

Gerência de Configuração

Gilmar Ferreira Arantes¹

¹FASAM - Faculdade Sul-Americana

Especialização em Engenharia de Software

Agenda

1 Gerência de Configuração

Definições Iniciais

Conceitos e Terminologia

Controle de Modificações

Ferramentas de Apoio

GCS - Modelos de Maturidade

2 Referências

Gerência de Configuração

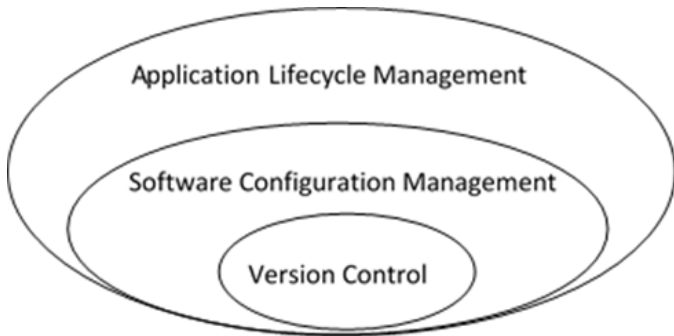


Figure : 1

Ciclo de Vida da Gerência de Configuração

Fonte: [Ebichondo, 2013].

EAP - GCS

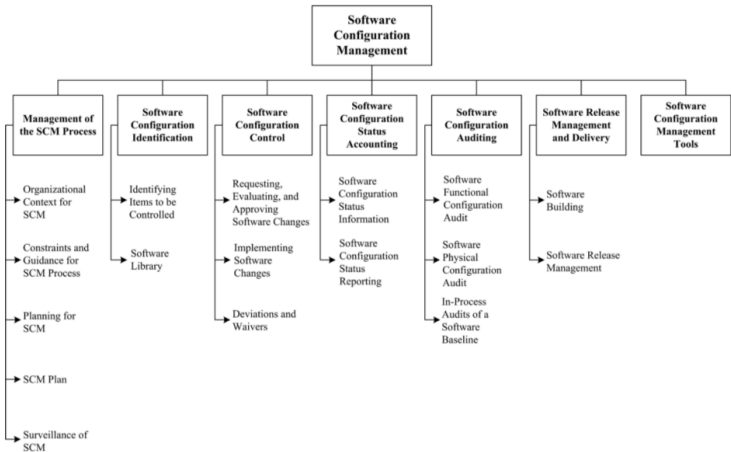


Figure 6.1. Breakdown of Topics for the Software Configuration Management KA

Fonte: [SWEBOK 2010].

Gerência de Configuração

Baseado em [Boller Filho et al, 2010].

Mudanças

Ao longo do ciclo de vida de um software, diversas modificações podem ocorrer em seu projeto original. Os motivos e origens destas modificações podem ser os mais variados possíveis e podem ocorrer em qualquer época.

Gerência de Configuração

A primeira lei da engenharia de sistemas diz:

“Independente de onde você está no ciclo de vida de um sistema, o sistema vai se modificar e o desejo de modificá-lo vai persistir ao longo de todo o ciclo de vida”.
Bersoff (1980 apud [Pressman, 2006], p. 600).

Gerência de Configuração

Existem quatro fontes fundamentais de modificações de software:

- 1 Novas condições de negócio e/ou mercado, modificam regras de negócio.

Gerência de Configuração

Existem quatro fontes fundamentais de modificações de software:

- 1 Novas condições de negócio e/ou mercado, modificam regras de negócio;
- 2 Novas necessidades do cliente exigem modificações de funcionalidades.

Gerência de Configuração

Existem quatro fontes fundamentais de modificações de software:

- 1 Novas condições de negócio e/ou mercado, modificam regras de negócio;
- 2 Novas necessidades do cliente exigem modificações de funcionalidades;
- 3 Reorganização e/ou crescimento/diminuição do negócio.

Gerência de Configuração

Existem quatro fontes fundamentais de modificações de software:

- 1 Novas condições de negócio e/ou mercado, modificam regras de negócio;
- 2 Novas necessidades do cliente exigem modificações de funcionalidades;
- 3 Reorganização e/ou crescimento/diminuição do negócio;
- 4 Restrições de orçamento ou cronograma, causando redefinição do sistema.

Gerência de Configuração de Software

- GCS

É uma atividade abrangente que ocorre ao longo de todo o ciclo de vida de um software e que gerencia e controla sua evolução, através do controle de versões e solicitações de mudanças, permitindo que os diversos envolvidos na sua criação e manutenção tenham acesso ao histórico destas modificações, fornecendo-lhes subsídios para o entendimento do sistema na sua forma atual, e também nas suas formas anteriores.

Atividades da GCS

- 1 Identificar modificações;
- 2 Controlar modificações;
- 3 Garantir a implementação adequada das modificações;
- 4 Relatar as modificações a outros que possam ter interesse.

Duração da GCS

Segundo ([Pressman, 2006], p. 599), diferentemente da atividade de **suporte de software**, que ocorre depois que o software foi entregue ao cliente, a **GCS** é um conjunto de atividades de acompanhamento e controle que começam quando o projeto de engenharia de software tem início e só terminam quando o software é retirado de operação.

Importância da GCS

A Gerência de Configuração de Software tem especial importância na **garantia da qualidade do software** e no apoio a gestão de projetos, sendo imprescindível sua aplicação nas empresas que desejam obter a certificação do Capability Maturity Model Integration (**CMMI**) **nível 2** ou na obtenção da certificação do modelo para Melhoria de Processo do Software Brasileiro (**MPSBR**) **nível F**.

Processo da GCS

Segundo ([Pressman, 2006], p. 606), o processo de gestão de configuração de software define uma série de tarefas que têm quatro objetivos principais:

- 1 Identificar todos os itens de configuração;
- 2 Gerir modificações em um ou mais destes itens;
- 3 Facilitar a construção de diferentes versões de um mesmo produto;
- 4 Garantir que a qualidade do software seja mantida a medida que a configuração evolui ao longo do tempo.

Configuração de Software

Diversos elementos são gerados como produto de um processo de software. Estes elementos podem ser:

- 1 Programas de computador (tanto na forma de código fonte como executável);
- 2 Documentos que podem descrever programas, procedimentos regras de negócio etc. (Documentação técnica e de usuário final);
- 3 Estrutura de dados que podem estar contidos em programas ou externos a ele;

Itens de Configuração de Software

Segundo ([Molinari, 2007], p. 44), um item de configuração é o menor item de controle num processo de GCS.

Um item de configuração de Software (ICS) ou *Software Configuration Item* (SCI) é cada elemento criado durante o processo de engenharia de software, ou que para este processo seja necessário. Pode ser um arquivo, uma aplicação corporativa, uma parte de um documento, uma seqüência de casos de teste, um hardware ou um componente de programa. Além dos ICS's gerados no processo de engenharia de software, também poderão ser considerados como ICS's ferramentas de software como editores, compiladores, navegadores e outras ferramentas que forem necessárias à correta geração do software.

Item de Configuração de Software

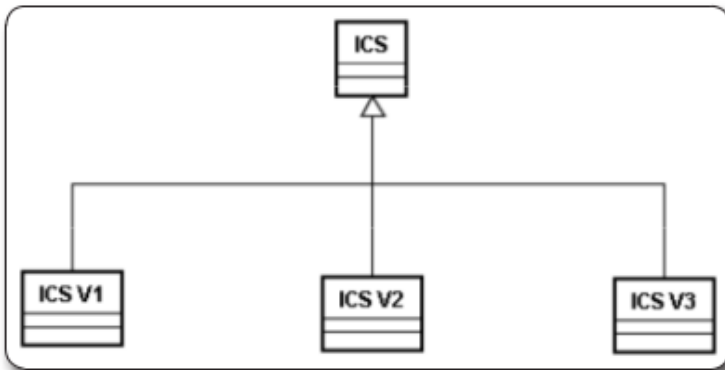


Figure : 2

Item de Configuração

Fonte: [Boller Filho et al, 2010].

Centralized version control

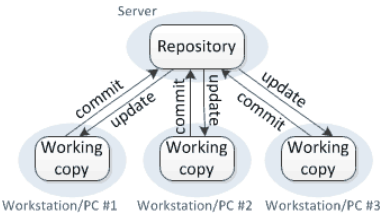


Figure : 3

Controle de versão centralizado
Fonte: [Ernst, 2015].

Repositório

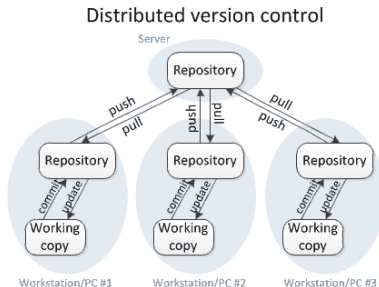


Figure : 4

Controle de versão distribuído

Fonte: [Ernst, 2015].

Repositório

Segundo Forte (1989 apud [Pressman, 2006], 2006, p. 604), o repositório é o conjunto de mecanismos e estruturas que permite a uma equipe de software gerir modificação de modo efetivo. Fornece as funções óbvias de um sistema de gestão de banco de dados, mas, além disso executa ou propicia as seguintes funções:

Repositório

- **Integridade de dados** – Valida entradas no repositório, garante consistência entre os ICS's relacionados, executa modificações em cascata quando uma modificação em um objeto exige modificações em objetos a ele relacionados;

- **Compartilhamento de informações** – Fornece mecanismo para compartilhar a informação entre vários desenvolvedores e ferramentas, controla o acesso aos ICS's por diferentes usuários através de política de bloqueio e desbloqueio de modo que as modificações não sejam sobrepostas umas as outras;

Repositório

- **Integração de ferramenta** – Provê uma estrutura de dados que permite acesso à várias ferramentas de engenharia de software;

Repositório

- **Integração de dados** – Fornece funções de banco de dados que permitem que várias tarefas de GCS sejam executadas em um ou mais ICS's;

Repositório

- **Imposição de metodologia** – Define um modelo entidade relacionamento que implica um modelo de processo específico de engenharia de software. O relacionamento entre objetos definem um conjunto de passos a ser seguido para construir o conteúdo do repositório;

- **Padronização de documentação** – Normalização para a criação de documentos de engenharia de software.

Repositório

Os ICS's são retirados e inseridos nos repositórios, através das seguintes operações:

- 1 **LOCK**: Garante que apenas um usuário detém o acesso para alterar um determinado ICS. Apesar de resolver o problema de atualização simultânea, o lock serializa o trabalho dos desenvolvedores.
- 2 **CHECK-OUT**: Recupera a última versão de um item de configuração guardado no repositório, copiando para a área de trabalho do desenvolvedor.
- 3 **CHECK-IN**: Insere ou atualiza um item de configuração no repositório, incrementando a versão do ICS, e fazendo o registro das informações de mudança.

Identificação dos ICS's.

- 1 A identificação dos ICs é uma área de extrema importância dentro da Gerência de Configuração de Software. Sem uma identificação correta de um item, é impossível gerenciar este processo.
- 2 Cada ICS deve ser identificado através de características distintas que o caracterizam unicamente. Estas características podem ser um nome, uma descrição, uma lista de recursos etc. É necessário que cada organização defina suas convenções de identificação.

Baseline

- 1 Um *baseline* pode ser descrito como a situação de uma coleção de ICS's similares em um momento específico do ciclo de vida de um software que foram aprovados e armazenados em uma biblioteca controlada.
- 2 Pode ser descrito também como a conexão de um item de configuração com um determinado marco no projeto (*milestone*). Um *baseline* pode ser visto como um conjunto de itens de configuração identificados e liberados para uso, independente de suas versões.

O desenvolvimento com *baselines* pode, então, ser resumido nos seguintes pontos:

Baseline

- Caracterização do ciclo de vida, identificando-se as fases pelas quais o desenvolvimento do software irá passar;

- Definição do conjunto de *baselines*, estabelecendo-se quais serão os ICS's que a irão compor;

Baseline

- Estabelecer o marco, que a *baseline* irá representar. Uma nova *baseline* é estabelecida no final de cada fase do ciclo de vida do software;

Baseline

- Durante cada fase, a modificação de uma configuração base somente pode ser feita de forma controlada, mediante um processo bem definido;

Baseline

- Ao ser estabelecida, cada *baseline* incorpora integralmente a anterior.

Baseline

- O estabelecimento de cada *baseline* somente é realizado após ser aprovada por procedimentos de consistência interna, verificação e validação;

Item de Configuração	V1.1	V1.2	V1.3	V1.4	V1.5	Baseline At	Ultima versão de cada item de configuração
IC01	?	☀					V1.2
IC02	?	?	☀				V1.3
IC03	☀						V1.1
IC04	?	?	?	?	☀		V1.5
IC05	?	?	?	☀			V1.4

Figure : 5

Linha de Base

Fonte: [Boller Filho et al, 2010], Adaptada de [Molinari, 2007].

Controle de Versão



Figure : 6

Controle de Versões
Fonte: Google Images.

Controle de Versão

Segundo ([Pressman, 2006], p. 608), controle de versões combina procedimentos e ferramentas para gerir diferentes versões de objetos de configuração que são criados durante o processo de software. Um sistema de controle de versão implementa ou está diretamente integrado com quatro capacidades principais:

Controle de Versão

- Um banco de dados de projeto (repositório) que guarda todos os objetos de configuração relevantes.

Controle de Versão

- Um banco de dados de projeto (repositório) que guarda todos os objetos de configuração relevantes;
- Uma capacidade de gestão de versão que guarda todas as versões de um objeto de configuração.

Controle de Versão

- Um banco de dados de projeto (repositório) que guarda todos os objetos de configuração relevantes;
- Uma capacidade de gestão de versão que guarda todas as versões de um objeto de configuração;
- Uma facilidade de construir que permite ao engenheiro de software coletar todos os objetos de configuração relevantes e construir uma versão específica do software;

Controle de Versão

- Um banco de dados de projeto (repositório) que guarda todos os objetos de configuração relevantes;
- Uma capacidade de gestão de versão que guarda todas as versões de um objeto de configuração;
- Uma facilidade de construir que permite ao engenheiro de software coletar todos os objetos de configuração relevantes e construir uma versão específica do software;
- Capacidade de acompanhamento de tópicos (também chamado de acompanhamento de bugs).

Controle de Versão

Segundo ([Molinari, 2007], p.168), independente do que cada um pensa sobre controle de versões, algumas das suas características perceptíveis são:

Controle de Versão

- 1 Mantém e disponibiliza cada versão já produzida de item do projeto.

Controle de Versão

- 1 Mantém e disponibiliza cada versão já produzida de item do projeto;
- 2 Possui mecanismos para gerenciar diferentes ramos de desenvolvimento possibilitando a existência de diferentes versões de maneira simultânea.

Controle de Versão

- 1 Mantém e disponibiliza cada versão já produzida de item do projeto;
- 2 Possui mecanismos para gerenciar diferentes ramos de desenvolvimento possibilitando a existência de diferentes versões de maneira simultânea;
- 3 Estabelece uma política de sincronização de mudanças que evita a sobreposição de mudanças.

Controle de Versão

- 1 Mantém e disponibiliza cada versão já produzida de item do projeto;
- 2 Possui mecanismos para gerenciar diferentes ramos de desenvolvimento possibilitando a existência de diferentes versões de maneira simultânea;
- 3 Estabelece uma política de sincronização de mudanças que evita a sobreposição de mudanças;
- 4 **Fornece um histórico completo de alterações sobre cada item do projeto.**

Política de Versionamento

Trava-Modifica-Destrava:

Apenas uma pessoa por vez detém o direito de alterar um determinado arquivo. O sistema trava o arquivo quando uma pessoa retira uma cópia para manutenção e só libera a cópia para edição à outra pessoa, quando a primeira publica sua nova versão. Esta política é limitada e pode causar problemas administrativos e forçar a serialização do trabalho, entretanto, é a solução indicada para organizações que precisam de um alto grau de formalização.

Política de Versionamento

Copia-Modifica-Resolve:

Não utiliza travamento de arquivos. Nesta política, cada desenvolvedor trabalha de forma independente em sua cópia de trabalho, e ao final, as alterações são mescladas no repositório, formando a versão final. Esta política pode parecer caótica, mas na prática funciona bem, devido ao baixo índice de conflitos. Os conflitos são causados na maioria das vezes, por falta de comunicação entre os desenvolvedores.

Branch

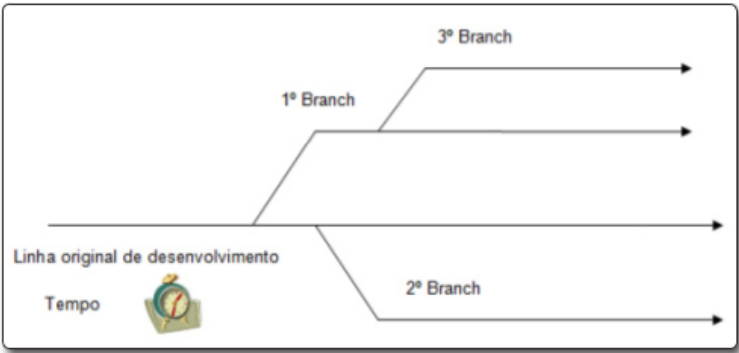


Figure : 7

Branch

Fonte: [Boller Filho et al, 2010].

Branch

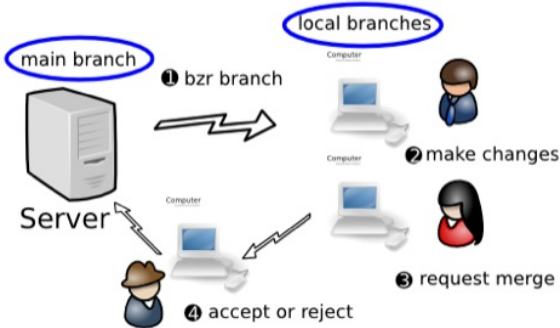


Figure : 8

Ciclo de Vida de um Branch
Fonte: Google Images.

Branch

Branching (ou ramificação em português) é uma parte fundamental do controle de versões. O conceito de branching reside em se iniciar uma nova linha de desenvolvimento em paralelo à uma principal já existente.

Esta técnica é extremamente útil em situações onde temos que fazer grandes alterações em objetos de configuração e que provavelmente levarão muito tempo, sem contudo afetar a linha principal de manutenção do sistema. Nesta situação, cria-se uma segunda cópia do objeto em questão e trabalha-se em paralelo à linha principal.

Merge

O Merge faz a fusão de dois arquivos que estão sendo alterados simultaneamente por desenvolvedores diferentes garantindo que a versão final contenha todas as alterações.

- 1 Um outro tipo de controle de versão são as tags.
- 2 Tags são como fotografias de um projeto, em um determinado instante do tempo.
- 3 São rótulos associados a um conjunto de arquivos usados para denominar um projeto ou uma nova versão, e que rotulam estes arquivos.
- 4 As tags nas versões de um conjunto de itens de configuração, existente em diversos sistemas de controle de versões, pode ser utilizado para implementar o conceito de baseline.

Necessidade de se controlar

- Mudanças são inevitáveis.
- Todo produto de software sofre mudanças durante o seu ciclo de vida, seja por erro de construção, ou por necessidade de mudanças solicitadas pelos clientes.
- Em projetos de grande porte, fazer modificações sem um esquema formal de controle pode levar ao caos.
- Alguns dos problemas que podem ser causados em projetos devido a mudanças não controladas são:

Problemas

- Aumento do custo do projeto;
- Atrasos em entregas planejadas;
- Impacto em outros objetos de configuração;
- Degradação da qualidade do software;
- Retrabalho.

Processo Formal

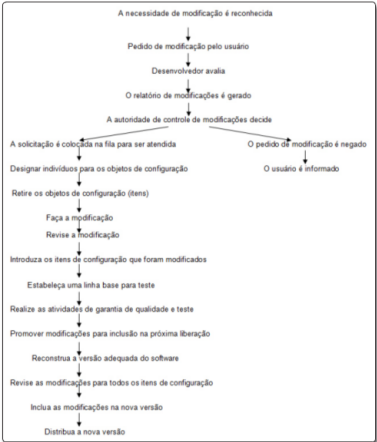


Figure : 9

Processo de Controle de Mudanças
Fonte: [Pressman, 2006].

Auditoria de Configuração

Além dos controles de identificação, versão e modificação vistos anteriormente, precisamos implementar algum mecanismo que nos garanta que as modificações foram feitas de forma adequada. Este mecanismo envolve dois aspectos:

Auditoria de Configuração

- 1 Revisões técnicas formais;
- 2 Auditoria de configuração de software.

Revisão Técnica Formal

- 1 A revisão técnica tem como foco a correção técnica do ICS que foi modificado a partir de uma requisição.
- 2 Revisores avaliam o ICS para determinar se o mesmo é consistente.
- 3 Revisões técnicas formais normalmente são feitas para todas as modificações, exceto as mais triviais.

Auditoria de Configuração

A auditoria de configuração de software tem como objetivo complementar a revisão técnica formal e trata dos seguintes aspectos:

- 1 Verifica se a modificação técnica especificada foi realizada e se foi incorporada alguma modificação adicional;
- 2 Verifica se uma revisão técnica formal foi feita para avaliar a correção técnica;
- 3 Verifica se o processo de software foi seguido;
- 4 Verifica se constam anotados no ICS modificado, dados de rastreabilidade tais como data da modificação, autor, no de requisição etc.;
- 5 Verifica se os procedimentos de GCS para registrar e relatar a modificação, foram seguidos;
- 6 Verifica se todos os ICS's relacionados foram atualizados de forma adequada.

Relatório de Estado

A geração dos relatórios de estado é uma tarefa de GCS que tem por objetivo responder às seguintes questões:

- 1 O que aconteceu?
- 2 Quem fez?
- 3 Quando aconteceu?
- 4 O que mais será afetado?

Ferramentas

Tipo de ferramenta	Nome
Controle de Versão	CVS
	SubVersion
Controle de mudança	Mantis
	Bugzilla
Integração (Build)	Scons

Figure : 10

Ferramentas de Apoio
Fonte: [Boller Filho et al, 2010].



Figure : 11

CVS

Fonte: Google Images.

CVS - Limitações

- 1 arquivos em um repositório CVS não podem ser renomeados, eles devem ser explicitamente removidos e readicionados.

CVS - Limitações

- 1 Os arquivos em um repositório CVS não podem ser renomeados, eles devem ser explicitamente removidos e readicionados;
- 2 O protocolo do CVS não permite que os diretórios sejam movidos ou renomeados. Cada arquivo do subdiretório em questão deve ser individualmente removido e readicionado.

CVS - Limitações

- 1 Os arquivos em um repositório CVS não podem ser renomeados, eles devem ser explicitamente removidos e readicionados;
- 2 O protocolo do CVS não permite que os diretórios sejam movidos ou renomeados. Cada arquivo do subdiretório em questão deve ser individualmente removido e readicionado;
- 3 Não permite “lock” (permite que dois usuários alterem o mesmo arquivo ao mesmo tempo) e em alguns casos pode ser mais custoso resolver o conflito do que evitar que ele ocorra.

Subversion



Figure : 12

Subversion

Fonte: Google Images.

Subversion

O Subversion é bem conhecido na comunidade que utiliza código aberto e é utilizado em diversos projetos incluindo Apache Software Foundation, KDE, Free Pascal, FreeBSD, GCC Python, Django, Ruby, Mono, SourceForge.net, ExtJS e Tigris.org. O Google Code também faz a hospedagem de seus projetos de código aberto utilizando o Subversion.

Algumas de suas facilidades são:

Subversion

- 1 Operações de “Commit” são atômicas. Interrupções nestas operações não causam inconsistências no repositório;

Subversion

- 1 Operações de “Commit” são atômicas. Interrupções nestas operações não causam inconsistências no repositório;
- 2 Mudanças de nomes, cópias, e movimentações de arquivos são guardadas como histórico de revisões;

Subversion

- 1 Operações de “Commit” são atômicas. Interrupções nestas operações não causam inconsistências no repositório;
- 2 Mudanças de nomes, cópias, e movimentações de arquivos são guardadas como histórico de revisões;
- 3 Mudanças de nomes em diretórios são guardados também como versões. Árvores de diretórios inteiras podem ser movidas ou copiadas com rapidez e tem estas operações guardadas no histórico de revisões;

Subversion

- 1 Operações de “Commit” são atômicas. Interrupções nestas operações não causam inconsistências no repositório;
- 2 Mudanças de nomes, cópias, e movimentações de arquivos são guardadas como histórico de revisões;
- 3 Mudanças de nomes em diretórios são guardados também como versões. Árvores de diretórios inteiras podem ser movidas ou copiadas com rapidez e tem estas operações guardadas no histórico de revisões;
- 4 **Servidor Apache HTTP como servidor de rede;**

Subversion

- 1 Operações de “Commit” são atômicas. Interrupções nestas operações não causam inconsistências no repositório;
- 2 Mudanças de nomes, cópias, e movimentações de arquivos são guardadas como histórico de revisões;
- 3 Mudanças de nomes em diretórios são guardados também como versões. Árvores de diretórios inteiras podem ser movidas ou copiadas com rapidez e tem estas operações guardadas no histórico de revisões;
- 4 Servidor Apache HTTP como servidor de rede;
- 5 Operações de Branch e Tag são menos custosas, independente do tamanho do arquivo. O Subversion não faz distinção entre Tag, Branch e um diretório;

Subversion

- 1 Operações de “Commit” são atômicas. Interrupções nestas operações não causam inconsistências no repositório;
- 2 Mudanças de nomes, cópias, e movimentações de arquivos são guardadas como histórico de revisões;
- 3 Mudanças de nomes em diretórios são guardados também como versões. Árvores de diretórios inteiras podem ser movidas ou copiadas com rapidez e tem estas operações guardadas no histórico de revisões;
- 4 Servidor Apache HTTP como servidor de rede;
- 5 Operações de Branch e Tag são menos custosas, independente do tamanho do arquivo. O Subversion não faz distinção entre Tag, Branch e um diretório;
- 6 Permite operações de “lock” para arquivos que não podem ser fundidos (sofrer operação de “merge”).



Figure : 13

Mantis Bug Tracker
Fonte: Google Images

Mantis é um sistema de controle de mudança open source, baseado em Web, bastante popular. Foi escrito em linguagem PHP e trabalha com banco de dados MySQL, MS SQL e PostgreSQL e um webserver. Pode ser instalado em Windows, Linux, Mac OS, OS/2 e outros e funciona na maioria dos web browsers como um cliente.

Algumas de suas características são:

- É um software livre (licença GPL);

Algumas de suas características são:

- É um software livre (licença GPL);
- **Fácil de instalar;**

Algumas de suas características são:

- É um software livre (licença GPL);
- Fácil de instalar;
- **Fácil de avaliar;**

Algumas de suas características são:

- É um software livre (licença GPL);
- Fácil de instalar;
- Fácil de avaliar;
- **É baseado em Web;**

Algumas de suas características são:

- É um software livre (licença GPL);
- Fácil de instalar;
- Fácil de avaliar;
- É baseado em Web;
- **Suporta qualquer plataforma que roda PHP;**

Algumas de suas características são:

- É um software livre (licença GPL);
- Fácil de instalar;
- Fácil de avaliar;
- É baseado em Web;
- Suporta qualquer plataforma que roda PHP;
- **Suporta múltiplos projetos por instancia;**

Algumas de suas características são:

- É um software livre (licença GPL);
- Fácil de instalar;
- Fácil de avaliar;
- É baseado em Web;
- Suporta qualquer plataforma que roda PHP;
- Suporta múltiplos projetos por instancia;
- **Suporte para Projetos, Subprojetos e categorias;**

Algumas de suas características são:

- É um software livre (licença GPL);
- Fácil de instalar;
- Fácil de avaliar;
- É baseado em Web;
- Suporta qualquer plataforma que roda PHP;
- Suporta múltiplos projetos por instancia;
- Suporte para Projetos, Subprojetos e categorias;
- **Suporte a diferentes níveis de acesso de usuário por projetos;**

Algumas de suas características são:

- É um software livre (licença GPL);
- Fácil de instalar;
- Fácil de avaliar;
- É baseado em Web;
- Suporta qualquer plataforma que roda PHP;
- Suporta múltiplos projetos por instancia;
- Suporte para Projetos, Subprojetos e categorias;
- Suporte a diferentes níveis de acesso de usuário por projetos;
- **Suporta log de mudanças;**

Algumas de suas características são:

- É um software livre (licença GPL);
- Fácil de instalar;
- Fácil de avaliar;
- É baseado em Web;
- Suporta qualquer plataforma que roda PHP;
- Suporta múltiplos projetos por instancia;
- Suporte para Projetos, Subprojetos e categorias;
- Suporte a diferentes níveis de acesso de usuário por projetos;
- Suporta log de mudanças;
- **Possui diversos relatórios e gráficos;**

Algumas de suas características são:

- É um software livre (licença GPL);
- Fácil de instalar;
- Fácil de avaliar;
- É baseado em Web;
- Suporta qualquer plataforma que roda PHP;
- Suporta múltiplos projetos por instancia;
- Suporte para Projetos, Subprojetos e categorias;
- Suporte a diferentes níveis de acesso de usuário por projetos;
- Suporta log de mudanças;
- Possui diversos relatórios e gráficos;
- **Usuários podem monitorar as solicitações;**

Algumas de suas características são:

- É um software livre (licença GPL);
- Fácil de instalar;
- Fácil de avaliar;
- É baseado em Web;
- Suporta qualquer plataforma que roda PHP;
- Suporta múltiplos projetos por instancia;
- Suporte para Projetos, Subprojetos e categorias;
- Suporte a diferentes níveis de acesso de usuário por projetos;
- Suporta log de mudanças;
- Possui diversos relatórios e gráficos;
- Usuários podem monitorar as solicitações;
- **Envio de notificações por email;**

Algumas de suas características são:

- É um software livre (licença GPL);
- Fácil de instalar;
- Fácil de avaliar;
- É baseado em Web;
- Suporta qualquer plataforma que roda PHP;
- Suporta múltiplos projetos por instancia;
- Suporte para Projetos, Subprojetos e categorias;
- Suporte a diferentes níveis de acesso de usuário por projetos;
- Suporta log de mudanças;
- Possui diversos relatórios e gráficos;
- Usuários podem monitorar as solicitações;
- Envio de notificações por email;
- **Permite histórico de mudanças;**

Algumas de suas características são:

- É um software livre (licença GPL);
- Fácil de instalar;
- Fácil de avaliar;
- É baseado em Web;
- Suporta qualquer plataforma que roda PHP;
- Suporta múltiplos projetos por instancia;
- Suporte para Projetos, Subprojetos e categorias;
- Suporte a diferentes níveis de acesso de usuário por projetos;
- Suporta log de mudanças;
- Possui diversos relatórios e gráficos;
- Usuários podem monitorar as solicitações;
- Envio de notificações por email;
- Permite histórico de mudanças;
- **Integração com SVN e CVS.**



Figure : 14

Bugzilla. Fonte: Google Images.

Bugzilla

Bugzilla é uma ferramenta baseada em Web e email, que dá suporte ao desenvolvimento do projeto Mozilla, rastreando defeitos de software, e servindo também como plataforma para solicitações e controle de mudanças. Como projeto de software livre, é mantido por voluntários, sendo utilizado por diversos outros projetos e empresas.

Algumas de suas características são:

- Capacidade de pesquisa avançada;

Algumas de suas características são:

- Capacidade de pesquisa avançada;
- **Notificações por email controladas pelas preferências de usuários;**

Algumas de suas características são:

- Capacidade de pesquisa avançada;
- Notificações por email controladas pelas preferências de usuários;
- Lista de defeitos em diversos formatos;

Algumas de suas características são:

- Capacidade de pesquisa avançada;
- Notificações por email controladas pelas preferências de usuários;
- Lista de defeitos em diversos formatos;
- **Relatórios periódicos (diários, semanais etc.) enviados por email;**

Algumas de suas características são:

- Capacidade de pesquisa avançada;
- Notificações por email controladas pelas preferências de usuários;
- Lista de defeitos em diversos formatos;
- Relatórios periódicos (diários, semanais etc.) enviados por email;
- **Controle de tempo de execução de uma solicitação;**

Algumas de suas características são:

- Capacidade de pesquisa avançada;
- Notificações por email controladas pelas preferências de usuários;
- Lista de defeitos em diversos formatos;
- Relatórios periódicos (diários, semanais etc.) enviados por email;
- Controle de tempo de execução de uma solicitação;
- **Sistema de atribuição de tarefas;**

Algumas de suas características são:

- Capacidade de pesquisa avançada;
- Notificações por email controladas pelas preferências de usuários;
- Lista de defeitos em diversos formatos;
- Relatórios periódicos (diários, semanais etc.) enviados por email;
- Controle de tempo de execução de uma solicitação;
- Sistema de atribuição de tarefas;
- **Diversas versões de localização (incluindo Português – Brasil).**

Scons

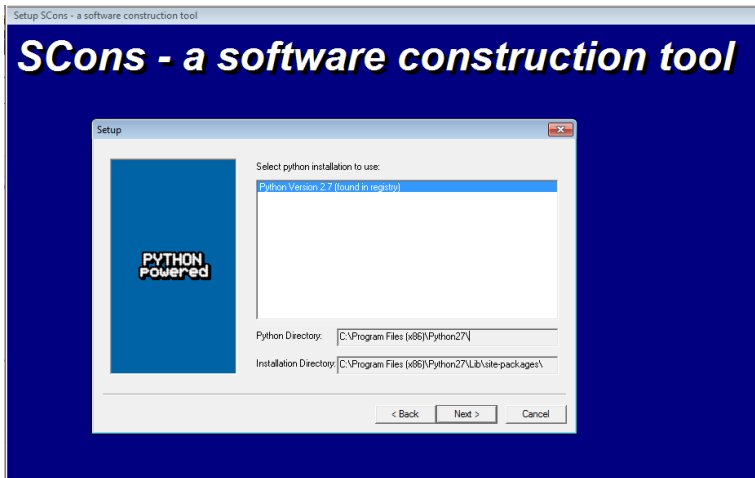


Figure : 15

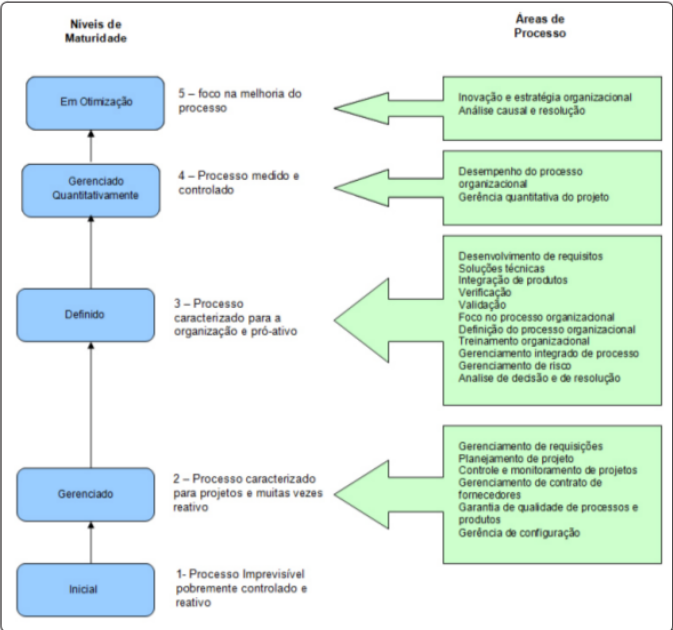
Scons. Fonte: Google Imagens

Scons - Definição

SCons é uma ferramenta de construção de software open source escrita em Python. O SCons provê uma ponte entre plataformas e é o substituto para o utilitário clássico Make com funcionalidades similares ao autoconf/automake, e possui compiladores integrados. Algumas de suas características são:

Scons - Características

- 1 Seus arquivos de configuração são scripts Python;
- 2 Análise automática de dependências, nativa para C, C++ e Fortran, extensível para outras linguagens;
- 3 Suporte nativo para adquirir arquivos fonte do CVS;
- 4 Suporte nativo ao MS Visual Studio .NET
- 5 Detecção de mudanças de construção usando assinaturas baseadas em MD5.



As **metas e práticas** específicas da gestão de configuração definidas na CMMI são:

- SG 1. Estabelecer baselines.
 - SP 1.1 Identificar itens de configuração
 - SP 1.2 Estabelecer um Sistema de Gestão de Configuração
 - SP 1.3 Criar ou liberar baselines
- SG 2. Rastrear e controlar mudanças
 - SP 2.1 Rastrear requisição de mudanças
 - SP 2.2 Controlar itens de configuração
- SG 3. Estabelecer integridade
 - SP 3.1 Estabelecer registros de Gerencia da Configuração
 - SP 3.2 Realizar auditorias de configuração

Nível	Processos
A	
B	Gerência de Projetos – GPR (evolução)
C	Gerência de Riscos – GRI Desenvolvimento para Reutilização – DRU Gerência de Decisões – GDE
D	Verificação – VER Validação – VAL Projeto e Construção do Produto – PCP Integração do Produto – ITP Desenvolvimento de Requisitos – DRE
E	Gerência de Projetos – GPR (evolução) Gerência de Reutilização – GRU Gerência de Recursos Humanos – GRH Definição do Processo Organizacional – DFP Avaliação e Melhoria do Processo Organizacional – AMP
F	Medição – MED Garantia da Qualidade – GQA Gerência de Portfólio de Projetos – GPP Gerência de Configuração – GCO Aquisição – AQU
G	Gerência de Requisitos – GRE Gerência de Projetos – GPR

Segundo definição do 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. A GCS é encarada pelo MPS.BR como sendo um processo do nível de maturidade “F” e portanto deve ter resultados esperados. Os resultados esperados da GCS são:

1 Um Sistema de Gerência de Configuração é estabelecido e Mantido;

- 1 GCO1 - Um Sistema de Gerência de Configuração é estabelecido e Mantido;
- 2 GCO2 - Os itens de configuração são identificados com base em critérios estabelecidos;

- 1 GCO1 - Um Sistema de Gerência de Configuração é estabelecido e Mantido;
- 2 GCO2 - Os itens de configuração são identificados com base em critérios estabelecidos;
- 3 GCO3 - Os itens de configuração sujeitos a um controle formal são colocados sob baseline;

- 1 GCO1 - Um Sistema de Gerência de Configuração é estabelecido e Mantido;
- 2 GCO2 - Os itens de configuração são identificados com baseem critérios estabelecidos;
- 3 GCO3 - Os itens de configuração sujeitos a um controle formal são colocados sob baseline;
- 4 GCO4 - A situação dos itens de configuração e das baselines é registrada ao longo do tempo e disponibilizada;

- 1 GCO1 - Um Sistema de Gerência de Configuração é estabelecido e Mantido;
- 2 GCO2 - Os itens de configuração são identificados com baseem critérios estabelecidos;
- 3 GCO3 - Os itens de configuração sujeitos a um controle formal são colocados sob baseline;
- 4 GCO4 - A situação dos itens de configuração e das baselines é registrada ao longo do tempo e disponibilizada;
- 5 GCO5 - Modificações em itens de configuração são controladas;

- ❶ GCO1 - Um Sistema de Gerência de Configuração é estabelecido e Mantido;
- ❷ GCO2 - Os itens de configuração são identificados com baseem critérios estabelecidos;
- ❸ GCO3 - Os itens de configuração sujeitos a um controle formal são colocados sob baseline;
- ❹ GCO4 - A situação dos itens de configuração e das baselines é registrada ao longo do tempo e disponibilizada;
- ❺ GCO5 - Modificações em itens de configuração são controladas;
- ❻ GCO6 - O armazenamento, o manuseio e a liberação de itens de configuração e baselines são controlados consistentes;

MPS-BR

- 1 GCO1 - Um Sistema de Gerência de Configuração é estabelecido e Mantido;
- 2 GCO2 - Os itens de configuração são identificados com baseem critérios estabelecidos;
- 3 GCO3 - Os itens de configuração sujeitos a um controle formal são colocados sob baseline;
- 4 GCO4 - A situação dos itens de configuração e das baselines é registrada ao longo do tempo e disponibilizada;
- 5 GCO5 - Modificações em itens de configuração são controladas;
- 6 GCO6 - O armazenamento, o manuseio e a liberação de itens de configuração e baselines são controlados consistentes;
- 7 GCO7 - Auditorias de configuração são realizadas objetivamente para assegurar que as baselines e os itens de configuração estejam íntegros, completos e consistentes

Referências Bibliográficas



Boller Filho, A.; Spínola, R. O.; Costa, M. N.; Kalinowski, M.

Gerência de Configuração - Definições Iniciais, Ferramentas e Processo

Engenharia de Software Magazine, edição 24, 2010



Ebichondo, Stephen

Application Lifecycle Management

<http://blogs.msdn.com/b/africaapps/archive/2013/05/29/>

[application-lifecycle-management-part-1-of-5.aspx](http://blogs.msdn.com/b/africaapps/archive/2013/05/29/application-lifecycle-management-part-1-of-5.aspx), acesso em 08/03/2015.



Ernst, Michael

Version control concepts and best practices

<http://homes.cs.washington.edu/~mernst/advice/version-control.html>, acesso em 08/03/2015.



IEEE Computer Society

SWEBOK Version 3.0, 2014



MOLINARI, Leonardo.

Gerência de configuração: Técnicas e práticas no desenvolvimento do software.

Florianópolis : Visual Books, 2007. 208p.



PRESSMAN, Roger S.

Engenharia de software.

6a. ed. São Paulo: McGraw-Hill, 2006. xxxi, 720p.