

# Gerência da Configuração

# Plano da Disciplina

# Plano da Disciplina

- Ementa

- Compreender técnicas e metodologias para o gerenciamento de configuração de projetos;
- Contribuir para a elaboração de um plano de gerência de configuração;
- Gerenciar a configuração de módulos de software usando ferramentas.

# Plano da Disciplina

<b>UNIDADE I</b>
INTRODUÇÃO A GERÊNCIA DE CONFIGURAÇÃO
RELEVÂNCIA DA GERÊNCIA DE CONFIGURAÇÃO PARA OS PROJETOS
PAPEL DO GERENTE DE CONFIGURAÇÃO
IDENTIFICAÇÃO DOS ITENS DE CONFIGURAÇÃO
IDENTIFICAÇÃO DOS ARTEFATOS DO PROJETO
VERSIONAMENTO DOS ITENS DE CONFIGURAÇÃO E ARTEFATOS DO PROJETO
CONTROLE DE CONFIGURAÇÃO - LABELS OU RÓTULOS
COMITÊ DE MUDANÇAS
<b>UNIDADE II</b>
GERAÇÃO DE BASELINES
CRIAÇÃO DE RELEASES DE UM PROJETO
GERENCIAMENTO DE REQUISIÇÕES DE MUDANÇAS
CICLO DE VIDA DAS SOLICITAÇÕES DE MUDANÇA
PROCEDIMENTOS PARA MUDANÇAS
AUDITORIA DE CONFIGURAÇÃO
PLANO DE CONTIGÊNCIA
FERRAMENTAS PARA O GERENCIAMENTO DE MUDANÇAS

- Avaliação – 1º N.I. – 03/04
  - Prova
    - Pontuação: 10,0 pts
- Avaliação – 2º N.I. – 05/06
  - Prova Colegiada
    - Pontuação: 10,0 pts

- Avaliação – 2º Chamada – 12/06
  - Prova Objetiva
    - Pontuação: 10,0 pts
- Avaliação – Final – 19/06
  - Prova Objetiva
    - Pontuação: 10,0 pts

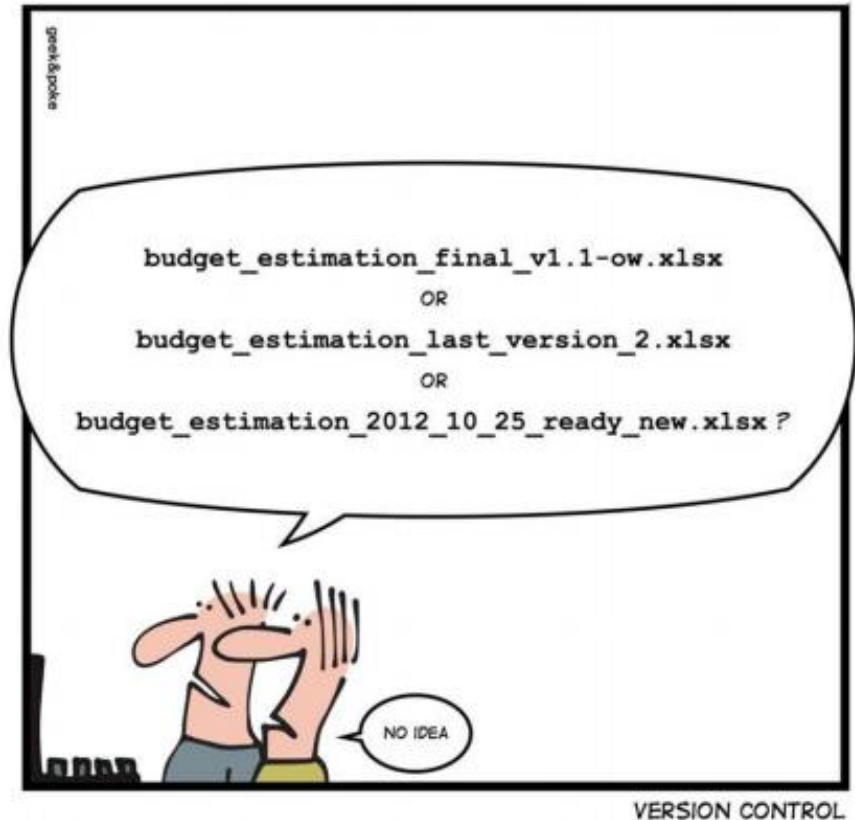
# Introdução

# Introdução

## "FINAL".doc
























## SIMPLY EXPLAINED



Idea from Jen Simmons and John Albin Wilkins during episode #40 of "Web Ahead" about Git:  
<http://5by5.tv/webahead/40>










# Introdução

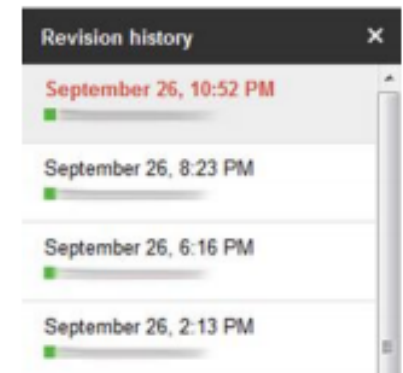
 qualificacao-doutorado05.01.2016.zip  
 qualificacao-doutorado06.01.2016.zip  
 qualificacao-doutorado07.01.2016.zip  
 qualificacao-doutorado08.01.2016.zip  
 qualificacao-doutorado09.01.2016.zip  
 qualificacao-doutorado10.01.2016.zip  
 qualificacao-doutorado11.01.2016m.zip  
 qualificacao-doutorado11.01.2016n.zip  
 qualificacao-doutorado12.01.2016n.zip  
 qualificacao-doutorado13.01.2016.zip  
 qualificacao-doutorado14.01.2016.zip  
 qualificacao-doutorado15.01.2016m.zip  
 qualificacao-doutorado15.01.2016noite.zip  
 qualificacao-doutorado15.01.2016t.zip  
 qualificacao-doutorado16.01.2016m.zip  
 qualificacao-doutorado17.01.2016m.zip  
 qualificacao-doutorado18.01.2016m.zip  
 qualificacao-doutorado19m.zip  
 qualificacao-doutorado20m.zip  
 qualificacao-doutorado20n\_aplicadaRevSIMONE.zip  
 qualificacao-doutorado22.01.2016n.zip



**Dropbox**

 Crash_Course_on_GIT_and_GitHub	29-Apr-15 23:55
 Crash_Course_on_GIT_and_GitHub	30-Apr-15 01:11
 Crash_Course_on_GIT_and_GitHub_rev01	30-Apr-15 09:11
 Crash_Course_on_GIT_and_GitHub_rev02	30-Apr-15 13:27
 Crash_Course_on_GIT_and_GitHub_rev03	30-Apr-15 20:29
 Crash_Course_on_GIT_and_GitHub_rev04	05-May-15 13:36
 Crash_Course_on_GIT_and_GitHub_rev05	07-May-15 19:21

Google docs



# ...e no desenvolvimento de software?

- Mudanças:
  - não são **analisadas** antes de ser feitas
  - não são **registradas** antes de ser implementadas
  - não são **relatadas** àqueles que precisam saber
  - ou não são **controladas** de forma que melhore a qualidade e reduza erros



# Gerenciamento de Configuração de Software (GCS)

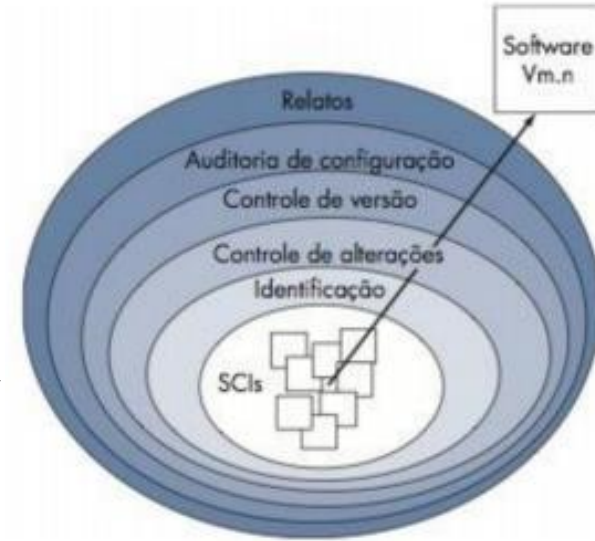
- Durante o desenvolvimento do software queremos saber:
  - O que mudou e quando? (controle de versão)
  - Por que mudou? (controle de mudanças)
  - Quem fez a mudança? (auditoria de configuração)
  - Podemos reproduzir esta mudança? (auditoria de configuração)

# Gerenciamento de Configuração de Software

- Coordenar o desenvolvimento para minimizar a confusão!
  - “Arte de *identificar, organizar e controlar* modificações no software que está sendo criado, maximizando a produtividade e reduzindo os erros” [Pressman, 2011]

# Gerenciamento de Configuração de Software

- As atividades são desenvolvidas para:
  - Identificar a alteração
  - Controlar a alteração
  - Assegurar que a alteração seja implementada corretamente
  - Relatar as alterações aos outros interessados



- Problemas pela falta de GCS.
  - 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.
  - Requisitos já documentados desaparecem.
  - Requisitos implementados desaparecem do código.
  - O programa em execução e o seu código fonte estão em diferentes versões

# Gerenciamento de Configuração de Software

- Benefícios do 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 (requisito, análise, projeto, código, programa executável).
  - Documentação sobre a evolução do sistema.

# O que é uma Configuração?

- Um projeto de desenvolvimento de software produz os seguintes itens:
  - Programas (código fonte, programas executáveis, bibliotecas de componentes, etc.)
  - Documentação (manuais do usuário, documento de requisitos, modelo de análise e projeto, etc.)
  - Dados (dados de teste e do projeto)
- Esses conjuntos de itens são chamados, coletivamente, de configuração do software



# Outros conceitos

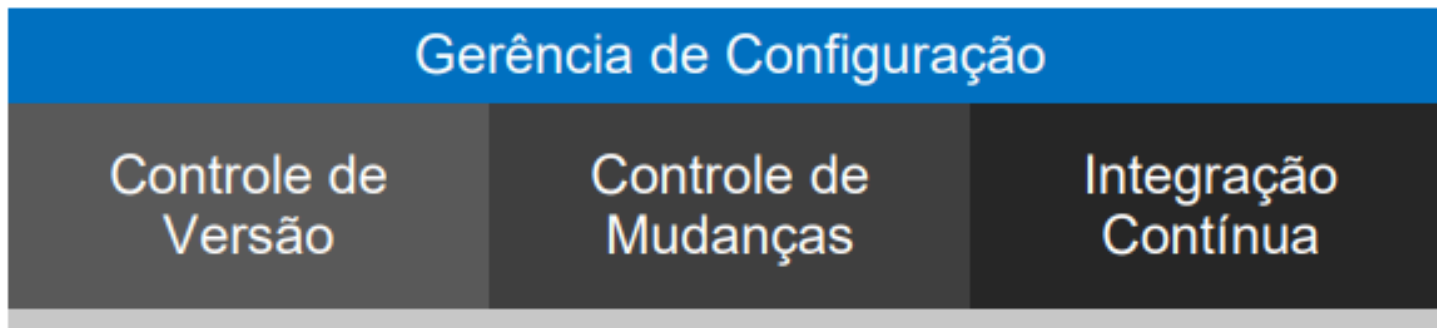
- **Item de configuração:** Elemento unitário ou um grupo de elementos para efeito de controle de versão.
  - Código
  - Documentação
  - Diagramas, planos, ferramentas, casos de teste e etc
- Um item de configuração está sujeito a mudanças e essas devem obedecer às políticas estabelecidas.
- Normalmente, um item de configuração é estabelecido para cada pedaço de software que pode ser projetado, implementado e testado de forma independente.

# Outros conceitos

- **Baseline:** configuração formalmente aprovada que servirá de referência para desenvolvimento posterior.
- A configuração do software em um ponto discreto no tempo.
- Quando um conjunto de artefatos de software se torna um item de configuração?

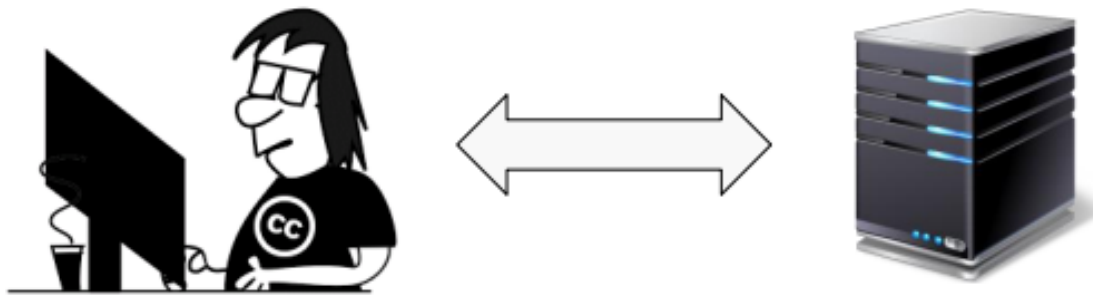


# Qual a importância do controle de versões?



# Problema exemplo

- Você precisa editar um site hospedado em um servidor:
  - 1) Você faz o download via FTP
  - 2) Faz as alterações necessárias
  - 3) Envia os arquivos alterados para o servidor via FTP



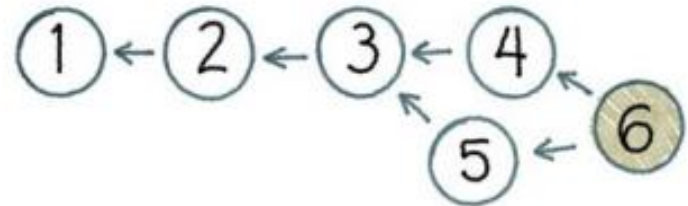
# Problema exemplo

- Agora outro desenvolvedor também deve fazer alterações no site...
  - 1) Baixa o mesmo arquivo junto com você
  - 2) Edita e manda para o servidor depois de você
  - 3) Sobrescreve suas alterações!!!



# Controle de Versões

- Principais objetivos do Controle de Versões do Sistema [Sink, 2011]
  - Trabalhar simultaneamente
  - Evitar conflitos em alterações
  - Manter versões



# Controle de Mudanças

- **Objetivo:** Garantir que todas as mudanças ocorram de maneira controlada e otimizada
  - Manter histórico de mudança
  - Justificar mudança
- **Exemplo:**
  - a) Mudanças da versão 2.2 para a versão 2.3:
    - correção do defeito D345
    - correção do defeito D346
    - adicionada a funcionalidade do RF44
  - b) Pendências para uma versão posterior:
    - melhorar usabilidade da interface I43

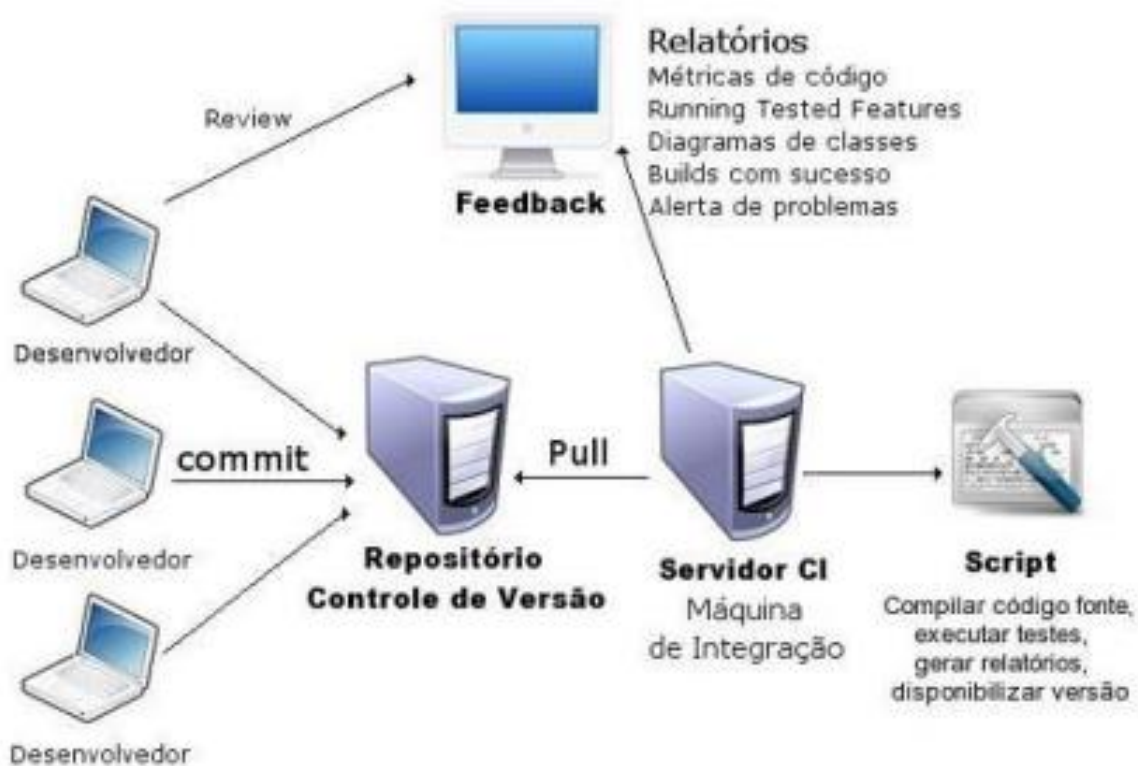


# Integração contínua

- Uma das práticas mais importantes do **desenvolvimento ágil!**
- Agilizar tarefas demoradas como a compilação de um projeto e a execução dos seus testes automatizados.
- Tarefas são executadas a cada mudança no repositório de código e, em caso de erros de compilação ou falhas nos testes automatizados, todos os desenvolvedores são alertados rapidamente



# Integração contínua



# Ferramentas de apoio

Tipo de Ferramenta	Open Source	Comercial
Controle de Versão	Mercurial   <b>Git</b>   Subversion   CVS	Team Foundation Server - Microsoft   Team Concert - IBM/Rational   ClearCase   StarTeam   Perforce   BitKeeper
Controle de Mudança	Trac   <b>Redmine</b>   Mantis   Bugzilla	JIRA   FogBUGZ   CaliberRM   Perforce
Integração Contínua	<b>Jenkins</b>   Bitten   Scons   Ant   Maven   CruiseControl   Gump   TinderBox	AntHill Pro   FinalBuilder   BuildForge

# Dúvidas??

