



Universidade Federal do Vale do São Francisco

Padrões de Qualidade de Software

e

Métricas de Software

Engenharia de Software I

Aula 3 e 4

Ricardo Argenton Ramos

Agenda da Aula

- Introdução (Qualidade de Software)
- Padrões de Qualidade de Software
- Alguns Padrões de Qualidade de Software
 - CMM
 - ISO/IEC 15504 (SPICE)
- Métricas de Software
- Processo de Medição
- Métricas de Produto
- Pontos Chaves
- Exercícios

Objetivos da Aula

- Compreender os conceitos básicos de qualidade de software. Conhecer os padrões de Qualidade CMM, ISO/IEC 15504 (SPICE).
- Compreender o que são métricas de software e as diferenças entre métricas de predição e métricas de controle.

Introdução

- O **principal objetivo** da **Engenharia de Software** (ES) é ajudar a **produzir software de qualidade**;
- **Empresas** que desenvolvem software de qualidade são mais **competitivas**;
- Empresas que tem **qualidade em seus processos** podem, em geral, oferecer um melhor serviço a um **preço mais competitivo**.

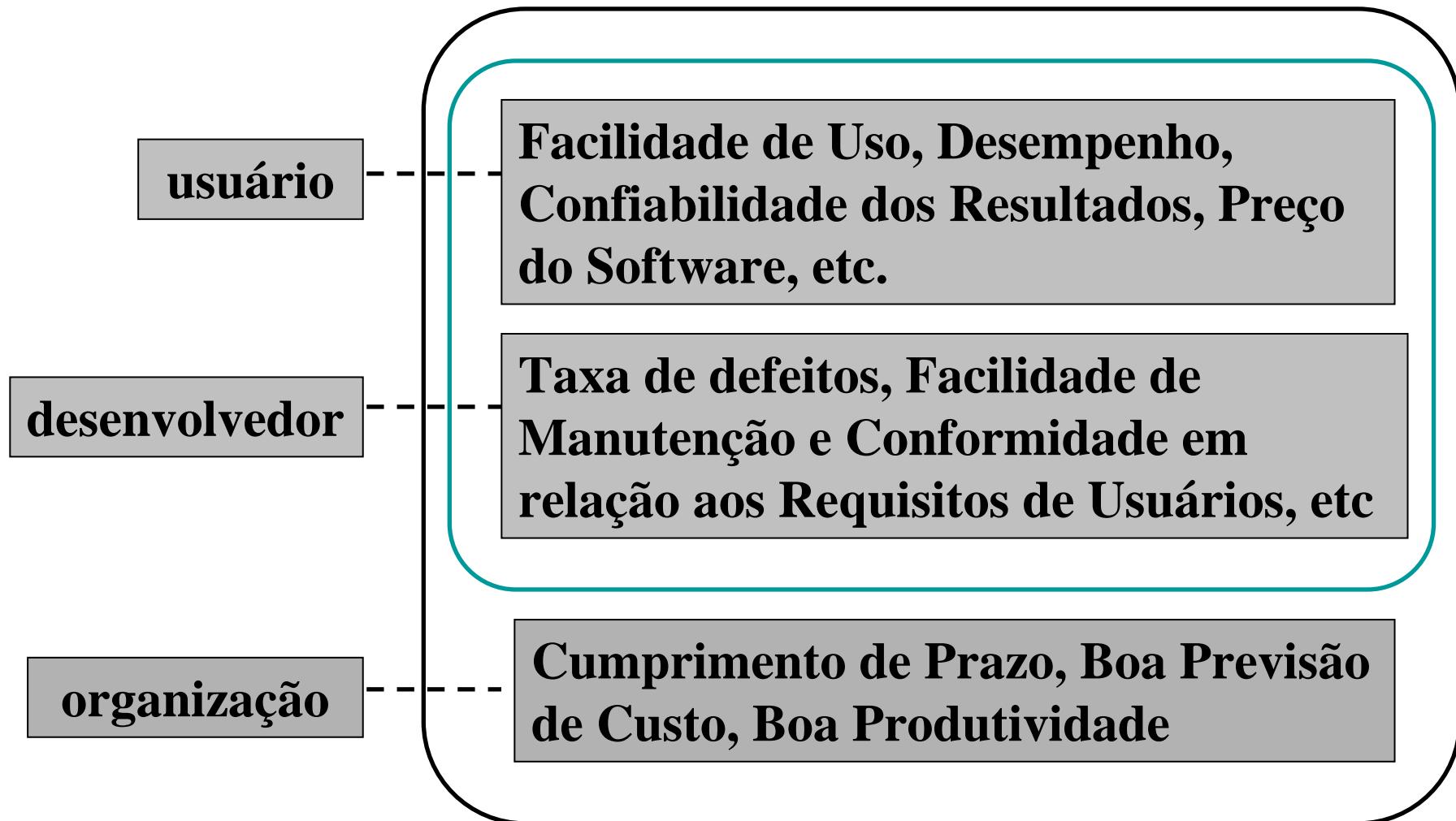
Definições sobre Qualidade

- Definição genérica:
 - “Propriedade, atributo ou condição das coisas ou das pessoas capaz de distingui-las das outras e de lhes determinar a natureza” (Aurélio).
- Outras definições para software:
 - Qualidade é estar em **conformidade** com os **requisitos dos clientes**;
 - Qualidade é antecipar e satisfazer os desejos dos clientes;

Qualidade x Confiança

- Confiança do cliente em estar obtendo um **produto que atende a suas expectativas**
- Confiança do fornecedor em poder **planejar a construção de produtos** em relação aos custos

Visões da Qualidade de Software



Como garantir a qualidade?

Padrões de Qualidade de Software

Matéria da próxima aula

- Padrões de produto:
 - Se aplicam ao produto de software em desenvolvimento. Incluem padrões de documentos, como a estrutura de um documento de requisitos até como uma linguagem de programação deve ser usada.
- Padrões de processo:
 - Definem os processos que devem ser seguidos durante o desenvolvimento de software.

Padrões de Qualidade de Software

[importância] 1/2

1. São baseados no conhecimento sobre as **melhores** e mais **apropriadas** **práticas** para a empresa. Esse conhecimento freqüentemente é adquirido somente após um grande número de tentativas e erros. Ajudam a empresa a **evitar a repetição de erros cometidos no passado**.

Padrões de Qualidade de Software

[importância] 2/2

2. Provêem um **framework conceitual** para a implementação **do processo de garantia de qualidade**.

Considerando que esses padrões **englobam as melhores práticas**, a garantia da qualidade envolve assegurar que padrões apropriados foram selecionados e usados.

Alguns Padrões de Qualidade de Software

- CMM – Capability Maturity Model
 - Desenvolvido pelo SEI (*Software Engineering Institute*)
- SPICE (Software Process Improvement & Capability dEtermination)
 - Desenvolvido pela “*International Organization for Standardization and the International Electrotechnical Commission*” (ISO/IEC)

CMM*

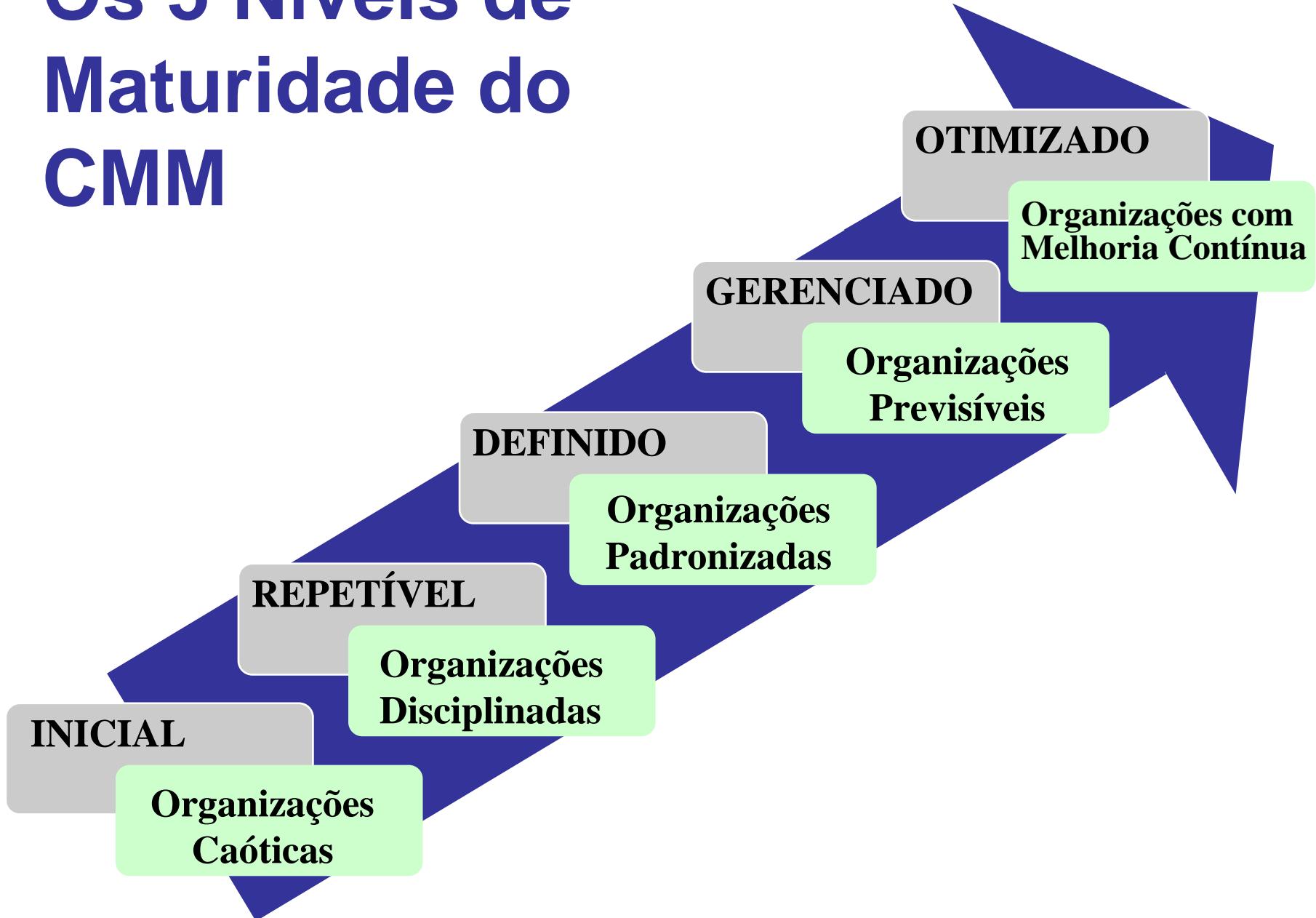
(Modelo de Maturidade de Competência)

- Uma estrutura conceitual que **descreve os elementos chaves de um processo de software eficaz.**
- Um caminho de melhoramento evolucionário (**5 níveis de maturidade**) para organizações de software mudarem de um processo de software imaturo, ad hoc, para um processo maduro, disciplinado.

<http://www.sei.cmu.edu/cmm>

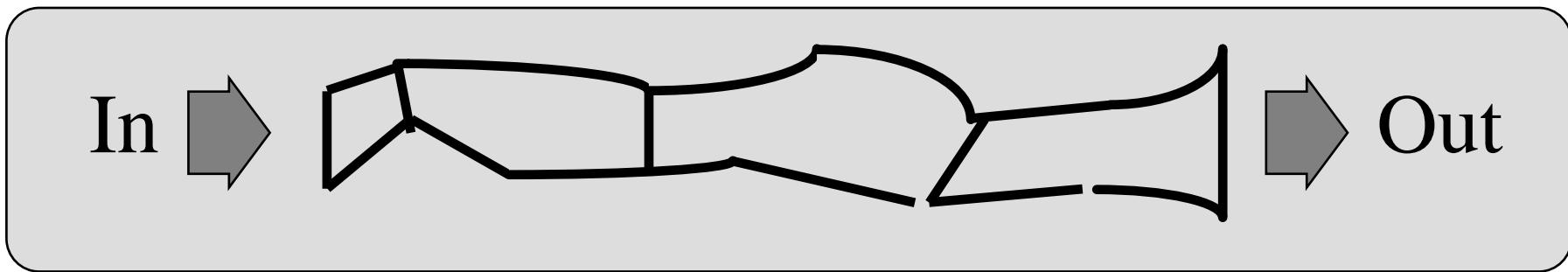
**A versão mais atual é o CMMI
Capability Maturity Model Integration*

Os 5 Níveis de Maturidade do CMM



CMM Nível 1- Inicial

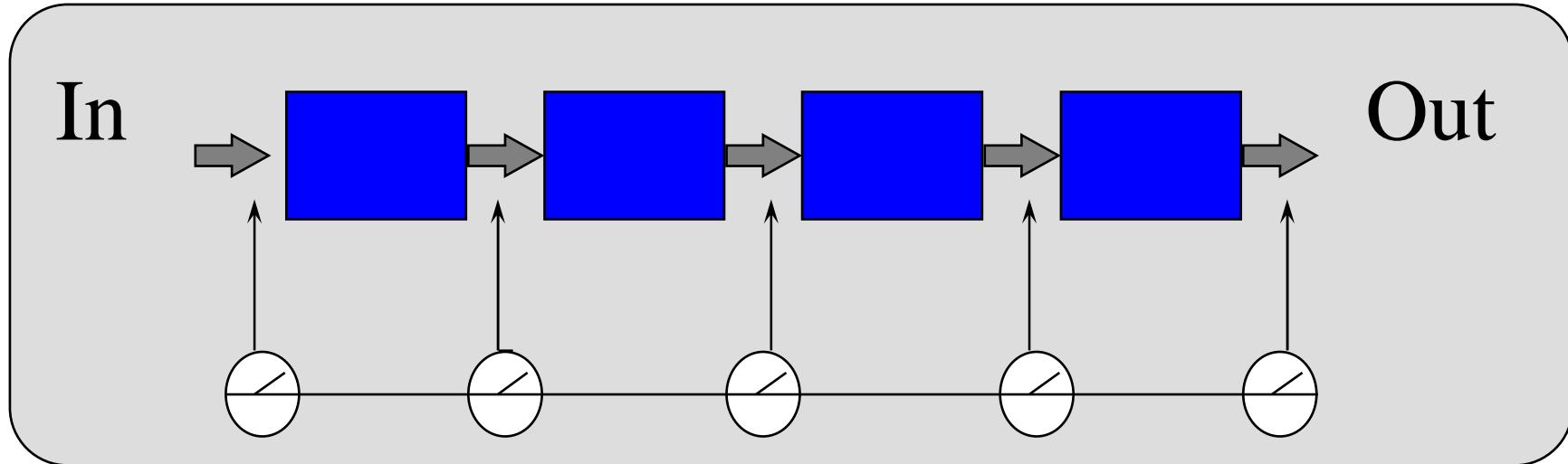
*O Gerenciamento de Software é
uma Caixa Preta*



- Requisitos fluem para dentro
- Um produto de software é (normalmente) produzido através de **algum processo disforme**
- O produto flui para fora e (espera-se) funciona

CMM Nível 2 - Repetível

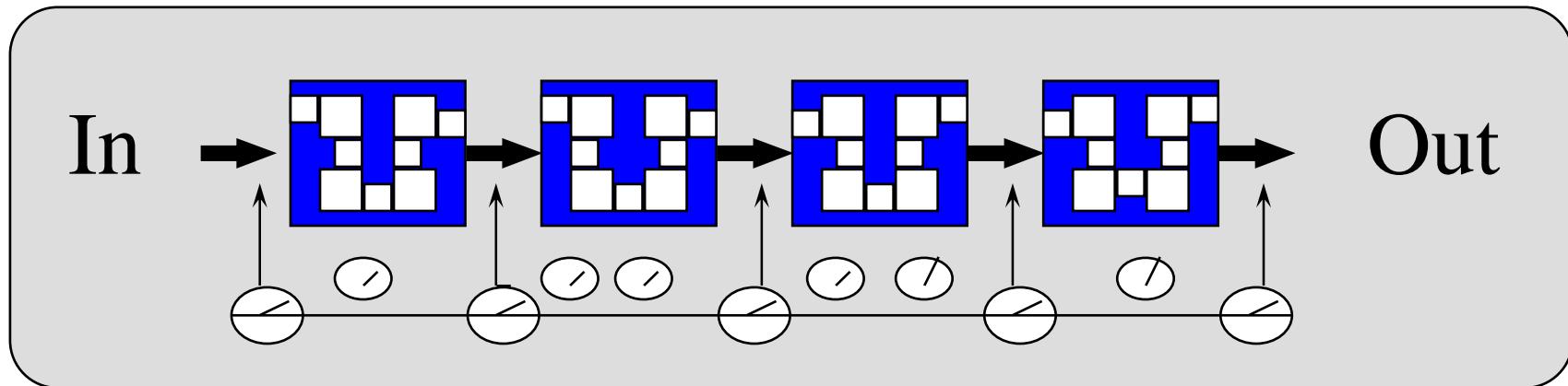
Está em vigor um Sistema de Gerenciamento de Projeto



- **Processo** de construção de software é uma série de **caixas pretas** com **pontos de verificação** definidos

CMM Nível 3 - Definido

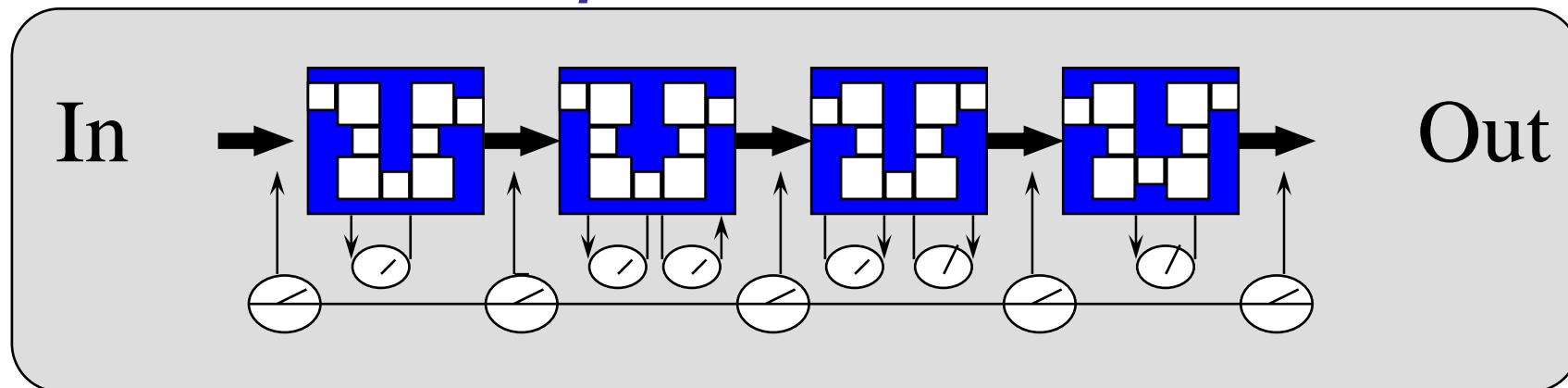
Desenvolvimento de software de acordo com um processo bem definido



- **Funções e responsabilidades** no processo são bem entendidas
- A **produção do produto** de software é **visível** através do processo de software

CMM Nível 4 - Gerenciado

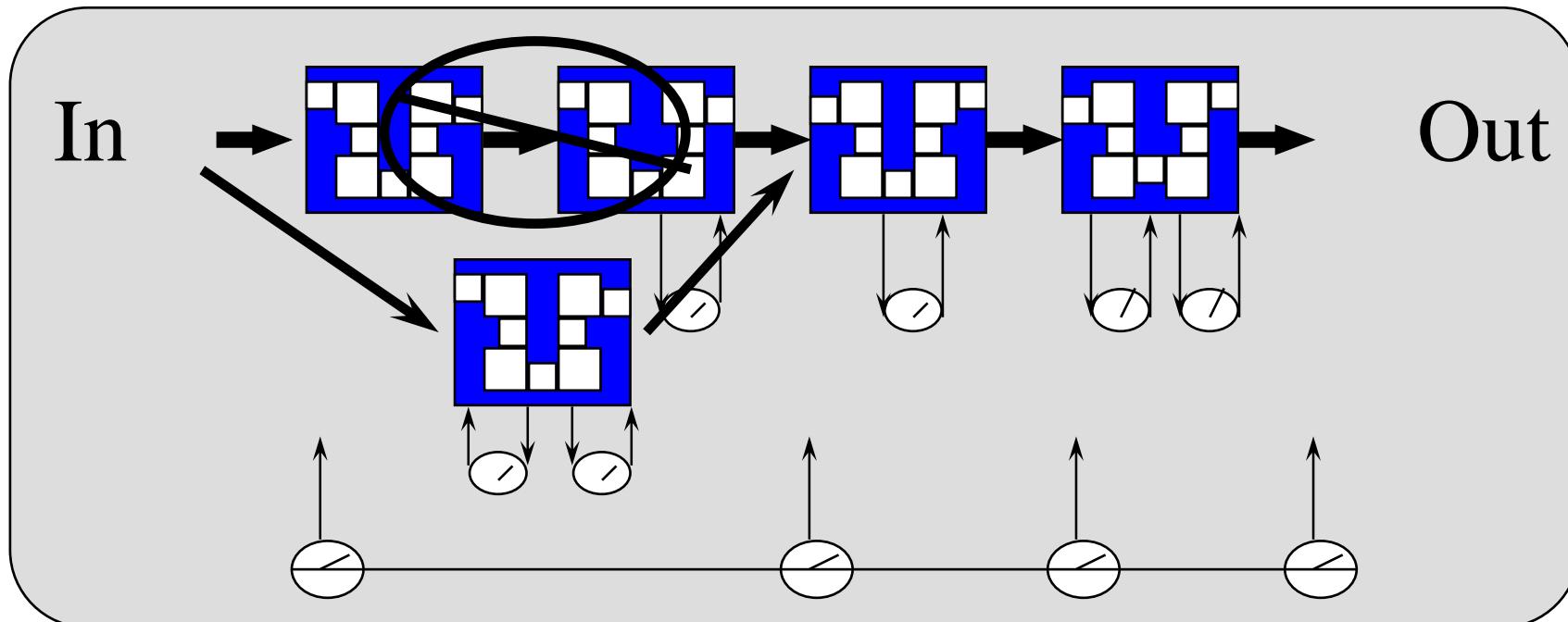
Produto e processo são gerenciados quantitativamente



- A **gerência** tem **bases objetivas** para tomada de decisão
- A gerência é capaz de **prever o desempenho** dentro de limites quantificados

CMM Nível 5 - Otimizado

Foco na melhoria contínua do processo



- Mudança disciplinada é um meio de vida

CMM

[Áreas-chave de Processo (KPA)]

- Indicam as áreas que uma organização deveria enfocar para melhorar seu processo de software;
- O CMM define 18 KPA's distribuídas nos seus níveis de maturidade;
- **Cada KPA é descrita em termos de práticas que contribuem para satisfazer seus objetivos.**
 - descrevem a infra-estrutura e atividades que contribuem para a implementação e institucionalização da KPA.

CMM

[Áreas-chave de Processo (KPA)]

- **Inicial - 1**: não existem KPA's para este nível;
- **Repetível - 2**: interesses relacionados ao **estabelecimento do controle básico** de **administração de projeto**;
- **Definido - 3**: **problemas organizacionais** e de projeto;
- **Gerenciado - 4**: estabelecer um **entendimento quantitativo** do processo de software e do produto;
- **Otimizado - 5**: cobrem os problemas que a organização e os projetos devem endereçar para implementar uma **melhora contínua e mensurável do processo de software**.

Alguns Padrões de Qualidade de Software

- CMM – Capability Maturity Model
 - Desenvolvido pelo SEI (*Software Engineering Institute*)
- SPICE (Software Process Improvement & Capability dEtermination)
 - Desenvolvido pela “*International Organization for Standardization and the International Electrotechnical Commission*” (ISO/IEC)

ISO/IEC 15504 (SPICE)

[definição]

“Melhoria do processo e determinação da capacidade do processo”

- Consiste de um *framework* de avaliação
 - Facilita o auto-julgamento
 - Desperta consciência do contexto
 - Produz um perfil do processo
 - Direciona a adequação das atividades
 - Apropriado para organizações de diversos tamanhos

ISO/IEC 15504 (SPICE)

- Melhoria dos Processos
 - As empresas poderão **identificar quais os processos que devem melhorar**, o que deverá ser feito para este fim e **deduzir onde devem investir em primeiro lugar**, com vista à obtenção de retornos rápidos e significativos.

ISO/IEC 15504 (SPICE)

- Determinação da Capacidade dos Processos
 - Permite às empresas **avaliar** o estado dos seus **processos** em **comparação com as melhores práticas**, através da identificação das suas forças, fraquezas e riscos.
 - Com base nesta avaliação poderão decidir se têm a capacidade para empreender um determinado projeto.

ISO/IEC 15504 (SPICE)

[documentos]

- O SPICE é composto por 9 partes:
 - parte 1: Conceitos e Guia Introdutório
 - parte 2: Modelo de Gerenciamento de Processo
 - parte 3: Avaliação do Processo
 - parte 4: Guia para Condução de uma Avaliação
 - parte 5: Construção, Seleção e Uso das Ferramentas de Avaliação
 - parte 6: Qualificação e Treinamento dos Avaliadores
 - parte 7: Guia para o Processo de Melhoria
 - parte 8: Guia para Orientação da Determinação da Capacidade do Processo
 - parte 9: Dicionários

Comparando os dois Padrões 1/2

Aspectos	CMM (SEI)	SPICE
Abordagem	Classificação das organizações em níveis de maturidade crescente.	Avaliação dos processos com o objetivo de determinar a capacitação da organização e propor melhoria.
Meta/ Objetivo	Determinar a capacitação da empresa e apoiar sua evolução de acordo com os 5 níveis.	Determinar a capacitação da organização e apoiar sua evolução de acordo com os objetivos da organização.
Avaliação de nível de capacitação	Questionário.	Fornece orientações para montar questionário.

Comparando os dois Padrões 2/2

Aspectos	CMM (SEI)	SPICE
Benefícios	Estabelecimento de um roteiro para a melhoria contínua	Expansão e flexibilização dos padrões CMM, ISO 9000 e outros.
Limitações	Pouca consideração à diversidade das organizações. Dificuldade de aplicação em pequenas organizações.	Dificuldade de aplicação devido à grande quantidade de informações.
Empresas Alvo	Organizações de grande porte que necessitam de uma certificação.	Organizações em geral.

<http://www.sei.cmu.edu/cmmi/faq/15504-faq.html>

Métricas de Software

[utilidade]

- Mediçãoes de software podem ser usadas para:
 1. fazer previsões gerais sobre um sistema
 2. identificar componentes anômalos

Métricas

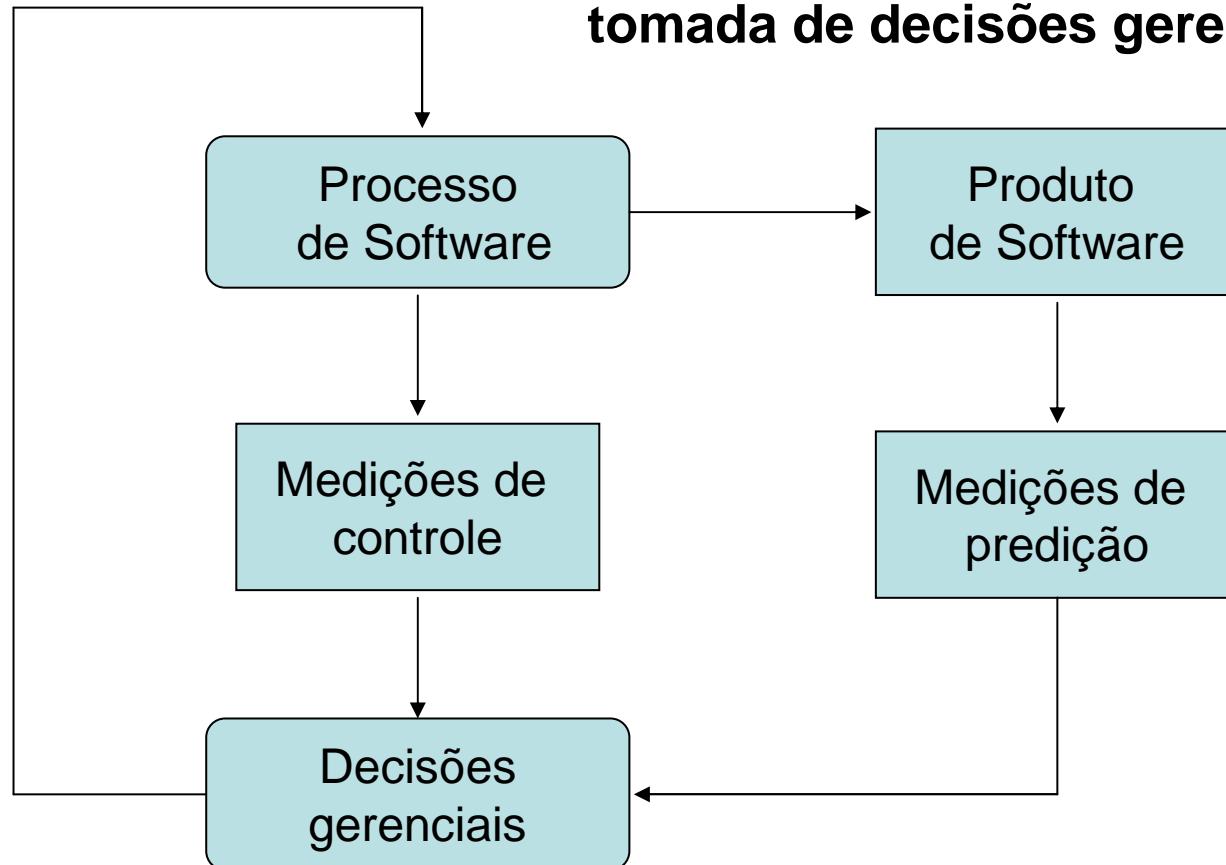
[controle e predição]

- Controle – são associadas com processos de software. Exemplos:
 - Tempo médio para reparar um defeito reportado
 - Tempo para rastrear um módulo
- Predição – são associadas com produtos de software. Exemplo:
 - Complexidade de um módulo

Métricas

[controle e predição]

Influenciam no processo de tomada de decisões gerenciais



[Sommerville]

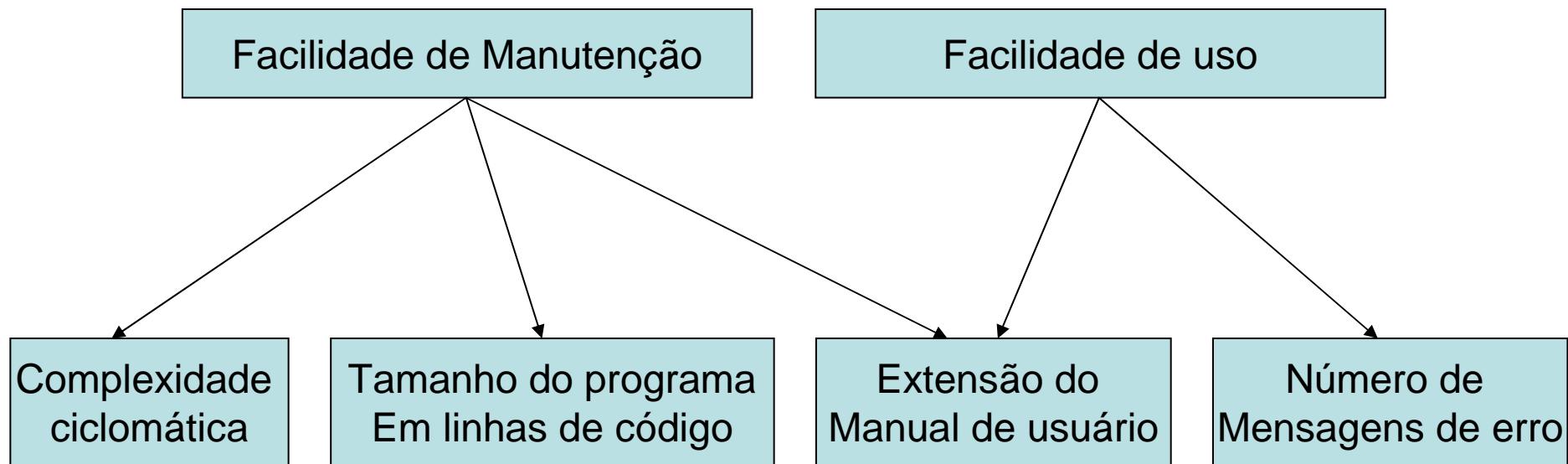
Métricas de Software

[como medir]

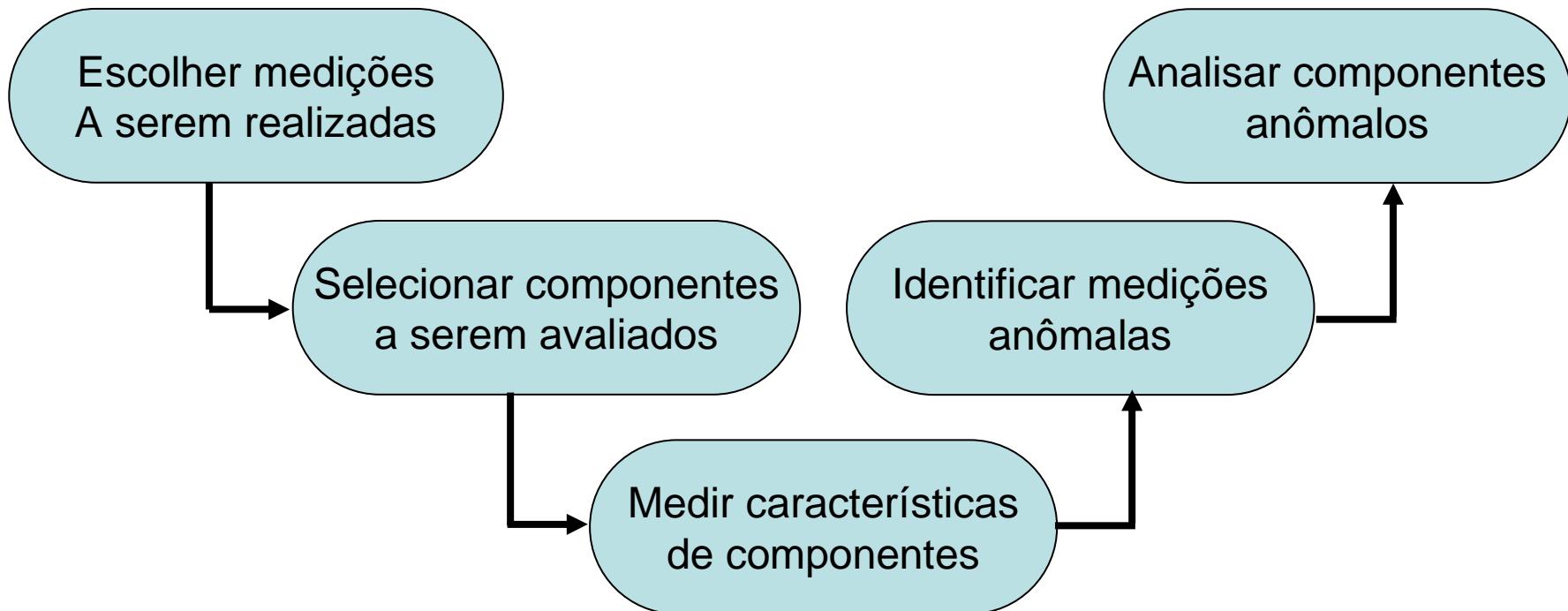
- Geralmente é impossível medir os atributos de software diretamente.
- Atributos como facilidade de manutenção, facilidade de compreensão e facilidade de uso são os atributos externos.

Métricas de Software

[relacionamento de atributos]



Processo de Medição



Métricas de Produto

- Dinâmicas
 - Coletadas por meio de medições realizadas em um programa em execução. Exemplo:
 - Tempo para iniciar um programa
- Estáticas
 - Coletadas por meio de medições realizadas em representações do sistema, como projeto, código fonte ou documentação.

Algumas Métricas de Produto

Fan-in/fan-out	Fan-in é uma medida do número de funções ou métodos que chamam alguma outra função ou método (digamos x). Fan-out é o número de funções chamadas pela função x.
Tamanho do código	Mede o tamanho em relação a linhas de códigos.

Pontos Chaves

- Os padrões de software são importantes para a garantia da qualidade, pois representam uma identificação das melhores práticas.
- As métricas de qualidade de produto são particularmente valiosas para destacar componentes anômalos que podem ter problemas de qualidade.
- Não existem métricas de softwares padronizadas e universalmente aplicáveis.

Exercícios

- Explique por que as métricas de projetos são, por si sós, um método inadequado para prever a qualidade do projeto.
- Pesquisa para ser entregue na próxima aula. Diferenças entre:
 - ISO/IEC 15504, e
 - SPICE, CMMI e CMM.

Referências

- KOSCIANSKI A. e SOARES, M. Qualidade de Software – Ed. Novatec, 2007.
- ROCHA, A. MALDONADO, J. WEBER, K. A Qualidade de software - Teoria e Prática. Prentice Hall. 2001.
- Sommerville, I. Engenharia de Software. Adisson-Wesley, São Paulo, 2007.
- Presman, R. Engenharia de Software. Makron Books. 1995.
- Kautz, K. “Making Sense of Measurement for Small Organizations”. Transactions of IEEE Software, março/abril de 1999.
- SEI - Software Engineering Institute, <http://www.sei.cmu.edu/cmm>
- ISO - International Organization for Standardization, <http://www.iso.org>
- Versão disponível do SPICE <http://www.sqi.gu.edu.au/spice/suite/>
- SEI - Software Engineering Institute, <http://www.sei.cmu.edu/cmm>
- ISO/IEC - <http://www.isospice.com>