

Engenharia de Software I

Aula 03: Capability Maturity Model (CMM) e Definição de Processo de Desenvolvimento de Software

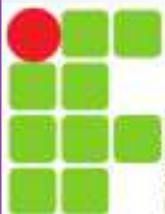
Breno Lisi Romano

<http://sites.google.com/site/blromano>

Instituto Federal de São Paulo – IFSP São João da Boa Vista

Bacharelado em Ciência da Computação – BCC (ENSC5)

Tecnologia em Sistemas para Internet – TSI (ESWI5)



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus São João da Boa Vista



Sumário

- Motivação
- *Capability Maturity Model* (CMM)
- Estudo dos Níveis de Maturidade Propostos pelo CMM
- Visibilidade da Gerência dentro dos Processos de Software em cada Nível de Maturidade
- Áreas Chave de Processo por Nível de Maturidade
- Processos de Desenvolvimento de Software



Problema

- Se você tivesse que investir 1.000.000...
 - Em que você investiria?
 - Treinamentos?
 - Máquinas poderosas?
 - Contratar funcionários?
 - Programadores?
 - Gerentes de projeto?
 - Analistas?



História

- Departamento de Defesa dos USA
 - Quase 90% dos projetos falham. Porquê?
 - Pesquisa pela Carnegie-Mellon
 - Depois de muito tempo a resposta:
 - Projetos falham por gerência inadequada!
 - » Ausência de processos que auxiliem no controle das atividades de desenvolvimento de software
 - Nascimento do *Capability Maturity Model* - CMM (atual CMM-I)
 - Níveis de maturidade de uma empresa de desenvolvimento
 - Classificação de 1 a 5



Capability Maturity Model – CMM (1)

- **História:**
 - **1984:** criação do SEI (*Software Engineering Institute*):
 - Melhorar as práticas de Engenharia de Software
 - Começou a **desenvolver a estrutura de maturidade** do processo que ajudaria as **organizações a melhorar** seus processos de software
 - Esse esforço foi iniciado em resposta a uma **solicitação de fornecer ao governo federal um método para avaliar a capacidade** de software de seus fornecedores de software
 - **1991:** Liberação da Versão 1.1



Capability Maturity Model – CMM (2)

- **Proposta:**
 - Ser baseado em **Experiência Prática** de Empresas de Software
 - Refletir o melhor do **estado** da **prática**
 - Atender as **necessidades** daqueles que realizam **melhoria** do **processo** de software e **avaliação** do processo de software
 - Ser documentado e estar disponível publicamente



Capability Maturity Model – CMM (3)

- **Empresas Imaturas:**

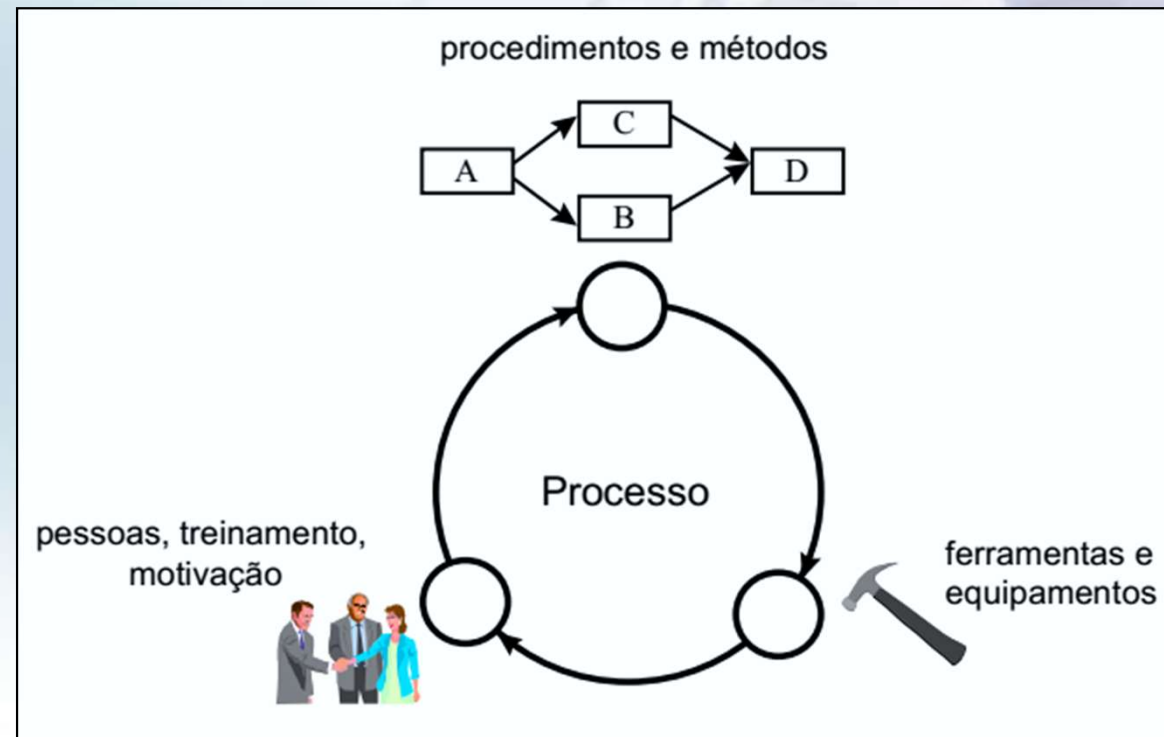
- O trabalho é feito em **regime de emergência** (apagar incêndio)
- Compromissos de **prazo e custo não são cumpridos**
- **Planejamento não** é feito com base em estimativas **realistas**
- Como os **processos não são** bem **definidos** todas as iniciativas de melhoria não se sustentam e não se perpetuam
- Quando o **projeto é pressionado por prazo**, a **qualidade e a funcionalidade são sacrificadas**
- O **sucesso** de um projeto **depende** de **especialistas** (“gurus”) para resolver grandes problemas
- **Novas tecnologias** são adotadas como **solução milagrosa**

Metáfora: Time de Várzea

- Sem coordenação em que uns correm desordenadamente e outros observam!
- Mas podem produzir bons produtos → “jogadores excepcionais”
- Porém com resultados imprevisíveis e custos fora do controle

Capability Maturity Model – CMM (4)

- Componentes de um Processo:



- Todos **componentes** do **tripé** são importantes:
 - Se ênfase em treinamento (pessoas) ou ferramentas (CASE): benefício cresce no início mas satura → Métodos não podem ser desprezados
- **Evolução** da empresa: melhor **equilíbrio** entre os **três componentes**
- **Premissa**: Bons Processos → Bons Produtos



Capability Maturity Model – CMM (5)

- **Definição:**
 - CMM fornece às organizações de software um guia de como obter controle em seus processos para desenvolver e manter software e como evoluir em direção a uma cultura de Engenharia de Software e excelência de gestão
 - Objetivo de **auxiliar** na **seleção** das **estratégias** de **melhoria**, determinando a **maturidade atual** do processo e **identificando** as **questões** mais **críticas** para a **qualidade e melhoria** do processo de software
- **CMM não é um processo de desenvolvimento de software**

CMM: Níveis de Maturidade (1)

Site para consultar empresas CMMI:

<https://cmminstitute.com/pars/>



CMM: Níveis de Maturidade (2)



CMM: Níveis de Maturidade (3)

- O processo de software é caracterizado como *ad hoc*, e ocasionalmente até mesmo caótico
- Poucos processos são definidos e o sucesso depende de esforços individuais e heróicos

INICIAL

Organizações
Caóticas



CMM: Níveis de Maturidade (4)

- Processos administrativos básicos são estabelecidos para acompanhar custo, cronograma e funcionalidade.
- A disciplina de processo está em repetir sucessos anteriores em **projetos com aplicações similares**.

REPETÍVEL

Organizações
Disciplinadas



CMM: Níveis de Maturidade (5)

DEFINIR

- Os processos, tanto para atividades administrativas quanto para de engenharia de software estão **documentados, padronizados e integrados em um processo de software padrão** para a organização
- Todos os projetos usam uma versão aprovada do processo de software padrão da organização para desenvolvimento e manutenção de software



CMM: Níveis de Maturidade (6)

GERENCIADO

Organizações
Previsíveis

- São coletadas medidas detalhadas da qualidade do processo e do produto
- Tanto o processo de software quanto os produtos são quantitativamente compreendidos e controlados



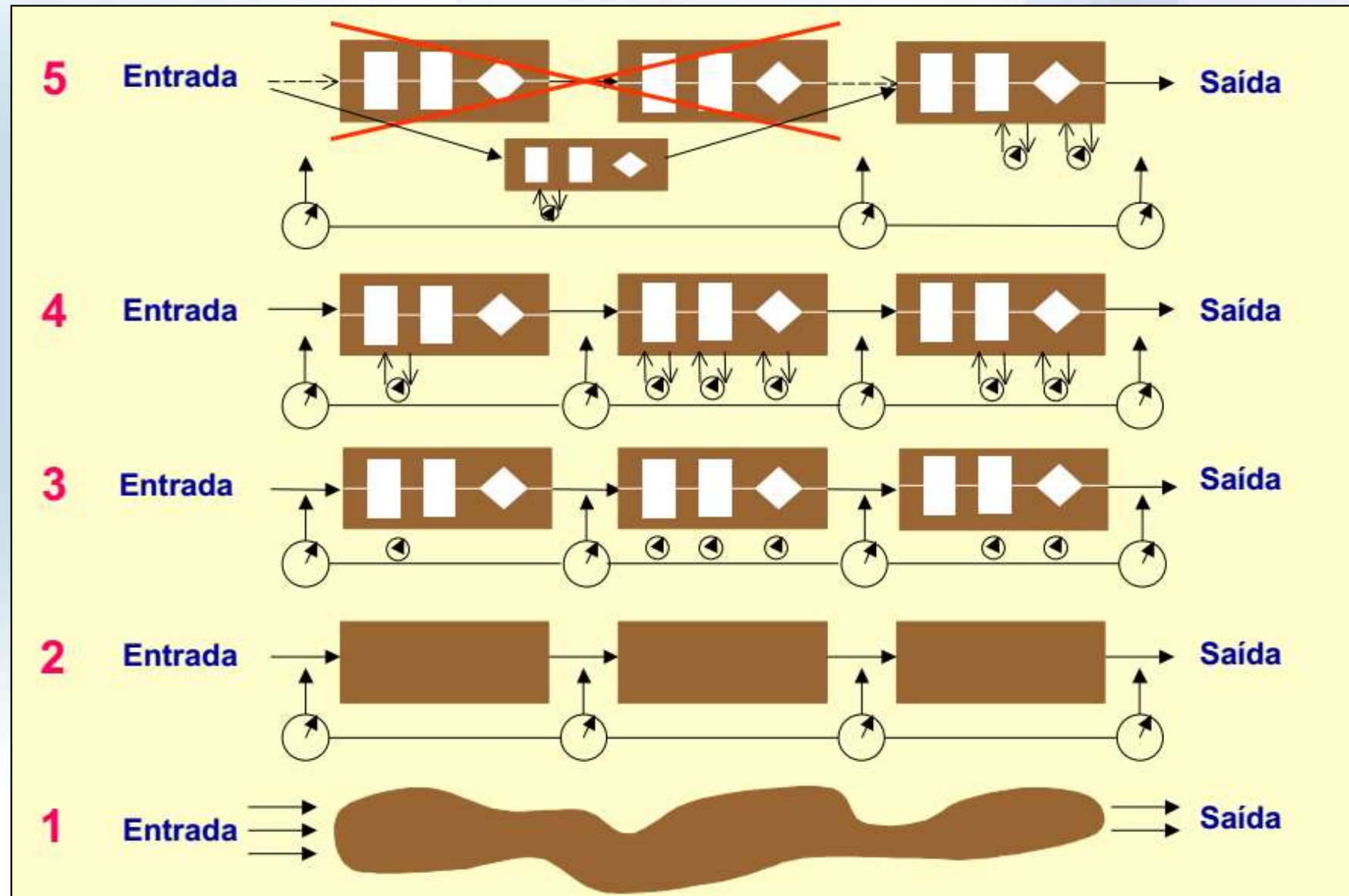
CMM: Níveis de Maturidade (7)

OTIMIZADO

Organizações com
Melhoria Contínua

- Melhorias contínuas são realizadas no processo, utilizando-se as medidas quantitativas de qualidade do processo e produto, e também aplicando-se ideias e tecnologias inovadoras

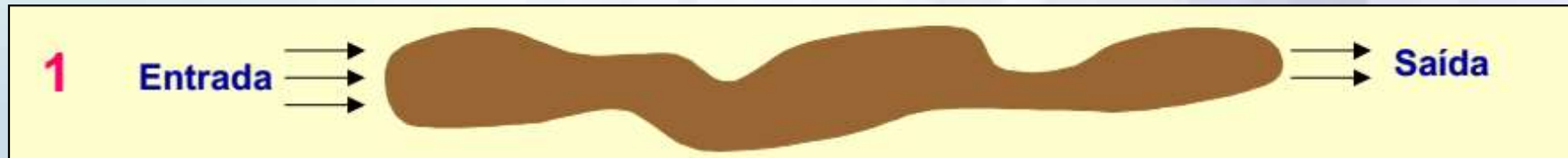
Visibilidade da Gerência Dentro dos Processos de Software em cada Nível de Maturidade (1)





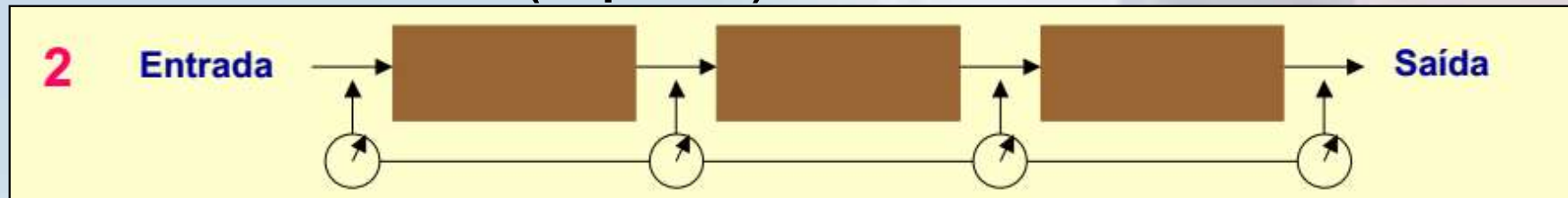
Visibilidade da Gerência Dentro dos Processos de Software em cada Nível de Maturidade (2)

- **Visibilidade Nível 1 (Inicial):**



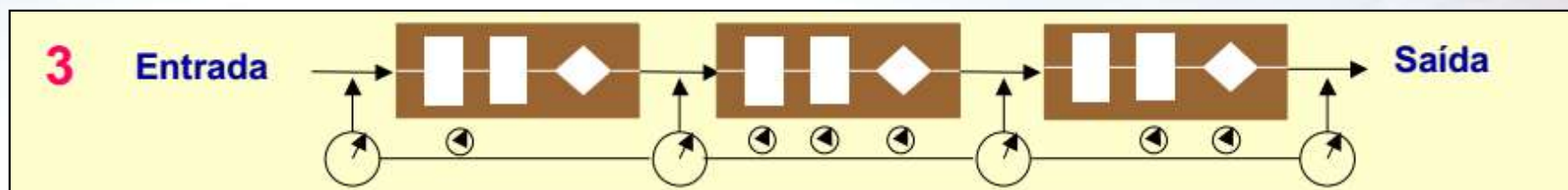
- Caixa preta
- Resultado pode acabar saindo, com prazos e custos fora do controle

- **Visibilidade Nível 2 (Repetível):**



- Pontos de verificação nas saídas de fases
- Tomada de ações corretivas

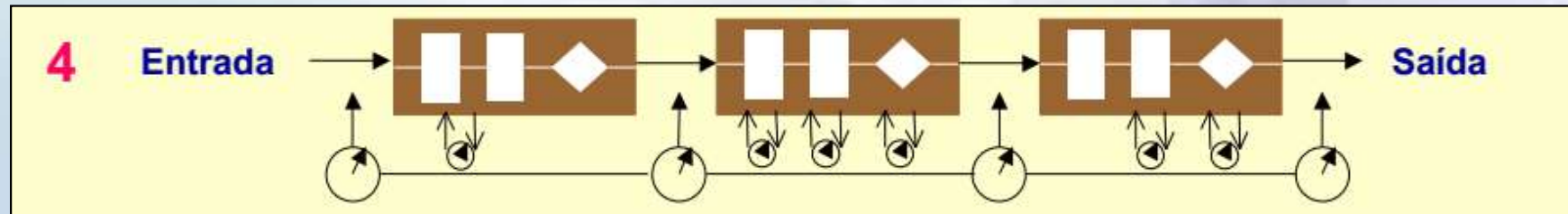
- **Visibilidade Nível 3 (Definido):**



- Cada fase passa a ter visibilidade interna
- Processos definidos

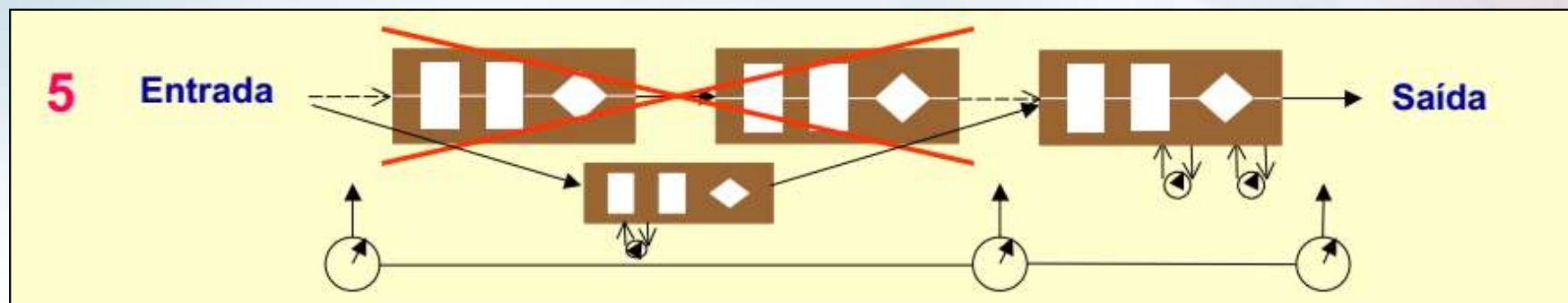
Visibilidade da Gerência Dentro dos Processos de Software em cada Nível de Maturidade (3)

- Visibilidade Nível 4 (Gerenciado):



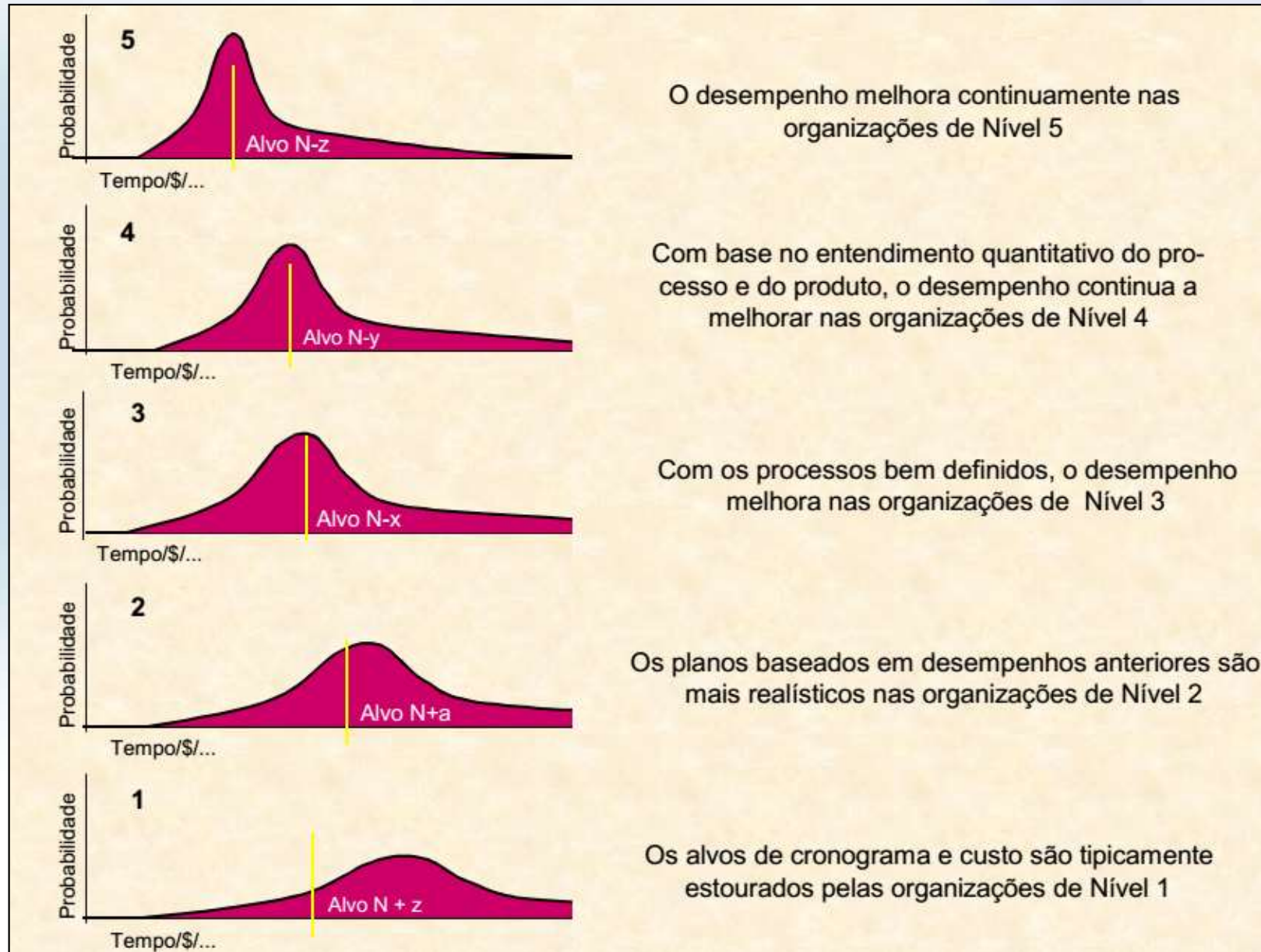
- Práticas de medida são institucionalizadas
- Pontos de verificação internos, externos
- Monitoração e controle

- Visibilidade Nível 5 (Otimizado):



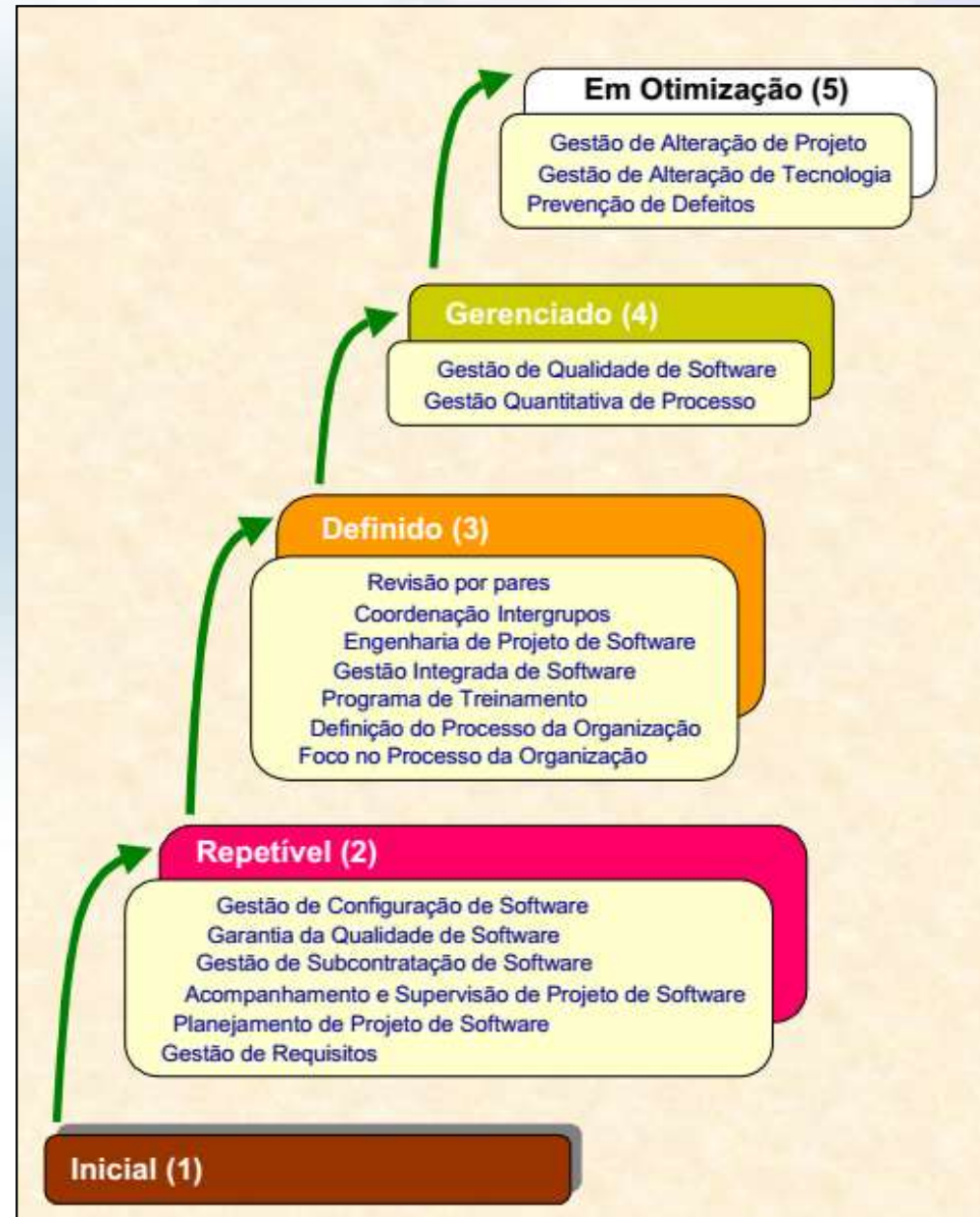
- Evolução controlada de tecnologia e processos
- Fases completas podem ser substituídas

A Capacidade do Processo como indicado pelo Nível de Maturidade



As Áreas Chave de Processo por Nível de Maturidade

As áreas-chave de processo identificam os assuntos que devem ser tratados para se obter um determinado nível de maturidade





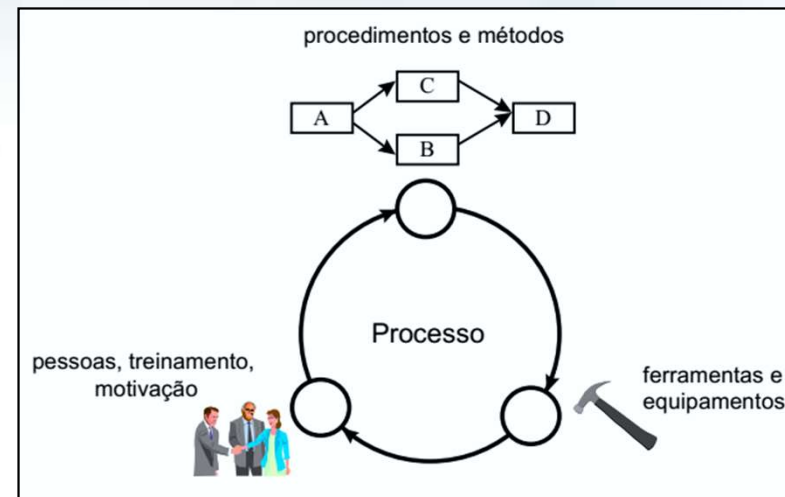
Processos de Desenvolvimento

- Fundamentais para qualidade
 - Processos ajudam mas...
 - No *silver bullet*!
 - Modelos de processos
 - Codifica-Remenda
 - Cascata
 - Evolucionário
 - Prototipagem Evolutiva
 - Espiral
 - Processo Unificado
 - Orientado à Modelos
 - Ágeis



O que é um *processo de desenvolvimento de software*?

É um **conjunto de atividades** necessárias para **transformar as necessidades** dos usuários e suas expectativas em um sistema baseado em **software**





O que é um *modelo* de processo de desenvolvimento de software? (1)

- É uma **referência**, abordagem que caracteriza e guia quanto as atividades de melhoria de um processo de desenvolvimento de software
- Possibilita a **avaliação da maturidade** do processo de desenvolvimento de software
- Referência e estabelece parâmetros de mercado quanto a **capacidade em produzir software** de qualidade



O que é um *modelo* de processo de desenvolvimento de software? (2)

Um modelo de processo é um conjunto estruturado de boas práticas que descrevem as características de **efetividade** de um processo

– Boas práticas referem-se aquelas comprovadas pela experiência

- **Eficiência** é a capacidade de obter bons produtos como **produtividade** e **desempenho**, utilizando a menor quantidade de recursos possíveis, como tempo, mão-de-obra e material.
- **Eficácia** é a capacidade de **fazer aquilo que é preciso**, que é certo para se alcançar determinado objetivo, escolhendo os melhores meios e produzir um produto adequado ao mercado.
- A **eficiência** envolve a **forma com que uma atividade é feita**, a **eficácia** se refere ao **resultado da mesma**.

Exemplo: ter a produção de um produto com eficiência, isto é, rapidamente e com baixos custos, mas que não é adequado ao contexto e à situação econômica das pessoas. Nesse caso, temos eficiência, mas não eficácia



Atividade para Sala de Aula

Faça uma análise de um dos Cases disponibilizados sobre melhorias de processo e avaliação CMM de empresas e responda:

1. Apresentar a Empresa e o Problema enfrentado;
2. Apresentar os principais pontos de melhorias obtidos
3. Quais foram as principais melhorias de processos apresentadas?
4. Quais foram os principais ganhos da empresa?
5. Quais lições aprendidas pela empresa que você considera aplicável para outras empresas?

- **Cases sobre CMMi:**

- Case 1: Allianz ([Download](#))
- Case 2: Chemtech ([Download](#))
- Case 3: Diligent ([Download](#))
- Case 4: Graham Tech ([Download](#))
- Case 5: Honeywell ([Download](#))
- Case 6: Perficient ([Download](#))
- Case 7: Unisys Brazil ([Download](#))

Engenharia de Software I

Aula 03: Capability Maturity Model (CMM) e Definição de Processo de Desenvolvimento de Software

Dúvidas?

Breno Lisi Romano

<http://sites.google.com/site/blromano>

Instituto Federal de São Paulo – IFSP São João da Boa Vista

Bacharelado em Ciência da Computação – BCC (ENSC5)

Tecnologia em Sistemas para Internet – TSI (ESWI5)



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus São João da Boa Vista