****

**UNIVERSIDADE DO SUL DE SANTA CATARINA**

**FABIO DELA BRUNA**

**MÁRCIO OZÓRIO TEIXEIRA**

**GERENCIAMENTO DE CONFIGURAÇÃO E MUDANÇAS**

Tubarão

2009

**SUMÁRIO**

**1 INTRODUÇÃO**......................................................................................................................2

**2 PLANEJAMENTO DE GERENCIAMENTO DE CONFIGURAÇÕES**.........................3

* 1. IDENTIFICAÇÃO DE ÍTEM DE CONFIGURAÇÃO......................................................3
  2. BANCO DE DADOS DE CONFIGURAÇÕES...................................................................4

1. **GERENCIAMENTO DE MUDANÇAS**...........................................................................5

**4 GERENCIAMENTO DE VERSÕES E RELEASES**.........................................................5

* 1. IDENTIFICAÇÃO DE VERSÕES.......................................................................................6

4.2 GERENCIAMETO DE RELEASES....................................................................................6

**5 CONSTRUÇÃO DE SISTEMAS**.........................................................................................7

**6 FERRAMENTAS PARA GERENCIAMENTO DE CONFIGURAÇÃO**........................7

6.1 BORLANDSTARTEAM.....................................................................................................7

6.2 TORTOISESVN...................................................................................................................9

**7 CONCLUSÃO**......................................................................................................................10

**REFERÊNCIAS**......................................................................................................................11

**1 INTRODUÇÃO**

Segundo Pressman (2006), a gerência de configuração de software é o conjunto de atividades projetadas para controlar as mudanças pela identificação dos produtos do trabalho que serão alterados, estabelecendo um relacionamento entre eles, definindo o mecanismo para o gerenciamento de diferentes versões destes produtos, controlando as mudanças impostas, e auditando e relatando as mudanças realizadas.

Neste trabalho serão abordadas as principais características que envolvem o gerenciamento de configurações e mudanças, assim como a importância de utilizar o mesmo para sistemas complexos.

1. **PLANEJAMENTO DE GERENCIAMENTO DE CONFIGURAÇÕES**

Segundo Sommerville (2003), o plano de gerenciamento de configurações descreve os padrões e procedimentos que devem ser usados para o gerenciamento. O ponto de partida para o desenvolvimento de plano deve ser um conjunto de padrões de configuração, que devem ser adaptados para se atender aos requisitos e às restrições de cada projeto específico. O plano de Gerenciamento de configurações deve ser organizado em um número de seções que:

1. Define o que será gerenciado (itens de configuração) e o esquema que se deve usar para identificar essas entidades.
2. Estabelece quem é responsável pelos procedimentos de gerenciamento de configuração e pela submissão de itens controlados para a equipe de gerenciamento de configuração.
3. Define as políticas de gerenciamento de configuração que todos os membros da equipe devem adotar para o controle de mudanças e gerenciamento de versões.
4. Especifica as ferramentas que se deve usar para o gerenciamento de configuração e os processos para uso dessas ferramentas.
5. Descreve a estrutura do banco de dados de configuração usada para registrar as informações de configuração e as informações que devem ser mantidas nesse banco de dados (os registros de configuração).
   1. IDENTIFICAÇÃO DE ÍTEM DE CONFIGURAÇÃO

Em um grande sistema de software, pode haver módulos de milhares de códigos-fonte, scripts de testes, documentos de projeto etc. Eles são produzidos por pessoas diferentes e, quando criado, podem ser denominados com nomes similares ou idênticos.

Durante o processo de planejamento de gerenciamento de configuração, você decide exatamente quais itens serão controlados. Planos de projeto, especificações, projetos, programas e massa de dados de teste são normalmente mantidos como itens de configuração.

O esquema de identificação de itens de configuração deve atribuir um único nome para todos os documentos sob controle de configuração. Esse nome pode refletir o tipo do item, uma parte do sistema ao qual ele se aplica, o criador do item, etc. Portanto, você pode definir um esquema de hierarquia com nomes como:

PCL-TOOLS/EDIT/FORMS/DISPLAY/AST-INTERFACE/CODE

PCL-TOOLS/EDIT/HELP/QUERY/HELPFRAMES/FR-1

A parte inicial do nome é o nome do projeto, PCL-TOOLS. Nesse projeto, há algumas ferramentas distintas em desenvolvimento, assim, o nome da ferramenta (EDIT) é usado como a próxima parte do nome. Cada ferramenta inclui módulos denominados diferentemente e cujos nomes definem o próximo componente do identificador do item (FORM, HELP). Itens como telas de auxílio são também gerenciados e têm diferentes nomes.

1. BANCO DE DADOS DE CONFIGURAÇÕES

O banco de dados de configuração é utilizado para registrar todas as informações relevantes sobre as configurações e os itens de configuração. Você usa o banco de dados para auxiliar a avaliação do impacto das mudanças de sistema e para gerar relatórios para a gerência sobre o processo de gerenciamento de controle.

De acordo com Sommerville (2003), um banco de dados de configuração não inclui apenas informações sobre itens de configuração. Pode também registrar informações sobre usuários de componentes, clientes de sistemas, plataformas de execução, mudanças propostas etc.

1. **GERENCIAMENTO DE MUDANÇAS**

Pressman (2005), explica que procedimentos de gerenciamento de mudança dizem respeito à análise de custo e benefício das mudanças propostas, à aprovação das mudanças viáveis e à rastreabilidade de quais componentes do sistema foram alterados. O processo de gerenciamento de mudança deve surtir efeito quando o software ou a documentação associada são colocados em baseline pela equipe de gerenciamento de configurações.

Durante o desenvolvimento, quando novas versões do sistema são criadas por meio de construções diárias de sistemas, é usado um processo de gerenciamento de mudanças simples. Problemas e mudanças devem ser registrados, mas as mudanças que afetam somente componentes e módulos individuais não necessitam ser avaliadas independentemente. Elas são passadas diretamente para o desenvolvedor do sistema. O desenvolvedor do sistema pode aceitá-las ou justificar por que elas não são necessárias.

1. **GERENCIAMENTO DE VERSÕES E RELEASES**

Segundo Sommerville (2003), os processos envolvidos no gerenciamento de versões e releases preocupam-se com a identificação e a manutenção da rastreabilidade das versões de um sistema.

Uma versão de um sistema é uma instância de um sistema que difere, de alguma maneira, de outras instâncias. Versões de sistemas podem ter funcionalidades distintas, desempenhos aprimorados ou defeitos de software reparados. Versões com somente pequenas alterações são às vezes chamadas de *variantes.*

1. IDENTIFICAÇÃO DE VERSÕES

De acordo com Sommerville (2003), para criar uma versão específica de um sistema, você precisa especificar as versões dos componentes de sistema que devem ser incluídas nele. Em sistemas grandes de software, há centenas de componentes de software, cada um pode existir em versões diferentes. Há três técnicas básicas usadas para a identificação da versão de componente:

1. *Numeração de versões*. O componente recebe um número explícito e único de versão. Isso é o mais comumente usado no esquema de identificação.
2. *Identificação baseada em atributos*. Cada componente tem um nome e um grupo de atributos associados para cada versão.
3. *Identificação orientada a mudanças*. Cada componente é denominado como na identificação baseada em atributos, mas também é associado com uma ou mais solicitações de mudanças. Ou seja, considera-se que cada versão de componente foi criada em resposta a uma ou mais solicitações de mudanças.
4. GERENCIAMETO DE RELEASES

Um release de sistema é uma versão do sistema distribuído para os clientes. Os gerentes de releases de sistemas são responsáveis por decidir quando um sistema pode ser liberado para os clientes, gerenciar o processo de criação do release e de meios de distribuição e documentar o release para assegurar que ele pode ser recriado exatamente como foi distribuído, se for necessário (Pressman, 2005).

Um release de sistema não é somente um código executável do sistema. Segundo Sommerville (2003), ele pode incluir:

1. *Arquivos de configuração* que definem como o release pode ser configurado para instalações específicas.
2. *Arquivo de dados* necessários para a operação do sistema com sucesso.
3. *Um programa de instalação* usado para auxiliar a instalação do sistema no hardware-alvo.
4. *Documentação eletrônica e em papel* que descreve o sistema.
5. *Empacotamento e publicidade associada* projetados para release.

Os gerentes de release não podem considerar que os clientes sempre instalarão releases novos de sistema. Alguns usuários de sistemas podem se sentires satisfeitos com um sistema existente. Eles podem considerar que não vale a pena o custo de alteração para um novo release. Novos releases de sistema não podem, portanto, confiar na instalação de releases anteriores.

**5 CONSTRUÇÃO DE SISTEMAS**

De acordo com Cardozo (2007), a construção de sistemas é um processo de geração, compilação, ligação e empacotamento do código. Dependências entre componentes do sistema podem ser verificados durante a construção. E durante a construção pode-se proceder ainda à geração de documentação (exemplo: Javadoc), a análise estática (exemplo: FreeBugs) e o gerenciamento de versões (exemplo: Subversion). Ferramentas de construção de sistemas: Ant e Maven (ambas da Fundação Apache).

**6 FERRAMENTAS PARA GERENCIAMENTO DE CONFIGURAÇÃO**

Apresentamos abaixo o conceito e suas características de duas ferramentas especificas para o gerenciamento de configuração.

## 6.1 BORLANDSTARTEAM

Segundo Borland Software Corporation, o Borland StarTeam é uma ferramenta amplamente integrada e economicamente viável para o gerenciamento de mudanças e configuração de software e foi desenhada para ambientes de desenvolvimento centralizados ou geograficamente distribuídos.

Uma plataforma robusta para coordenar e gerenciar o processo de criação do software como um todo, o Borland StarTeam promove a comunicação e colaboração da equipe através do controle centralizado de todas as atividades e ativos digitais de um projeto. Ele fornece gerenciamento integrado de requisitos e de mudança e configuração de softwares, gerenciamento de projeto e tarefas, acompanhamento de defeitos, controle de versão de arquivos e discussões em formato de threads. Workflows customizáveis com formulários dinâmicos para tarefas de desenvolvimento estratégicas, unificando as equipes ainda mais no ambiente centralizado. E sua arquitetura efetiva e de fácil uso reduz seu custo total de propriedade.

O Borland StarTeam oferece uma gama completa de capacidades para o gerenciamento de mudanças e configuração de software em uma plataforma integrada que suporte equipes de desenvolvimento de todos os tamanhos, algumas destas capacidades são detalhadas abaixo:

* Gerenciamento Aberto do Ciclo de Vida da Aplicação (Open ALM): Workflows customizáveis, amplo suporte a sistemas operacionais e integração com ferramentas da Borland e de terceiros para o gerenciamento do ciclo de vida da aplicação – tudo para ajudar a dinamizar o processo de criação de softwares.
* Forte segurança: O Borland StarTeam criptografa todas as comunicações de cliente-servidor e seu repositório centralizado garante que todos os ativos digitais sejam armazenados de modo seguro, com controle dos acessos e das modificações através de regras definidas pelo administrador.
* Alta disponibilidade e performance: Agentes de cachê permitem que você crie um ambiente altamente distribuído com um servidor centralizado e ainda assim forneça uma performance similar à de redes aos clientes.
* Escalabilidade: O Borland StarTeam está disponível em duas edições: StarTeam Enterprise, para equipes de desenvolvimento de softwares de pequenas a médias; e StarTeam Enterprise Advantage, para equipes de desenvolvimento grandes e geograficamente distribuídas.

6.2 TORTOISESVN

De acordo com Elísio o TortoiseSVN é cliente gratuito e de código aberto para o sistema de controle de versão Subversion. Daqui já dá para ver que Subversion é um sistema de controle de versão. Sistema de controle de versão é fácil de definir: É um sistema que controla as alterações no código fonte de um projeto. Para além de controlar, ele ainda pode armazenar as alterações, para que a qualquer momento possamos voltar a uma dada versão. Acertei? É isto mesmo, TortoiseSVN gerencia arquivos e diretórios enquanto desenvolvemos um projeto. Os arquivos são armazenados num repositório central. O repositório é algo parecido com um servidor de arquivos normal, com a diferença de que o repositório armazena todas as mudanças excetuadas nos arquivos e diretórios. O repositório torna possível que você recupere versões antigas dos seus arquivos, e visualize o histórico completo sobre  como e quando o arquivo foi alterado, e ainda saber quem alterou este arquivo. É por estes e outros motivos que o Subversion e outros sistemas de controle de versão são  vistos como “A máquina do tempo”.

Elísio ainda define que existem muitas ferramentas para o Subversion, mas o TortoiseSVN tem uma característica, que é a interface gráfica, e a integração total com o Windows Explorer. Lembra que dissemos que era a melhor ferramenta de controle de versão para iniciantes? Para além da integração com o Windows Explorer, o que facilita muito a nossa vida, nós que estamos viciados no sistema de janelas do Windows, também é facilmente integrável a outras aplicações.

O TortoiseSVN possui formas específicas de indicar os arquivos que foram alterados num projeto, através de ícones em pastas e arquivos. A sincronização, criação de arquivos e outras funções também estão muito bem criadas e simplificadas no TortoiseSVN.

**7 CONCLUSÃO**

Pôde-se concluir através deste trabalho que a gerencia de configuração e mudança é algo fundamental em qualquer empresa de software, pois a utilização desta forma de gerenciamento proporciona uma forma eficaz de se gerenciar todo o processo de desenvolvimento, porém, identificamos que a gerencia de configuração não é um processo que pode ser realizado apenas com a utilização de uma ferramenta adequada, necessita-se também de conhecimento e processos definidos, que não podem ser realizados automaticamente. Ressaltamos aqui também que o controle de versão é, neste meio, uma das chaves do gerenciamento de configuração, pois reduz custos e incrementa a confiabilidade dos processos.

**REFERÊNCIAS**

## BORLAND SOFTWARE CORPORATION. Borland StarTeam: Ferramenta Completa para o Gerenciamento de Mudanças e Configuração de Softwares. Disponível em: < http://www.borland.com/br/products/starteam/>. Acesso em: 18 jun. 2009.

CARDOZO, Eleri. **Gerenciamento de Configuração.** 2007.Disponível em: < http://143.106.50.145:8080/Cursos/EA976/02-07/ea976-cm.pdf>. Acesso em: 18 jun. 2009.

ELISIO, Leonardo. **Projeto** : TortoiseSVN, SVN, Subversion e ferramentas de controle de versão. Disponível em: < http://infomoz.net/modules/news/article.php?storyid=278 >. Acesso em: 18 jun. 2009.

PETERS, James F. e PEDRYCZ, Witold. **Engenharia de software – teoria e prática.** Rio de Janeiro: Campus, 2001. 602p.

PRESSMAN, Roger S. **Engenharia de software.** São Paulo: Makron Books, 1995. 1056 p.

\_\_\_\_\_\_. **Engenharia de software**. 6. ed. São Paulo: McGraw-Hill, 2006. 720 p

SOMMERVILLE, Ian. **Engenharia de software.** 6. ed. São Paulo: Addison-Wesley, 2003. 592 p.