

# Unidade II – Normas e Padrões de Qualidade de Software

## **Título:** CMM e CMMI 2.0 – Parte 1

Profa. Ana Carolina Gondim Inocêncio

# Roteiro

- Histórico
- SW-CMM (*Capability Maturity Model for Software*)
- CMMI (*Capability Maturity Model Integration*) Versão 2.0

# Histórico

# Histórico

- O **SW-CMM** (*Capability Maturity Model for Software*) é um modelo de capacitação de processos de software, desenvolvido pelo **SEI** (*Software Engineering Institute*) e patrocinado pelo **Departamento de Defesa Americano** (DoD), para a **avaliação da capacidade dos fornecedores de software** deste último.

# Histórico

- Início dos trabalhos deu-se em 1986, tendo sido publicada a versão 1.0 do SW-CMM em agosto de 1991.
- Em fevereiro de 1993, foi publicada a versão 1.1.

# Histórico

- Por ser específico para a área de software, o SW-CMM **não contemplava outras áreas importantes** das organizações, tais como **Recursos Humanos e Engenharia de Sistemas**.

# Histórico

- Com o sucesso do SW-CMM, **outros modelos semelhantes** foram criados para outras áreas, tais como:
  - Gestão de Recursos Humanos (People-CMM),
  - Aquisição de Software (SA-CMM) e
  - Engenharia de Sistemas (SE-CMM).

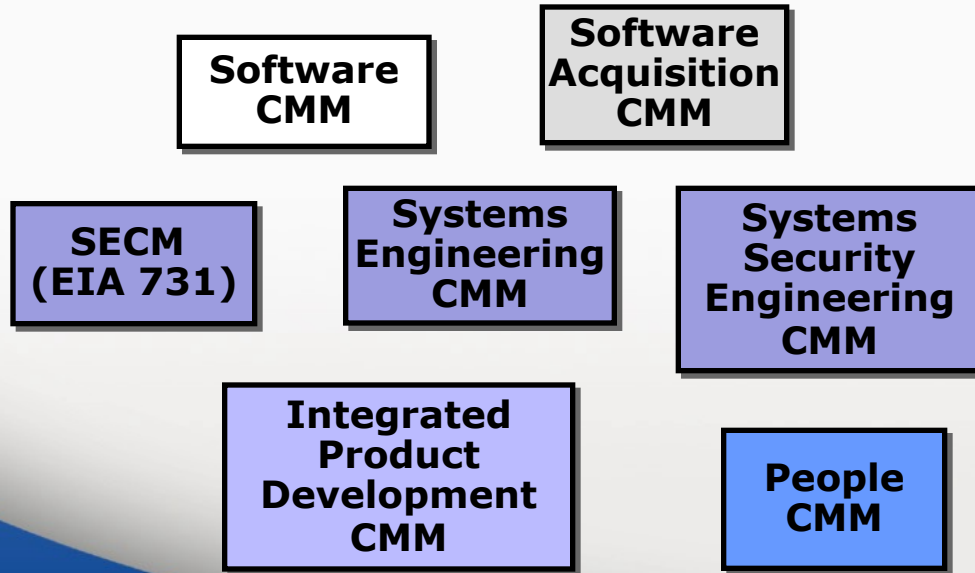
# Histórico

- Entretanto, os diversos modelos apresentavam estruturas, formatos e termos diferentes, dificultando sua aplicação conjunta.



# Histórico

- Proliferação de Modelos e Padrões em diversas áreas



- **Diferentes estruturas, formatos, termos, maneiras de medir maturidade**
- **Causa confusão, especialmente quando mais de um modelo é utilizado**
- **Difícil de integrar em um único programa de melhoria**

# Histórico

- O CMMI (*Capability Maturity Model Integration*) foi criado, então, com a finalidade de integrar os diversos modelos CMM.

# Histórico

- Em 1999, foi publicado o esboço (*draft*), versão 0.2: CMMI-SE/SW (*Capability Maturity Model -Integrated – System / Software Engineering*).
- Versões do CMMI:
  - Versão 1.0: Agosto de 2000
  - Versão 1.1: Março de 2002
  - Versão 1.2: Agosto de 2006 (*CMMI for Development*)
  - Versão 1.3: Novembro de 2011, passou a valer apenas esta versão.

# SW-CMM

# SW-CMM

- Modelo de Maturidade de Capacitação para Software
- **Objetivo Principal:** guiar organizações a conhecerem e melhorarem seus processos de software.
- Identifica **práticas para um processo de software maduro**, definindo as características de um processo de software efetivo.
- Descreve **como as práticas de engenharia de software evoluem** sob certas condições.
- **Organiza os estágios de evolução** da melhoria dos processos em **cinco níveis de maturidade**.

# SW-CMM: Estrutura

- Cada nível de maturidade, com exceção do primeiro, é composto por **áreas-chave de processo** (*Key Process Areas* – KPAs).
- Cada KPA **identifica atividades relacionadas** que, quando executadas adequadamente, atingem determinados **objetivos considerados importantes** para o aumento da capacidade do processo.
- As **KPAs são os requisitos para a obtenção de um nível no CMM**.
- As **KPAs são cumulativas**, isto é, para uma organização atingir um determinado nível de maturidade, ela deve satisfazer todas as KPAs daquele nível e de seus inferiores.

# SW-CMM: Estrutura

- Cada KPA é descrita em termos de práticas-chave (*Key Practices*).
- Uma prática-chave descreve as atividades e a infraestrutura necessárias para a efetiva implementação e institucionalização de uma KPA.
- Uma prática-chave descreve “o quê” deve ser feito, e não “como” deve ser feito.

CMM não deve ser entendido  
como sendo uma metodologia,  
pois o CMM não diz como fazer,  
mas sim o que deve  
ser feito (melhores práticas)



# SW-CMM: Estrutura

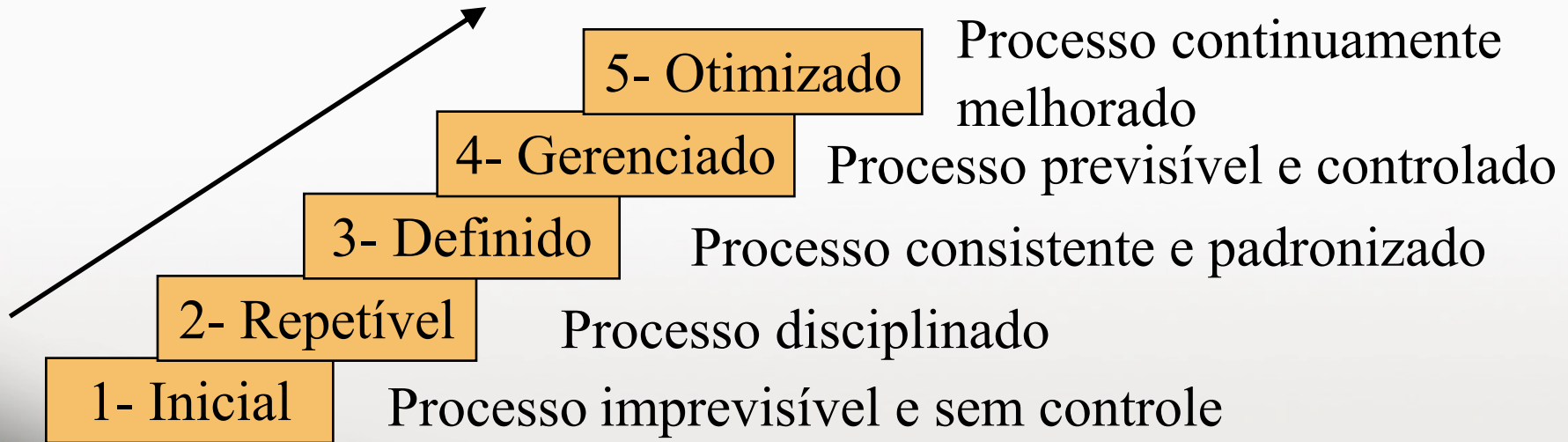
- Para cada KPA **há metas a serem alcançadas**, que caracterizam o seu conteúdo, escopo e limite.
- Metas são usadas para **determinar se a organização ou projeto efetivamente implantou a KPA** em questão.
- Em uma **avaliação de conformidade com o CMM**, o mais importante é **verificar se todas as metas da KPA foram atingidas**

# SW-CMM – Níveis de Maturidade

- Um nível de maturidade é um **patamar evolutivo bem definido**, que visa a **alcançar um processo de software maduro**.
- Os níveis são uma **forma de priorizar as ações de melhoria**, de tal forma que se **auamente a maturidade do processo de software**.
- No nível 2 por exemplo, são **focados aspectos gerenciais dos projetos**.

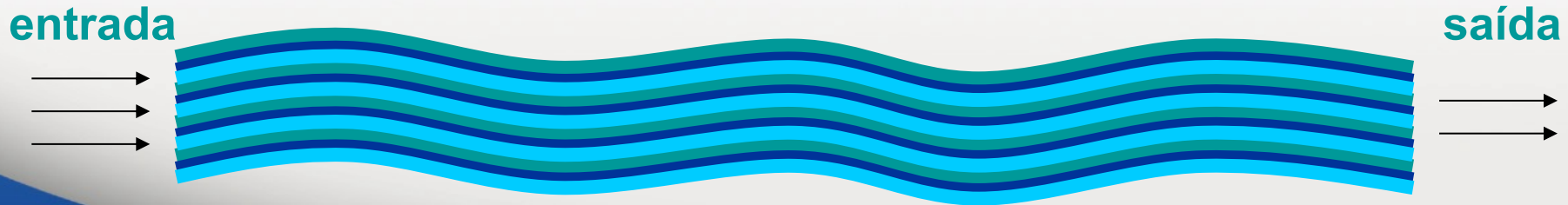
# SW-CMM – Níveis de Maturidade

- O conceito de maturidade é baseado na noção de que alguns processos proveem mais estrutura e controle do que outros.*



# SW-CMM: Nível 1 (Inicial)

- O processo de software é caracterizado como sendo **imprevisível** e **ocasionalmente caótico**.
- Poucos processos são definidos e o **sucesso depende de esforços individuais** e, muitas vezes, **heróicos**.
- O **processo de software é uma caixa preta**, de forma que **somente as entradas e os produtos finais podem ser vistos com clareza**.



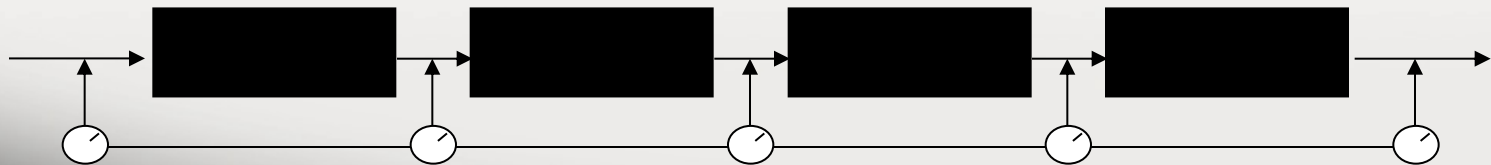
# SW-CMM: Nível 1

- Organizações no nível 1 **apresentam deficiências de planejamento e enfrentam dificuldades ao realizarem previsões.**
- Cronogramas e planos são irrealistas.
- Como não há credibilidade no planejamento, **mesmo aquilo que foi planejado não é seguido.**
- **Não há controle de requisitos e o cliente só os avalia na entrega do produto.**
- É comum **passar diretamente dos requisitos à codificação.**
- A **documentação é encarada como algo inútil.**
- São comuns **reações intransigentes à coleta de dados e ao uso de padrões, documentação e ferramentas.**

# SW-CMM: Nível 2 (Repetível)

- Processos **básicos de gerência de projetos são estabelecidos** para controle de custos, prazos e escopo.
- É **possível repetir sucessos de projetos anteriores** em aplicações similares.
- Ao invés do processo **ser uma única caixa preta**, ele passa a **ser uma sequência de caixas pretas** que asseguram a **visibilidade em determinados pontos**, os marcos do projeto.

entrada



saída

# SW-CMM: Nível 2

- Neste nível, organizações têm maior probabilidade de cumprir compromissos de requisitos, prazos e custos, mas desde que sejam semelhantes a outros realizados anteriormente.
- A organização é disciplinada, mas não está bem preparada para mudanças.
- Há preocupação com a gerência do projeto.
  - Os gerentes acompanham custos, cronogramas e funcionalidades de cada um dos projetos.
  - Porém, a gerência ainda não é pró-ativa, tomando ações normalmente quando se está diante de uma crise.

# SW-CMM: Nível 2

- Os projetos **podem ter processos diferentes**. No entanto, **existe uma política para guiar os projetos no estabelecimento** desses processos.
- Controla-se a **evolução dos requisitos**, permitindo **avaliações ao final de cada marco do projeto**, e
- Controla-se, também, **a evolução das configurações do software**.

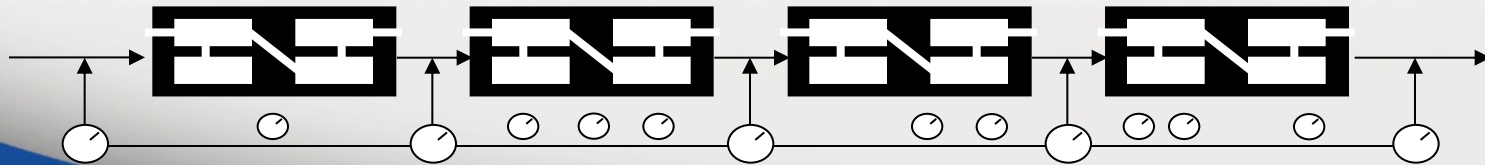


# SW-CMM: Nível 3 (Definido)

- Um processo de software, **composto por atividades de gerência e engenharia**, é documentado, padronizado e integrado em um processo de software padrão da organização.
- Todos os **projetos utilizam uma versão aprovada e adaptada do processo organizacional** para desenvolvimento e manutenção de software.
- A **organização interna das tarefas** está **definida e visível**

entrada

saída



# SW-CMM: Nível 3

- Processos utilizados são **estabelecidos e padronizados em toda a organização**.
- Os **processos pertencem à organização** e não aos projetos.
- O **Grupo de Processos** (*Software Engineering Process Group* - SEPG) **é responsável pelos processos da organização**.
- Apesar da padronização, é **possível adaptar os processos para as necessidades particulares de um projeto**.

# SW-CMM: Nível 3

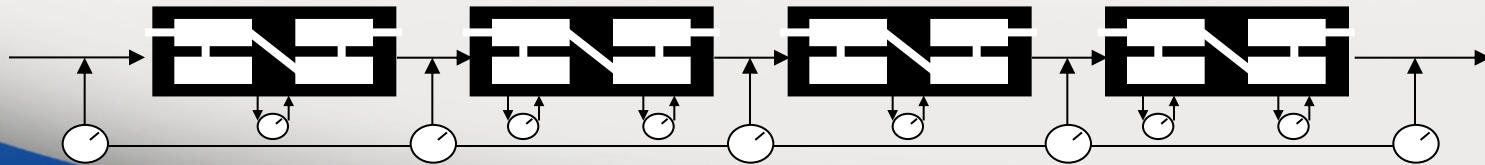
- Processos de engenharia de software são considerados ao lado dos processos gerenciais.
- Há treinamento técnico e gerencial.
- A organização consegue se manter dentro do processo mesmo em períodos de crise.
- Como o processo é bem definido, caso um desenvolvedor abandone o projeto antes de seu término, o impacto é relativamente menor que nos níveis anteriores.
- **Passagem do nível 2 para o 3:** a padronização realizada é a oportunidade de escolher as melhores práticas existentes na organização.

# SW-CMM: Nível 4 (Gerenciado)

- Métricas detalhadas do processo de software e da qualidade do produto são coletadas.
- Tanto o processo como o produto de software são quantitativamente compreendidos e controlados.

entrada

saída



# SW-CMM: Nível 4

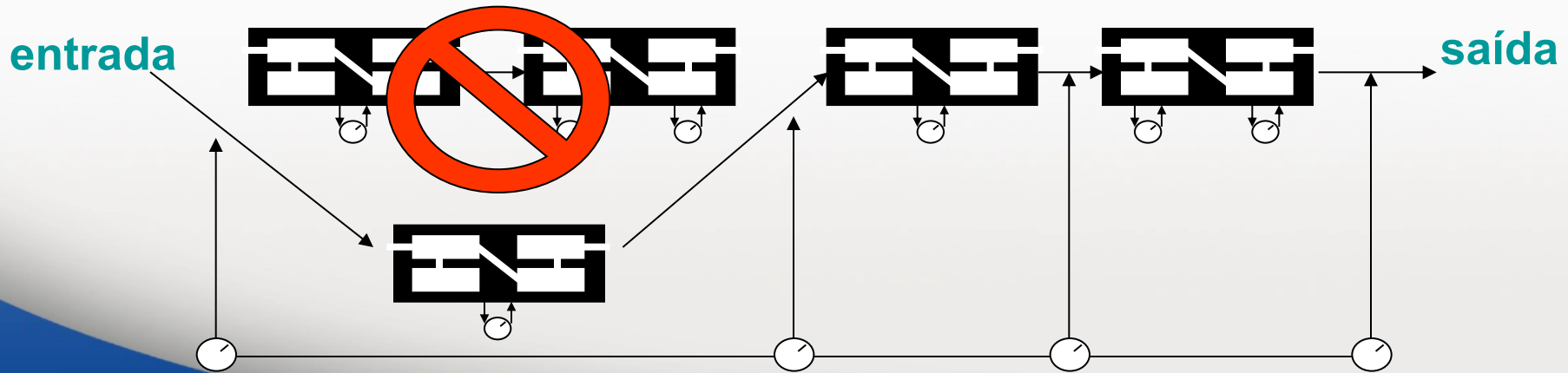
- A organização estabelece **metas quantitativas de qualidade e produtividade** para as **atividades do processo** e **para os produtos produzidos** são estabelecidas para cada projeto.
- **Medidas de qualidade e produtividade** são coletadas em **todos os projetos** como parte de um processo organizacional de medição e
- **Estabelecem uma base quantitativa** para que os gerentes possam avaliar o progresso do **desenvolvimento** e a ocorrência de problemas.

# SW-CMM: Nível 4

- Os projetos melhoram o seu controle sobre os produtos e processos e a **variância das medidas é diminuída**.
- É estabelecido o **controle estatístico de processos**.
- Uma organização no nível 4 **passa a ter uma gestão feita com bases quantitativas**.

# SW-CMM: Nível 5 (Otimizado)

- A melhoria contínua do processo é estabelecida por meio de sua avaliação quantitativa, e da implantação planejada e controlada de tecnologias e ideias inovadoras.



# SW-CMM: Nível 5

- A organização está **engajada na melhoria contínua de seus processos**, possuindo **meios para identificar fraquezas e fortalecer o processo de forma pró-ativa**, prevenindo defeitos.
- O **entendimento** do processo **ultrapassa os processos praticados**, possibilitando **compreender os efeitos de alterações potenciais** no processo.
- **Melhorias em processos e tecnologias** são planejadas e executadas como parte das atividades de rotina.
- **Mudanças mais significativas de processos ou de tecnologias são feitas a partir de análises de custo / benefício** com base em dados quantitativos cuja coleta iniciou-se no nível 4.



# IMPLEMENTANDO CMMI NÍVEL 3 DE MATURIDADE

# Obrigada!