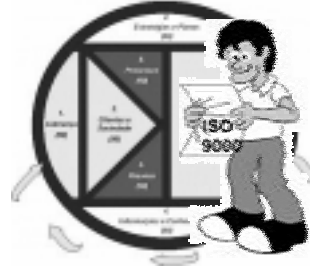




Modelo de Excelência do PNQ®



Engenharia de Software

Tema da Aula
Normas e Padrões de
Qualidade em Software – II
CMM

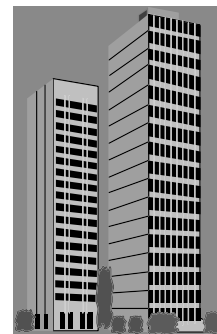
Prof. Cristiano R R Portella
portella@widesoft.com.br



CMM - Capability Maturity Model

Desenvolvido pelo *SEI*
(Instituto de Engenharia
de Software)
Carnegie Mellon
University, Pittsburgh,
PA. A partir de 1984
financiado pelo DoD.

SEI



Software Engineering Institute

Missão da SEI:

*Exercer liderança nos estágios avançados da prática
de engenharia de software para melhorar a
qualidade de sistemas que dependam de software.*

CMM - Capability Maturity Model

- ✓ **Capability Maturity Model:**
(Modelo de Maturidade da Competência).
- ✓ Maturidade da Competência : Competência em controlar o Processo de Software (desenvolvimento, gerenciamento e manutenção).
- ✓ A qualidade do processo de software pode ser analisada através do nível de maturidade do processo.
- ✓ A maturidade dos processos de software de uma organização influencia na sua capacidade de atingir metas de custo, qualidade e cronograma.

O que é o CMM

Uma estrutura que descreve os elementos chaves de um processo de software eficaz.

Um caminho de melhoramento evolucionário (cinco níveis de maturidade) para organizações de software mudarem de um processo de software imaturo (ad hoc), para um processo maduro, disciplinado.

A Evolução do CMM

- ✓ **1986** - Início do desenvolvimento de um modelo de maturidade de processo, para ajudar as organizações a melhorar seus processos de software (por solicitação do governo federal).
- ✓ **Junho/1987** - liberação de breve descrição do modelo de maturidade de processo de software.
- ✓ **Setembro/1987** - versão preliminar do questionário de maturidade.
- ✓ **1991** - 1ª versão do CMM (Versão 1.0).

A Evolução do CMM

- ✓ **1993** - Após 5 anos de experiência, o modelo de maturidade evoluiu para um modelo completamente definido, usando conhecimento adquirido das avaliações de processo de software e de extensivo retorno das indústrias e do governo.
- ✓ **Fevereiro/1993** - Versão 1.1 do CMM- Capability Maturity Model for Software.
- ✓ **1997**: Versão 2.0 do SW-CMM (draft). Disponibiliza dois modelos:
 - CMMI Contínuo: pode avaliar qualquer área de processo (como SPICE).
 - CMMI Escalonado: abordagem tradicional do CMM (5 níveis e KPA's associadas).

Premissa Básica

Premissa básica que está por baixo do trabalho da **SEI** sobre maturidade de processo:

A qualidade de um produto de software é profundamente determinada pela qualidade do processo de *desenvolvimento* e de *manutenção* usados.

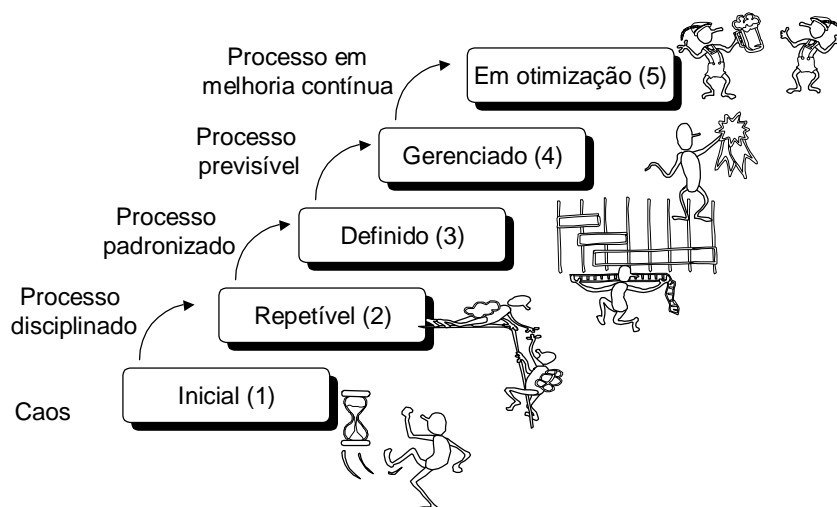
Visão Geral do Modelo CMM

- ✓ Modelo de 5 níveis que orienta uma organização em como “amadurecer” seus processos de software.
- ✓ O modelo descreve um caminho evolucionário que vai de um processo indisciplinado para um processo disciplinado.
- ✓ Sem a disciplina descrita no modelo, programas de melhoria podem mostrar-se ineficientes porque os fundamentos necessários para apoiar os melhoramentos sucessivos não foram estabelecidos.

Visão Geral do Modelo CMM

- ✓ Os 5 níveis de maturidade descrevem fundamentos sucessivos para melhoria contínua do processo e definem uma escala ordinal para medir a maturidade de processo de uma organização.
- ✓ As vantagens dos níveis de maturidade é que eles fornecem prioridades claras, as quais orientam na seleção de algumas atividades de melhoramento que serão muito úteis se implementadas imediatamente
- ✓ Isso é importante porque a maioria das organizações podem focalizar somente algumas poucas atividades de melhoramento de cada vez.

Os 5 Níveis de Maturidade

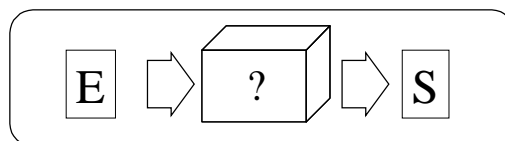




- ✓ O processo de software é caracterizado como “*ad hoc*”, e ocasionalmente até mesmo caótico.
- ✓ Poucos processos são definidos e o sucesso depende de esforços individuais e heróicos.



- ✓ Um produto de software é (normalmente) produzido através de algum processo disforme (caixa preta).
 - Requisitos fluem para dentro da caixa preta (processo de software).
 - O produto flui para fora da caixa preta e espera-se que funcione.

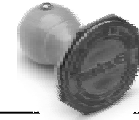




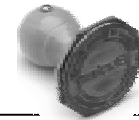
- A organização não provê um ambiente estável para o desenvolvimento e manutenção de software.
- Cronogramas e orçamentos são freqüentemente abandonados por não serem baseados em estimativas realistas.
- Numa crise para cumprir cronograma, etapas planejadas do ciclo de vida não são realizadas prejudicando a qualidade do software.



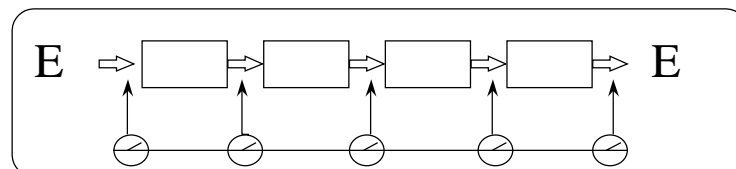
- ✓ Desempenho basicamente em função da competência e heroísmo das pessoas que fazem o trabalho.
- ✓ O processo de software é imprevisível, já que é constantemente alterado no decorrer do projeto.
- ✓ Os maiores problemas com os quais se defrontam as organizações de software são gerenciais e não técnicos.

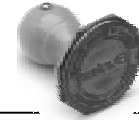


- ✓ Enfoque gerencial: estabelece processos para Gerenciamento de Projetos.
- ✓ Processos são diferentes entre projetos.
- ✓ Organização define Políticas para os projetos.
- ✓ Planos baseados em dados históricos (mais realistas).
- ✓ Processos administrativos básicos são estabelecidos para acompanhar custo, cronograma e funcionalidade.
- ✓ A disciplina de processo está em repetir sucessos anteriores em projetos com aplicações similares.

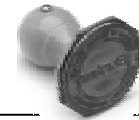


- ✓ Está em vigor um Sistema de Gerenciamento de Projeto
- ✓ O Processo de construção de software é uma série de caixas pretas com pontos de verificação definidos





- ✓ Caracterizado pela existência de um processo efetivo de planejamento e gerenciamento do projeto de software onde os controles sobre os procedimentos, compromissos e atividades são bem fundamentados
- ✓ Os processos de planejamento e gerenciamento do projeto de software devem ser praticados na organização, documentados, treinados e controlados
- ✓ Neste nível ainda não há preocupação com o processo de engenharia de software



- O planejamento e gerenciamento de novos projetos são baseados na experiência obtida com projetos similares, que tenham obtido sucesso no passado.
- Um fator relevante para a organização nesse nível é a dependência das experiências anteriores.
- O desenvolvimento de novos tipos de produtos pode causar um desequilíbrio no projeto, nas estimativas de custos e nos cronogramas.



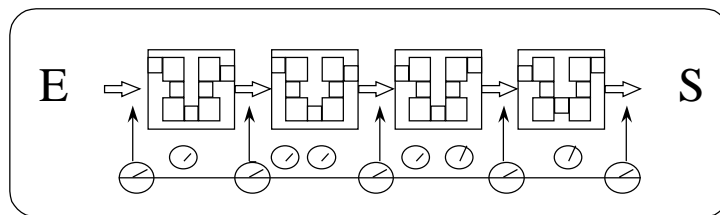
- ✓ Processo bem definido.
- ✓ Enfoque no processo de desenvolvimento:
 - Processos da engenharia de produto
 - Processos da gerência de software.
- ✓ Programa de treinamento para toda a Organização.
- ✓ Processo de Software Padrão da Organização.



- ✓ Os processos de software, tanto para atividades administrativas quanto para de engenharia 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.



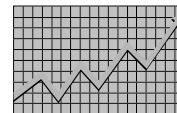
- ✓ Desenvolvimento de software de acordo com um processo bem definido.
- Funções e responsabilidades no processo são bem entendidas.
- A produção do produto de software é visível através do processo de software.



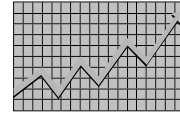
- Caracterizado principalmente pela existência de um processo de engenharia de software bem definido, documentado e padrão para a empresa.
- As saídas de uma atividade fluem naturalmente para as entradas da próxima atividade.
- Cada projeto de software utiliza o processo padrão da organização como base para implementar seu próprio processo.



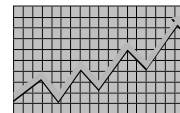
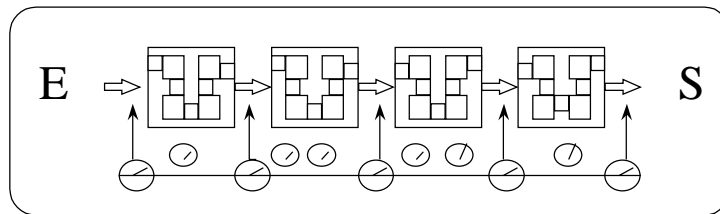
- Existe um grupo para processos de software (SEPG) responsável por facilitar atividades de definição e melhoria de processos.
- Existe um programa de treinamento que assegura que todos tenham o conhecimento e a capacidade requerida para desenvolver suas tarefas, utilizando as ferramentas e os métodos disponíveis.
- Existem políticas que dêem poderes às pessoas para realizarem o trabalho.



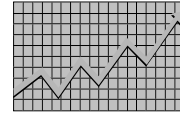
- ✓ Estabelecimento de práticas quantitativas
- ✓ Melhoria contínua
- ✓ São coletadas medidas detalhadas da qualidade do processo e do produto.
- ✓ Tanto o processo de software quanto os produtos são quantitativamente compreendidos e controlados.



- ✓ Produto e processo são gerenciados quantitativamente.
- A gerência tem bases objetivas para tomada de decisão.
- A gerência é capaz de prever o desempenho dentro de limites quantificados.



- Caracterizado pela existência de processos de software passíveis de medida.
- A produtividade e a qualidade são medidas em todas as etapas do processo de software e para todos os projetos da organização.
- O controle sobre produtos e processos de todos os projetos são adquiridos através da diminuição da variação do seu desempenho para dentro de limites quantitativos aceitáveis.



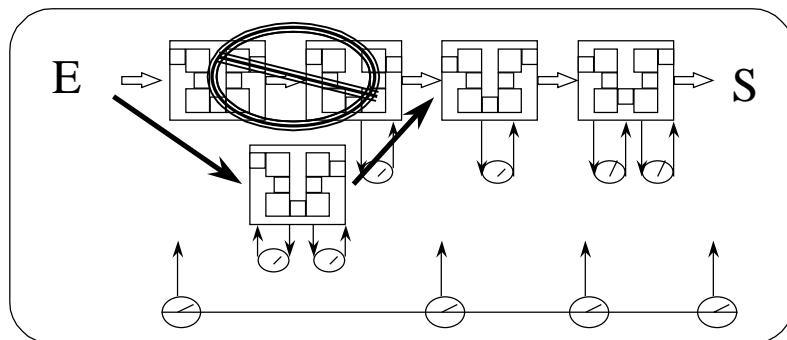
- A organização começa a aplicar métricas de controle de qualidade para aumentar a qualidade e a produtividade do software entregue aos clientes.
- À medida que a organização adquire mais conhecimento sobre o produto, tem a oportunidade de remover várias fontes de comprometimento da qualidade final.
- Isto proporciona a oportunidade de colocar o produto sob um controle estatístico de qualidade.



- ✓ Melhoria contínua visando a otimização do processo.
- ✓ Contínua melhoria de processo é possível por retornos quantitativos dos processos e das idéias e tecnologias inovadoras.



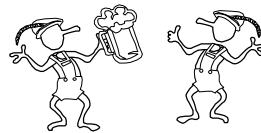
- ✓ Foco na melhoria contínua do processo.
- Mudança disciplinada é um meio de vida.



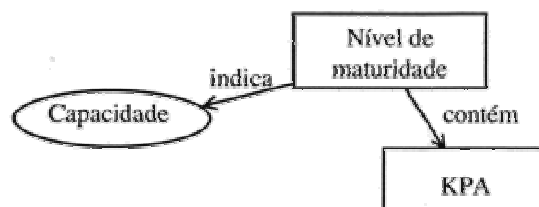
- Caracterizado pela existência de processos de software com contínua melhoria.
- Os processos de software são avaliados para prevenir tipos de defeitos conhecidos devido à recorrência, e as lições aprendidas são disseminadas para outros projetos.
- Tecnologias que proporcionem mais retorno para processos específicos, utilizados pela organização, são selecionadas para serem introduzidas de maneira gerenciável na organização.



- Apesar de o processo ser maduro, ele é alvo de contínuas melhorias.
- Os grupos de projetistas analisam o rendimento do projeto para determinar as causas dos defeitos.
- Nesse nível foi atingido um ambiente de excelência em engenharia de software.



Cada nível de maturidade (5) indica uma capacidade do processo e é composto por várias **Áreas-Chaves** de processo ou **KPA's** (Key Process Area).

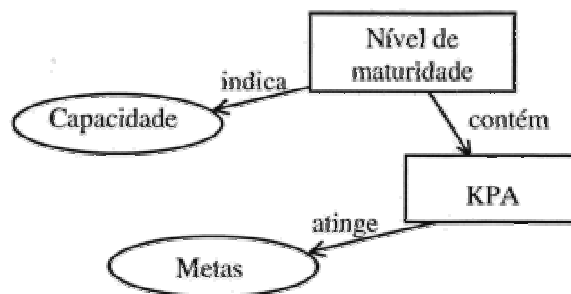


Estrutura do CMM AS 18 KPA's

Nível 5 - OTIMIZADO	<ul style="list-style-type: none"> Prevenção de defeitos Gerenciamento de mudança de tecnologia Gerenciamento de mudança de processo 	3
Nível 4 - GERENCIADO	<ul style="list-style-type: none"> Gerenciamento quantitativo do processo Gerenciamento da qualidade do software 	2
Nível 3 - DEFINIDO	<ul style="list-style-type: none"> Foco no processo da organização Definição do processo da organização Programa de treinamento Gerenciamento de software integrado Engenharia do produto de software Coordenação intergrupo Revisões (peer reviews) 	7
Nível 2 - REPETÍVEL	<ul style="list-style-type: none"> Gerenciamento de requisitos Planejamento de projeto de software Acompanhamento de projeto de software Gerenciamento de subcontrato de software Garantia da qualidade de software Gerenciamento da configuração de software 	6
Nível 1 - INICIAL	<ul style="list-style-type: none"> Sem KPA's - caos 	

Estrutura do CMM

Cada KPA permite alcançar um conjunto de **Metas** ou **Objetivos**, cuja finalidade é determinar se a organização atingiu um determinado nível de maturidade.



Estrutura do CMM

Cada KPA é organizada em um conjunto de **Práticas Base** (ou Práticas-Chave).

- As práticas chave descrevem as atividades e a infraestrutura que mais contribuem para a efetiva implementação e institucionalização das áreas chave de processo.
- As práticas chave descrevem “o que” é para ser feito e não “como” o processo deve ser implementado.
- Existem **316** práticas chave no CMM.

Estrutura do CMM As Práticas Bases (ou Práticas-Chave)

- | | |
|---|---|
| <ul style="list-style-type: none"> • Prevenção de defeitos • Gerenciamento de mudança de tecnologia • Gerenciamento de mudança de processo • Gerenciamento quantitativo do processo • Gerenciamento da qualidade do software • Foco no processo da organização • Definição do processo da organização • Programa de treinamento • Gerenciamento de software integrado • Engenharia do produto de software • Coordenação intergrupo • Revisões (peer reviews) • Gerenciamento de requisitos • Planejamento de projeto de software • Acompanhamento de projeto de software • Gerenciamento de subcontrato de software • Garantia da qualidade de software • Gerenciamento da configuração de software • Sem KPA's - caos | <ul style="list-style-type: none"> • 8 Práticas Chave • 8 Práticas Chave • 10 Práticas Chave • 7 Práticas Chave • 5 Práticas Chave • 7 Práticas Chave • 6 Práticas Chave • 6 Práticas Chave • 11 Práticas Chave • 10 Práticas Chave • 7 Práticas Chave • 3 Práticas Chave • 3 Práticas Chave • 15 Práticas Chave • 13 Práticas Chave • 13 Práticas Chave • 8 Práticas Chave • 10 Práticas Chave <p>TOTAL = 316 Práticas Chave</p> |
|---|---|

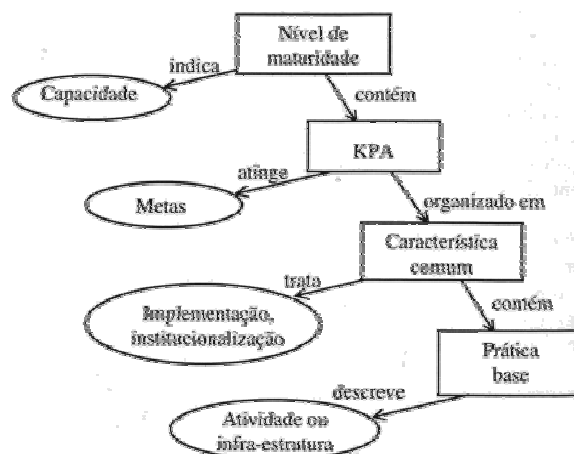
Estrutura do CMM

As Práticas Bases são organizadas, por conveniência, em 5 tipos de **Características Comuns**:

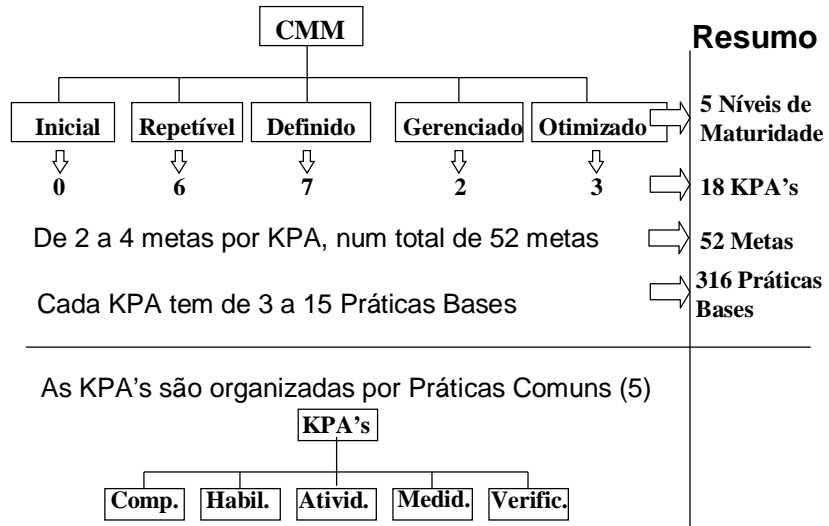
- Comprometimento para a Realização;
- Condições para Realizar;
- Atividades Realizadas;
- Medições e Análises; e
- Verificação da Implementação.

Estrutura do CMM

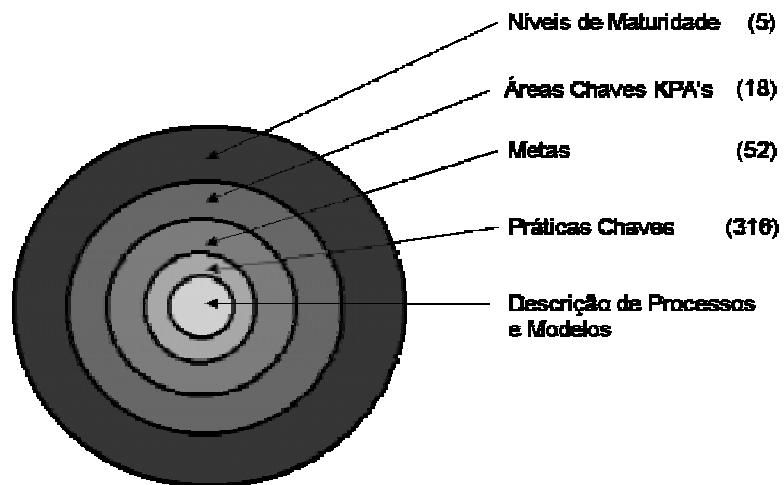
- Estrutura completa do CMM.



Estrutura do CMM

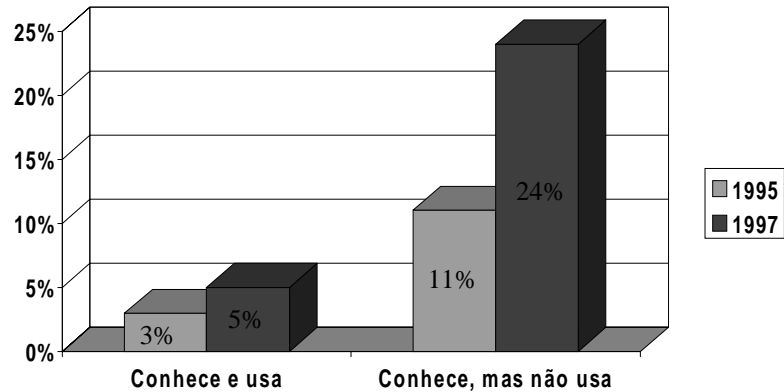


Estrutura do CMM



Evolução do CMM - Brasil

- Amostra: 445 empresas em 1995, 589 em 1997
- NEC, Xerox, Credicard, Citibank, Siemens, Motorola etc.

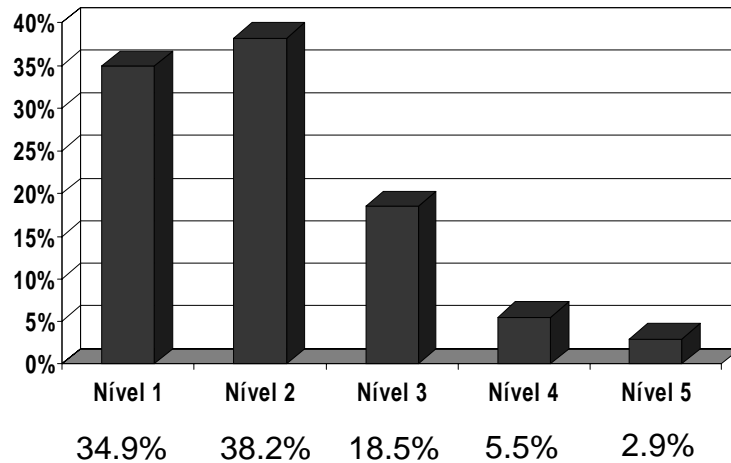


Evolução do CMM - Mundo

Caracterização da amostra:

- ✓ 901 organizações
- ✓ 276 empresas
- ✓ 4174 projetos
- ✓ Pesquisa realizada pela CMU (*Carnegie Mellon University*), no período de 1996-2000, Brasil e mundo

Evolução do CMM - Mundo



Cuidados Não omitir nenhum nível

- Processos dos níveis mais altos de maturidade podem ser realizados até mesmo por organizações do nível 1 (provavelmente de maneira ineficaz).
- Competência em processos é construída em estágios, uma vez que alguns processos não são eficazes enquanto outros não estão estáveis.
- Cada nível oferece um fundamento necessário para melhorias a serem implementadas no nível seguinte.
- Sem a disciplina de gerenciamento o processo de engenharia é sacrificado.
- Medidas detalhadas são inconsistentes sem um processo definido.

Cuidados Analisar o Contexto Organizacional

- ✓ O CMM é aplicável a diferentes tipos de organizações.
- ✓ Cada organização tem uma resposta diferente às mudanças.
- ✓ Cada organização deve interpretar níveis de excelência no contexto do ambiente de negócios da organização. O CMM funciona de forma ideal quando as práticas são interpretadas de uma maneira que faça sentido para a organização.

- ✓ Lembre-se: **Qualidade é COMPORTAMENTAL !**

Afinal, qual modelo usar ?

- Usar o modelo mais adequado à sua instituição.
Analisar:
- ✓ Os modelos em relação:
 - Adequação (tipo de negócio, tipos de aplicações)
 - Estabilidade e estado da prática (qtos. usam)
 - Suporte (treinamento, consultoria, livros)
 - Custo (tamanho da infra-estrutura)
 - ✓ O produto e a equipe de software:
 - Tamanho e complexidade dos produtos
 - Tamanho e nível da equipe

Afinal, qual modelo usar ?

- ✓ Todos os modelos guardam grande semelhança entre si e têm áreas de intersecção entre eles.
- ✓ Todos eles têm um custo para a organização. Sua viabilização depende de um estudo em relação ao “Custo da Não-Qualidade” (horas de manutenção, horas paradas, custo das falhas em si, imagem da empresa perante o cliente e o mercado etc).

<http://softeng.cs.mcgill.ca/>

www.sei.cmu.edu/sema/pub_ml.html

www.sei.cmu.edu/cmm/docs/biblio.html