



Gerência de Configuração: Introdução

Leonardo Gresta Paulino Murta

leomurta@ic.uff.br

Histórico

- 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
 - Mudança de foco (MIL → EIA, IEEE, ISO, etc.)
 - Surgimento das primeiras normas internacionais
 - Assimilação por organizações não militares

Principais normas

- AFSCM 375-1 (1962)
 - Somente hardware é considerado
- MIL Std 483 (1971)
 - Software passa a ser considerado
- DOD Std 2167A (1985)
 - Consolidação das normas existentes
- MIL Std 973 (1992, 1993, 1995)
 - Usada amplamente em contratos governamentais
 - Descontinuada em 2000
- EIA 649 (1998, 2004)
 - Substitui a MIL Std 973
 - Compatível com ISO 12207

Principais normas

- IEEE Std 828 (1983, 1990, 1998, 2005)
 - Trata da confecção de planos de GCS
 - Principal referência em ambientes não militares
- IEEE Std 1042 (1987)
 - Consiste em um guia para aplicação da IEEE Std 828
 - É considerada uma das normas internacionais mais completas sobre GCS
 - Reafirmada em 1993
 - Descontinuada em 2000
- ISO 10007 (1995, 2003)
 - Baseada na MIL Std 973
 - Visa satisfazer ISO 9000
 - Será possivelmente substituída pela EIA 649

Principais normas

- ISO 12207
 - Processo de GC baseado em
 - IEEE Std 828
 - IEEE Std 1042
 - ISO 10007
- CMMI
 - Área de processo nível 2
- MPS.BR
 - Processo nível F
- ISO/IEC TR 15846 (1998)
 - Relatório técnico (não normativo)

Principais livros

- “Software Configuration Management Handbook”
 - Alexis Leon (2004)
- “Software Configuration Management”
 - Jessica Keyes (2004)
- “Configuration Management Principles and Practice”
 - Anne Hass (2002)
- “Software Configuration Management Patterns: Effective Teamwork, Practical Integration”
 - Stephen Berczuk e Brad Appleton (2002)

Principais artigos

- “Concepts in Configuration Management Systems”
 - Susan Dart, *SCM* (1991)
- “Software Configuration Management: A Roadmap”
 - Jacky Estublier, *ICSE – Future of SE Treck* (2000)
- “Impact of Software Engineering Research on the Practice of Software Configuration Management”
 - J. Estublier, D. Leblang, A. van der Hoek, R. Conradi, G. Clemm, W. Tichy, D. Wiborg-weber, *IEEE TOSEM* (2005)
- “Version Models for Software Configuration Management”
 - R. Conradi e B. Westfechtel, *ACM Comp. Surveys*, 30, 2 (1998)

Principais referências

- <http://www.cmcrossroads.com>
 - Principal comunidade de GCS na web
- <http://www.cmcrossroads.com/yp>
 - Páginas amarelas (livros, ferramentas, treinamento, etc)
- <http://www.cmcrossroads.com/jobs>
 - Canal para empresas buscarem por funcionários de GCS
- <http://www.bradapp.net/acme>
 - Padrões de GCS
- <http://www.cmbok.com>
 - *Body of Knowledge* de GCS
- <http://www.cmwiki.com>
 - Wiki sobre GCS

Principais referências

- <http://www.ucmcentral.com>
 - Documentação introdutória
 - Métricas para GCS
 - Formulários (*templates*) para GCS
- <http://www.icmhq.com>
 - Instituto de Gerência de Configuração
 - Não articula com as demais iniciativas
 - Cria suas certificações (CMII)
- <http://www.cmcommunity.com>
 - Comunidade voltada para CMII
- <http://dblp.uni-trier.de/db/conf/scm>
 - Principal conferência de GCS
 - Existe desde 1988
 - Conta com contribuições da academia e indústria

Contra-definição

- GC não é (somente) controle de versões!
- GC não é configuração de conteúdo/dados (ver PDM)!
- GC não é backup!

- GC não é simples!
- GC não é impossível!
- GC não é modismo!
- GC não é opcional!
- GC não é uma panacéia!
- GC não evita que ocorram modificações!
- GC não termina nela mesma!
- GC não é somente para sistemas grandes e complexos!
- GC não é somente para grandes equipes geograficamente distribuídas!

Definição

- Wayne Babish
 - A arte de coordenar o desenvolvimento de software para minimizar confusão é chamada GC. O objetivo é maximizar a produtividade e minimizar os erros
- Susan Dart
 - GCS é uma disciplina para o controle da evolução de sistemas de software
- Steve McConnell
 - GC é a prática de lidar com modificações de forma sistemática, permitindo que o sistema tenha a sua integridade mantida com o passar do tempo
- Walter Tichy
 - GC é a disciplina para o controle e evolução de sistemas complexos

Definição

- Jacky Estublier
 - GC é a **disciplina** que nos permite **evoluir produtos de software de forma controlada**, e, desta forma, contribui na **satisfação de restrições de qualidade e de tempo**
- IEEE Std 610
 - GC é uma **disciplina** que aplica **procedimentos técnicos e administrativos** para **identificar e documentar** as características físicas e funcionais de um item de configuração, **controlar as alterações** nessas características, **armazenar e relatar** o processamento das modificações e o estágio da implementação e **verificar** a compatibilidade com os requisitos especificados

Definição

- ISO 10007
 - Atividades **técnicas e organizacionais** compreendendo: **identificação** de configuração; **controle** de configuração; **contabilização da situação** de configuração; **auditoria** de configuração
- ISO 12207
 - O processo de gerência de configuração é um processo de aplicação de procedimentos **administrativos e técnicos, por todo o ciclo de vida de software**, destinado a: **identificar e definir** os itens de software em um sistema, e **estabelecer suas baselines; controlar** as modificações e **liberações dos itens; registrar e apresentar** a situação dos itens e dos pedidos de modificação; **garantir a completeza, a consistência e a correção dos itens;** e **controlar o armazenamento, a manipulação e a distribuição** dos itens

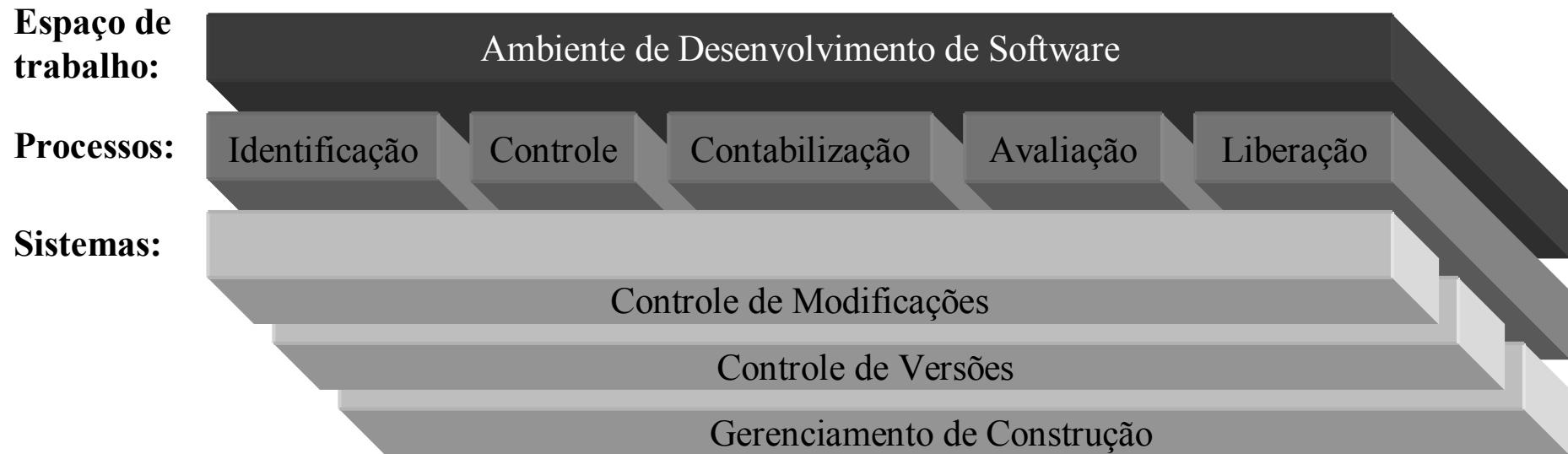
Definição

- CMMI
 - O propósito de GC é **estabelecer e manter a integridade** dos produtos de trabalho utilizando **identificação** da configuração, **controle** da configuração, **contabilização da situação** da configuração e **auditoria** da configuração
- MPS.BR
 - O propósito do processo de Gerência de Configuração é **estabelecer e manter a integridade** de todos os produtos de trabalho de um **processo ou projeto** e **disponibilizá-los** a todos os envolvidos

Foco de atuação

- Controle sobre a evolução de produtos de trabalho
 - Produtos de trabalho do projeto
 - Produtos de trabalho dos processos
- Fonte de informações para outros processos
 - Ex. 1: dados históricos para medição e análise
 - Ex. 2: laudo de defeitos para análise de causa
- Garantia de que foi feito o que deveria ter sido feito

Perspectivas



 **Perspectiva de integração**

 **Perspectiva gerencial**

 **Perspectiva de desenvolvimento**

Cenário atual

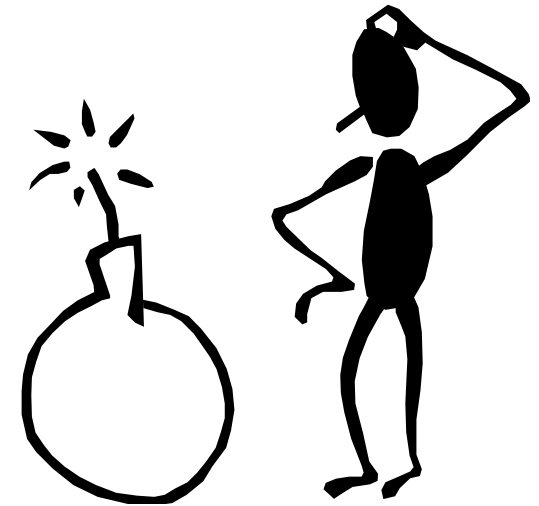
- Projetos cada vez mais complexos em relação ao tamanho, sofisticação e tecnologias envolvidas
- Grandes equipes geograficamente dispersas
- Requisitos de execução do sistema em diferentes plataformas de hardware e software
- Suporte a diferentes línguas e culturas
- Sabores específicos para equacionar custo/benefício (eg.: *Desktop, Standard, Professional e Enterprise*)

Cenário atual

- Crescente velocidade da comunicação entre clientes sobre a ocorrência de defeitos
- Diminuição dos tempos de desenvolvimento e de correção de defeitos para preservar a reputação da empresa
- Necessidade de **lidar com o inevitável caos** relacionado à atividade criativa, buscando métodos e ferramentas para **maximizar a produtividade e minimizar os erros**

Problemas pela falta de GC

- Perda de código-fonte
- Bibliotecas inesperadamente não funcionam mais
- Impossibilidade de determinar o que aconteceu com um programa, ou parte dele
- Impossibilidade de determinar quem, porque e quando foram efetuadas modificações



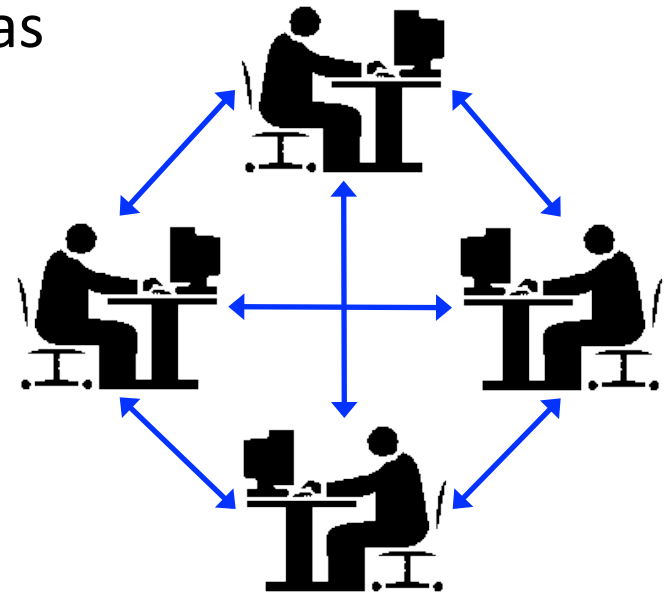
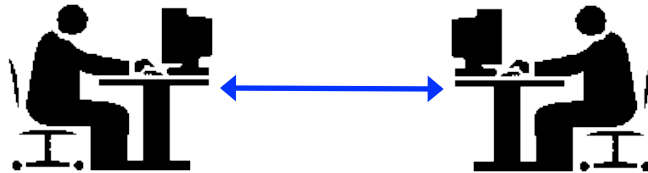
Problemas pela falta de GC

- Requisitos já documentados desaparecem
- Requisitos implementados desaparecem do código
- O programa em execução e o seu código fonte estão em diferentes versões



Evolução das técnicas de GC

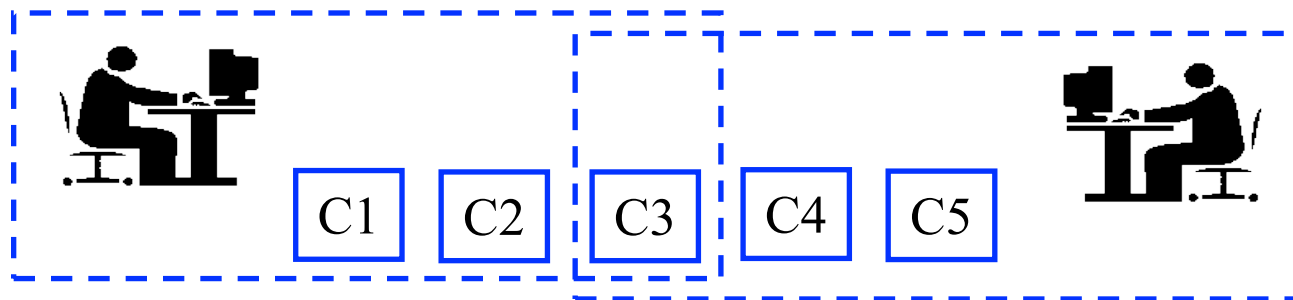
- Sistemas pequenos, desenvolvidos por somente uma pessoa são menos dependentes de GC
- O crescimento dos sistemas implica no aumento da equipe, o que acarreta problemas relacionados com comunicação



[Leon, 2000] Communications breakdown problem

Evolução das técnicas de GC

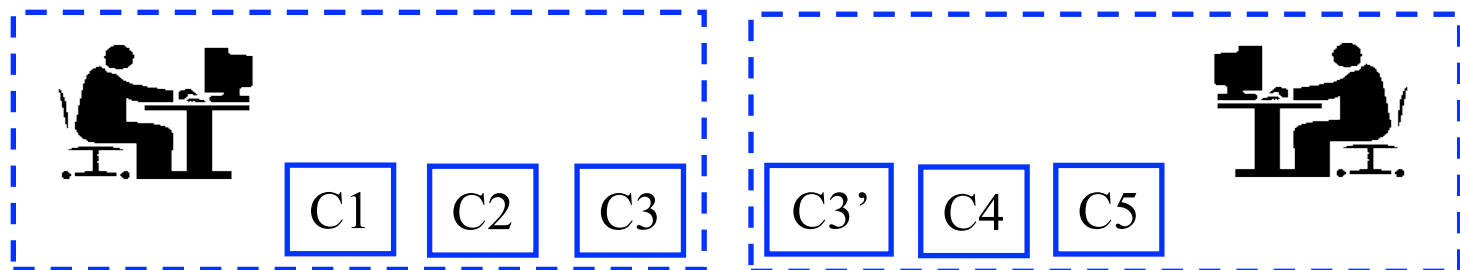
- Artefatos de software são compartilhados por diversos desenvolvedores
- Modificações efetuadas por outros desenvolvedores não são comunicadas a todos os interessados nos artefatos, gerando incompatibilidades de versões



[Leon, 2000] Shared data problem

Evolução das técnicas de GC

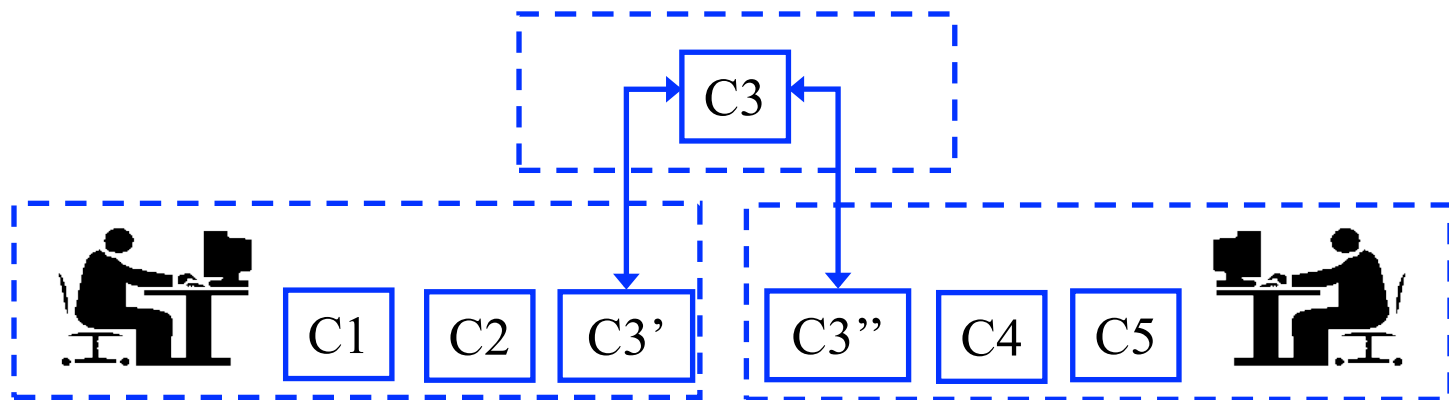
- Solução inicial consiste em criar várias cópias do mesmo artefato compartilhado
- Falta de controle sobre as cópias existentes
- Retrabalho nas diferentes cópias para implementar os mesmos requisitos e corrigir os mesmos defeitos



[Leon, 2000] Multiple maintenance problem

Evolução das técnicas de GC

- Criação de bibliotecas centralizadas;
- Retorno ao problema de artefatos de compartilhados
- Perda de trabalho devido a falta de exclusão mútua nos acessos às bibliotecas centralizadas



[Leon, 2000] Simultaneous update problem

Evolução das técnicas de GC

- Criação de mecanismos de controle sobre as bibliotecas centralizadas
- Definição dos atuais **repositórios compartilhados**
- Aplicação, em conjunto, de **métodos e ferramentas** que suportam o processo de Gerência de Configuração
- Ganho de produtividade devido ao **aumento da disciplina e diminuição do retrabalho** e dos defeitos

Benefícios com o uso de GC

- Aumento da Memória Organizacional da empresa
- Desenvolvimento dependente do processo, e não de pessoas
- Controle sobre o desenvolvimento
- Rastreabilidade entre os diferentes níveis de abstração de um mesmo componente (requisito, análise, projeto, código, programa executável)
- Documentação sobre a evolução do sistema



Gerência de Configuração: Introdução

Leonardo Gresta Paulino Murta

leomurta@ic.uff.br