

Métodos de Desenvolvimento de Software

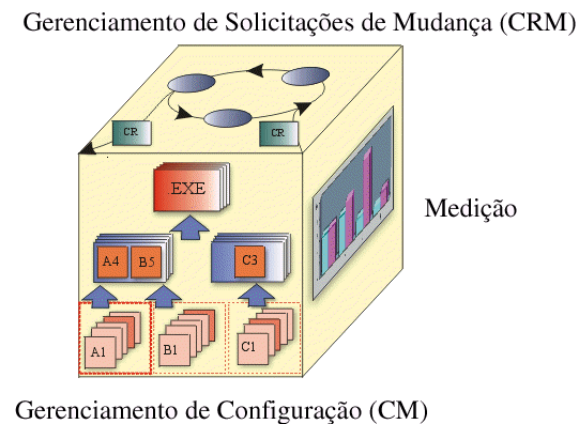
Introdução a Gerência de Configuração e Mudança / Manutenção

Histórico de GCS

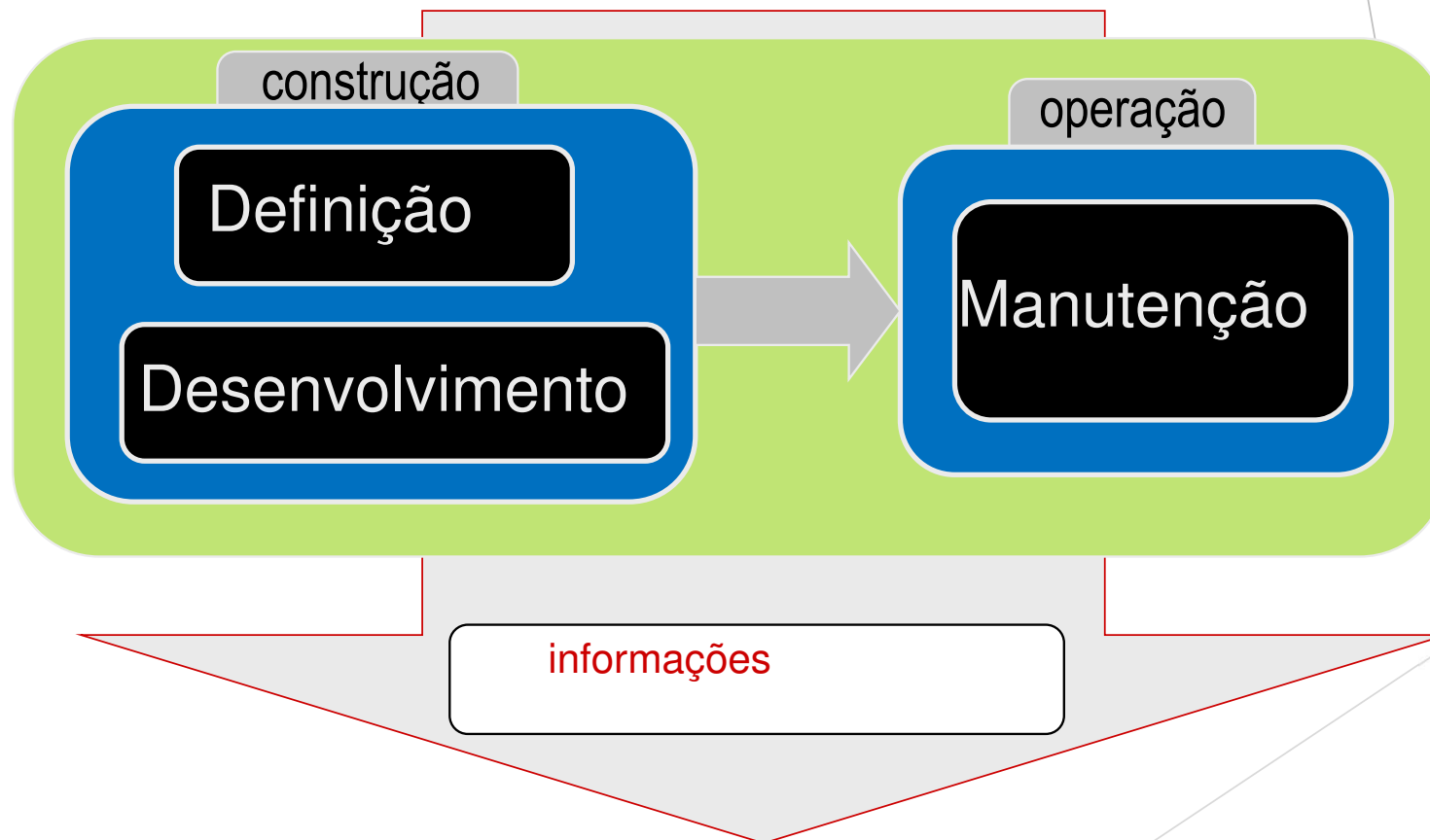
- ▶ Anos 50
 - ▶ GC para produção de aviões de guerra e naves espaciais
- ▶ Anos 60 e 70
 - ▶ Surgimento de GCS (S = Software)
 - ▶ Foco ainda em aplicações militares e aeroespaciais
- ▶ Anos 80 e 90
 - ▶ Surgimento das primeiras normas internacionais (IEEE, ISO, etc.)
 - ▶ Assimilação por organizações não militares

Histórico de GCS

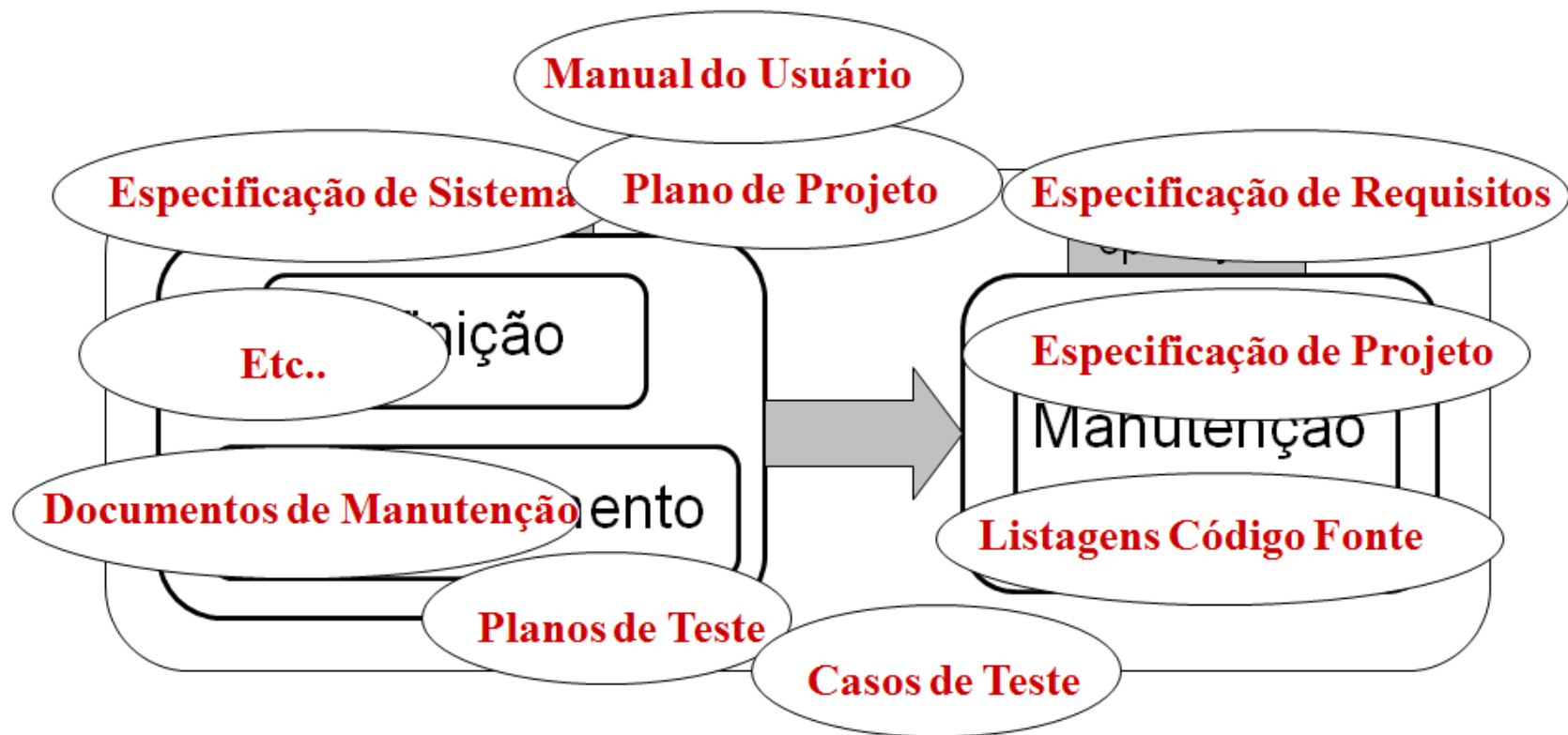
- ▶ O processo de GCS não mudou muito nos últimos 10 anos.
- ▶ Contudo, o ambiente no qual a GCS está inserida têm mudado significativamente
 - ▶ Mainframe, Web-Based, Aplicações críticas e seguras, Mobile...
 - ▶ Processos: cascata, espiral, unificado, ágil.



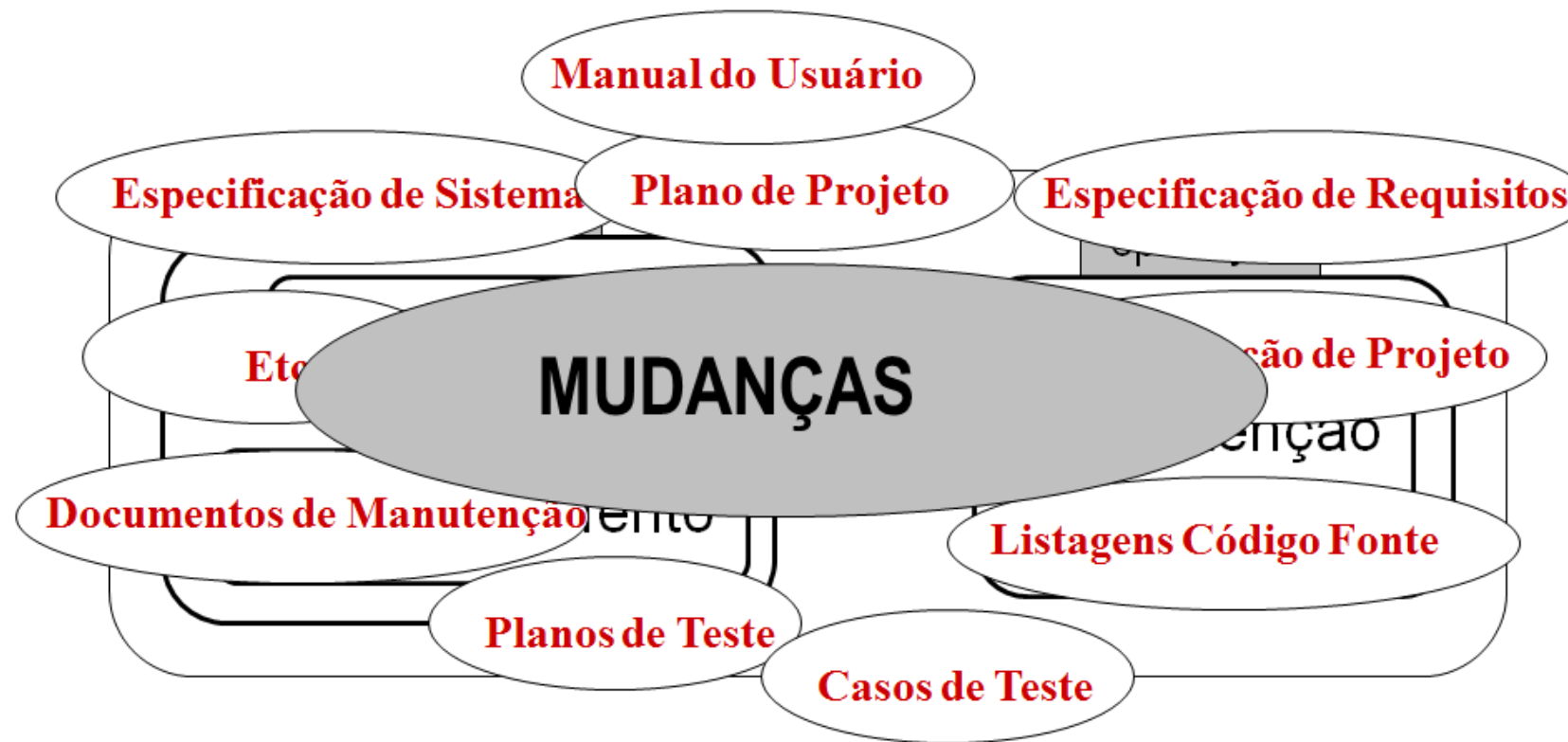
Desenvolvimento de Software



Desenvolvimento de Software (cont.)



Desenvolvimento de Software (cont.)



Desenvolvimento de Software (cont.)



Sintomas do Caos

- Dificuldades imensas durante a integração dos diversos módulos para compor o produto, por problemas de interface;
- Após uma modificação qualquer do produto de software, erros removidos anteriormente reaparecem, ou então, funcionalidades acrescentadas desaparecem, inexplicavelmente;

Sintomas do Caos (cont.)

- Dificuldades em se identificar a última versão do produto de software;
- Incapacidade de se reproduzir, no ambiente de desenvolvimento, um problema informado pelo cliente;
- Perda dos arquivos-fonte que compõem o sistema distribuído ao cliente.

Problema 1: Compartilhamento de Dados

Desenvolvedor A



Desenvolvedor B



Programa de A

A1 A2 A3

**Componente
Compartilhado**

Programa de B

B1 B2 B3

Contextualização para GCS

- ▶ Cenário: Compartilhamento de Dados
 - ▶ O desenvolvedor “A” modifica o componente compartilhado
 - ▶ Mais tarde, o desenvolvedor “B” realiza algumas alterações no mesmo
 - ▶ Ao tentar compilar o componente, erros são apontados pelo compilador, mas nenhum deles ocorre na parte que “B” alterou
 - ▶ O desenvolvedor “B” não tem a menor ideia sobre a causa do problema

Contextualização para GCS

- ▶ Cenário: Compartilhamento de Dados
 - ▶ Solução simplista:
 - ▶ cada desenvolvedor trabalha em uma cópia “local” do componente
 - ▶ resolve o Problema dos Dados Compartilhados, mas cria um novo problema

Problema 2: Manutenção Múltipla

Desenvolvedor A



Desenvolvedor B



Programa de A
A1 A2 A3

**Componente
Compartilhado**

**Componente
Compartilhado**

Programa de B
B1 B2 B3

Versão A do
componente
compartilhado

Versão B do
componente
compartilhado

Contextualização para GCS

- ▶ Cenário: Manutenção Múltipla
 - ▶ Cada desenvolvedor trabalha com uma cópia “local” do que seria o mesmo componente
 - ▶ Dificuldade para saber:
 - ▶ Que funcionalidades foram implementadas em quais versões do componente
 - ▶ Que defeitos foram corrigidos

Contextualização para GCS

- ▶ Cenário: Manutenção Múltipla

- ▶ Solução

- ▶ Criação de uma biblioteca central de componentes compartilhados
 - ▶ Nesse esquema, cada componente é copiado para a biblioteca sempre que alterado
 - ▶ Resolve o Problema da Manutenção Múltipla, mas...

Problema 3: Atualização Simultânea



Contextualização para GCS

► Cenário 1: Atualização Simultânea

- Desenvolvedor **“A”** encontra e corrige um defeito em sua versão do componente compartilhado
- Uma vez corrigido, o componente modificado é copiado para a biblioteca central
- O desenvolvedor **“B”** encontra e corrige o mesmo defeito em sua versão do componente por não saber que **“A”** já tinha feito isso
- O trabalho de **“A”** é desperdiçado

Contextualização para GCS

- ▶ Cenário 2: Atualização Simultânea (cont.)
 - ▶ O desenvolvedor “**A**” encontra e corrige um defeito em sua versão do componente compartilhado
 - ▶ Uma vez corrigido, o componente modificado é copiado para a biblioteca central
 - ▶ O desenvolvedor “**B**” encontra e corrige um outro defeito em sua versão do componente, sem saber do defeito corrigido por “**A**”

Contextualização para GCS

- ▶ Cenário 2: Atualização Simultânea (cont.)
 - ▶ O desenvolvedor “**B**” copia sua versão do componente para a biblioteca central
 - ▶ Além do trabalho de “**A**” ser desperdiçado, a versão do componente que se encontra na biblioteca central continua apresentando um defeito
 - ▶ O desenvolvedor “**A**” julga o problema como resolvido

Contextualização para GCS

- ▶ Atualização Simultânea

- ▶ Solução

- ▶ O problema da atualização simultânea não pode ser resolvido simplesmente copiando componentes compartilhados para uma biblioteca central
 - ▶ Algum mecanismo de controle é necessário para gerenciar a entrada e saída dos componentes

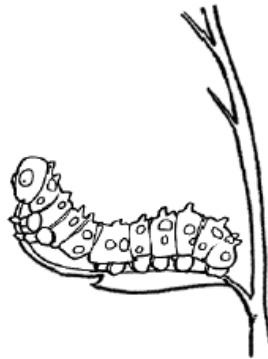
Motivação para GCS

“Nada é permanente, exceto a mudança”. (Heráclito)

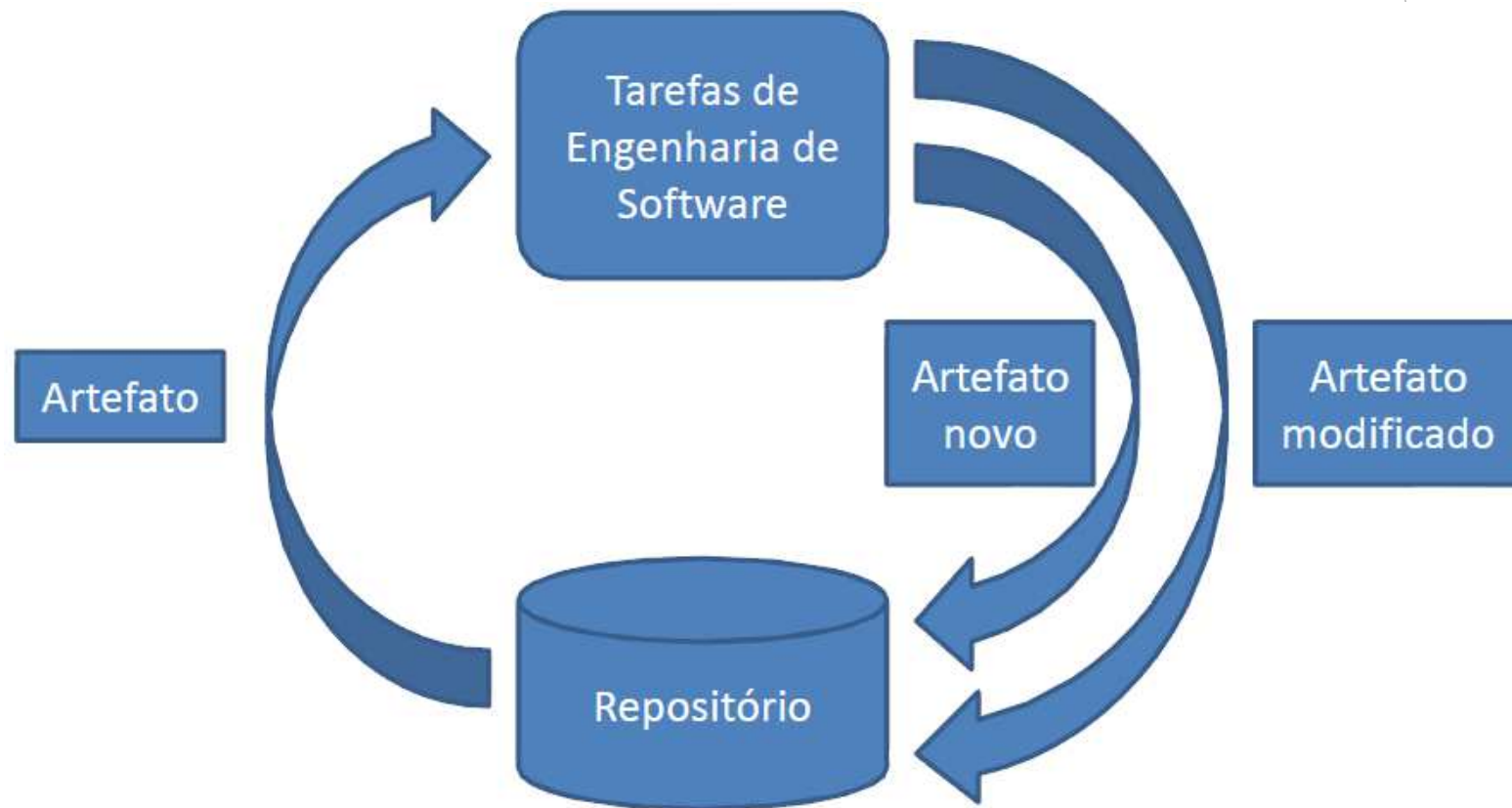
- Eliminar mudanças é eliminar oportunidades e pode significar obsolescência. Pode ser opcional ou não, mas todas precisam ser avaliadas.
- **Fontes de mudanças:** requisitos, orçamento, tecnologia, cronograma, novas oportunidades de melhoria do sistema...

Motivação para GCS

- ▶ Dentro das várias metodologias de desenvolvimento de software existe um ponto comum:
 - ▶ Refinamentos sucessivos de artefatos



Motivação para GCS



Gerência de Configuração de Software

Identifica a configuração do sistema (características documentadas do hardware e software que o compõem) em pontos discretos no tempo, de modo a controlar sistematicamente suas mudanças e manter sua integridade e rastreabilidade durante o ciclo de vida do sistema.

Fonte: Swebok, 2004

Gerência de Configuração de Software

► Finalidade:

- Prover uma base para mudança e melhoria de processos
- Ser um processo que torna possível que os membros de uma equipe trabalhem juntos em um ambiente estável, porém flexível.



Gerência de Configuração de Software

- ▶ Propósito:
 - ▶ Estabelecer e manter a integridade dos produtos de trabalho usando:
 - ▶ Identificação da configuração;
 - ▶ Controle da configuração;
 - ▶ Relato do status da configuração;
 - ▶ Auditoria de configuração.
 - ▶ Disponibilizar os produtos de trabalho de um processo ou projeto a todos os envolvidos.

Gerência de Configuração de Software

► Responsabilidade:

- Controlar a evolução de sistemas de software [DART, 1991]

► Aplicação:

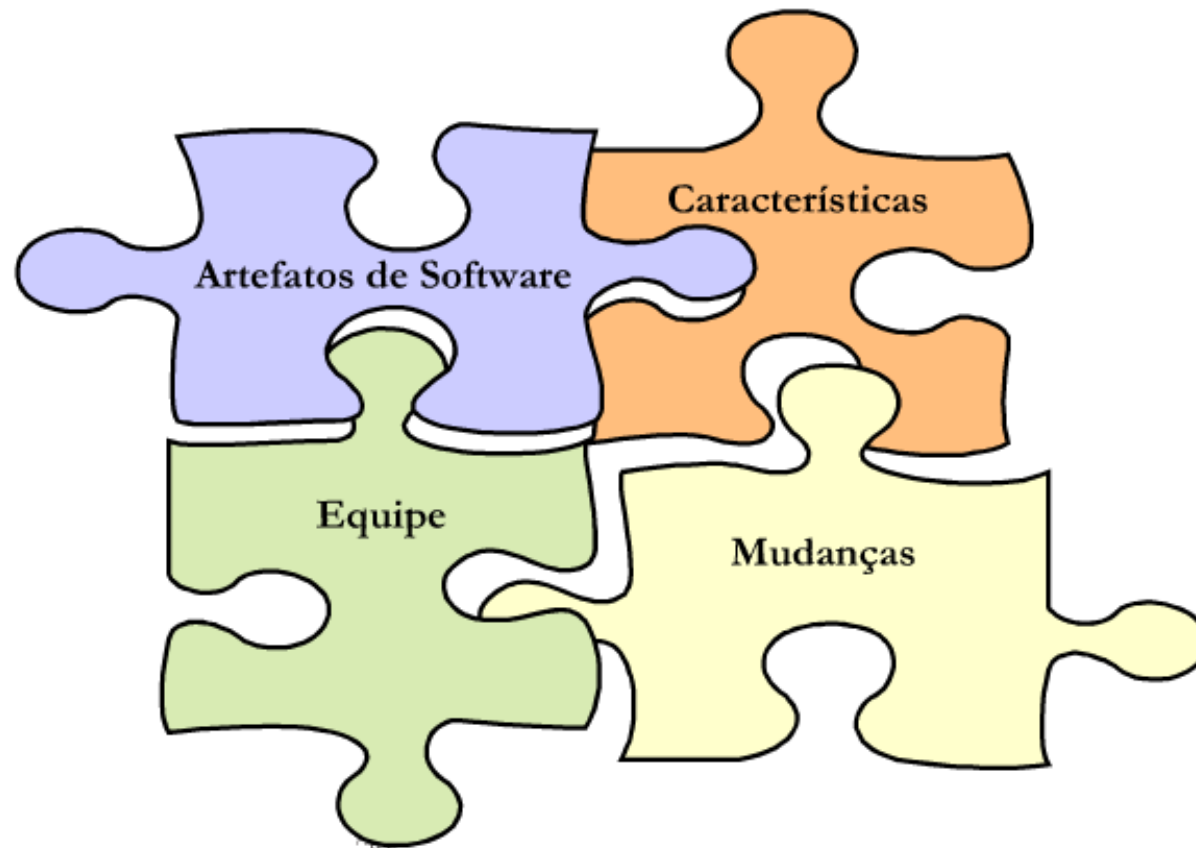
- Durante todo o ciclo de vida de desenvolvimento e manutenção de software [IEEE Std 1042, 1987].



Gerência de Configuração de Software

- ▶ GCS é uma disciplina essencial nas atividades diárias do desenvolvimento de um software
- ▶ CGS **NÃO** é um simples controle de versões!
- Um efetivo processo de GCS endereça os requisitos de qualidade
 - uma vez que rastreia cada requisito desde sua concepção até a entrega para o usuário

O que compõe GCS ?



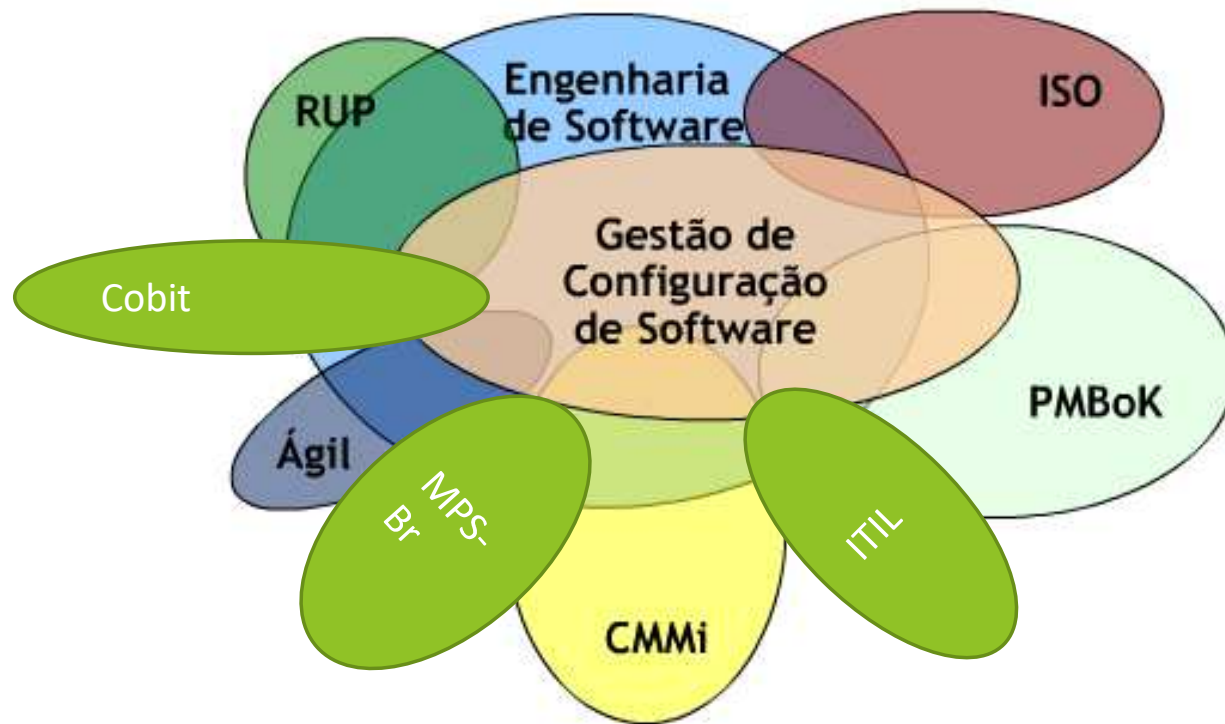
Fonte: Misael Santos, 2010

Onde está inserida GCS ?



Fonte: Molinari, 2007

Onde está inserida GCS ? (cont.)



Fonte: Misael Santos, 2010

Onde está inserida GCS ? (cont.)

Projeto X						
Preparação	Desenvolvimento				Operação e Manutenção	
	Requisitos	Projeto	Codificação	Teste		
Gerenciamento de Projeto						
Garantia da Qualidade						
Gestão de Configuração						

Onde está inserida GCS ? (cont.)



Benefícios de CGS

- ▶ Organiza atividades para manter a integridade do produto;
- ▶ Ajuda a reduzir custos;
- ▶ Auxilia na entrega de um produto dentro do cronograma e orçamento;
- ▶ Possibilita conformidade com requisitos;

Benefícios de CGS (cont.)

- ▶ Provê habilidade de rastrear mudanças;
- ▶ Garante a configuração correta do produto;
- ▶ Possibilita uma fiscalização legal através do registros de ações;
- ▶ Provê um entendimento de quando o produto está pronto;

Benefícios de CGS (cont.)

- ▶ Ajuda na produção de um software de qualidade;
- ▶ Provê um ambiente estável de desenvolvimento;
- ▶ Garante acompanhamento e registro de status;
- ▶ Garante que os engenheiros estão implementando mudanças na versão ou baseline correta.

Custos de CGS

- ▶ Investimentos com a adoção, que envolvem **capacitação** (aquisição de informações, consultorias, etc.) e **treinamento** (custos com treinamento de pessoal envolvido com a adoção de GCS).
- ▶ Investimentos para a implementação, que englobam **recursos humanos** (custos com pessoal envolvido com a implementação de GCS), **recursos computacionais** (computadores e dispositivos a serem utilizados pela GCS) e **ferramentas de GCS** (ferramentas de software para automatizar o processo de GCS).

Custos de CGS (cont.)

- ▶ Investir em GCS é semelhante a se investir em seguros: custos e benefícios devem ser avaliados criteriosamente;
- ▶ Não há solução ou fórmula prontas para resolver esta relação;
- ▶ O quanto deve-se investir em GCS dependerá do tipo e valor do software a ser controlado e, principalmente, de seu grau de criticidade.

Impedimentos de CGS

- ▶ **Custos**

- ▶ Implica em gastos com pessoal, equipamentos, etc.

- ▶ **Cultura da Empresa**

- ▶ Resistência das pessoas à mudança e dificuldade de seguir padrões.

- ▶ **Aumento da formalidade e burocracia**

- ▶ Haverá procedimentos a serem seguidos.

- ▶ **Falta de conhecimento sobre o assunto**

- ▶ A implantação de GCS envolve conhecimento de todas as pessoas envolvidas na equipe, e nem todas elas possuem as mesmas informações.

Perguntas de GCS

- ▶ **Quem** fez a mudança?
- ▶ **Que** mudanças foram feitas?
- ▶ **Quando** as mudanças foram feitas?
- ▶ **Por que** as mudanças tiveram de ser feitas?

Quem trabalha com GCS ?

- ▶ Gestor de Configuração de Software (GCS)
- ▶ Programadores
- ▶ Analistas de Requisitos
- ▶ Analistas de Teste
- ▶ Analistas de Suporte
- ▶ Analistas de Negócio
- ▶ Gerentes de Projeto
- ▶ **Toda a Equipe !**



Fonte: Misael Santos, 2010

Conceitos Fundamentais

- ▶ Configuração
- ▶ Item de Configuração
- ▶ Repositório
- ▶ Check-in / Check-out
- ▶ Baseline
- ▶ Características Físicas e Funcionais
- ▶ Controle de versão e de Mudanças
- ▶ Níveis de Controle
- ▶ CCB

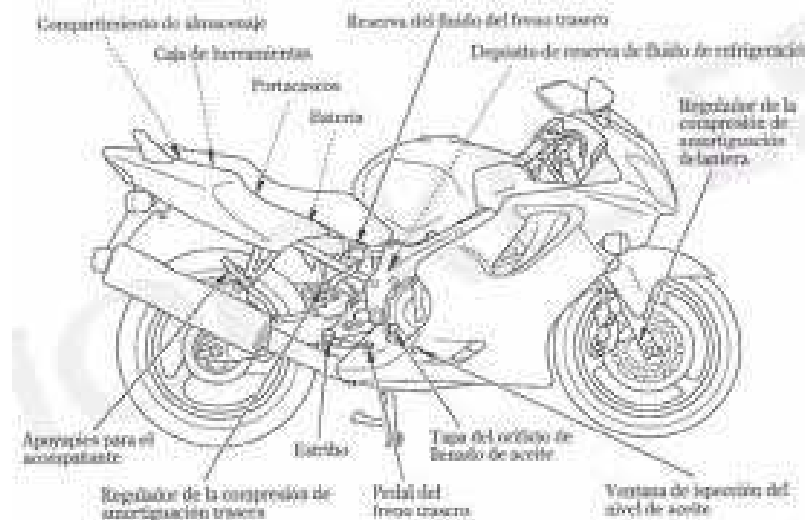
Conceitos Fundamentais (cont.)

- ▶ Configuração
 - ▶ Conjunto de itens que constituem um produto



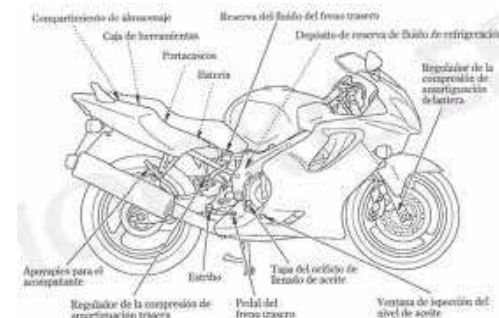
Conceitos Fundamentais (cont.)

► Item de Configuração



Conceitos Fundamentais (cont.)

- ▶ Item de Configuração (cont.)
 - ▶ **Menor componente** de uma configuração que será controlado via Gerência de Configuração.
 - ▶ Um produto de trabalho, ou um agregado de produtos de trabalho que são **tratados como uma entidade única**, que está sob os rigores do processo de Gerência de Configuração.
 - ▶ **NÃO É APENAS CÓDIGO!**



Conceitos Fundamentais (cont.)

► Item de Co

1. Especificação do Sistema
2. Plano de Projeto de Software
3. Especificação de Requisitos do Software
4. Manual Preliminar do Usuário
5. Especificação do Projeto
 - a) Descrição do Projeto de Dados
 - b) Descrição do Projeto Arquitetural
 - c) Descrições do Projeto Modular
 - d) Descrições do Projeto de Interface
 - e) Descrições de Objetos (se forem usadas técnicas orientadas a objetos)
6. Listagem do código-fonte
7. Planos, Procedimentos, Casos de Testes e Resultados Registrados
8. Manuais Operacionais e de Instalação
9. Programa Executável e Módulos Interligados
10. Descrição do Banco de Dados
 - a) Esquema e estrutura de arquivo
 - b) Conteúdo inicial
11. Manual do Usuário
12. Documentos de Manutenção
 - a) Relatórios de problemas de software
 - b) Solicitações de manutenção
 - c) Pedidos de mudança
13. Padrões e procedimentos para engenharia de software
14. Ferramentas de produção de software (editores, compiladores, CASE, etc.)

**Lista de itens de configuração
sugerida por Pressman (2005)**

Conceitos Fundamentais (cont.)

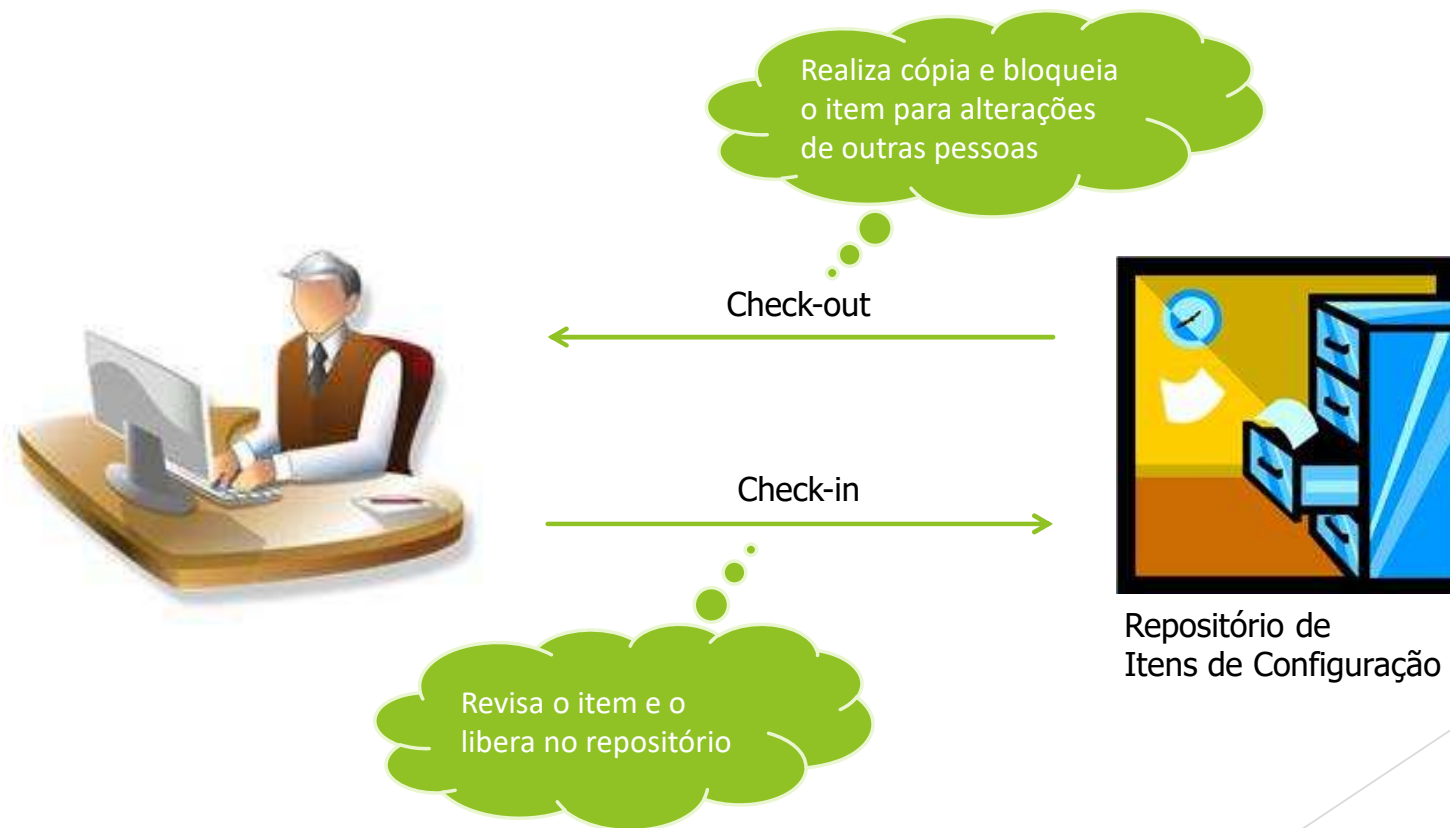
- ▶ Repositório
 - ▶ Lugar seguro onde artefatos são depositados
 - ▶ Permitem armazenamento, busca e recuperação de artefatos
 - ▶ Servem como um ponto de referência
 - ▶ Apoiam no aumento da memória organizacional



Conceitos Fundamentais (cont.)

- ▶ Check-in / Check-out
 - ▶ Método utilizado para trabalhar com os itens de configuração que já se encontram no repositório
 - ▶ **Check-out:** gera uma cópia do item de configuração e não permite que outras pessoas alterem o mesmo item (*controle de concorrência*).
 - ▶ **Check-in:** revisa o item de configuração alterado e o recoloca no repositório, liberando para alteração de outras pessoas.

Conceitos Fundamentais (cont.)



Conceitos Fundamentais (cont.)

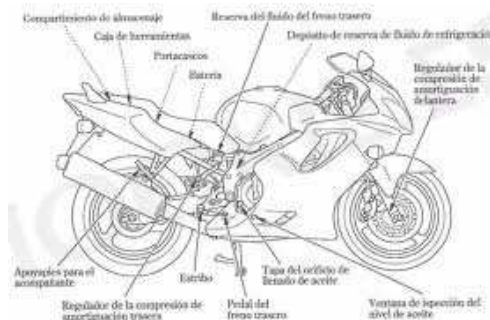
- ▶ Baseline
 - ▶ Uma versão formalmente **aprovada** de um item de configuração ou um conjunto de itens formalmente definida e fixada em um determinado momento durante o ciclo de vida.
 - ▶ **Serve de base para trabalhos posteriores.**



Conceitos Fundamentais (cont.)

Características Físicas

A **auditoria física** examina a estrutura de todos os itens de configuração que compõem a baseline. Desta forma, o seu foco principal está na **completude** da baseline, ou seja, se ela **contém todos os itens** de configuração especificados.



Características Funcionais

A **auditoria funcional** ocorre por meio da revisão dos planos, dados, metodologia e resultado de testes, com o objetivo de verificar a **corretude** da baseline, ou seja, se ela **cumpe o que foi especificado**.



Conceitos Fundamentais (cont.)

Controle de Mudanças

Trata do uso sensato de meios para efetuar uma mudança ou proposta de mudança em produto ou serviço.

Como? Quando? Por quê?

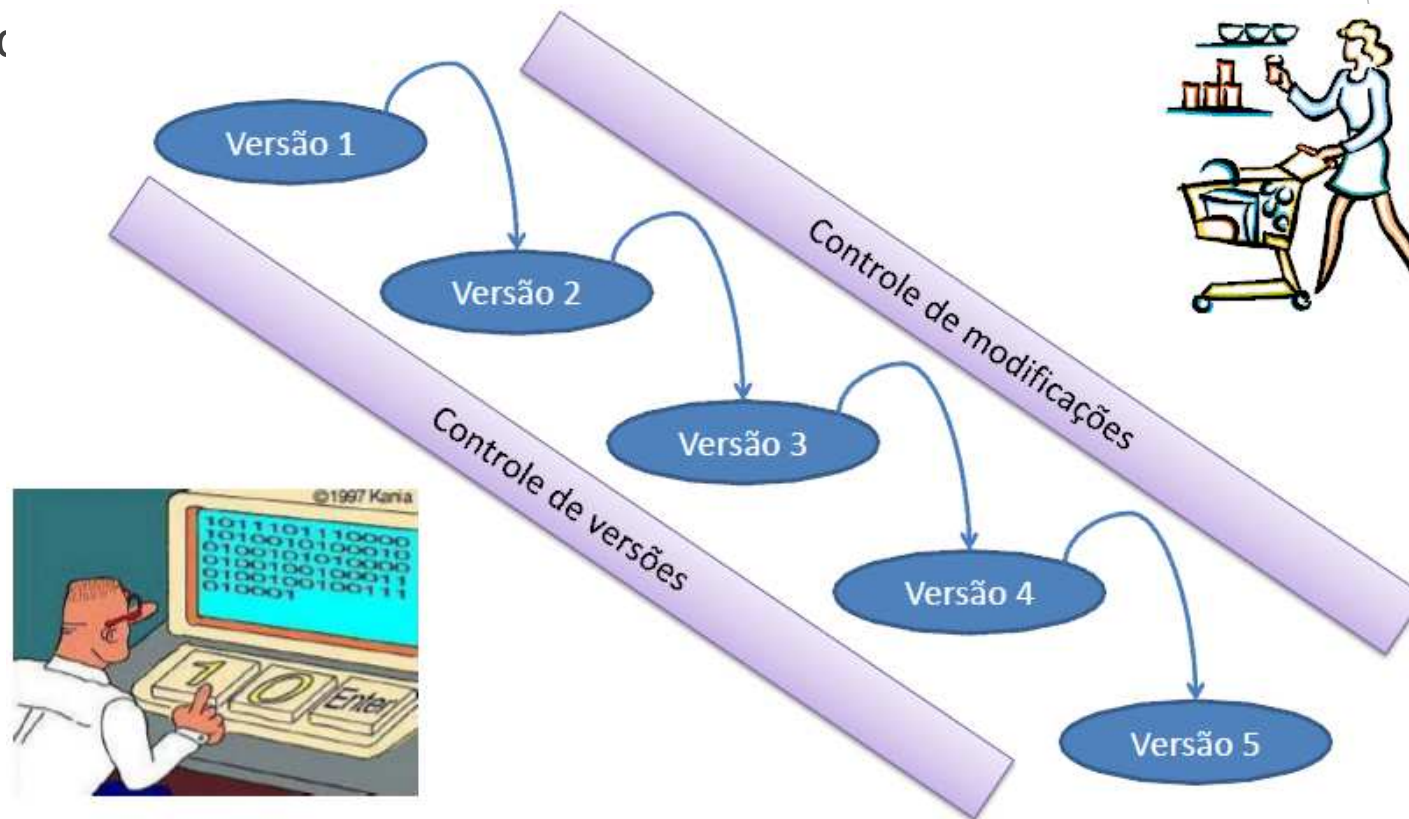
Controle de Versão

Trata do estabelecimento, manutenção e identificação de mudanças em baselines que tornam possível retornar a uma baseline anterior.

A mudança foi realizada. Como retornar a uma versão anterior?

Conceitos Fundamentais (cont.)

► Controle de



Conceitos Fundamentais (cont.)

- Níveis de Controle
 - Diz respeito ao rigor com que cada item de configuração será tratado pela Gerência de Configuração
 - **Controle de Armazenamento**
 - *(guarde com cuidado!)*
 - **Controle de Versão**
 - *(guarde com cuidado, e controle as diversas versões!)*
 - **Controle de Mudança**
 - *(guarde com cuidado, e controle as diversas versões e modificações!)*

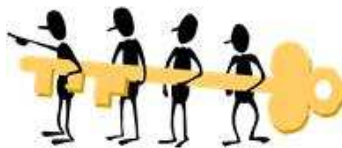
Conceitos Fundamentais (cont.)

- CCB

- ▶ *Configuration Control Board (CCB)*

- Equipe que possui a autoridade por aceitar ou rejeitar uma solicitação de mudança
 - Em projetos pequenos essa autoridade pode ser delegada a uma ou duas pessoas

Fonte: Swebok, 2004



Medição, Métricas e GCS

- ▶ A contabilidade na GCS fornece visibilidade do estado dos produtos relacionados ao desenvolvimento de software
- ▶ Medições e métricas podem ser usadas para identificar áreas do processo que requerem mais atenção!



Medição, Métricas e GCS (cont.)



Métodos de Desenvolvimento de Software

Introdução a Gerência de Configuração e Mudança / Manutenção

