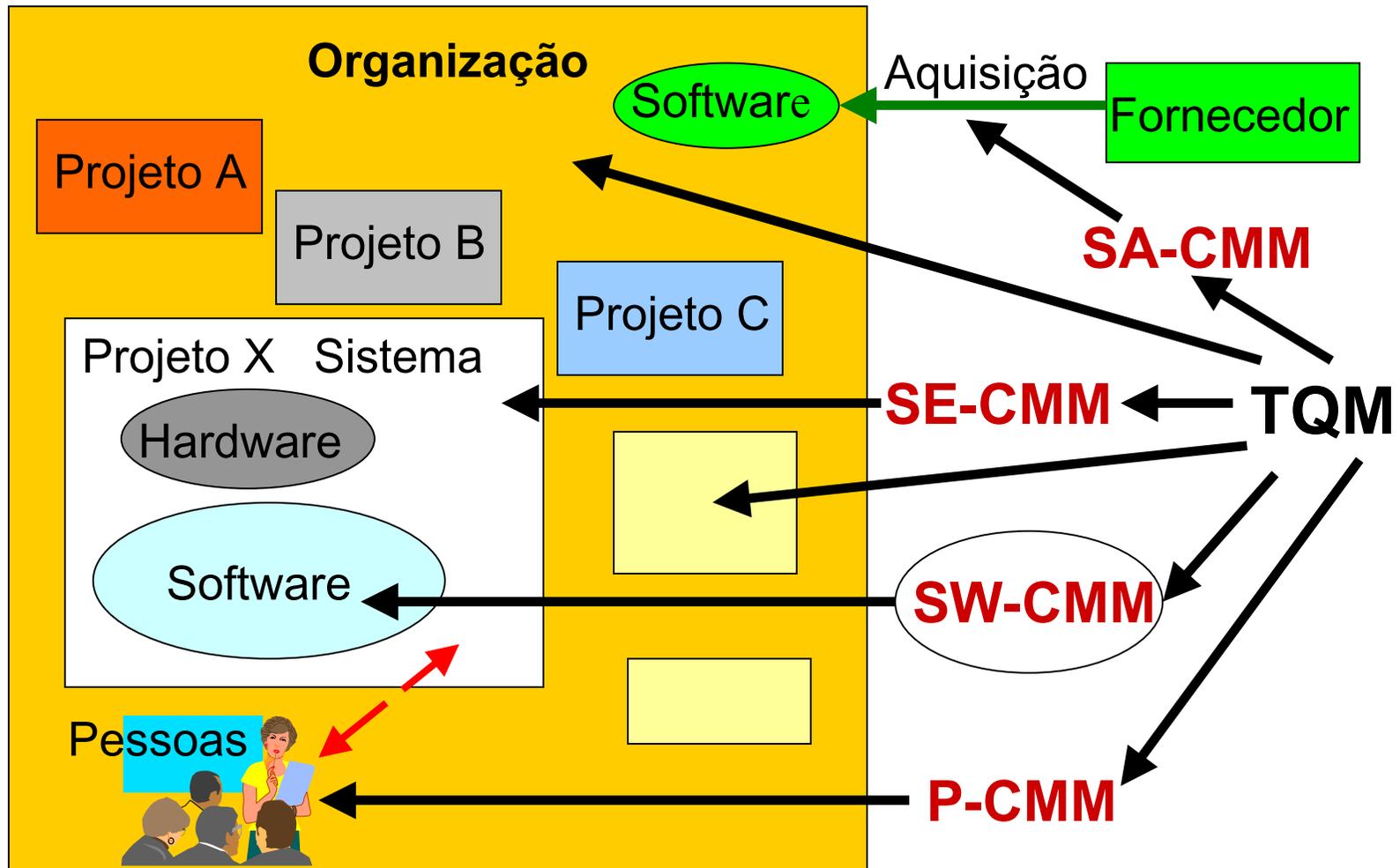
Os CMMs

- z **SE-CMM:** Systems Engineering Capability Maturity Model
- z **IPD-CMM:** Integrated Product Development Capability Maturity Model

CMMI

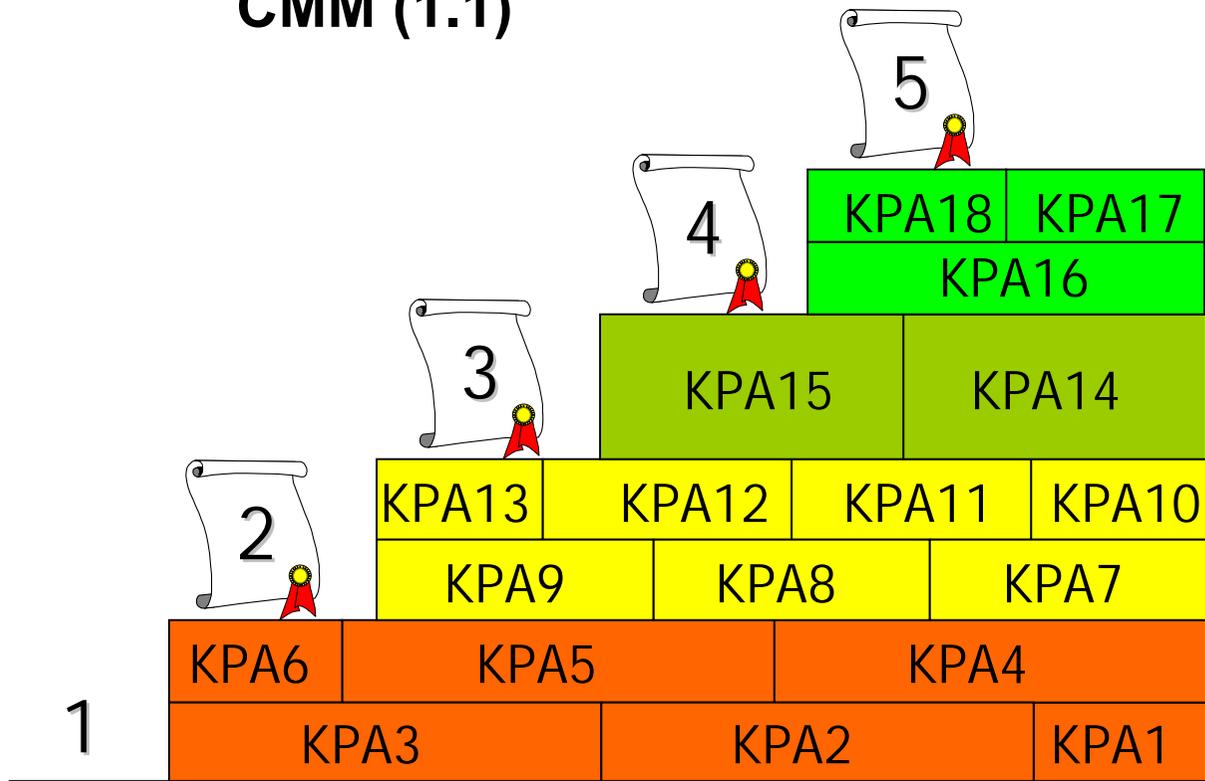


TQM e os CMMs

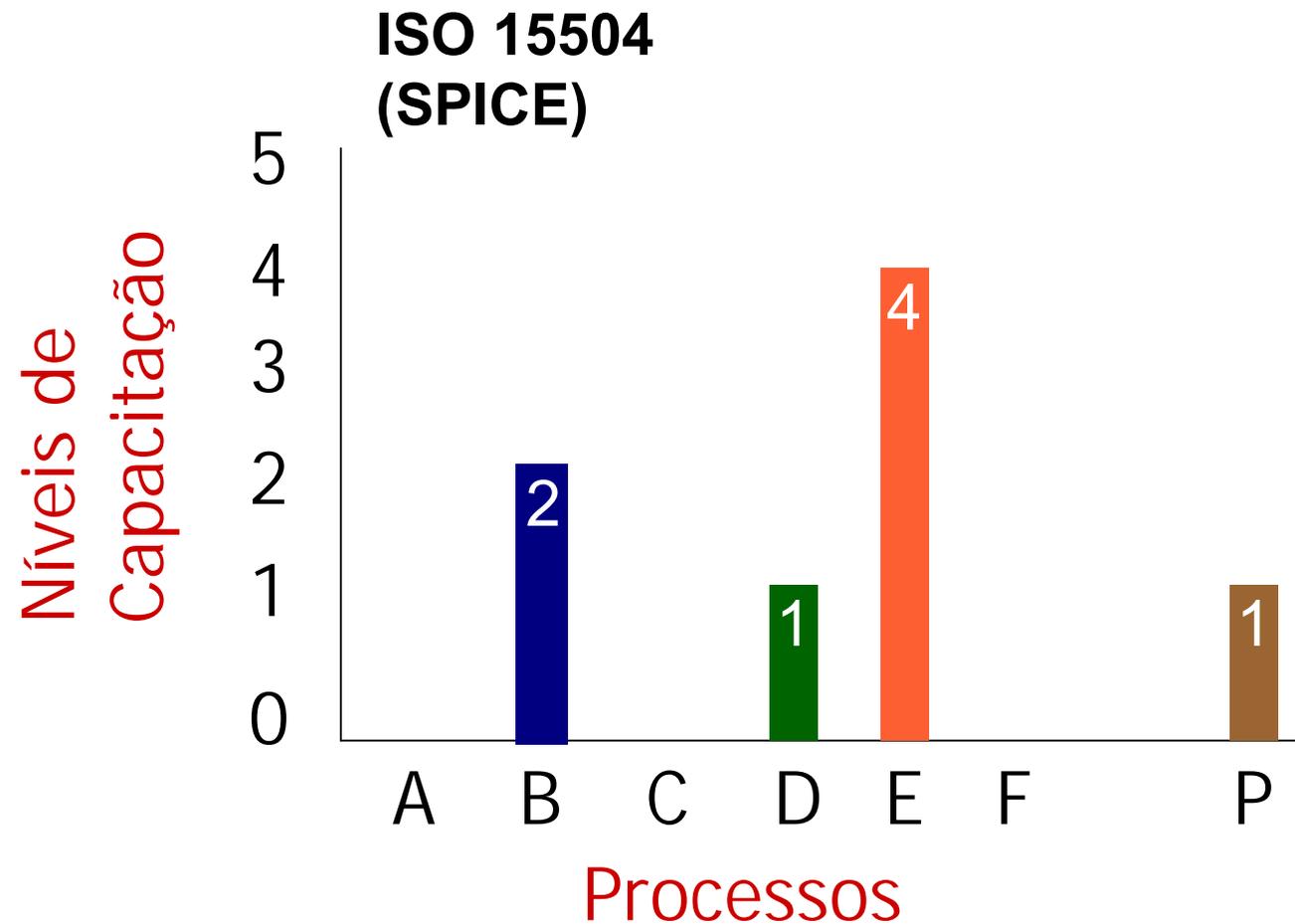


CMM x ISO/IEC 15504

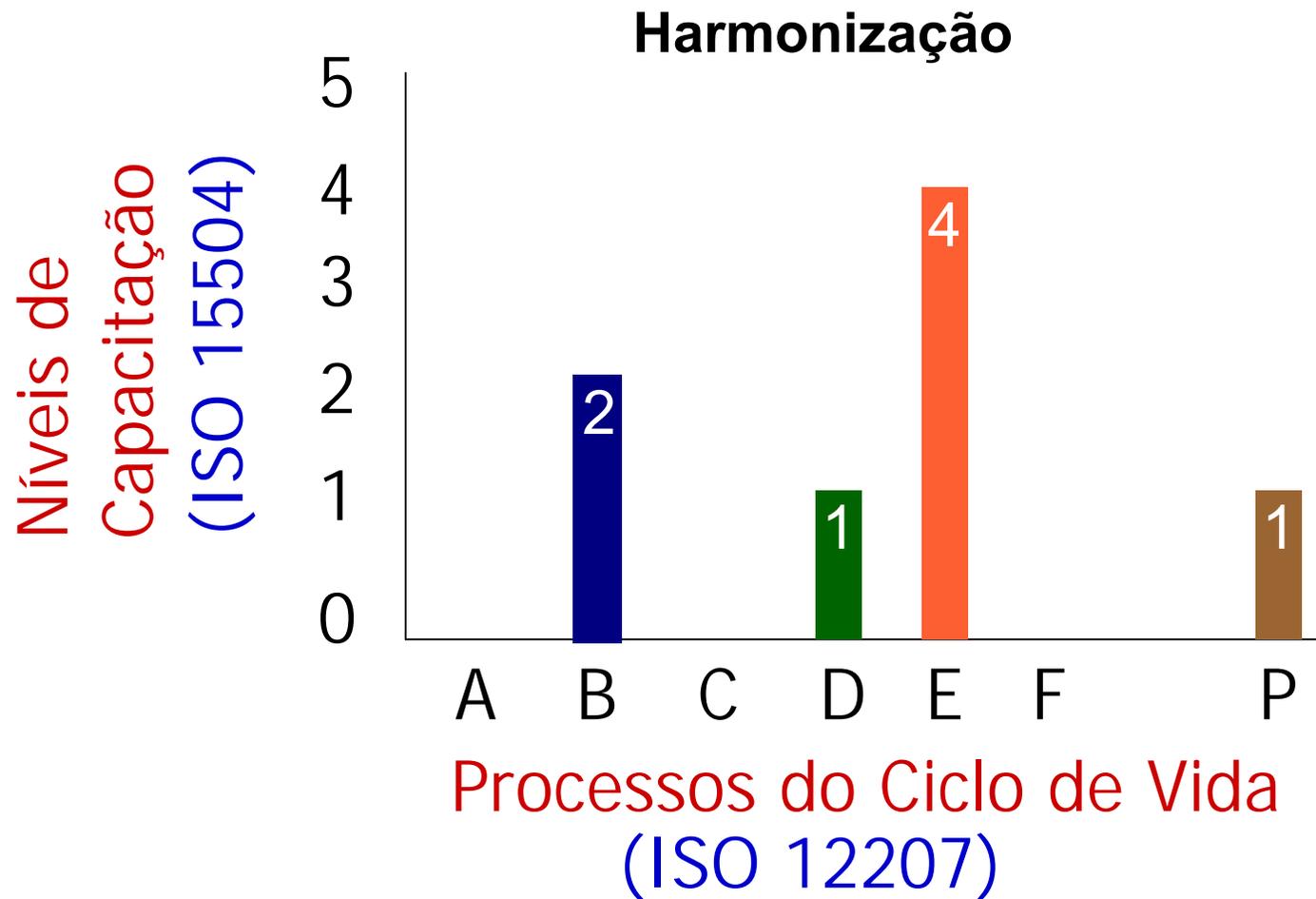
CMM (1.1)



CMM x ISO/IEC 15504



ISO/IEC 15504 x 12207



Futuro

- z SW-CMM (CMM v 2.0) compatível com a ISO 15504
- z ISO 15504 harmonizada com a ISO 12207
- z ISO 9000 versão 2000 harmonizada com a ISO 12207

Para mais informações

- z *Site* do SEI/CMU - www.sei.cmu.edu
- z *Site* de José Barreto Jr. (Ceará) - www.barreto.com.br

Bibliografia

- GOLDESON Dennis R. HERBSLEB James D.
After the appraisal: a systematic survey of process improvement its benefits, and factors that influence success. Pittsburgh: SEI/Carnegie Mellon University, 1995.
- HAYES, Will. ZUBROW, Dave. Moving ou up: data and experience doing CMM-based process improvement. Pittsburgh: SEI/Carnegie Mellon University, 1995.

Bibliografia

HERBSLEB, James et al. Benefits of CMM-based Software process improvement: initial results. Pittsburgh: SEI/Carnegie Mellon University, 1994.

NBR ISO/IEC 12207 Tecnologia de informação - Processos de ciclo de vida de software. Rio de Janeiro: ABNT, 1998.

Bibliografia

McFEELEY, Bob. IDEAL: a user's guide for software process improvement. Pittsburgh: SEI/Carnegie Mellon University, 1996.

PAULK, Mark C. et al. The capability maturity model: guidelines for improving the software process. Pittsburgh: Addison Wesley, 1997.

Bibliografia

PAULK, Mark C. et al. Capability maturity model for software, version 1.1. Pittsburgh: SEI/Carnegie Mellon University, 1993.

Introduction to the capability maturity model. Pittsburgh: SEI/Carnegie Mellon University.

Bibliografia

WINTERS, George R. Executive overview of CMM based software process improvement. SEI/Carnegie Mellon University, Pittsburgh, 1997.

WINTERS, George R. Software process improvement overview. SEI/Carnegie Mellon University, Pittsburgh, 1997.